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 VerCadInvertida (cad IN VARCHAR2) IS**

**BEGIN**

**dbms\_output.put\_line('La cadena '||cad||' invertida es:');**

**FOR VAR IN REVERSE 1..LENGTH(cad)**

**LOOP**

**dbms\_output.put(SUBSTR(cad,VAR,1));**

**END LOOP;**

**dbms\_output.put\_line(''); -- Para liberar el buffer**

**END;**

**-- Llamada al método**

**set serveroutput on;**

**begin**

**vercadinvertida('hola que tal');**

**end;**

**/**

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

**create or replace FUNCTION CadInvertida (cad IN VARCHAR2) RETURN VARCHAR2 IS**

**inv VARCHAR2(40);**

**BEGIN**

**FOR VAR IN REVERSE 1..LENGTH(cad)**

**LOOP**

**INV:=INV||SUBSTR(cad,VAR,1);**

**END LOOP;**

**RETURN INV;**

**END;**

**-- Llamada a la funcion**

**set serveroutput on;**

**declare**

**cad varchar2(20):='Hoy es Viernes';**

**begin**

**dbms\_output.put\_line('La cadena '||cad||' invertida es :' ||cadinvertida(cad));**

**end;**

**/**

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 BorraEmple (nEmple emple.emp\_no%type) AS

VNOMBRE EMPLE.NOMBRE%TYPE;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(' SE PROCEDE A BORRAR AL EMPLEADO '||NEMPLE);

SELECT NOMBRE INTO VNOMBRE FROM EMPLE WHERE EMP\_NO=NEMPLE;

DELETE FROM EMPLE WHERE EMP\_NO=nEmple;

DBMS\_OUTPUT.PUT\_LINE(' EMPLEADO '||VNOMBRE||' BORRADO ');

END;

set serveroutput on;

DECLARE

NEMP EMPLE.EMP\_NO%TYPE;

BEGIN

NEMP:=&DAME\_EMPLEADO\_A\_BORRAR;

BORRAEMPLE(NEMP);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE(' nO EXISTE EL EMPLEADO '||NEMP);

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE(' ERROR '||SQLERRM);

END;

/

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 procedimiento **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.

CREATE OR REPLACE PROCEDURE pverempleoficio (POFICIO emple.OFICIO%type) AS

TYPE INFO IS RECORD (VNOMBRE EMPLE.NOMBRE%TYPE, VAPEL EMPLE.APELLIDO%TYPE, VOFICIO EMPLE.OFICIO%TYPE, VNOMBRED DEPART.DNOMBRE%TYPE);

infoEmple INFO;

BEGIN

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;

**\*\*nota: Si hay más de un empleado con ese oficio concreto se producirá el error TOO\_MANY\_ROWS. Habría que utilizar un cursor.**

SET SERVEROUTPUT ON;

DECLARE

OFICIO EMPLE.OFICIO%TYPE;

BEGIN

OFICIO:='&DAME\_OFICIO';

pverempleoficiot(OFICIO);

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;

/

Otra forma:

create or replace PROCEDURE pverempleoficio2 (POFICIO

emple.OFICIO%type) AS

TYPE INFO IS RECORD (VNOMBRE EMPLE.NOMBRE%TYPE,

VAPEL EMPLE.APELLIDO%TYPE, VOFICIO EMPLE.OFICIO%TYPE,

VNOMBRED DEPART.DNOMBRE%TYPE);

infoEmple INFO;

n int;

BEGIN

select count(\*) into n FROM EMPLE E, depart d WHERE

e.dept\_no=d.dept\_no and e.oficio=POFICIO;

if n=0 then

Dbms\_Output.Put\_Line ('No hay empleados del oficio '||

poficio);

elsif n>1 then

Dbms\_Output.Put\_Line ('hay '||n||' empleados del oficio '||

poficio);

else

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;

END;

Programa Principal:

SET SERVEROUTPUT ON;

DECLARE

OFICIO EMPLE.OFICIO%TYPE;

BEGIN

OFICIO:='&DAME\_OFICIO';

pverempleoficio2(OFICIO);

END;

create or replace PROCEDURE pverempleoficio (POFICIO emple.OFICIO%type) AS

TYPE INFO IS RECORD (VNOMBRE EMPLE.NOMBRE%TYPE, VAPEL EMPLE.APELLIDO%TYPE, VOFICIO EMPLE.OFICIO%TYPE, VNOMBRED DEPART.DNOMBRE%TYPE);

infoEmple INFO;

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

BEGIN

OPEN COFICIO;

