Page No.
Date

Assignment - VII

Title :- PL/SQL stored procedure & stored function

Problem Statement :- Write a stored procedure namely proc-Grade for the categorization of customer.

Objective :-

i) Understand PL/SQL stored procedure
ii) Understand PL/SQL stored function
iii) Write PL/SQL block code using stored procedure & stored function

Outcome :-

Student shall be able to
i) Implement PL/SQL stored procedure
ii) Implement PL/SQL stored function
iii) Implement PL/SQL block code using stored procedure.

S/W & H/W requirements :-

MySQL, 64 bit OS, computer system.

Theory :-

· PL/SQL -

PL/SQL stands for procedural language structured query language. PL/SQL offers set of procedural commands organized within block that implement & extend reach of mySQL.

· Stored Procedure -

A stored procedure is simple a proc is a named PL/SQL block which performs one or more specific task. This is similar to a procedure in other ~~procedure~~ progra-mming languages. A procedure has a header & a body. Header consist of the name of the procedure & the parameters or variable passed to procedure. The body consist of declaration section, execution section & exception section similar to general PL/SQL block.

· Procedure : passing parameters -

We can pass parameters to procedure in 3 ways.

i) IN parameters
ii) OUT parameters
iii) IN OUT parameters

A procedure may or maynot return any value.

Page No.
Date

• General syntax to create a procedure -

```
CREATE [OR REPLACE] procedure proc_name[
list of parameters]
 IS
      Declaration section
 Begin
      Excecution section
 Exception
      Exception section
 END..
```

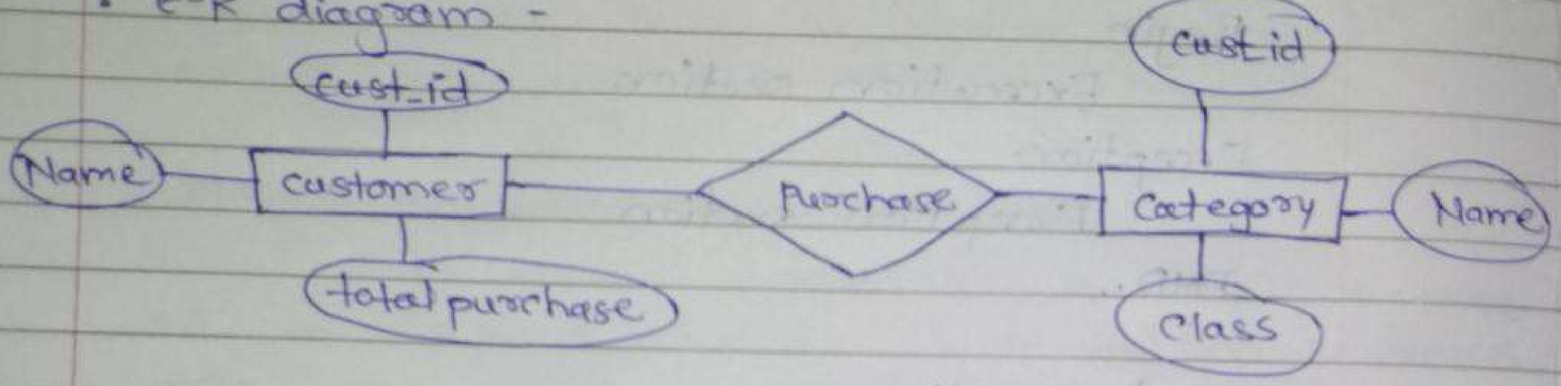
• stored function -
A function is a named PL/SQL block which is similar to a procedure. The major difference between a procedure & a function is that a function must al-ways return a value but a procedure may or maynot return a value.

• General syntax to create function -

```
CREATE [OR REPLACE] Function fun_name [
parameters]
 RETURN returndatatype,
 IS
 Declaration section
 Begin
 Execution section
 return return_variable
 END
```

• Return Type -
The header section defines the return type of function. The return datatype can be any of the datatype like varchar, number etc.

