NAME: Atharva Chavan
TE-A IT
ROLL NO: T1851010
PRN: 71901316L

Group B: SQL & PL/SQL

<u>Assignment No. 9</u>

Aim: Write and execute PL/SQL stored procedure and function to perform a suitable task on the database. Demonstrate its use.

Objective:

- To study and implement PL/SQLprocedure.
- To study and implement PL/SQLfunction.

Theory :

A **subprogram** is a program unit/module that performs a particular task. These subprograms are combined to form larger programs. This is basically called the 'Modular design'. A subprogram can be invoked by another subprogram or program which is called the **calling program**.

A subprogram can be created −

- At the schemalevel
- Inside apackage
- Inside a PL/SQLblock

At the schema level, subprogram is a **standalone subprogram**. It is created with the CREATE PROCEDURE or the CREATE FUNCTION statement. It is stored in the database and can be deleted with the DROP PROCEDURE or DROP FUNCTION statement.

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 avalue.
- **Procedures** − These subprograms do not return a value directly; mainly used to perform an action.

Parts of a PL/SQL Subprogram

Each PL/SQL subprogram has a name, and may also have a parameter list. Like anonymous PL/SQL blocks, the named blocks will also have the following three parts −

1	**Executable Part** This is a mandatory part and contains statements that perform the designated action.
2	**Exception-handling** This is again an optional part. It contains the code that handles run-time errors.

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>
```

Where,

- *procedure-name* specifies the name of theprocedure.
- [OR REPLACE] option allows the modification of an existingprocedure.
- The optional parameter list contains name, mode and types of the parameters. **IN** represents the value that will be passed from outside and OUT represents the parameter that will be used to return a value outside of theprocedure.
- *procedure-body* contains the executablepart.
- The AS keyword is used instead of the IS keyword for creating a standalone procedure.

Example

The following example creates a simple procedure that displays the string 'Hello World!' on the screen when executed.

```
CREATE OR REPLACE PROCEDURE greetings
```

```
AS
BEGIN
   dbms_output.put_line('Hello World!');
END;
/
```

Deleting a Standalone Procedure

A standalone procedure is deleted with the **DROP PROCEDURE** statement. Syntax for deleting a procedure is −

```
DROP PROCEDURE procedure-name;
```

You can drop the greetings procedure by using the following statement −

```
DROP PROCEDURE greetings;
```

Parameter Modes in PL/SQL Subprograms

The following table lists out the parameter modes in PL/SQL subprograms −

S.No	Parameter Mode & Description
1	**IN** An IN parameter lets you pass a value to the subprogram. **It is a read-only parameter**. Inside the subprogram, an IN parameter acts like a constant. It cannot be assigned a value. You can pass a constant, literal, initialized variable, or expression as an IN parameter. You can also initialize it to a default value; however, in that case, it is omitted from the subprogram call. **It is the default mode of parameter passing. Parameters are passed byreference**.
2	**OUT** 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** An **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 the value can be read. The actual parameter corresponding to an IN OUT formal parameter must be a

	variable, not a constant or an expression. Formal parameter must be assigned a value. **Actual parameter is passed byvalue.**

Functions:

A function is same as a procedure except that it returns a value. Therefore, all the discussions of the previous chapter are true for functionstoo.

Creating a Function

A standalone function is created using the **CREATE FUNCTION** statement. The simplified syntax for the **CREATE OR REPLACE PROCEDURE** statement is as follows −

```
CREATE [OR REPLACE] FUNCTION function_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
RETURN return_datatype
{IS | AS}
BEGIN
  <function_body>
END [function_name];
```

Where,

- *function-name* specifies the name of thefunction.
- [OR REPLACE] option allows the modification of an existingfunction.
- The optional parameter list contains name, mode and types of the parameters. IN represents the value that will be passed from outside and OUT represents the parameter that will be used to return a value outside of theprocedure.
- The function must contain a **return**statement.
- The *RETURN* clause specifies the data type you are going to return from thefunction.
- *function-body* contains the executablepart.
- The AS keyword is used instead of the IS keyword for creating a standalonefunction.

Conclusion:-

We have studied and executed PL/SQL stored procedures and functions.

Code & Output: -

```
Atharva@BRAINMETRON:~$ sudomysql -u root #

mysql> use Atharva;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> delimiter #
mysql> show tables#

