I.E.S. Jacarandá
Gestión de Base de Datos
PL/SQL
Relación de ejercicios 6

SIEMPRE QUE SE DIGA EL NOMBRE Y APELLIDO DE UNA PERSONA, DEBERÁN SER DOS PARÁMETRO O CAMPOS DISTINTOS.

1. Realizar un procedure que se llame insertar_alumno, que recibirá como parámetro el nombre y apellido de una persona, e inserte de forma automática esa persona como alumno.

```
CREATE OR REPLACE PROCEDURE insertar_alumno(pNombre PERSONA.NOMBRE%TYPE, pApellido PERSONA.APELLIDO
%TYPE)
IS

vDni persona.dni%TYPE;
vIdalumno alumno.idalumno%TYPE;

BEGIN

SELECT dni INTO vDni FROM PERSONA WHERE persona.nombre = pNombre AND persona.apellido = pApellido;
vIdalumno := 'A' || SUBSTR(vDni,1, 6);

INSERT INTO ALUMNO (idalumno, dni)
VALUES (vIdalumno, vDni);

exception
when no_data_found then
dbms_output.put_line('ESTA PERSONA ES MENTIRA, NO TIENE DENE Y');

END;
/

BEGIN
insertar_alumno('Rosario', 'Diaz');
END;
/
```

2. Realizar una función que reciba como parámetro el nombre y el apellido de una persona, también debe recibir un parámetro que podrá ser un uno (debe insertarlo un alumno) o un dos (debe insertarlo como profesor), y un parámetro de entrada salida en el que deberá devolver el código del profesor o alumno insertado. La función deberá devolver un 1 si se ha podido realizar la inserción, y un cero si ha ocurrido algún error.

```
CREATE OR REPLACE FUNCTION EJ2(nombreP PERSONA.NOMBRE%TYPE, apellidoP PERSONA.APELLIDO%TYPE,
    num NUMBER, codP IN OUT VARCHAR2)
RETURN VARCHAR2
IS

codPA PERSONA.DNI%TYPE;

BEGIN

SELECT PERSONA.DNI INTO codPA
FROM PERSONA
WHERE nombreP = PERSONA.NOMBRE AND apellidoP = PERSONA.APELLIDO;

   IF (num = 1) THEN
      codP := ('A' || SUBSTR(codPA, 1,6));
      INSERT INTO ALUMNO (IDALUMNO, DNI)
      VALUES(codP, codPA);
      RETURN 1;
   ELSIF(num = 2) THEN

      codP := ('P' || SUBSTR(codPA, 1,3));
      INSERT INTO PROFESOR(IDPROFESOR, DNI)
      VALUES(codP, codPA);
      RETURN 1;
   ELSE
      codP := NULL;
      RETURN 0;
END IF;
END;
/

SET SERVEROUTPUT ON;
DECLARE
   codigo VARCHAR2(7);
BEGIN

   DBMS_OUTPUT.PUT_LINE(EJ2('Jorge', 'Saenz', 1, codigo));

END;
/
```

3. Crear un procedure para que se llame tres o más veces a la función anterior, mostrando el mensaje el alumno tal ha sido insertado, o el alumno tal no ha sido insertado, y lo mismo con profesores.

```
CREATE OR REPLACE PROCEDURE EJER3
IS

idA ALUMNO.IDALUMNO%TYPE;
idP PROFESOR.IDPROFESOR%TYPE;
vNombre PERSONA.NOMBRE%TYPE;
aux NUMBER := EJ2('Mara','Gutierrez',1, idA);
aux2 NUMBER := EJ2('Luis','Ramirez',2, idP);
aux3 NUMBER := EJ2('Laura','Beltran',1, idA);

BEGIN

IF (aux = 1) THEN
SELECT persona.nombre INTO vNombre
from ALUMNO, persona
WHERE alumno.dni = persona.dni
AND idalumno = idA;
DBMS_OUTPUT.PUT_LINE('Se ha insertado el alumno ' || vNombre);

ELSIF (aux = 0) THEN
dbms_output.put_line('No se ha insertado');
END IF;

IF (aux2 = 1) THEN
SELECT persona.nombre INTO vNombre
from PROFESOR, persona
WHERE profesor.dni = persona.dni
AND idprofesor = idP;
DBMS_OUTPUT.PUT_LINE('Se ha insertado el profesor ' || vNombre);

ELSIF (aux2 = 0) THEN
dbms_output.put_line('No se ha insertado');
END IF;

IF (aux3 = 1) THEN
SELECT persona.nombre INTO vNombre
from alumno, persona
WHERE alumno.dni = persona.dni
AND idalumno = idA;
DBMS_OUTPUT.PUT_LINE('Se ha insertado el profesor ' || vNombre);

ELSIF (aux3 = 0) THEN
dbms_output.put_line('No se ha insertado');
END IF;

END;
/

SET SERVEROUTPUT ON
BEGIN
EJER3();
END;
/
```

