PL/SQL Functions

PL/SQL functions block create using CREATE FUNCTION statement. The major difference between PL/SQL function or procedure, function return always value where as procedure may or may not return value.

When you create a function or procedure, you have to define IN/OUT/INOUT parameters parameters.

1. IN : IN parameter referring to the procedure or function and allow to overwritten the value of parameter.
2. OUT : OUT parameter referring to the procedure or function and allow to overwritten the value of parameter.
3. IN OUT : Both IN OUT parameter referring to the procedure or function to pass both IN OUT parameter, modify/update by the function or procedure and also get returned.

IN/OUT/INOUT parameters you define in function argument list that get returned back to a result. When you create the function default IN parameter is passed in argument list. It's means value is passed but not returned. Explicitly you have define OUT/IN OUT parameter in argument list.

#### **PL/SQL Functions Syntax**

CREATE [OR REPLACE] FUNCTION [SCHEMA..] function\_name

[ (parameter [,parameter]) ]

RETURN return\_datatype

IS | AS

[declaration\_section

variable declarations;

constant declarations;

]

BEGIN

[executable\_section

PL/SQL execute/subprogram body

]

[EXCEPTION]

[exception\_section

PL/SQL Exception block

]

END [function\_name];

/

### Function Example

In this example we are creating a function to pass employee number and get that employee name from table. We have emp1 table having employee information,

|  |  |  |  |
| --- | --- | --- | --- |
| EMP\_NO | EMP\_NAME | EMP\_DEPT | EMP\_SALARY |
| 1 | Forbs ross | Web Developer | 45k |
| 2 | marks jems | Program Developer | 38k |
| 3 | Saulin | Program Developer | 34k |
| 4 | Zenia Sroll | Web Developer | 42k |

#### **Create Function**

So lets start passing IN parameter (no). Return datatype set varchar2. Now inside function SELECT ... INTO statement to get the employee name.

*fun1.sql*

SQL>edit fun1

CREATE or REPLACE FUNCTION fun1(no in number)

RETURN varchar2

IS

name varchar2(20);

BEGIN

select ename into name from emp1 where eno = no;

return name;

END;

/

#### **Execute Function**

After write the PL/SQL function you need to execute the function.

**SQL>@fun1**  
Function created.  
  
PL/SQL procedure successfully completed.

#### **PL/SQL Program to Calling Function**

This program call the above define function with pass employee number and get that employee name.

*fun.sql*

SQL>edit fun

DECLARE

no number :=&no;

name varchar2(20);

BEGIN

name := fun1(no);

dbms\_output.put\_line('Name:'||' '||name);

end;

/

#### **PL/SQL Program Result**

**SQL>@fun**  
no number &n=2  
Name: marks jems  
  
PL/SQL procedure successfully completed.

### PL/SQL Drop Function

You can drop PL/SQL function using DROP FUNCTION statements.

#### **Functions Drop Syntax**

DROP FUNCTION function\_name;

#### **Functions Drop Example**

**SQL>DROP FUNCTION fun1;**  
  
Function dropped.