FETCH COFICIO INTO infoEmple;

while coficio%FOUND

LOOP

DBMS\_OUTPUT.PUT\_LINE (infoEmple.VNOMBRE||' '||infoEmple.Vapel||' '||infoEmple.Voficio||' '||infoEmple.VNOMBRED);

FETCH COFICIO INTO infoEmple;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('SE HAN PROCESADO '||COFICIO%ROWCOUNT||' REGISTROS');

CLOSE COFICIO;

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.

create or replace Procedure Pcalcularsueldosdep (Ndep Emple.Dept\_No%Type) As

Numemp Number;

Sums Number;

Sumc Number;

Begin

Select Count(\*) , Sum(Salario) , NVL(Sum(Comision),0) Into Numemp,Sums,Sumc From Emple Where Dept\_No=Ndep;

Dbms\_Output.Put\_Line ('NUMERO DE EMPLEADOS: '||NUMEMP||' DEL DEPARTAMENTO '||NDEP||' TOTAL DE SUELDOS '||SUMS||' TOTAL COMISIONES '||SUMC);

End;

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.

- 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%.

Create Or Replace Procedure Paumentosalario2 (Ndep Emple.Dept\_No%Type) As

Cursor Cemple Is Select \* From Emple Where Dept\_No=Ndep **FOR UPDATE;**

Reg Emple%Rowtype;

INC NUMBER:=1;

Begin

Open Cemple;

Fetch Cemple Into Reg;

While Cemple%Found

Loop

Dbms\_Output.Put\_Line (Reg.Nombre||' '||Reg.Apellido||' '||Reg.Oficio||' ' ||Reg.Dept\_No||' '||Reg.Salario);

Case Reg.Oficio

When 'PRESIDENTE' Then Inc:=1;

When 'ANALISTA' Then Inc:=1.005;

When 'VENDEDOR' Then Inc:=1.003;

When 'DIRECTOR' Then Inc:=1.01;

Else Inc:=1.02;

End Case;

UPDATE EMPLE SET SALARIO=SALARIO\*INC **WHERE CURRENT OF CEMPLE;**

Fetch Cemple Into Reg;

End Loop;

Dbms\_Output.Put\_Line ('NUMERO DE EMPLEADOS DEL DEPARTAMENTO : '||Ndep||' ES '||Cemple%Rowcount);

CLOSE CEMPLE;

End;

SET SERVEROUTPUT ON;

DECLARE

NDEP EMPLE.DEPT\_NO%TYPE;

BEGIN

NDEP:=&DAME\_NUMDEPT;

paumentosalario2(NDEP);

EXCEPTION

WHEN OTHERS THEN dbms\_output.put\_line('eRROR: '||SQLERRM);

end;

/

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

create or replace PROCEDURE pverApellidos AS

vapel emple.apellido%type;

vfecha emple.fecha\_alt%type;

CURSOR CAPELLIDO IS SELECT apellido, fecha\_alt FROM EMPLE order by 1;

BEGIN

OPEN CAPELLIDO;

FETCH CAPELLIDO INTO vapel,vfecha;

while CAPELLIDO%FOUND

LOOP

DBMS\_OUTPUT.PUT\_LINE (vapel||' '||Vfecha);

FETCH CAPELLIDO INTO vapel,vfecha;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('SE HAN PROCESADO '||CAPELLIDO%ROWCOUNT||' REGISTROS');

CLOSE CAPELLIDO;

END;

* Llamada:

SET SERVEROUTPUT ON;

BEGIN

PVERAPELLIDOS;

EXCEPTION

WHEN OTHERS THEN dbms\_output.put\_line('eRROR: '||SQLERRM);

END;

/

9.- Escribe un procedimiento que reciba una cadena y visualice el apellido y el número de empleado, de todos los empleados cuyo apellido contenga la cadena especificada. Al finalizar, visualiza el número de empleados mostrados.

create or replace PROCEDURE pverempleApellido (PAPELLIDO emple.APELLIDO%type) AS

TYPE INFO IS RECORD (VNOMBRE EMPLE.NOMBRE%TYPE, VAPEL EMPLE.APELLIDO%TYPE, VOFICIO EMPLE.OFICIO%TYPE, VNOMBRED DEPART.DNOMBRE%TYPE);

infoEmple INFO;

CAD VARCHAR2(30);

CURSOR CAPELLIDO IS SELECT NOMBRE,apellido,oficio,DNOMBRE FROM EMPLE E, depart d WHERE e.dept\_no=d.dept\_no and e.APELLIDO LIKE CAD;

