**6. Aumentar un 10% el salario de los empleados que ganan menos de 1650.**

DECLARE

CURSOR cursorEmp IS

SELECT \*

FROM employees;

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

IF v\_regEmp.msal < 1650 THEN

UPDATE employees SET msal = msal \*1.10

WHERE CURRENT OF cursorEmp;

END LOOP;

END;

**7. Crear un procedimiento o bloque que recorra los empleados en un cursor, por cada empleado vea su grado salarial y si está por debajo de grado 4, actualice su salario en 500, si está por encima de grado 4 en 1000**

DECLARE

CURSOR cursorEmp IS

select e.ename, e.msal, s.grade

from employees e

join salgrades s

on (e.msal between s.lowerlimit and s.upperlimit);

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

IF v\_regEmp.grade<4 THEN

UPDATE employees SET msal = msal +500

WHERE CURRENT OF cursorEmp;

ELSE IF v\_regEmp.grade>4 THEN

UPDATE employees SET msal = msal +1000

WHERE CURRENT OF cursorEmp;

END IF;

END LOOP;

END;

**8. Procedimiento para actualizar el salario en 100€ a los empleados del departamento ventas si su grado salarial es superior a 2 usando una única query (un solo cursor) y sin bloquear la tabla departamentos**

create or replace PROCEDURE CURSORES\_08

(

CURSOR cursorEmp IS

select e.ename, e.msal, s.grade, d.deptno

from employees e

join salgrades s

on (e.msal between s.lowerlimit and s.upperlimit)

join departments d

on (e.deptno=d.deptno)

where lower(d.dname)='sales'

FOR UPDATE OF msal NOWAIT;

) AS

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

IF v\_regEmp.grade<2 THEN

UPDATE employees SET msal = msal +1000

WHERE CURRENT OF cursorEmp;

END IF;

END LOOP;

END CURSORES\_08;

**9. Crear un procedimiento que reciba un parámetro como porcentaje e incremente el salario de los empleados cuyo salario es superior a la media de salarios de la empresa.**

create or replace PROCEDURE CURSORES\_09

(

vPorcentaje IN NUMBER(4),

CURSOR cursorEmp IS

select msal

from employees

where msal >

(select avg(msal)

from employees);

) AS

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

UPDATE employees SET msal = msal + (msal\*vPorcentaje);

WHERE CURRENT OF cursorEmp;

END LOOP;

END CURSORES\_09;

**10. Crear un procedimiento que reciba el número de empleado por parámetro y, si es jefe, muestre los cursos a los que ha asistido (código de curso, fecha y localización).**

create or replace PROCEDURE CURSORES\_10

(

vEmpno IN employees.empno%type,

CURSOR cursorEmp IS

select e.empno, r.course, r.begindate, o.location

from employees e

join registrations r

on (e.empno=r.attendee)

join offerings o

on (r.course=o.course)

where empno=vEmpno AND empno in(

select mgr

from employees);

) AS

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

DBMS\_OUTPUT.PUT\_LINE(

v\_regEmp.empno||','||

v\_regEmp.course||','||

v\_regEmp.begindate||','||

v\_regEmp.location);

END LOOP;

END CURSORES\_10;

**11. Crear un procedimiento que reciba por parámetro el código de un curso y me muestre en qué localidades se ha impartido y nombre de los trainer que lo han impartido.**

create or replace PROCEDURE CURSORES\_11

(

vCourse IN offerings.course%type :=&v,

CURSOR cursorEmp IS

select o.location, e.ename

from offerings o

join employees e

on (o.trainer=e.empno)

where o.course=vCourse;

) AS

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

DBMS\_OUTPUT.PUT\_LINE(

v\_regEmp.location||','||

v\_regEmp.ename);

END LOOP;

END CURSORES\_11;

**12. Realizar una función que reciba como parámetro el número de un empleado y devuelva lo que ha ganado en el último año. Desde un procedimiento, usando dicha función, se sacarán todos los empleados que hayan ganado más que el empleado en el último año.**

**FUNCIÓN:**

create or replace FUNCTION CURSORES\_12\_F

(

vEmpno IN employees.empno%type,

CURSOR cursorEmp IS

select (msal\*12)+comm as ganancias

from employees

where empno=vEmpno;

)

RETURN varchar2

AS

v\_ganancias number;

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

v\_ganancias:=v\_regEmp.ganancias;

RETURN v\_ganancias;

END LOOP;

END CURSORES\_12\_F;

**PROCEDIMIENTO:**

create or replace PROCEDURE CURSORES\_12\_P

(

vEmpno IN employees.empno%type :=&v,

CURSOR cursorEmp IS

select \*

from employees

where (msal\*12)+comm > CURSORES\_12\_F(vEmpno);

) AS

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

DBMS\_OUTPUT.PUT\_LINE(

v\_regEmp.empno||','||

v\_regEmp.msal);

END LOOP;

END CURSORES\_12\_P;

**13. Realizar un procedimiento que reciba como parámetro un salario mínimo y un salario máximo. Sacar la categoría o grado correspondiente a ese intervalo de salario y modificar los empleados que están dentro de esa categoría de sueldo aumentando su sueldo en 100 €.**

create or replace PROCEDURE CURSORES\_13

(

vMin\_msal IN employees.msal%type :=&v,

vMax\_msal IN employees.msal%type :=&v,

CURSOR cursorEmp IS

select \*

from employees

where msal between vMin\_msal and vMax\_msal;

) AS

BEGIN

FOR v\_regEmp IN cursorEmp LOOP

UPDATE employees SET msal = msal +100;

WHERE CURRENT OF cursorEmp;

END LOOP;

END CURSORES\_13;