• E-R diagram -

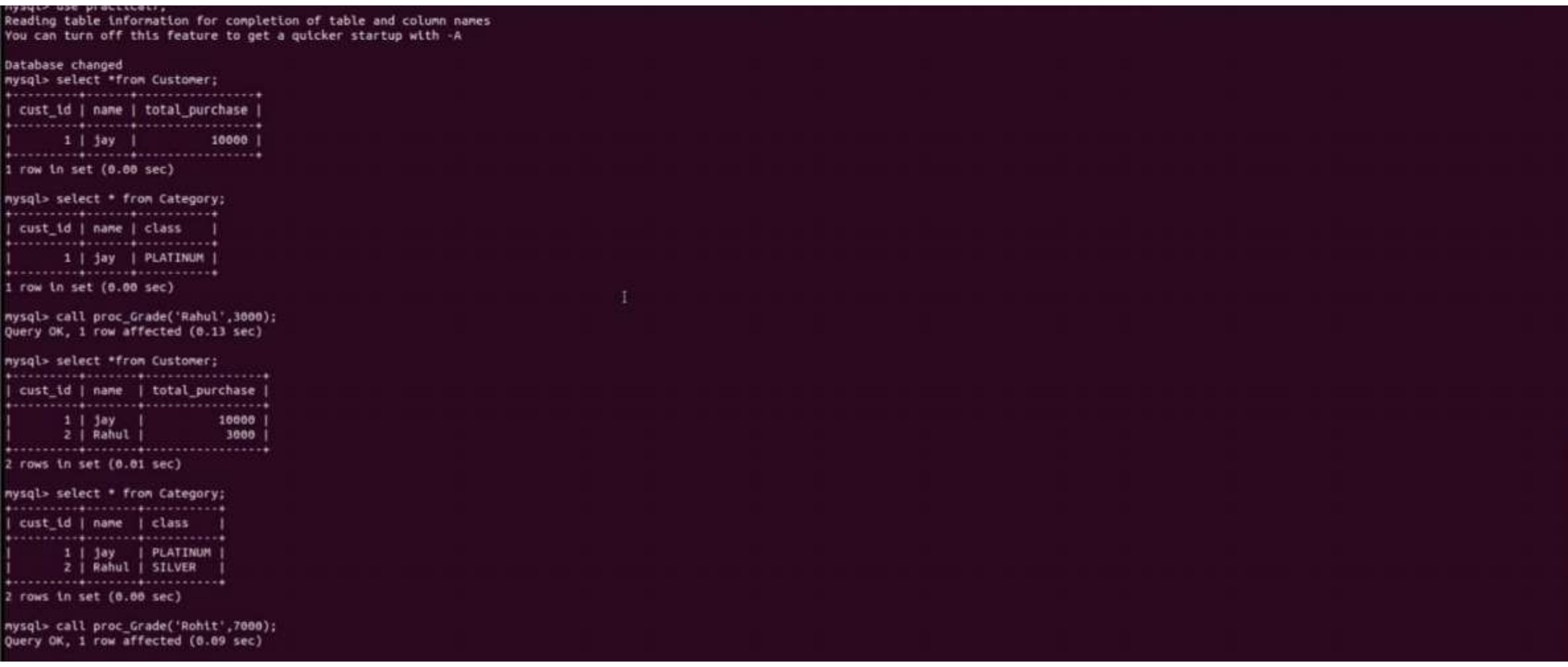


• Test cases :-

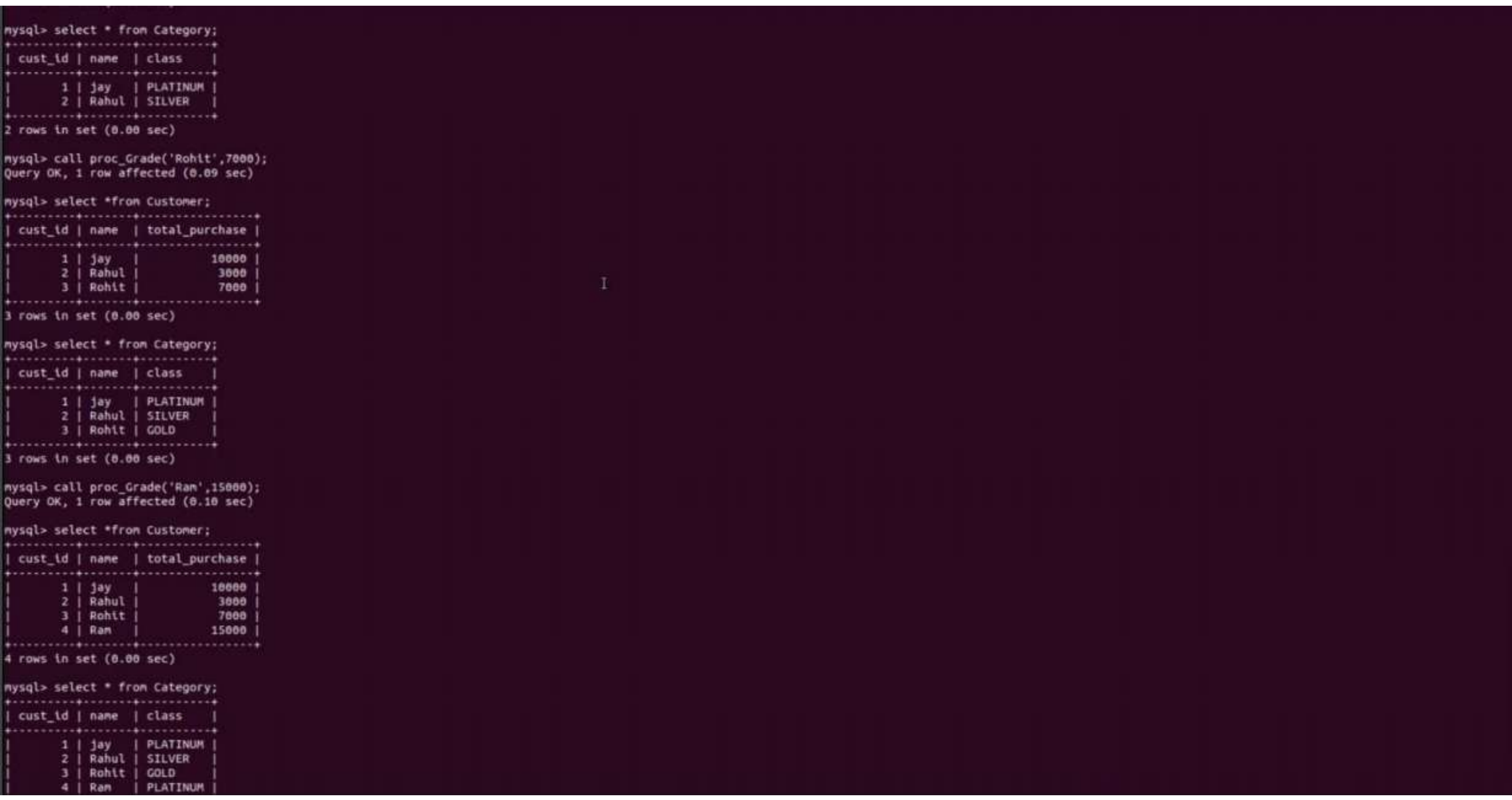
I/P	O/P	Expected O/P	Result
1] Call proc_name("Jay", 1500)	none	none	success
2] call proc name("A", 3000)	Silver	Silver	Success

Conclusion :-
In this assignment, we learnt implementation of stored procedure & function.

