

# Prácticas ORACLE PL/SQL

## Práctica de FUNCIONES

1. Crear una función que tenga como parámetro un número de departamento y que devuelva la suma de los salarios de dicho departamento. La imprimimos por pantalla.
  - Si el departamento no existe debemos generar una excepción con dicho mensaje

```
--practica de funciones 1
CREATE OR REPLACE FUNCTION CALCULO_DEPA(P_DEPA IN EMPLOYEES.DEPARTMENT_ID%TYPE) RETURN NUMBER
IS
SUMA_SALARY NUMBER;
BEGIN
    SELECT SUM(SALARY) INTO SUMA_SALARY FROM EMPLOYEES WHERE DEPARTMENT_ID = P_DEPA;
    IF SUMA_SALARY IS NULL THEN
        RAISE_APPLICATION_ERROR(-20002, 'EL DEPARTAMENTO NO EXISTE');
    END IF;
    RETURN SUMA_SALARY;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('NO HAY DATOS');
END;
```

```
set serveroutput on
DECLARE
A NUMBER;
begin
A := CALCULO_DEPA(10);
DBMS_OUTPUT.PUT_LINE('LA SUMA DEL SALARIO DEL DEPARTAMENTO ES ' || A );
end;
```

Salida de Script x Resultado de la Consulta x  
Tarea terminada en 0.092 segundos

```
A NUMBER;
begin
A := CALCULO_DEPA(99);
DBMS_OUTPUT.PUT_LINE('LA SUMA DEL SALARIO DEL DEPARTAMENTO ES ' || A );
end;
Informe de error -
ORA-20002: EL DEPARTAMENTO NO EXISTE
ORA-06512: en "HR.CALCULO_DEPA", línea 7
ORA-06512: en línea 4
```

```

set serveroutput on
DECLARE
  A NUMBER;
begin
  A := CALCULO_DEPA(10);
  DBMS_OUTPUT.PUT_LINE('LA SUMA DEL SALARIO DEL DEPARTAMENTO ES ' || A );
end;

```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0.08 segundos

LA SUMA DEL SALARIO DEL DEPARTAMENTO ES 4400

Procedimiento PL/SQL terminado correctamente.

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

```

--practica de funciones 1
CREATE OR REPLACE FUNCTION CALCULO_DEPA(P_DEPA IN EMPLOYEES.DEPARTMENT_ID%TYPE,P_CONTADOR OUT NUMBER ) RETURN NUMBER
IS
  SUMA_SALARY NUMBER;
  CONTADOR NUMBER;
BEGIN
  SELECT SUM(SALARY), COUNT(SALARY) INTO SUMA_SALARY,CONTADOR FROM EMPLOYEES WHERE DEPARTMENT_ID = P_DEPA;
  IF SUMA_SALARY IS NULL THEN
    RAISE_APPLICATION_ERROR(-20002,'EL DEPARTAMENTO NO EXISTE');
  END IF;
  P_CONTADOR := CONTADOR;
  RETURN SUMA_SALARY;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_line('NO HAY DATOS');
END;

```

```

set serveroutput on
DECLARE
  A NUMBER;
  VCONTADOR NUMBER;
begin
  A := CALCULO_DEPA(50,VCONTADOR);
  DBMS_OUTPUT.PUT_LINE('LA SUMA DEL SALARIO DEL DEPARTAMENTO ES ' || A );
  DBMS_OUTPUT.PUT_LINE('LA SUMA DE LOS SALARIOS PROVIENE DE ' || VCONTADOR || ' SALARIOS SUMADOS ');
end;

```

Procedimiento PL/SQL terminado correctamente.

LA SUMA DEL SALARIO DEL DEPARTAMENTO ES 206400

LA SUMA DE LOS SALARIOS PROVIENE DE 46 SALARIOS SUMADOS

Procedimiento PL/SQL terminado correctamente.

### 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 cuál 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 (P_NOMBRE IN VARCHAR2 ) RETURN NUMBER
IS
V_ID_REGION NUMBER ;
BEGIN
    SELECT COALESCE(MAX(REGION_ID), 0) + 1 INTO V_ID_REGION FROM REGIONS;
    INSERT INTO REGIONS (REGION_ID,REGION_NAME) VALUES (V_ID_REGION,P_NOMBRE );
    RETURN V_ID_REGION;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO INSERTAR: ' || SQLERRM);
END;
```

```
set serveroutput on
DECLARE
VNOMBRE VARCHAR(25);
begin
    VNOMBRE:= 'ASIA 5';
    DBMS_OUTPUT.PUT_LINE('LA NUEVA REGION TIENE UN ID = ' || CREAR_REGION(VNOMBRE));
end;
```

```
Procedimiento PL/SQL terminado correctamente.
LA NUEVA REGION TIENE UN ID = 5
```

```
Procedimiento PL/SQL terminado correctamente.
```

Recursos Humanos.sql Recursos Humanos1.sql PHPCRUD.sql Recursos Humanos REGIONS

Columnas Datos Model Restricciones Permisos Estadísticas Disparadores Flashback Dependencias Detalles Particiones Índices SQL

Ordenar... Filtrar:

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