Zeal College of Engineering and Research

Subject: DBMSL

Name: Jenil Girish Rathod Class: TE

Div: B Batch: B1

Roll No: T212009

Group A: Practical No: 6

Program Statement:

Named PL/SQL Block: PL/SQL Stored Procedure and Stored Function. Write a Stored Procedure namely proc_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and 900 category is first class, if marks 899 and 825 category is Higher Second Class. Write a PL/SQL block to use procedure created with above requirement. Stud_Marks(name, total_marks)

Result(Roll,Name, Class

Code:

```
mysql> CREATE DATABASE Grade;
Query OK, 1 row affected (0.00 sec)

mysql> USE Grade;
Database changed
mysql> CREATE TABLE Stud_Marks (name VARCHAR(50), total_marks INT);
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Result (Roll INT AUTO_INCREMENT PRIMARY KEY, Name
VARCHAR(50), Class VARCHAR(30));
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO Stud_Marks (name, total_marks)
    -> VALUES
    -> ('Zaid', 1450),
    -> ('Jenil', 980),
    -> ('Sudeep', 860),
    -> ('Durva', 820);
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE proc_Grade (IN p_name VARCHAR(50), IN p_total_marks INT)
   -> BEGIN
   -> DECLARE student_class VARCHAR(30);
   -> IF p_total_marks <= 1500 AND p_total_marks >= 990  THEN
   -> SET student_class = 'Distinction';
   -> ELSEIF p_total_marks <= 989 AND p_total_marks >= 900 THEN
   -> SET student_class = 'First Class';
   -> ELSEIF p_total_marks <= 899 AND p_total_marks >= 825 THEN
   -> SET student_class = 'Higher Second Class';
   -> ELSEIF p_total_marks <= 824 AND p_total_marks >= 525 THEN
   -> SET student_class = 'Pass';
   -> ELSE
   -> SET student_class = 'Fail';
   -> END IF;
   -> INSERT INTO Result (Name, Class) VALUES (p_name, student_class);
   -> END;
   -> //
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> DELIMITER //
mysql> CREATE FUNCTION func_CallGrade(p_name VARCHAR(50), p_total_marks INT)
   -> RETURNS VARCHAR(100)
   -> DETERMINISTIC
   -> BEGIN
   -> CALL proc_Grade(p_name, p_total_marks);
   -> RETURN CONCAT(p_name, ' has been graded successfully.');
   -> END;
   -> //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;
mysql> SELECT func_CallGrade('Zaid', 1450);
+-----------------------------------+
| func_CallGrade('Zaid', 1450)      |
+-----------------------------------+
| Zaid has been graded successfully. |
+-----------------------------------+
1 row in set (0.01 sec)

mysql> SELECT func_CallGrade('Jenil', 980);
+-----------------------------------+
| func_CallGrade('Jenil', 980)      |
+-----------------------------------+
| Jenil has been graded successfully. |
+-----------------------------------+
1 row in set (0.00 sec)

mysql> SELECT func_CallGrade('Sudeep', 860);
+-----------------------------------+
| func_CallGrade('Sudeep', 860)     |
+-----------------------------------+
| Sudeep has been graded successfully. |
```

```
+-------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT func_CallGrade('Durva', 820);
+-------------------------------------+
| func_CallGrade('Durva', 820)        |
+-------------------------------------+
| Durva has been graded successfully. |
+-------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM Result;
+------+--------+--------------------+
| Roll | Name   | Class              |
+------+--------+--------------------+
|    1 | Zaid   | Distinction        |
|    2 | Jenil  | First Class        |
|    3 | Sudeep | Higher Second Class |
|    4 | Durva  | Pass               |
+------+--------+--------------------+
4 rows in set (0.00 sec)

mysql>
```