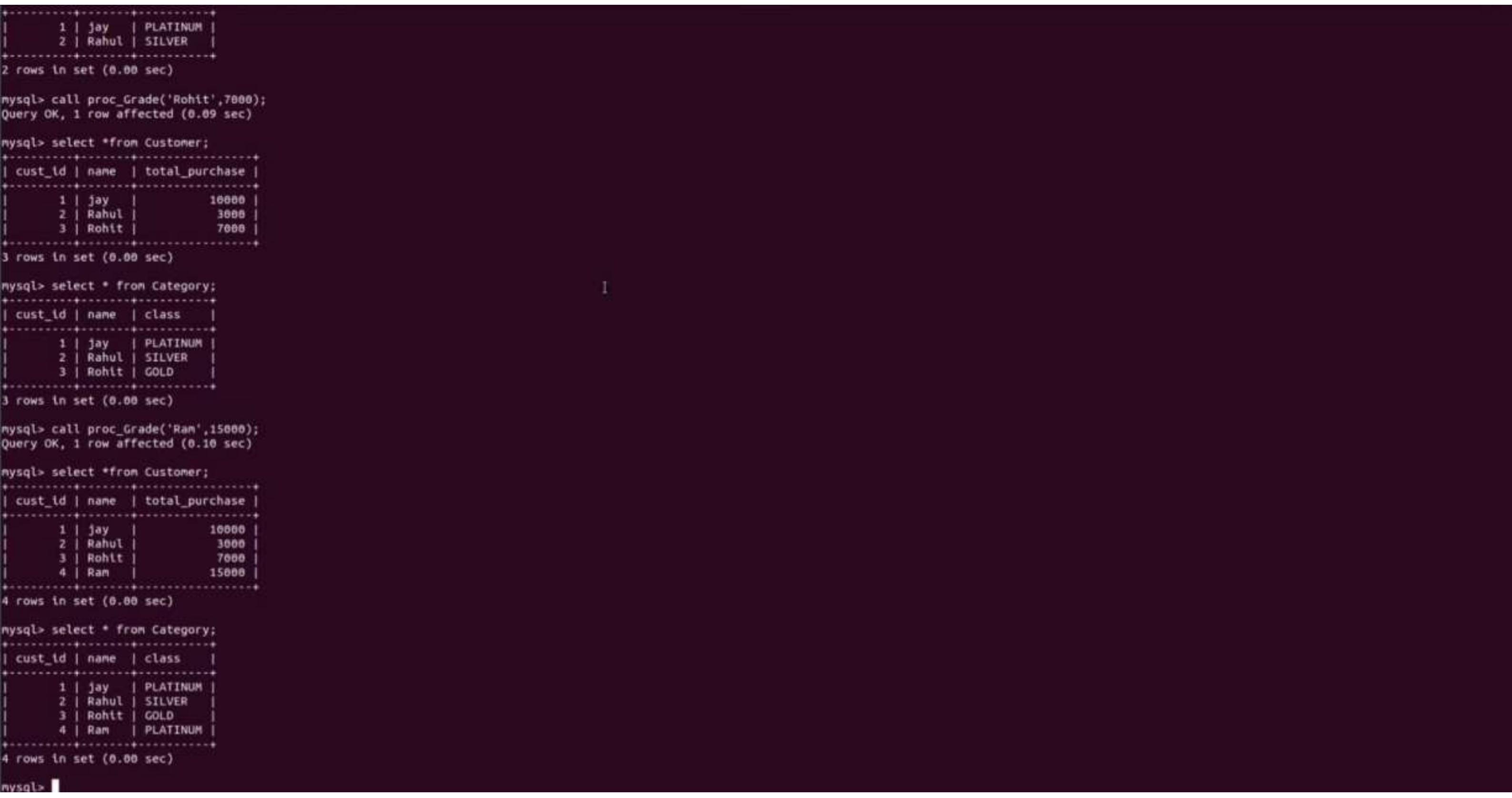
```
show databases;
create database asgn7;
use asgn7;
create table Customer ( cust_id int primary key auto_increment , name varchar(100),
total_purchase int );
create table Category ( cust_id int primary key auto_increment , name varchar(100),
class varchar(100));
show tables;

delimiter $$
create function cust_class( credit int )
returns varchar (100)
deterministic
begin
 DECLARE customerLevel VARCHAR(100);

 IF credit > 20000 THEN
  SET customerLevel = 'Not Define';
 ELSEIF (credit >= 10000 AND credit <= 20000 ) THEN
  SET customerLevel = 'PLATINUM';
 ELSEIF (credit >= 5000 AND credit <= 9999 ) THEN
  SET customerLevel = 'GOLD';
 ELSEIF (credit >= 2000 AND credit <= 4999 ) THEN
  SET customerLevel = 'SILVER';
 ELSEIF  credit<2000 THEN
  SET customerLevel = 'Not Define';
 END IF;
  --return the customer level
  RETURN(customerLevel);
 END$$
 DELIMITER;

show function status where db='asgn7';

delimiter$$
create procedure proc_Grade (in cust_name varchar(100),in purchase int)
begin
 declare class varchar(100);
 insert into Customer (name,total_purchase) values (cust_name,purchase);
 set class = cust_class(purchase);
 insert into Category(name,class) values (cust_name,class);
 end$$
delimiter;

call proc_Grade('jay',10000);
select * from Customer;
select * from Category;
drop procedure proc_Grade;
drop function cust_class;
drop table Customer;
drop table Category;
```



```
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> select *from Customer;
+---------+------+----------------+
| cust_id | name | total_purchase |
+---------+------+----------------+
|       1 | jay  |          10000 |
+---------+------+----------------+
1 row in set (0.00 sec)

mysql> select * from Category;
+---------+------+----------+
| cust_id | name | class    |
+---------+------+----------+
|       1 | jay  | PLATINUM |
+---------+------+----------+
1 row in set (0.00 sec)

mysql> call proc_Grade('Rahul',3000);
Query OK, 1 row affected (0.13 sec)

mysql> select *from Customer;
+---------+-------+----------------+
| cust_id | name  | total_purchase |
+---------+-------+----------------+
|       1 | jay   |          10000 |
|       2 | Rahul |           3000 |
+---------+-------+----------------+
2 rows in set (0.01 sec)

mysql> select * from Category;
+---------+-------+----------+
| cust_id | name  | class    |
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
+---------+-------+----------+
2 rows in set (0.00 sec)

mysql> call proc_Grade('Rohit',7000);
Query OK, 1 row affected (0.09 sec)
```



```
mysql> select * from Category;
+---------+-------+----------+
| cust_id | name  | class    |
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
+---------+-------+----------+
2 rows in set (0.00 sec)

