
8.- CREACIÓN DE FUNCIONES

- 8.1.- Introducción.
- 8.2.- Ejecución de funciones.
- 8.3.- Más sobre funciones.
- 8.4.- Procedimientos y funciones.

8.1.- INTRODUCCIÓN

Una función es un bloque nombrado PL/SQL que devuelve un valor, solamente tiene parámetros de entrada. Está almacenada en la Base de Datos como un objeto de la misma para repetidas ejecuciones.

Sintaxis:

```
CREATE [ OR REPLACE] FUNCTION nb_función
    (argumento1 [modo1] tipo_dato1,
    argumento2 [modo2] tipo_dato2,
    ...)
RETURN tipo_de_dato
IS | AS
Bloque PL/SQL;
/
```

Donde:

ARGUMENTO: Nombre de la variable PL cuyo valor se pasa en la función.

TIPO_DATO: tipo de dato del parámetro.

RETURN TIPO_DE_DATO: Indica el tipo de dato que la función debe devolver.

MODO: siempre es IN y se puede omitir.

Creación de una función

Para crear una función es conveniente seguir los siguientes pasos:

1. Introducir el texto en el editor y guardarlo en un SCRIPT. SQL.
2. Desde SQL*Plus ejecutar el SCRIPT para compilar el código y que éste quede almacenado en la base de datos.
3. Llamar a la función cuando necesite ser ejecutada.

Si se crea una función dentro de un bloque PL no se escribirá la cláusula CREATE, únicamente FUNCTION nb_función.

Ejemplo:

Crear una función que devuelva el salario de un empleado cuyo código es introducido como parámetro. La función se llamará GET_SAL;

```
CREATE OR REPLACE FUNCTION GET_SAL
(V_ID EMP.EMPNO%TYPE)
RETURN NUMBER
IS
V_SALARY EMP.SAL%TYPE := 0;
BEGIN
    SELECT SAL INTO V_SALARY FROM EMP WHERE EMPNO= V_ID;
    RETURN V_SALARY;
END GET_SAL;
/
```

8.2.- EJECUCIÓN DE FUNCIONES

A.- Mediante una variable HOST:

1. Definir una variable host: VARIABLE nb_variable tipo_dato;
2. Ejecutar de la siguiente forma:
EXECUTE :nb_variable := nb_función (valor_de_entrada);
3. Imprimir la variable host: PRINT nb_variable.

B.- Invocarla desde un bloque PL, cargándola en una variable.

<VARIABLE> := nb_funcion (parámetro)

Ejemplo 1:

Diseñar una función que devuelva el número de años comprendidos entre dos fechas, las cuales son introducidas como parámetros.

```
CREATE OR REPLACE FUNCTION FUNC_FECHA (FECHA1 DATE, FECHA2 DATE)
RETURN NUMBER
AS
    V_AÑOS NUMBER(6);
BEGIN
    V_AÑOS := ABS(TRUNC(MONTHS_BETWEEN(FECHA2,FECHA1)/12));
    RETURN (V_AÑOS);
END FUNC_FECHA;
/
```

Ejemplo 2:

Crear un bloque anónimo que llama a la función y devuelva su valor.

```
DECLARE
    V_RES NUMBER;
BEGIN
    V_RES:= FUNC_FECHA ('03/12/95', '08/08/99');
```

```
DBMS_OUTPUT.PUT_LINE(V_RES);  
END;  
/
```

Una función de usuario puede ser invocada desde una orden SQL:

- Como columna de una SELECT.
- En las condiciones WHERE y HAVING.
- En las cláusulas ORDER BY y GROUP BY.
- Cláusula VALUES en el comando INSERT.
- En la cláusula SET del comando UPDATE.

8.3.- MÁS SOBRE FUNCIONES

Para borrar funciones se utiliza la siguiente orden:

```
DROP FUNCTION nb_funcion;
```

Las funciones las encontramos en la tabla USER_OBJECTS del Diccionario de Datos.

```
SELECT OBJECT_NAME, OBJECT_TYPE, STATUS  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE = 'FUNCTION';
```

La tabla USER_SOURCE del Diccionario de Datos, permite ver el código fuente de las funciones, mediante la siguiente sentencia:

```
SELECT LINE, SUBSTR (TEXT,1,60)  
FROM USER_SOURCE  
WHERE NAME = 'NB_FUNCION';
```

Para volver a compilar una función utilizamos la orden ALTER:

```
ALTER {PROCEDURE|FUNCTION} nb_subprograma COMPILE;
```

Esto vale tanto para procedimientos como para funciones.

8.4.- PROCEDIMIENTOS Y FUNCIONES

Se crea un procedimiento para almacenar una serie de acciones para ejecuciones posteriores, pudiendo contener cero o más parámetros, que pueden transferirse desde el entorno de llamada, no teniendo que devolver un valor.

Se crea una función cuando se quiere calcular un valor, el cual, debe ser devuelto al entorno de llamada. Una función puede contener varios parámetros de entrada pero una sola salida al programa principal.

Los beneficios de los procedimientos y las funciones son:

1. Mejorar el rendimiento.
2. Mejorar el mantenimiento.
3. Mejorar la seguridad e integridad de los datos.