PL/SQL Procedures

PL/SQL procedures create using CREATE PROCEDURE 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 procedure argument list that get returned back to a result. When you create the procedure 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 Procedure Syntax**

CREATE [OR REPLACE] PROCEDURE [SCHEMA..] procedure\_name

[ (parameter [,parameter]) ]

IS

[declaration\_section

variable declarations;

constant declarations;

]

BEGIN

[executable\_section

PL/SQL execute/subprogram body

]

[EXCEPTION]

[exception\_section

PL/SQL Exception block

]

END [procedure\_name];

/

### PL/SQL Procedure Example

In this example we are creating a procedure to pass employee number argument and get that employee information 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 PROCEDURE**

In this example passing IN parameter (no) and inside procedure SELECT ... INTO statement to get the employee information.

*pro1.sql*

SQL>dit pro1

CREATE or REPLACE PROCEDURE pro1(no in number,temp out emp1%rowtype)

IS

BEGIN

SELECT \* INTO temp FROM emp1 WHERE eno = no;

END;

/

#### **Execute PROCEDURE**

After write the PL/SQL Procedure you need to execute the procedure.

**SQL>@pro1**  
Procedure created.  
  
PL/SQL procedure successfully completed.

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

This program (pro) call the above define procedure with pass employee number and get that employee information.

*pro.sql*

SQL>edit pro

DECLARE

temp emp1%rowtype;

no number :=&no;

BEGIN

pro1(no,temp);

dbms\_output.put\_line(temp.eno||' '||

temp.ename||' '||

temp.edept||' '||

temp.esalary||' '||);

END;

/

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

**SQL>@pro**  
no number &n=2  
2    marks jems    Program Developer    38K  
  
PL/SQL procedure successfully completed.

### PL/SQL Drop Procedure

You can drop PL/SQL procedure using DROP PROCEDURE statement,

#### **Functions Drop Syntax**

DROP PROCEDURE procedure\_name;

#### **Procedure Drop Example**

**SQL>DROP PROCEDURE pro1;**  
  
Procedure dropped.