4. Realizar una función que devuelva un uno o un cero, si el alumno con dni que se le pasa como argumento está matriculado en la asignatura cuyo nombre se le pasa también como argumento. La función también deberá tener un parámetro de entrada salida en donde se devuelva el nombre y apellido del profesor que le da clase en esa asignatura, si el alumno está matriculado. Procedure o bloque anónimo para comprobar que funciona

```
CREATE OR REPLACE FUNCTION EJ4 (dniA ALUMNO.DNI%TYPE, nombreA ASIGNATURA.NOMBRE%TYPE, nombreP
IN OUT VARCHAR2, apellidoP IN OUT VARCHAR2) RETURN NUMBER
IS
vNombreP VARCHAR2(50);
vApellidoP VARCHAR2(50);
BEGIN

SELECT persona.nombre, persona.apellido INTO vNombreP,vApellidoP
FROM alumno, asignatura, profesor, persona, alumno_asignatura
WHERE alumno.dni = dniA
AND alumno.idalumno = alumno_asignatura.idalumno
AND asignatura.idasignatura = alumno_asignatura.idasignatura
AND asignatura.nombre = nombreA
AND persona.dni = profesor.dni
AND profesor.idprofesor = asignatura.idprofesor;

IF vNombreP is not NULL AND vApellidoP is not NULL THEN
   DBMS_OUTPUT.PUT_LINE(vNombreP || ' ' || vApellidoP);
   RETURN 1;
END IF;

exception
when no_data_found then
RETURN 0;

END;
/

SET SERVEROUTPUT ON;
DECLARE
vNombreP VARCHAR2(50);
vApellidoP VARCHAR(50);
BEGIN
DBMS_OUTPUT.PUT_LINE(EJ4('20202020A', 'Quimica Fisica', vNombreP, vApellidoP));
END;
/
```

5. Realizar una función que devuelva un 1 si el nombre y apellido de la persona que se le pasa por parámetro es un alumno, un dos si es un profesor y un 0 si no está en la base de datos.

```
CREATE OR REPLACE FUNCTION EJER5(vNombre PERSONA.NOMBRE%TYPE, vApellido PERSONA.APELLIDO
%TYPE)RETURN NUMBER
IS

vDNI PERSONA.DNI%TYPE;
vIDP PROFESOR.IDPROFESOR%TYPE;
vIDA ALUMNO.IDALUMNO%TYPE;
vID VARCHAR(20);

BEGIN
SELECT PERSONA.DNI INTO vDNI
FROM PERSONA
WHERE vNombre = persona.nombre
AND vApellido = persona.apellido;

BEGIN
SELECT alumno.idalumno INTO vIDA
FROM alumno
WHERE vDNI = alumno.dni;

exception
when no_data_found THEN
dbms_output.put_line('');
END;

BEGIN
SELECT profesor.idprofesor INTO vIDP
FROM profesor
WHERE vDNI = profesor.dni;

exception
when no_data_found then
dbms_output.put_line('');
END;

vID := vIDA || vIDP;

IF (LENGTH(vID))>6 THEN
RETURN 1;
ELSIF (LENGTH(vID))<6 THEN
RETURN 2;
END IF;

exception
when no_data_found then
RETURN 0;

END;
/

SET SERVEROUTPUT ON;
BEGIN
DBMS_OUTPUT.PUT_LINE(EJER5('Juan', 'Sanchez'));
END;
/
```