BEGIN

CAD:='%'||PAPELLIDO||'%';

OPEN CAPELLIDO;

FETCH CAPELLIDO INTO infoEmple;

while CAPELLIDO%FOUND

LOOP

DBMS\_OUTPUT.PUT\_LINE (infoEmple.VNOMBRE||' '||infoEmple.Vapel||' '||infoEmple.Voficio||' '||infoEmple.VNOMBRED);

FETCH CAPELLIDO INTO infoEmple;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('SE HAN PROCESADO '||CAPELLIDO%ROWCOUNT||' REGISTROS');

CLOSE CAPELLIDO;

END;

SET SERVEROUTPUT ON;

DECLARE

APEL EMPLE.APELLIDO%TYPE;

BEGIN

APEL:='&DAME\_APELLIDO';

pverempleAPELLIDO(APEL);

EXCEPTION

WHEN OTHERS THEN dbms\_output.put\_line('eRROR: '||SQLERRM);

END;

/

10.- Codifica un procedimiento que muestre el nombre de cada departamento y el número de empleados que tiene.

create or replace PROCEDURE verNombreDep IS

vnombreD depart.dnombre%type;

cont number;

CURSOR CNOMBRED IS SELECT DNOMBRE, COUNT(emp\_no) FROM EMPLE E, DEPART D WHERE E.DEPT\_NO=D.DEPT\_NO

GROUP BY DNOMBRE;

BEGIN

OPEN CNOMBRED;

FETCH CNOMBRED INTO VNOMBRED,CONT;

while CNOMBRED%FOUND

LOOP

DBMS\_OUTPUT.PUT\_LINE (VNOMBRED||' => '||CONT);

FETCH CNOMBRED INTO VNOMBRED,CONT;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE ('SE HAN PROCESADO '||CNOMBRED%ROWCOUNT||' DEPARTAMENTOS');

CLOSE CNOMBRED;

END;

SET SERVEROUTPUT ON;

begin

verNombreDep;

end;

/

11.- Escribe un programa que visualice el apellido y el salario de los cinco empleados que tienen el salario más alto.

SET SERVEROUTPUT ON;

DECLARE

CURSOR SALMASALTO IS SELECT APELLIDO,SALARIO FROM EMPLE ORDER BY SALARIO DESC;

begin

FOR VAR IN SALMASALTO

LOOP

DBMS\_OUTPUT.PUT\_LINE(VAR.APELLIDO||' '||VAR.SALARIO);

IF SALMASALTO%ROWCOUNT =5 THEN

EXIT;

END IF;

END LOOP;

end;

/

SET SERVEROUTPUT ON;

DECLARE

CURSOR SALMASALTO IS SELECT APELLIDO,SALARIO FROM EMPLE ORDER BY SALARIO DESC;

VAR SALMASALTO%ROWTYPE;

I NUMBER:=0;

begin

OPEN SALMASALTO;

FETCH SALMASALTO INTO VAR;

WHILE SALMASALTO%FOUND AND I < 5

LOOP

DBMS\_OUTPUT.PUT\_LINE(VAR.APELLIDO||' '||VAR.SALARIO);

FETCH SALMASALTO INTO VAR;

I:=I+1;

END LOOP;

CLOSE SALMASALTO;

end;

/

12.- Codifica un programa que visualice los dos empleados que ganan menos de cada oficio.

SET SERVEROUTPUT ON;

DECLARE

CURSOR SALMENOROFICIO IS SELECT APELLIDO, SALARIO,OFICIO FROM EMPLE ORDER BY OFICIO,SALARIO;

VAR SALMENOROFICIO%ROWTYPE;

I NUMBER:=0;

OFICIOANT EMPLE.OFICIO%TYPE;

begin

OPEN SALMENOROFICIO;

FETCH SALMENOROFICIO INTO VAR;

OFICIOANT:=' ';

WHILE SALMENOROFICIO%FOUND

LOOP

IF OFICIOANT != VAR.OFICIO THEN

I:=0;

OFICIOANT:=VAR.OFICIO;

DBMS\_OUTPUT.PUT\_LINE('OFICIO: '||VAR.OFICIO);

END IF;

IF I<2 THEN

DBMS\_OUTPUT.PUT\_LINE(VAR.APELLIDO|| ' ' ||VAR.SALARIO);

END IF;

FETCH SALMENOROFICIO INTO VAR;

I:=I+1;

END LOOP;

CLOSE SALMENOROFICIO;

end;

/

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.