+-------------------+
| Tables_in_Atharva |
+-------------------+
| department        |
| employee          |
| job               |
| location          |
+-------------------+
4 rows in set (0.00 sec)

mysql> delimiter #
mysql> show tables #
+-------------------+
| Tables_in_Atharva |
+-------------------+
| department        |
| employee          |
| job               |
| location          |
+-------------------+
4 rows in set (0.00 sec)

mysql> create procedure print()
    -> begin
    -> select 'My SQL procedure' as output;
    -> end;
    -> #
Query OK, 0 rows affected (0.07 sec)

mysql> print()#
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'print()' at line 1
mysql> call print()#



+------------------+
| output           |
+------------------+
| My SQL procedure |
+------------------+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.04 sec)

mysql> create procedure update1 ()
    -> begin
    -> update employee set salary=salary+10000;
    -> end;
    -> #

Query OK, 0 rows affected (0.01 sec)

mysql> select * from employee;
    -> #
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
| employee_id | lastname | firstname | middlename | job_id | manager_id | hiredate   | salary |
department_id |
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
|        7499 | allen    | kevin     | J.         |    670 |       7698 | 2085-02-20 |   1600 |            20 |
|        7499 | allen    | kevin     | J.         |    670 |       7698 | 2085-02-20 |   1600 |            20 |
|        7505 | doyle    | jean      | K.         |    671 |       7839 | 2085-04-04 |   2850 |            20 |
|        7506 | dennis   | lynn      | S.         |    671 |       7839 | 2085-05-15 |   2750 |            30 |
|        7507 | baker    | leslie    | D.         |    671 |       7839 | 2085-06-10 |   2200 |            40 |
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
5 rows in set (0.00 sec)

mysql> call update1()#
Query OK, 5 rows affected (0.09 sec)

mysql> select * from employee#





+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
| employee_id | lastname | firstname | middlename | job_id | manager_id | hiredate   | salary |
department_id |
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
|        7499 | allen    | kevin     | J.         |    670 |       7698 | 2085-02-20 |  11600 |            20 |
|        7499 | allen    | kevin     | J.         |    670 |       7698 | 2085-02-20 |  11600 |            20 |
|        7505 | doyle    | jean      | K.         |    671 |       7839 | 2085-04-04 |  12850 |            20 |
|        7506 | dennis   | lynn      | S.         |    671 |       7839 | 2085-05-15 |  12750 |            30 |
|        7507 | baker    | leslie    | D.         |    671 |       7839 | 2085-06-10 |  12200 |            40 |
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
5 rows in set (0.00 sec)

mysql> select firstname from employee;
    -> #
+-----------+
| firstname |
+-----------+
| kevin     |
| kevin     |
| jean      |
| lynn      |
| leslie    |
+-----------+
5 rows in set (0.00 sec)

mysql> delimiter -
mysql> select firstname from employee-
+-----------+
| firstname |
+-----------+
| kevin     |
| kevin     |
| jean      |
| lynn      |
| leslie    |
+-----------+
5 rows in set (0.00 sec)

mysql> create procedure pro1(n int)
    -> begin
    -> set @num=n;
    -> end;
    -> -
Query OK, 0 rows affected (0.00 sec)

mysql> call pro1(10)-
Query OK, 0 rows affected (0.00 sec)

mysql> select @num-
+------+
| @num |
+------+
|   10 |
+------+
1 row in set (0.00 sec)

mysql> create procedure pro2(out n int) begin set n=50; end;-
Query OK, 0 rows affected (0.00 sec)

mysql> select @n-
+------+
| @n   |
+------+
| NULL |
+------+
1 row in set (0.00 sec)

mysql> create procedure pro3(out n int) begin set @n=50; end;-
```

```
Query OK, 0 rows affected (0.00 sec)

mysql> call pro3(@10)-
Query OK, 0 rows affected (0.00 sec)

mysql> select @n-




+------+
| @n   |
+------+
|   50 |
+------+
1 row in set (0.00 sec)

mysql> call pro2(@10)-
Query OK, 0 rows affected (0.00 sec)

