**LABORATORIO 03 – FNC, SP & PACKAGE**

1. Cree un procedimiento almacenado llamado "SP\_AUMENTA\_SUELDO". Debe incrementar el sueldo de los empleados con cierta cantidad de años en la empresa (parámetro "ayear" de tipo numérico) en un porcentaje (parámetro "aporcentaje" de tipo numerico); es decir, recibe 2 parámetros.

\*El campo HIREDATE es la fecha de ingreso del empleado a la empresa

CREATE OR REPLACE PROCEDURE SP\_AUMENTA\_SUELDO (ayear in number , aporcentaje in number ) IS

BEGIN

update emp set sal= sal+ (aporcentaje/100)\*sal where trunc((sysdate-hiredate)/360) = ayear;

commit;

END;

Validación:

EXECUTE SP\_AUMENTA\_SUELDO(5,10);

Bloque de ejecución:

SET SERVEROUTPUT ON

DECLARE

AYEAR NUMBER:=5;

APORCENTAJE NUMBER:=10;

BEGIN

SP\_AUMENTA\_SUELDO(AYEAR, APORCENTAJE);

END;

1. Crear un Store Procedure que actualice el campo COMM de la tabla EMP con un 10% del Salario a los Empleados cuyo Departamentos tengan de 0 a 2 empleados, 15% del Salario a los Empleados cuyo Departamentos tengan de 3 a 5 empleados y 20% del Salario a los Empleados cuyo Departamentos tengan mas de 5 empleados (Usar cursores, %type o %rowtype) El procedure no necesita que ingrese variables.

CREATE OR REPLACE PROCEDURE SP\_ACTUALIZA\_COMM

AS

V\_DEPTNO DEPT.DEPTNO%TYPE;

CURSOR C1 IS

select d.deptno from emp e, dept d

where e.deptno=d.deptno

group by d.deptno, d.dname

having count(e.empno) between 0 and 2;

CURSOR C2 IS

select d.deptno from emp e, dept d

where e.deptno=d.deptno

group by d.deptno, d.dname

having count(e.empno) between 3 and 5;

CURSOR C3 IS

select d.deptno from emp e, dept d

where e.deptno=d.deptno

group by d.deptno, d.dname

having count(e.empno) >5;

BEGIN

OPEN C1;

LOOP

FETCH C1 INTO V\_DEPTNO;

EXIT WHEN C1%NOTFOUND;

UPDATE EMP SET COMM=(SAL\*0.10) WHERE DEPTNO=V\_DEPTNO;

COMMIT;

END LOOP;

CLOSE C1;

OPEN C2;

LOOP

FETCH C2 INTO V\_DEPTNO;

EXIT WHEN C2%NOTFOUND;

UPDATE EMP SET COMM=(SAL\*0.15) WHERE DEPTNO=V\_DEPTNO;

COMMIT;

END LOOP;

CLOSE C2;

OPEN C3;

LOOP

FETCH C3 INTO V\_DEPTNO;

EXIT WHEN C3%NOTFOUND;

UPDATE EMP SET COMM=(SAL\*0.20) WHERE DEPTNO=V\_DEPTNO;

COMMIT;

END LOOP;

CLOSE C3;

END;

1. Crear una función que al ingresar el código de un empleado me retorne el monto de su CTS a depositar en su cuenta bancaria

Formula para hallar la CTS= ((SALARIO \* 14 ) + 20% SALARIO ) / 6

NOTA: solo reciben CTS aquellos empleados que hayan ingresado a la empresa después del 01/06/1981 en caso no cumpla se retornara 0 y se pintara en consola un mensaje que diga "NO CORRESPONDE CTS"

CREATE OR REPLACE FUNCTION FNC\_CTS (v\_empno in emp.empno%type)

RETURN number

IS

cts number := 0;

BEGIN

select round(((sal\*14) + (0.2\*sal)/6),2) into cts from emp where empno=v\_empno and hiredate >= '01/06/1981';

if cts is null then

return 0;

***DBMS\_OUTPUT.PUT\_LINE***('NO CORRESPONDE CTS');

else

return cts;

end if;

END;

Script de Ejecucion:

set serveroutput on;

