Practical – 4

Stored Procedure

A stored procedure or in simple a proc is a named PL/SQL block which performs one or more specific task.

A procedure has a header and a body. The header consists of the name of the procedure and the parameters or variables passed to the procedure. The body consists or declaration section, execution section and exception section similar to a general PL/SQL Block.

Procedures do not return a value directly, mainly used to perform an action

We can pass parameters to procedures in three ways.

1) IN-parameters **:**

The parameter can be referenced by the procedure or function. The value of the parameter can not be **overwritten** by the procedure or function.

2) OUT-parameters **:**

An OUT parameter **returns a value** to the calling program. Inside the subprogram, an OUT parameter acts like a variable. You can change its value and reference the value after assigning it. The actual parameter must be variable and it is **passed by value**.

3) IN OUT-parameters **:**

IN OUT parameter passes an initial value to a subprogram and returns an updated value to the caller. It can be assigned a value and its value can be read. The actual parameter corresponding to an IN OUT formal parameter must be a variable, not a constant or an expression.

Syntax :

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...] ) ]
{IS | AS}
BEGIN
  < procedure_body >
END procedure_name;
```

Example

```
CREATE OR REPLACE PROCEDURE first_proc
AS
BEGIN
   dbms_output.put_line('Hello World!');
END;
/
```

Creating standalone procedure

- **Executing a Standalone Procedure**
  1. Using the EXECUTE keyword
  2. Calling the name of the procedure from a PL/SQL block

Syntax : **execute** <procedure_name>

Example

```
Sql> execute first_proc;
                    OR
Sql> begin
       first_proc;
     End;
      /
```

- **Declare Procedure within PL/SQL block**

  Procedure declare in pl/sql block will not allocate space in database. Its life is till the pl/sql block execute.

Example :

```
Sql> declare
       procedure first_proc is
     begin
        dbms_output.put_line('hello');
     end;

     begin
        first_proc;
     end;
     /
     Hello
     PL/SQL procedure successfully completed.
```

- **IN & OUT Mode**

Example:

```
 DECLARE
    a number(3);
    b number(3);
    c number(3);
PROCEDURE findMin (x IN number, y IN number, z OUT number) IS
  BEGIN
     IF x < y THEN
        z:= x;
     ELSE
        z:= y;
     END IF;
END;

BEGIN
  a:= &n1;
  b:= &n2;
  findMin(a, b, c);
  dbms_output.put_line(' Minimum  : ' || c);
END;
```

- **IN OUT prameters**

```
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;
/
```

- **Delete standalone procedure**

Syntax :

Drop procedure <procedure_name>

Example

```
Sql> create or replace procedure insert
        is
      begin
        insert into tmp values(n);
      end;
SQL> execute insert_proc(243);
243
```

- **Creating optional parameters**

```
SQL>create or replace procedure mul_proc(A in number , b in number :=10)
        as
        begin
          dbms_output.put_line(a*b);
        end;
          /
Sql> execute mul_proc(10);
```

Exercise

1	Create a procedure which find the factorial of given number.(use pl/sql block)
2	Write a pl/block which insert,delete ,and update the record of given empno. Create a procedure for insert record (pass values as a parameters). Create a procedure for delete record(pass empno as a parameter) Procedure for update the name of employee (pass empno and new value)
3	Write a procedure which find maximum from three given number.
4	Write a procedure which calculate area of shapes. variable Area should be out type parameter. Shape name and required values passed as a parameters
5	Write PL/sql block which takes the student roll number and mark of three subjects as a input. Make a procedure which calculate percentage and find class of student. If percentage >=75 -> Disctiction class If percentage >=60 and <75 First class If percentage >=50 and <60 Second class If percentage >=35 and <50 Pass class If percentage <35 Fail