mysql> select @n-mysql>
mysql> CREATE TABLE products (prod_id INT NOT NULL AUTO_INCREMENT, prod_name
VARCHAR(20) NOT NULL, prod_cost FLOAT NOT NULL DEFAULT 0.0, prod_price FLOAT NOT
NULL DEFAULT 0.0, PRIMARY KEY(prod_id));
    -> -
Query OK, 0 rows affected (0.43 sec)

mysql> INSERT INTO products (prod_name, prod_cost, prod_price) VALUES ('Basic
Widget',5.95,8.35),('Micro Widget',0.95,1.35),('Mega Widget',99.95,140.00);-
Query OK, 3 rows affected (0.08 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from product-
ERROR 1146 (42S02): Table 'Atharva.product' doesn't exist
mysql> select * from products-
+---------+--------------+-----------+------------+
| prod_id | prod_name    | prod_cost | prod_price |
+---------+--------------+-----------+------------+
|       1 | Basic Widget |      5.95 |       8.35 |
|       2 | Micro Widget |      0.95 |       1.35 |
|       3 | Mega Widget  |     99.95 |        140 |
+---------+--------------+-----------+------------+
3 rows in set (0.00 sec)

mysql> INSERT INTO products VALUES ('femto Widget',6.95,10.35)-
ERROR 1136 (21S01): Column count doesn't match value count at row 1
mysql> DELIMITER $$
mysql> CREATE FUNCTION calcProfit(cost FLOAT, price FLOAT) RETURNS DECIMAL(9,2)
    -> BEGIN

    ->   DECLARE profit DECIMAL(9,2);
    ->   SET profit = price-cost;
    ->   RETURN profit;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> SELECT *, calcProfit(prod_cost,prod_price) AS profit FROM products;
+---------+--------------+-----------+------------+--------+
| prod_id | prod_name    | prod_cost | prod_price | profit |
+---------+--------------+-----------+------------+--------+
|       1 | Basic Widget |      5.95 |       8.35 |   2.40 |
|       2 | Micro Widget |      0.95 |       1.35 |   0.40 |
|       3 | Mega Widget  |     99.95 |        140 |  40.05 |
+---------+--------------+-----------+------------+--------+
3 rows in set (0.03 sec)

mysql> DELIMITER $$
mysql> CREATE PROCEDURE procedureTest()
    -> BEGIN
    ->   SELECT prod_name FROM products;
    -> END$$
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> CALL procedureTest() ;
+--------------+
| prod_name    |
+--------------+
| Basic Widget |
| Micro Widget |
| Mega Widget  |
+--------------+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> select * from employee;

   -> /
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
| employee_id | lastname | firstname | middlename | job_id | manager_id | hiredate   | salary |
department_id |
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
|        7499 | allen    | kevin     | J.         |    670 |       7698 | 2085-02-20 |  11600 |            20 |
|        7499 | allen    | kevin     | J.         |    670 |       7698 | 2085-02-20 |  11600 |            20 |
|        7505 | doyle    | jean      | K.         |    671 |       7839 | 2085-04-04 |  12850 |            20 |
|        7506 | dennis   | lynn      | S.         |    671 |       7839 | 2085-05-15 |  12750 |            30 |
|        7507 | baker    | leslie    | D.         |    671 |       7839 | 2085-06-10 |  12200 |            40 |
+-------------+----------+-----------+------------+--------+------------+------------+--------+---------------+
5 rows in set (0.00 sec)

mysql> create procedure salout(in eid int, out sal int)
    ->begin
    ->select salary into sal from employee  where employee_id=eid;
    ->end;
    ->/
Query OK, 0 rows affected (0.33 sec)

mysql> call salout(7505, @salary)/
Query OK, 1 row affected (0.00 sec)

mysql> select @salary as sal/
+-------+
| sal   |
+-------+
| 12850 |
+-------+
1 row in set (0.00 sec)

mysql> alter table employee drop manager_id /
Query OK, 0 rows affected (1.31 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from employee;
    -> /



