1.- Dado el siguiente procedimiento almacenado:

CREATE OR REPLACE PROCEDURE **CrearDepart** (VNumDep depart.dept\_no%type, VNombreD depart.dnombre%type DEFAULT 'PROVISIONAL',VNUMCE depart.numce%type DEFAULT 7) IS

BEGIN

INSERT INTO DEPART (DEPT\_NO, DNOMBRE, NUMCE) VALUES (VNumDep,VNombreD,VNUMCE);

DBMS\_OUTPUT.PUT\_LINE('SE HA CREADO UN NUEVO DEPARTAMENTO '||VNOMBRED);

end;

Determinar cuál de las siguientes llamadas son correctas:

1. **CrearDepart;**
2. **CrearDepart(70);**
3. **CrearDepart(‘Compras’);**
4. **CrearDepart(70,’Compras’);**
5. **CrearDepart(‘Compras’,7);**
6. **CrearDepart(70,’Compras’,5);**
7. **CREARDEPART(90,null,3);**

2.- Codifica un procedimiento que reciba una cadena y la visualice al revés. E invocar al procedimiento.

3.- Realiza una función que reciba una cadena y devuelva la cadena invertida. Poned un ejemplo de llamada a la función.

4.- Procedimiento que permita borrar un empleado cuyo número se pasará en la llamada. Se visualizará un mensaje con el nombre del empleado borrado o un mensaje de error.

Visualiza todos los procedimientos y funciones del usuario almacenados en la BD y su situación (valid/invalid):

Select object\_name, object\_type, status from user\_objects where object\_type in ('PROCEDURE','FUNCTION');

5.- Crear una función **fverempleoficio** al que le paso un oficio y devuelve  ~~el nombre, apellidos, oficio, nombre departamento, nombre del departamento de los empleados que tienen ese oficio~~ el número de empleados de ese oficio.

create or replace FUNCTION fvercuantosoficio (POFICIO emple.OFICIO%type) RETURN INT

AS

TYPE INFO IS RECORD

(VNOMBRE EMPLE.NOMBRE%TYPE,

VAPEL EMPLE.APELLIDO%TYPE,

VOFICIO EMPLE.OFICIO%TYPE,

VNOMBRED DEPART.DNOMBRE%TYPE);

infoEmple INFO;

CONT INT;

BEGIN

SELECT COUNT(\*) INTO CONT FROM EMPLE WHERE oficio=pOFICIO;

IF CONT !=0 THEN

SELECT NOMBRE,apellido,oficio,DNOMBRE INTO infoEmple FROM EMPLE E, depart d WHERE e.dept\_no=d.dept\_no and e.oficio=POFICIO;

Dbms\_Output.Put\_Line (Infoemple.Vnombre||' '||Infoemple.Vapel||' '||Infoemple.Voficio||' '||Infoemple.Vnombred);

END IF;

RETURN CONT;

EXCEPTION

WHEN TOO\_MANY\_ROWS

THEN dbms\_output.put\_line('DEMASIADOS REGISTROS, HAY QUE USAR UN CURSOR !!!');

RETURN CONT;

END;

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  OFICIO EMPLE.OFICIO%TYPE;  CONT INT;  BEGIN  OFICIO:='&DAME\_OFICIO';  CONT:=fvercuantosoficio(OFICIO);  IF CONT !=0 THEN  dbms\_output.put\_line('HAY '||CONT || ' EMPLEADOS DEL OFICIO '||OFICIO);  ELSE  dbms\_output.put\_line('no HAY EMPLEADOS DEL OFICIO '||OFICIO);  END IF;  EXCEPTION  WHEN TOO\_MANY\_ROWS THEN dbms\_output.put\_line('DEMASIADOS REGISTROS, HAY QUE USAR UN CURSOR !!!');  WHEN NO\_DATA\_FOUND THEN dbms\_output.put\_line('NO HAY NINGUN REGISTRO, DE OFICIO : '||OFICIO);  WHEN OTHERS THEN dbms\_output.put\_line('eRROR: '||SQLERRM);  END; |

6.- Crear un procedimiento pcalcularsueldosdep al que se le pasa un número de departamento y muestra el total de sueldos, el total de comisiones, cuantos empleados hay.

7.- Crear el procedimiento paumentosalario2 se le pasa el número de departamento ~~y va mostrando uno por uno los empleados~~ y les actualice el sueldo en un 10%.

~~- Al presidente no se le sube el salario.~~

~~- Si son Analistas un 0,5 % el salario, si son Vendedores un 0,3%, si son Directores un 1% y si son otro tipo de oficio un 0,2%.~~

8. Desarrolla un procedimiento que visualice el apellido y la fecha de alta de todos los empleados ordenados por apellido. (Necesita Cursor).

9**.- Escribe un procedimiento que reciba una cadena y visualice el número de empleados cuyo apellido contenga la cadena especificada.**

10.- Codifica un procedimiento que muestre el **nombre del departamento pasado por parámetro** y el número de empleados que tiene, **utilizando una función que devuelva el número de empleados del departamento que se le pase como parámetro.**

11.- Escribe un programa que visualice el apellido y el salario del empleado con salario mayor ~~de los cinco empleados que tienen el salario más alto~~.

13.- Escribe un procedimiento que reciba todos los datos de un nuevo empleado y procese la transacción de alta, gestionando posibles errores. El procedimiento deberá gestionar en concreto los siguientes puntos:

* no\_existe\_departamento.
* no\_existe\_director.
* numero\_empleado\_duplicado.
* Salario nulo: con RAISE\_APPLICATION\_ERROR
* Otros posibles errores de Oracle visualizando códigos de error y el mensaje de error.