create or replace PROCEDURE DATOSEMPLE (VEMPLE IN EMPLE%ROWTYPE) IS

CURSOR CEMPLE IS SELECT \* FROM EMPLE;

VAR CEMPLE%ROWTYPE;

NODEP BOOLEAN:=TRUE;

NODIR BOOLEAN:=TRUE;

NOEMPLE BOOLEAN:=TRUE;

NOSAL BOOLEAN:=TRUE;

BEGIN

OPEN CEMPLE;

FETCH CEMPLE INTO VAR;

WHILE CEMPLE%FOUND

LOOP

IF VEMPLE.DEPT\_NO = VAR.DEPT\_NO THEN

NODEP:=FALSE;

DBMS\_OUTPUT.PUT\_LINE('DEPT: '||VAR.DEPT\_NO||' EXISTE');

END IF;

IF VAR.OFICIO ='DIRECTOR' AND VEMPLE.DEPT\_NO=VAR.DEPT\_NO THEN

NODIR:=FALSE;

DBMS\_OUTPUT.PUT\_LINE(VAR.APELLIDO|| 'HAY DIRECTOR ' );

END IF;

IF VAR.EMP\_NO =VEMPLE.EMP\_NO THEN

NOEMPLE:=FALSE;

DBMS\_OUTPUT.PUT\_LINE(VAR.EMP\_NO|| ' YA EXISTE ' );

RAISE\_APPLICATION\_ERROR(-20002,'EMPLEADO DUPLICADO !!!');

END IF;

IF VEMPLE.SALARIO IS NULL THEN

NOSAL:=FALSE;

DBMS\_OUTPUT.PUT\_LINE(VEMPLE.SALARIO|| ' ES NULO ' );

RAISE\_APPLICATION\_ERROR(-20003,'SALARIO NULO !!!');

END IF;

FETCH CEMPLE INTO VAR;

END LOOP;

IF NODEP=FALSE AND NODIR=FALSE AND NOEMPLE=TRUE THEN

DBMS\_OUTPUT.PUT\_LINE('LOS DATOS SON CORRECTOS ');

INSERT INTO EMPLE VALUES (VEMPLE.EMP\_NO,VEMPLE.NOMBRE,VEMPLE.APELLIDO,VEMPLE.OFICIO,VEMPLE.FECHA\_ALT,VEMPLE.SALARIO,0,VEMPLE.DEPT\_NO,NULL,NULL,0);

-- VEREMPLE(VEMPLE);

END IF;

CLOSE CEMPLE;

end;

SET SERVEROUTPUT ON;

DECLARE

VAR EMPLE%ROWTYPE;

BEGIN

VAR.EMP\_NO:=&NUMERO\_EMPLEADO;

VAR.NOMBRE:='&NOMBRE\_EMPLEADO';

VAR.APELLIDO:='&APELLIDO\_EMPLEADO';

VAR.OFICIO:='&OFICIO\_EMPLEADO';

VAR.SALARIO:=&SUELDO\_EMPLEADO;

VAR.FECHA\_ALT:=SYSDATE();

VAR.DEPT\_NO:=&NUMERO\_DEPARTAMENTO;

DATOSEMPLE(VAR);

end;

/

14.- Desarrolla un procedimiento que permita insertar nuevos departamentos según las siguientes especificaciones.

* Se pasará al procedimiento el nombre del departamento y la localidad.
* El procedimiento insertará la fila nueva asignando como número de departamento la decena siguiente al número mayor de la tabla
* Se incluirá la gestión de posibles errores.

CREATE OR REPLACE PROCEDURE INSERTADEP (NOMBRE DEPART.DNOMBRE%TYPE, NUMCEN DEPART.NUMCE%TYPE) IS

CURSOR DEP IS SELECT DNOMBRE, DEPT\_NO FROM DEPART;

NUEVODEP DEPART.DEPT\_NO%TYPE;

NUMDEP DEPART.DEPT\_NO%TYPE;

EXISTE BOOLEAN:=FALSE;

BEGIN

FOR VAR IN DEP

LOOP

IF VAR.DNOMBRE=NOMBRE THEN

DBMS\_OUTPUT.PUT\_LINE('EL DEPARTAMENTO '||NOMBRE||' YA EXISTE');

EXISTE:=TRUE;

EXIT;

END IF;

NUMDEP:=VAR.DEPT\_NO;

END LOOP;

IF EXISTE = FALSE THEN

NUEVODEP:=NUMDEP+10;

DBMS\_OUTPUT.PUT\_LINE('SE DA DE ALTA EL DEP. NUM: '||NUEVODEP||' DE NOMBRE '||NOMBRE);