mysql> call proc_Grade('Rohit',7000);
Query OK, 1 row affected (0.09 sec)

mysql> select *from Customer;
+---------+-------+----------------+
| cust_id | name  | total_purchase |
+---------+-------+----------------+
|       1 | jay   |          10000 |
|       2 | Rahul |           3000 |
|       3 | Rohit |           7000 |
+---------+-------+----------------+
3 rows in set (0.00 sec)

mysql> select * from Category;
+---------+-------+----------+
| cust_id | name  | class    |
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
|       3 | Rohit | GOLD     |
+---------+-------+----------+
3 rows in set (0.00 sec)

mysql> call proc_Grade('Ram',15000);
Query OK, 1 row affected (0.10 sec)

mysql> select *from Customer;
+---------+-------+----------------+
| cust_id | name  | total_purchase |
+---------+-------+----------------+
|       1 | jay   |          10000 |
|       2 | Rahul |           3000 |
|       3 | Rohit |           7000 |
|       4 | Ram   |          15000 |
+---------+-------+----------------+
4 rows in set (0.00 sec)

mysql> select * from Category;
+---------+-------+----------+
| cust_id | name  | class    |
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
|       3 | Rohit | GOLD     |
|       4 | Ram   | PLATINUM |
```



```
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
+---------+-------+----------+
2 rows in set (0.00 sec)

mysql> call proc_Grade('Rohit',7000);
Query OK, 1 row affected (0.09 sec)

mysql> select *from Customer;
+---------+-------+----------------+
| cust_id | name  | total_purchase |
+---------+-------+----------------+
|       1 | jay   |          10000 |
|       2 | Rahul |           3000 |
|       3 | Rohit |           7000 |
+---------+-------+----------------+
3 rows in set (0.00 sec)

mysql> select * from Category;
+---------+-------+----------+
| cust_id | name  | class    |
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
|       3 | Rohit | GOLD     |
+---------+-------+----------+
3 rows in set (0.00 sec)

mysql> call proc_Grade('Ram',15000);
Query OK, 1 row affected (0.10 sec)

mysql> select *from Customer;
+---------+-------+----------------+
| cust_id | name  | total_purchase |
+---------+-------+----------------+
|       1 | jay   |          10000 |
|       2 | Rahul |           3000 |
|       3 | Rohit |           7000 |
|       4 | Ram   |          15000 |
+---------+-------+----------------+
4 rows in set (0.00 sec)

mysql> select * from Category;
+---------+-------+----------+
| cust_id | name  | class    |
+---------+-------+----------+
|       1 | jay   | PLATINUM |
|       2 | Rahul | SILVER   |
|       3 | Rohit | GOLD     |
|       4 | Ram   | PLATINUM |
+---------+-------+----------+
4 rows in set (0.00 sec)

mysql> 
```