+-------------+----------+-----------+------------+--------+------------+--------+---------------+
| employee_id | lastname | firstname | middlename | job_id | hiredate   | salary | department_id |
+-------------+----------+-----------+------------+--------+------------+--------+---------------+
|        7499 | allen    | kevin     | J.         |    670 | 2085-02-20 |  11600 |            20 |
|        7499 | allen    | kevin     | J.         |    670 | 2085-02-20 |  11600 |            20 |
|        7505 | doyle    | jean      | K.         |    671 | 2085-04-04 |  12850 |            20 |
|        7506 | dennis   | lynn      | S.         |    671 | 2085-05-15 |  12750 |            30 |
|        7507 | baker    | leslie    | D.         |    671 | 2085-06-10 |  12200 |            40 |
+-------------+----------+-----------+------------+--------+------------+--------+---------------+
5 rows in set (0.00 sec)

mysql> alter table employee drop lastname /
Query OK, 0 rows affected (1.22 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table employee drop middlename /
Query OK, 0 rows affected (0.85 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from employee;
    -> /
+-------------+-----------+--------+------------+--------+---------------+
| employee_id | firstname | job_id | hiredate   | salary | department_id |
+-------------+-----------+--------+------------+--------+---------------+
|        7499 | kevin     |    670 | 2085-02-20 |  11600 |            20 |
|        7499 | kevin     |    670 | 2085-02-20 |  11600 |            20 |
|        7505 | jean      |    671 | 2085-04-04 |  12850 |            20 |
|        7506 | lynn      |    671 | 2085-05-15 |  12750 |            30 |
|        7507 | leslie    |    671 | 2085-06-10 |  12200 |            40 |
+-------------+-----------+--------+------------+--------+--------------+
5 rows in set (0.00 sec)

mysql> alter table employee add (status varchar(30)) /
Query OK, 0 rows affected (0.93 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create procedure stat ()
    -> begin
    -> update employee set status='platinum' where sal>12800;
    -> update employee set status='platinum' where sal<12800 and sal>11700;
    -> update employee set status='platinum' where sal<11700;
    -> end;
    -> /
Query OK, 0 rows affected (0.00 sec)

mysql> call stat/
ERROR 1054 (42S22): Unknown column 'sal' in 'where clause'
mysql> drop procedure stat;
    -> /
Query OK, 0 rows affected (0.01 sec)

mysql> create procedure stat () begin update employee set status='platinum' where sal>12800; update
employee set status='platinum' where sal<12800 and sal>11700; update employee set status='platinum'
where sal<11700; end;/
Query OK, 0 rows affected (0.00 sec)

mysql> drop procedure stat;
Query OK, 0 rows affected (0.00 sec)

mysql> create procedure stat () begin update employee set status='platinum' where salary>12800; update
employee set status='gold' where salary<12800 and salary>11700; update employee set status='silver'
where salary<11700; end;/
Query OK, 0 rows affected (0.00 sec)

mysql> call stat/
Query OK, 2 rows affected (0.18 sec)

mysql> select * from employee;
    -> /
```

```
+-------------+-----------+--------+------------+--------+---------------+----------+
| employee_id | firstname | job_id | hiredate   | salary | department_id | status   |
+-------------+-----------+--------+------------+--------+---------------+----------+
|        7499 | kevin     |    670 | 2085-02-20 |  11600 |            20 | silver   |
|        7499 | kevin     |    670 | 2085-02-20 |  11600 |            20 | silver   |
|        7505 | jean      |    671 | 2085-04-04 |  12850 |            20 | platinum |
|        7506 | lynn      |    671 | 2085-05-15 |  12750 |            30 | gold     |
|        7507 | leslie    |    671 | 2085-06-10 |  12200 |            40 | gold     |
+-------------+-----------+--------+------------+--------+---------------+----------+
5 rows in set (0.00 sec)

mysql> create function ct()
    -> returns int
    -> begin
    -> declare c int;
    -> select count(employee_id) into c from employee;
    -> return c;
    -> end;
    -> /
Query OK, 0 rows affected (0.00 sec)

mysql> select ct() as 'employee count'/
+----------------+
| employee count |
+----------------+
|              5 |
+----------------+
1 row in set (0.00 sec)

mysql> delimiter #

mysql> create function avg1(c int,d int, e int) returns int
     ->begin
     ->declare sum,a int;
     ->select c+d+e into sum;
     ->select sum/3 into a;
     ->return a;
     ->end;
     ->#
```

