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.

**create or replace procedure VerAlReves (cadena varchar2)**

**IS**

**BEGIN**

**for cont in reverse 1 .. length(cadena) loop**

**dbms\_output.put(substr(cadena,cont,1));**

**end loop;**

**dbms\_output.put\_line(' ');**

**end;**

set serveroutput on

declare

cadena varchar2(100):='&texto';

begin

veralreves(cadena);

end;

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.

create or replace procedure eliminaEmple (numeroEmple int)

is

begin

delete from emple where numeroEmple = emp\_no;

if (sql%rowcount)>0 then

dbms\_output.put\_line(' eliminado ' || sql%rowcount);

else

dbms\_output.put\_line('No existe ese usuario');

end if;

end;

-- no se puede poner exception ya que los delete no generan exceptions

--solo los select

set serveroutput on

declare

numEmple emple.emp\_no%type:=&Dime\_num;

begin

eliminaEmple(numEmple);

end;

--select \* from emple;

--rollback;

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 un funcion **pverempleoficio** al que le paso un oficio y muestra ~~el nombre, apellidos, oficio, nombre departamento, nombre del departamento de los empleados que tienen ese oficio~~ el número de empleados de ese oficio.

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.

create or replace procedure pcalcularsueldosdep (numDepart emple.dept\_no%type)

is

fTotalSueldos emple.salario%type;

fTotalComision emple.comision%type;

fnumempleados int;

begin

select sum(salario), nvl(sum (comision),0), count(\*) into

fTotalSueldos,fTotalComision,fnumempleados from emple

where dept\_no=num);

dbms(fTotalSueldos || fTotalComision || fnumempleados);

end;

declare

num\_emple.dept\_no%type := &DameNumDepartmanto;

begin

pcalcularsueldosdep(num);

end;

OJO a nvl(sum (comision),0) que nos muestra un 0 en lugar de vacio o null, si la suma no devuelve nada.

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%.~~

**Como función** si se hace correctamente deberá devolver true or false;

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.