6. Crear un procedure que reciba como parámetro el nombre de una titulación, el nombre de la asignatura, y el nombre y apellido del profesor (en dos parámetros distintos), y que inserte esos datos en la tabla de asignatura. Si se produce un error notitifcarlo. Los errores que deben notificarse son:
    - No existe la titulación
    - No existe la personas
    - La persona no es un profesor
    - El nombre de la asignatura ya está en la base de datos.

```
CREATE OR REPLACE PROCEDURE EJER6(nombreT TITULACION.NOMBRE%TYPE, nombreA ASIGNATURA.NOMBRE
%TYPE, nombrePro PERSONA.NOMBRE%TYPE, apellidoPro PERSONA.APELLIDO%TYPE)
IS

vIDTITULACION TITULACION.IDTITULACION%TYPE;
vIDPROFESOR PROFESOR.IDPROFESOR%TYPE;

BEGIN

SELECT IDPROFESOR INTO vIDPROFESOR
FROM PERSONA, PROFESOR
WHERE nombrePro = PERSONA.NOMBRE
AND apellidoPro = PERSONA.APELLIDO
AND persona.dni = PROFESOR.DNI;


SELECT IDTITULACION INTO vIDTITULACION
FROM TITULACION
WHERE nombreT = TITULACION.NOMBRE;

INSERT INTO ASIGNATURA(IDASIGNATURA, NOMBRE, IDPROFESOR, IDTITULACION)
VALUES('123456', nombreA, vIDPROFESOR, vIDTITULACION);

EXCEPTION
WHEN no_data_found THEN
DBMS_OUTPUT.PUT_LINE('ERROR');

END;
/

BEGIN
EJER6('Matematicas', 'SeguridVialica', 'Juan', 'Sanchez');
END;
/
```

7. Realizar una función que reciba un nombre de titulación y un porcentaje, y que realice la subida del precio en el porcentaje indicado de las asignaturas de esa titulación. La función también recibirá un parámetro de entrada salida en la que debe devolver la cantidad total que se ha subido en todas las asignaturas. La función debe devolver el número de asignatura que hay en esa titulación o un -1 si no hay ninguna.

```
CREATE OR REPLACE FUNCTION EJER6(nombreT TITULACION.NOMBRE%TYPE, porcentaje NUMBER, subido IN
OUT NUMBER) RETURN NUMBER
IS

numeroAsignaturas NUMBER;

BEGIN

SELECT (COUNT(asignatura.nombre)) INTO numeroAsignaturas
FROM ASIGNATURA, TITULACION
WHERE nombreT = TITULACION.NOMBRE
AND TITULACION.IDTITULACION = ASIGNATURA.IDTITULACION;

SELECT DISTINCT (SUM(COSTEBASICO*(porcentaje/100))) INTO subido
FROM ASIGNATURA, TITULACION
WHERE nombreT = TITULACION.NOMBRE
AND TITULACION.IDTITULACION = ASIGNATURA.IDTITULACION;

UPDATE ASIGNATURA
SET COSTEBASICO = COSTEBASICO + (COSTEBASICO * (porcentaje/100))
WHERE asignatura.idtitulacion = (SELECT titulacion.idtitulacion
                    FROM titulacion
                    WHERE nombreT = TITULACION.NOMBRE);

RETURN numeroAsignaturas;

EXCEPTION
WHEN no_data_found THEN
RETURN -1;

END;
/

SET SERVEROUTPUT ON;
DECLARE
subidoV NUMBER;
BEGIN
DBMS_OUTPUT.PUT_LINE(EJER6('Matematicas', 10, subidoV));
DBMS_OUTPUT.PUT_LINE('Se ha subido un total de ' || subidoV);
END;
/
```