SELECT FNC\_CTS('7654') FROM DUAL;

SELECT FNC\_CTS('7369') FROM DUAL

1. Crear el package PKG\_MANT\_DEPT con 3 Stores Procedures:

sp\_ingresa\_dept – Ingresa todos los campos de la tabla dept

\*Para el código del departamento DEPTNO utilizar una SEQUENCIA que empiece en 60 y el valor incremente de 10 en 10

sp\_actualiza\_dept – Actualiza los datos de un departamento a través del deptno

sp\_elimina\_dept – Elimina un departamento a través de la columna deptno

Tener en cuenta que se cumplan las siguientes validaciones:

* Si el departamento al ingresar ya existe enviar un mensaje en consola a través de una excepción “EL DEPARTAMENTO YA EXISTE”
* Si se requiere actualizar un departamento y este no existe enviar un mensaje en consola a través de una excepción “EL DEPARTAMENTO NO EXISTE”
* Si se requiere eliminar un departamento y este no existe enviar un mensaje en consola a través de una excepción “EL DEPARTAMENTO NO EXISTE O YA FUE ELIMINADO”

SET SERVEROUTPUT ON;

CREATE SEQUENCE SQ\_DEPT

INCREMENT BY 10

START WITH 10

MINVALUE 10

MAXVALUE 90

NOCACHE

NOCYCLE;

CREATE OR REPLACE PACKAGE "PKG\_MANT\_DEPT" AS

PROCEDURE sp\_ingresa\_dept (V\_DNAME DEPT.DNAME%TYPE,V\_LOC DEPT.LOC%TYPE);

PROCEDURE sp\_actualiza\_dept (V\_deptno dept.deptno%TYPE,V\_dname dept.dname%TYPE);

PROCEDURE sp\_elimina\_dept (V\_DEPTNO DEPT.DEPTNO%TYPE);

END PKG\_MANT\_DEPT;

CREATE OR REPLACE PACKAGE BODY "PKG\_MANT\_DEPT" AS

PROCEDURE sp\_ingresa\_dept (V\_DNAME DEPT.DNAME%TYPE,V\_LOC DEPT.LOC%TYPE)

AS

BEGIN

INSERT INTO DEPT VALUES (SQ\_DEPT.NEXTVAL, v\_dname, v\_loc);

***dbms\_output.put\_line***('EL DEPARTAMENTO '|| v\_dname||' FUE INGRESADO');

COMMIT;

END sp\_ingresa\_dept;

PROCEDURE sp\_actualiza\_dept (V\_deptno dept.deptno%TYPE,V\_dname dept.dname%TYPE)

AS

FILA DEPT%ROWTYPE;

BEGIN

SELECT \* INTO FILA FROM DEPDEPTNOT WHERE deptno = v\_deptno;

UPDATE DEPT SET DNAME = V\_DNAME WHERE deptno = V\_DEPTNO;

COMMIT;

***dbms\_output.put\_line***('EL DEPARTAMENTO '||fila.deptno||' FUE ACTUALIZADO');

exception

when no\_data\_found then

***dbms\_output.put\_line***('EL DEPARTAMENTO NO EXISTE');

END sp\_actualiza\_dept;

PROCEDURE sp\_elimina\_dept (V\_DEPTNO DEPT.DEPTNO%TYPE)

AS

V\_RECODEPT DEPT%ROWTYPE;

BEGIN

SELECT \* INTO V\_RECODEPT FROM DEPT WHERE DEPTNO=V\_DEPTNO;

DELETE FROM DEPT WHERE DEPTNO=V\_DEPTNO;

COMMIT;

***DBMS\_OUTPUT.PUT\_LINE***('EL DEPARTAMENTO'||' '||V\_RECODEPT.DEPTNO||' '||'FUE ELIMINADO');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

***DBMS\_OUTPUT.PUT\_LINE***('EL DEPARTAMENTO NO EXISTE O YA FUE ELIMINADO');

END sp\_elimina\_dept ;

END PKG\_MANT\_DEPT;

**Validacion:**

EXECUTE PKG\_MANT\_DEPT.sp\_ingresa\_dept('RRHH','LIMA');