**Procedures**

•A **subprogram** is a program unit/module that performs a particular task. These subprograms are combined to form larger programs.

PL/SQL subprograms are named PL/SQL blocks that can be invoked with a set of parameters. PL/SQL provides two kinds of subprograms −

* **Functions** − These subprograms return a single value; mainly used to compute and return a value.
* **Procedures** − These subprograms do not return a value directly; mainly used to perform an action.

Creating a Procedure

A procedure is created with the **CREATE OR REPLACE PROCEDURE** statement. The simplified syntax for the CREATE OR REPLACE PROCEDURE statement is as follows −

CREATE [OR REPLACE] PROCEDURE procedure\_name

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

{IS | AS}

BEGIN

< procedure\_body >

END procedure\_name;

**IN**

An IN parameter lets you pass a value to the subprogram. **It is a read-only parameter**.

**OUT**

An OUT parameter returns a value to the calling program.

**IN OUT**

An **IN OUT** parameter passes an initial value to a subprogram and returns an updated value to the caller.

IS/AS keyword:

Keyword ‘IS’ will be used when the stored procedure is nested into some other blocks.

Keyword ‘AS’ is used when the procedure is standalone.

**EXAMPLE 1:- PRINT HELLO WORLD**

CREATE OR REPLACE PROCEDURE greetings1

AS

BEGIN

dbms\_output.put\_line('Hello World!');

END;

/

## Executing a Standalone Procedure

A standalone procedure can be called in two ways −

* Using the **EXECUTE/exec** keyword

EXECUTE greetings;

* Calling the name of the procedure from a PL/SQL block

BEGIN

Greetings;

END;

/

## Deleting a Standalone Procedure

DROP PROCEDURE procedure-name;

PROGRAM:-

WAP THAT finds the minimum of two values. Here, the procedure takes two numbers using the IN mode and returns their minimum using the OUT parameters.

DECLARE

a number;

b number;

c number;

PROCEDURE findMin1(x IN number, y IN number, z OUT number) IS

BEGIN

IF x < y THEN

z:= x;

ELSE

z:= y;

END IF;

END findMin1;

BEGIN

a:= 23;

b:= 45;

findMin1(a, b, c);

dbms\_output.put\_line(' Minimum of (23, 45) : ' || c);

END;

/

OTHER WAY :-

CREATE OR REPLACE PROCEDURE findMin1(x IN number, y IN number, z OUT number) IS

BEGIN

IF x < y THEN

z:= x;

ELSE

z:= y;

END IF;

END;

DECLARE

a number;

b number;

c number;

BEGIN

a:= 23;

b:= 45;

findMin1(a, b, c);

dbms\_output.put\_line(' Minimum of (23, 45) : ' || c);

END;

/

EXAMPLE 2:-

WAP TO CREATE procedure to compute the square of value of a passed value.

DECLARE

a number;

PROCEDURE squareNum(x IN OUT number) IS

BEGIN

x := x \* x;

END;

BEGIN

a:= 23;

squareNum(a);

dbms\_output.put\_line(' Square of (23): ' || a);

END;

/