

Prácticas ORACLE PL/SQL

Práctica de FUNCIONES

1. Crear una función que tenga como parámetro un número de departamento y que devuelve la suma de los salarios de dicho departamento. La imprimimos por pantalla.
 - Si el departamento no existe debemos generar una excepción con dicho mensaje
 - Si el departamento existe, pero no hay empleados dentro, también debemos generar una excepción para indicarlo
2. Modificar el programa anterior para incluir un parámetro de tipo OUT por el que vaya el número de empleados afectados por la query. Debe ser visualizada en el programa que llama a la función. De esta forma vemos que se puede usar este tipo de parámetros también en una función
3. Crear una función llamada CREAR_REGION,
 - A la función se le debe pasar como parámetro un nombre de región y debe devolver un número, que es el código de región que calculamos dentro de la función
 - Se debe crear una nueva fila con el nombre de esa REGION
 - El código de la región se debe calcular de forma automática. Para ello se debe averiguar cual es el código de región más alto que tenemos en la tabla en ese momento, le sumamos 1 y el resultado lo ponemos como el código para la nueva región que estamos creando.
 - Si tenemos algún problema debemos generar un error
 - La función debe devolver el número que ha asignado a la región

Soluciones

1. Crear una función que tenga como parámetro un número de departamento y que devuelve la suma de los salarios de dicho departamento. La imprimimos por pantalla.

- Si el departamento no existe debemos generar una excepción con dicho mensaje
- Si el departamento existe, pero no hay empleados dentro, también debemos generar una excepción para indicarlo
-

```

create or replace function slarios_dept (dep number) return number
is
salary number;

depart hr.departments.department_id%type;
num_emple number;

begin

select department_id into depart from hr.departments where
department_id=dep;

select count(*) into num_emple from hr.employees where
department_id=dep;
if dep>0 then

select sum(salary) into salary from hr.employees where
department_id=dep group by department_id;
else
raise_application_error (-20730,'El departamento existe, pero no hay
empleados ' || dep);
end if;

return salary;

```

```
exception
```

```
when no_data_found then raise_application_error (-20730,'No existe el departamento ' || dep);
```

```
end;
```

--Probar la función

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
SAL NUMBER;
```

```
DEPT NUMBER:=100;
```

```
BEGIN
```

```
SAL:=salarios_dept(DEPT);
```

```
DBMS_OUTPUT.PUT_LINE('El salario total del departamento ' || DEPT || ' es: ' || SAL);
```

```
END;
```

```
/
```

2. Modificar el programa anterior para incluir un parámetro de tipo OUT por el que vaya el número de empleados afectados por la query. Debe ser visualizada en el programa que llama a la función. De esta forma vemos que se puede usar este tipo de parámetros también en una función

```
CREATE OR REPLACE FUNCTION salarios_dept1(dep_id NUMBER, n_empleados OUT NUMBER) RETURN NUMBER
```

```
IS
```

```
sal NUMBER;
```

```
BEGIN
```

```
--COMPROBAR QUE EL DEPARTAMENTO EXISTE. SI NO EXISTE SE DISPARA LA EXCEPCIÓN
```

```
SELECT DEPARTMENT_ID INTO DEPT FROM DEPARTMENTS WHERE DEPARTMENT_ID=DEP_ID;
```

```
--SI EL DEPARTAMENTO EXISTE CALCULAR TOTALES
```

```
SELECT sum(salary),count(salary) INTO SAL,n_empleados from employees where department_id=dep_id group by department_id;
```

```
RETURN sal;
```

```
END;
```

```

/

--PROBAR LA FUNCIÓN
set serveroutput on
declare
TOTAL_SAL NUMBER;
NUM_EMPLE NUMBER;
DEPT NUMBER:=110;
BEGIN
TOTAL_SAL:=salarios_dept1(DEPT,NUM_EMPLE);
DBMS_OUTPUT.PUT_LINE('El salario total del departamento ' ||
DEPT || ' es: ' || TOTAL_SAL);
DBMS_OUTPUT.PUT_LINE('El número total de empleados recabados
es : ' || NUM_EMPLE);
END;

```

3. Crear una función llamada CREAR_REGION,

- A la función se le debe pasar como parámetro un nombre de región y debe devolver un número, que es el código de región que calculamos dentro de la función
- Se debe crear una nueva fila con el nombre de esa REGION
- El código de la región se debe calcular de forma automática. Para ello se debe averiguar cual es el código de región más alto que tenemos en la tabla en ese momento, le sumamos 1 y el resultado lo ponemos como el código para la nueva región que estamos creando.
- Si tenemos algún problema debemos generar un error
- La función debe devolver el número que ha asignado a la región

```

create or replace FUNCTION CREAR_REGION (nombre varchar2)
RETURN NUMBER IS
regiones NUMBER;
NOM_REGION VARCHAR2(100);
BEGIN
    --AVERIGUAR SI EXISTE LA REGIÓN. SI YA EXISTE DAMOS
    ERROR. SI NO EXISTE PASAMOS A EXCEPTION Y SEGUIMOS
    CON EL PROGRAMA

    SELECT REGION_NAME INTO NOM_REGION FROM REGIONS
    WHERE REGION_NAME=UPPER(NOMBRE);

    raise_application_error(-20321,'Esta región ya existe!');

```

EXCEPTION

-- SI LA REGION NO EXISTE LA INSERTAMOS. ES UN EJEMPLO DE COMO PODEMOS USAR LA EXCEPCION PARA HACER ALGO CORRECTO

```
    WHEN NO_DATA_FOUND THEN
        SELECT MAX(REGION_ID)+1 INTO REGIONES from REGIONS;
        INSERT INTO REGIONS (region_id,region_name) VALUES
        (regiones,upper(nombre));
        RETURN REGIONES;
    END;
/
```

--PROBAR LA FUNCIÓN

```
DECLARE
N_REGION NUMBER;
BEGIN
N_REGION:=crear_region('NORMANDIA');
DBMS_OUTPUT.PUT_LINE('EL      NUMERO      ASIGNADO
ES:'||N_REGION);
END;
/
```