

## Práctica de PAQUETES

1. Crear un paquete denominado REGIONES que tenga los siguientes componentes:

- PROCEDIMIENTOS:

- - ALTA\_REGION, con parámetro de código y nombre Región.

Debe devolver un error si la región ya existe. Inserta una nueva región en la tabla. Debe llamar a la función EXISTE\_REGION para controlarlo.

- - BAJA\_REGION, con parámetro de código de región y que debe borrar una región. Debe generar un error si la región no existe, Debe llamar a la función EXISTE\_REGION para controlarlo

- - MOD\_REGION: se le pasa un código y el nuevo nombre de la región Debe modificar el nombre de una región ya existente. Debe generar un error si la región no existe, Debe llamar a la función EXISTE\_REGION para controlarlo

- FUNCIONES

- CON\_REGION. Se le pasa un código de región y devuelve el nombre

- EXISTE\_REGION. Devuelve verdadero si la región existe. Se usa en los procedimientos y por tanto es PRIVADA, no debe aparecer en la especificación del paquete

```

create or replace package regioness
is
procedure ALTA_REGION(codigo regions.region_id%type, nombre_region regions.region_name%type);
procedure BAJA_REGION(codigo regions.region_id%type);
procedure MOD_REGION(codigo regions.region_id%type, nombre_region regions.region_name%type);
end;

create or replace package body regioness
is

function EXISTE_REGION(codigo regions.region_id%type,nombre regions.region_name%type) RETURN boolean
is
existe boolean;
verificadorFilaRegion regions%rowtype;
begin
select * into verificadorFilaRegion from regions where region_id=codigo or lower(nombre)=lower(region_name);
existe := true;
return existe;
exception
WHEN NO_DATA_FOUND THEN
existe := false;
return existe;
end;

function CON_REGION(codigo regions.region_id%type) RETURN VARCHAR2
is
nombreRegion varchar2(200):='';
existe boolean:=EXISTE_REGION(codigo,'');
begin
if existe then
select region_name into nombreRegion from regions where region_id=codigo;
return nombreRegion;
else
nombreRegion:='No se encontro region con ese codigo';
return nombreRegion;
end if;
end;

procedure ALTA_REGION(codigo regions.region_id%type, nombre_region regions.region_name%type) is
begin
if EXISTE_REGION(codigo,nombre_region) then
DBMS_OUTPUT.PUT_LINE('la region con ese nombre o codigo ya existe');
else
insert into regions values(codigo, nombre_region);
DBMS_OUTPUT.PUT_LINE('insertado con exito');
end if;
end;

```

```

procedure BAJA_REGION(codigo regions.region_id%type) is
begin
    if EXISTE_REGION(codigo, '') then
        delete from regions where region_id=codigo;
        DBMS_OUTPUT.PUT_LINE('borrado con exito');
    else
        DBMS_OUTPUT.PUT_LINE('No hay region con ese codigo');
    end if;
end;

```

```

procedure MOD_REGION(codigo regions.region_id%type, nombre_region regions.region_name%type) is
begin
    if EXISTE_REGION(codigo, '') then
        UPDATE REGIONS SET REGION_NAME=nombre_region where region_id=codigo;
        DBMS_OUTPUT.PUT_LINE('Modificado con exito');
    else
        DBMS_OUTPUT.PUT_LINE('No hay region con ese codigo');
    end if;
end;

```

```

--
regiones.ALTA_REGION('7', 'europe');

```

la region con ese nombre o codigo ya existe

Procedimiento PL/SQL terminado correctamente.

```

begin
    regiones.ALTA_REGION('6', 'KAZAJISTAN');

```

insertado con exito

Procedimiento PL/SQL terminado correctamente.

	REGION_ID	REGION_NAME
1	6	KAZAJISTAN

```

execute regioness.mod_region('6', 'region');

```

Modificado con exito

Procedimiento PL/SQL terminado correctamente.

	REGION_ID	REGION_NAME
1	6	region

```
execute regioness.baja_region('6');
```

borrado con exito

Procedimiento PL/SQL terminado correctamente.

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa
5	5	TTTT

Crear un paquete denominado NOMINA que tenga sobrecargado

la función CALCULAR\_NOMINA de la siguiente forma:

- CALCULAR\_NOMINA(NUMBER): se calcula el salario del empleado restando un 15% de IRPF.
- CALCULAR\_NOMINA(NUMBER,NUMBER): el segundo parámetro es el porcentaje a aplicar. Se calcula el salario del empleado restando ese porcentaje al salario
- CALCULAR\_NOMINA(NUMBER,NUMBER,CHAR): el segundo parámetro es el porcentaje a aplicar, el tercero vale 'V' . Se calcula el salario del empleado aumentando la comisión que le pertenece y restando ese porcentaje al salario siempre y cuando el empleado tenga comisión.

```

create or replace package nomina
is
function CALCULAR_NOMINA(id_empleado NUMBER) return number;
function CALCULAR_NOMINA(id_empleado NUMBER, porcentaje_descuento varchar) return number;
function CALCULAR_NOMINA(id_empleado NUMBER, porcentaje_descuento number, V varchar2:='V') return number;
end;

create or replace package body nomina
is
function calcular_nomina(id_empleado NUMBER) return number is
salario employees.salary$TYPE;
begin
select salary into salario from employees where employee_id=id_empleado;
salario:=salario*0.85;
return salario;
exception
when no_Data_found then
dbms_output.put_line('empleado no encontrado');
return -1;
end;

function calcular_nomina(id_empleado NUMBER, porcentaje_descuento varchar) return number is
salario employees.salary$TYPE;
begin
select salary into salario from employees where employee_id=id_empleado;
salario:=salario*(1-(porcentaje_descuento/100));
return salario;
exception
when no_Data_found then
dbms_output.put_line('empleado no encontrado');
return -1;
end;

function calcular_nomina(id_empleado NUMBER, porcentaje_descuento number, V varchar2:='V') return number is
filaEmpleado employees$ROWTYPE;
noCommission exception;
begin
select * into filaEmpleado from employees where employee_id=id_empleado;
if(filaEmpleado.COMMISSION_PCT is null) then
raise noCommission;
end if;
filaEmpleado.salary:=(filaEmpleado.salary+(filaEmpleado.salary*filaEmpleado.COMMISSION_PCT));
filaEmpleado.salary:=filaEmpleado.salary*(1-(porcentaje_descuento/100));
return filaEmpleado.salary;
exception
when no_Data_found then
dbms_output.put_line('empleado no encontrado');
return -1;
when noCommission then
dbms_output.put_line('empleado no tiene commission');
return -1;
end;

end;

begin
dbms_output.put_line(nomina.calcular_nomina(100));
dbms_output.put_line(' ');
dbms_output.put_line(nomina.calcular_nomina(100,'20'));
dbms_output.put_line(' ');
dbms_output.put_line(nomina.calcular_nomina(100,20,'v'));
end;

```

---

20400

19200

empleado no tiene commission

-1