INSERT INTO DEPART (DEPT\_NO,DNOMBRE,NUMCE) VALUES (NUEVODEP,NOMBRE,NUMCEN);

END IF;

END;

SET SERVEROUTPUT ON;

DECLARE

NOMBRE DEPART.DNOMBRE%TYPE;

NUM DEPART.NUMCE%TYPE;

BEGIN

NOMBRE:='&NOMBRE\_DEPARTAMENTO';

NUM:=&NUMCENTRO;

INSERTADEP(NOMBRE,NUM);

end;

/

15.- Escribe un procedimiento que suba el sueldo de todos los empleados que ganen menos que el salario medio de su oficio. La subida será del 50 por cien de la diferencia entre el salario del y la media de su oficio. Se deberá hacer que la transacción no se quede a medias, y se gestionarán los posibles errores.

create or replace PROCEDURE SUBE\_SUELDO IS

CURSOR CEMPLE IS SELECT EMP\_NO, SALARIO, OFICIO FROM EMPLE FOR UPDATE;

VAR CEMPLE%ROWTYPE;

SAL\_MEDIO number(8,2):=0;

DIF\_SAL number(8,2);

BEGIN

OPEN CEMPLE;

FETCH CEMPLE INTO VAR;

IF CEMPLE%NOTFOUND THEN

dbms\_output.put\_line('NO HAY REGISTROS ');

END IF;

WHILE CEMPLE%FOUND

LOOP

SELECT AVG(SALARIO) INTO SAL\_MEDIO FROM EMPLE GROUP BY OFICIO HAVING OFICIO=VAR.OFICIO;

dbms\_output.put\_line('EL SALARIO MEDIO DEL OFICIO '||VAR.OFICIO ||' ES '||SAL\_MEDIO);

IF VAR.SALARIO <SAL\_MEDIO THEN

DIF\_SAL:=(SAL\_MEDIO-VAR.SALARIO)\*0.05;

UPDATE EMPLE SET SALARIO=SALARIO+DIF\_SAL WHERE CURRENT OF CEMPLE;

ELSE

dbms\_output.put\_line('EL SALARIO NO SE MODIFICA');

END IF;

FETCH CEMPLE INTO VAR;

END LOOP;

CLOSE CEMPLE;

END;

SET SERVEROUTPUT ON;

BEGIN

SUBE\_SUELDO;

END;

/

OTRA FORMA:

/\* SE ORDENA POR OFICIO Y SÓLO SE CALCULA LA MEDIA, CADA VEZ QUE SE CAMBIA DE OFICIO\*/

create or replace PROCEDURE SUBE\_SUELDO2 IS

CURSOR CEMPLE IS SELECT EMP\_NO, SALARIO, OFICIO FROM EMPLE FOR UPDATE ORDER BY OFICIO;

VAR CEMPLE%ROWTYPE;

OFICIO\_ANT EMPLE.OFICIO%TYPE;

PRIMER\_REGISTRO BOOLEAN:=TRUE;

SAL\_MEDIO number(8,2):=0;

DIF\_SAL number(8,2);

BEGIN

OPEN CEMPLE;

FETCH CEMPLE INTO VAR;

IF CEMPLE%NOTFOUND THEN

dbms\_output.put\_line('NO HAY REGISTROS ');

END IF;

OFICIO\_ANT:=VAR.OFICIO;

WHILE CEMPLE%FOUND

LOOP

IF OFICIO\_ANT != VAR.OFICIO OR PRIMER\_REGISTRO THEN

SELECT AVG(SALARIO) INTO SAL\_MEDIO FROM EMPLE GROUP BY OFICIO HAVING OFICIO=VAR.OFICIO;

dbms\_output.put\_line('EL SALARIO MEDIO DEL OFICIO '||VAR.OFICIO ||' ES '||SAL\_MEDIO);

OFICIO\_ANT:=VAR.OFICIO;

PRIMER\_REGISTRO:=FALSE;

END IF;

IF VAR.SALARIO <SAL\_MEDIO THEN

DIF\_SAL:=(SAL\_MEDIO-VAR.SALARIO)\*0.05;

UPDATE EMPLE SET SALARIO=SALARIO+DIF\_SAL WHERE CURRENT OF CEMPLE;

ELSE

dbms\_output.put\_line('EL SALARIO NO SE MODIFICA');

END IF;

FETCH CEMPLE INTO VAR;

END LOOP;

CLOSE CEMPLE;

END;