```
Query OK, 0 rows affected (0.00 sec)

mysql> select avg1(5,4,3)#
+-------------+
| avg1(5,4,3) |
+-------------+
|           4 |
+-------------+
1 row in set (0.00 sec)

mysql> create table student(roll int, name varchar(30), marks int);
    -> #
Query OK, 0 rows affected (0.39 sec)

mysql> delimiter ;
mysql> insert into student values(1,'ram',1700);
Query OK, 1 row affected (0.34 sec)

mysql> insert into student values(2,'radha',1000);
Query OK, 1 row affected (0.07 sec)

mysql> insert into student values(3,'raj',500);
Query OK, 1 row affected (0.07 sec)

mysql> insert into student values(4,'raj',950);
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values(5,'raja',940);
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values(5,'rinku',925);
Query OK, 1 row affected (0.06 sec)

mysql> insert into student values(5,'rinki',725);
Query OK, 1 row affected (0.06 sec)

mysql> select * from student;
+------+-------+-------+
| roll | name  | marks |
+------+-------+-------+
|    1 | ram   |  1700 |
|    2 | radha |  1000 |
|    3 | raj   |   500 |
|    4 | raj   |   950 |
|    5 | raja  |   940 |
|    5 | rinku |   925 |
|    5 | rinki |   725 |
+------+-------+-------+
7 rows in set (0.00 sec)

mysql> alter table student add (status varchar(20));
Query OK, 0 rows affected (0.85 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> delimiter #
mysql> create procedure stating ()
    -> begin
    -> update employee set status='platinum' where salary>12800;
    -> update employee set status='gold' where salary<12800 and salary>11700;
    -> update employee set status='silver' where salary<11700;
    -> update employee set status='silver' where salary<11700qddXAC;
    -> #
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near '' at line 6
mysql> create procedure stating ()
    -> begin
    -> update student set status='DISTINCTION' where marks<=1500 and marks>=990;
    -> update student set status='FIRST CLASS' where marks<=989 and marks>=900;
    -> update student set status='HIGHER SECOND CLASS' where marks<=899 and marks>=825;
    -> end;
    -> #
Query OK, 0 rows affected (0.00 sec)

mysql> select * from student#
+------+-------+-------+--------+
| roll | name  | marks | status |
+------+-------+-------+--------+
|    1 | ram   |  1700 | NULL   |
|    2 | radha |  1000 | NULL   |
|    3 | raj   |   500 | NULL   |
|    4 | raj   |   950 | NULL   |
|    5 | raja  |   940 | NULL   |
|    5 | rinku |   925 | NULL   |
|    5 | rinki |   725 | NULL   |
+------+-------+-------+--------+
7 rows in set (0.00 sec)

mysql> call stating()#
Query OK, 0 rows affected (0.11 sec)

mysql> select * from student#
+------+-------+-------+-------------+
| roll | name  | marks | status      |
+------+-------+-------+-------------+
|    1 | ram   |  1700 | NULL        |
|    2 | radha |  1000 | DISTINCTION |
|    3 | raj   |   500 | NULL        |
|    4 | raj   |   950 | FIRST CLASS |
|    5 | raja  |   940 | FIRST CLASS |
|    5 | rinku |   925 | FIRST CLASS |
|    5 | rinki |   725 | NULL        |
+------+-------+-------+-------------+
7 rows in set (0.00 sec)

mysql> create procedure stating1()
    ->begin
    ->update student set status='DISTINCTION' where marks>=990;
    ->update student set status='FIRST CLASS' where marks<=989 and marks>=900;
    ->update student set status='HIGHER SECOND CLASS' where marks<=899 and marks>=825;
    ->update student set status='FAIL' where marks<=825;
    ->end;
    ->#
Query OK, 0 rows affected (0.00 sec)

mysql> call stating1();
    -> #
Query OK, 2 rows affected (0.17 sec)
```

```
mysql> select * from student#
+------+-------+-------+-------------+
| roll | name  | marks | status      |
+------+-------+-------+-------------+
|    1 | ram   |  1700 | DISTINCTION |
|    2 | radha |  1000 | DISTINCTION |
|    3 | raj   |   500 | FAIL        |
|    4 | raj   |   950 | FIRST CLASS |
|    5 | raja  |   940 | FIRST CLASS |
|    5 | rinku |   925 | FIRST CLASS |
|    5 | rinki |   725 | FAIL        |
+------+-------+-------+-------------+
7 rows in set (0.00 sec)
```