Assignment no.: 5

```
USE te31145_db;

CREATE TABLE Stud_Marks (
        roll_no INT,
        name VARCHAR(50)      ,
        marks INT
);

CREATE TABLE Result (
        roll_no INT,
        class enum('Distinction', 'First Class', 'Higher Second Class', 'Pass')
);


DELIMITER $$

CREATE PROCEDURE ProcGrade (
        IN p_rollno INT
)
BEGIN
        DECLARE v_marks INT;
        DECLARE v_grade varchar(30);
        DECLARE exit handler for not found BEGIN
                SELECT 'No matching record found.' AS Message;
        END;

        SELECT  marks INTO v_marks
        FROM  Stud_Marks
        WHERE roll_no = p_rollno;

        IF v_marks <= 1500 AND v_marks >= 0 THEN
                IF v_marks >= 990 THEN
                        SET v_grade = 'Distinction';
                ELSEIF v_marks >= 900 THEN
                        SET v_grade = 'First Class';
                ELSEIF v_marks >= 825 THEN
                        SET v_grade = 'Higher Second Class';
                ELSE
                        SET v_grade = 'Pass';
                END IF;


                INSERT INTO Result (roll_no, class)
                VALUES (p_rollno, v_grade);


                SELECT  'Student graded successfully.' AS Message, v_grade AS grade;
        ELSE
                SELECT 'Marks Out of Range.' AS Message;
        END IF;
END $$

DELIMITER ;
```

INSERT INTO Stud_Marks (roll_no, name, marks)
VALUES (50, 'Song Joong Ki', 1450);

SELECT * FROM Stud_Marks;

```
+---------+---------------+-------+
| roll_no | name          | marks |
+---------+---------------+-------+
|      50 | Song Joong Ki |  1450 |
+---------+---------------+-------+
```

CALL ProcGrade(50);

```
+------------------------------+-------------+
| Message                      | grade       |
+------------------------------+-------------+
| Student graded successfully. | Distinction |
+------------------------------+-------------+
```

SELECT * FROM Result;

```
+---------+-------------+
| roll_no | class       |
+---------+-------------+
|      50 | Distinction |
+---------+-------------+
```

CALL ProcGrade(51);

```
+---------------------------+
| Message                   |
+---------------------------+
| No matching record found. |
+---------------------------+
```

INSERT INTO Stud_Marks (roll_no, name, marks)
VALUES (11, 'Kim Seon Ho', 950);

SELECT * FROM Stud_Marks;

```
+---------+---------------+-------+
| roll_no | name          | marks |
+---------+---------------+-------+
|      50 | Song Joong Ki |  1450 |
|      11 | Kim Seon Ho   |   950 |
+---------+---------------+-------+
```

CALL ProcGrade(11);

```
+------------------------------+-------------+
| Message                      | grade       |
+------------------------------+-------------+
| Student graded successfully. | First Class |
+------------------------------+-------------+
```

SELECT * FROM Result;

```
+---------+-------------+
| roll_no | class       |
+---------+-------------+
|      50 | Distinction |
|      11 | First Class |
+---------+-------------+
```

INSERT INTO Stud_Marks (roll_no, name, marks)
VALUES (33, 'Ji Chang-wook', 850);

SELECT * FROM Stud_Marks;

```
+---------+-----------------+-------+
| roll_no | name            | marks |
+---------+-----------------+-------+
|      50 | Song Joong Ki   |  1450 |
|      11 | Kim Seon Ho     |   950 |
|      33 | Ji Chang-wook   |   850 |
+---------+-----------------+-------+
```

CALL ProcGrade(33);

```
+------------------------------+---------------------+
| Message                      | grade               |
+------------------------------+---------------------+
| Student graded successfully. | Higher Second Class |
+------------------------------+---------------------+
```

SELECT * FROM Result;

```
+---------+---------------------+
| roll_no | class               |
+---------+---------------------+
|      50 | Distinction         |
|      11 | First Class         |
|      33 | Higher Second Class |
+---------+---------------------+
```

DELIMITER $$

CREATE FUNCTION ReturnGrade (
        marks INT
) RETURNS VARCHAR(30)
DETERMINISTIC

BEGIN
        DECLARE grade VARCHAR(30);

        IF marks > 1500 OR marks < 0 THEN
                SET grade = 'Invalid marks';
        ELSEIF marks >= 990 THEN
                SET grade = 'Distinction';
        ELSEIF marks >= 900 THEN
                SET grade = 'First class';
        ELSEIF marks >= 825 THEN
                SET grade = 'Higher second class';
        ELSE
                SET grade = 'Pass';
        END IF;
        RETURN grade;
END $$

DELIMITER ;

SELECT * FROM Stud_Marks;

```
+---------+---------------+-------+
| roll_no | name          | marks |
+---------+---------------+-------+
|      50 | Song Joong Ki |  1450 |
|      11 | Kim Seon Ho   |   950 |
|      33 | Ji Chang-wook |   850 |
+---------+---------------+-------+
```

SELECT ReturnGrade(830);

```
+---------------------+
| ReturnGrade(830)    |
+---------------------+
| Higher second class |
+---------------------+
```

SELECT ReturnGrade(
      (SELECT marks FROM Stud_Marks WHERE roll_no = 50)
) AS Grade;

```
+-------------+
| Grade       |
+-------------+
| Distinction |
+-------------+
```

SELECT ReturnGrade(
      (SELECT marks FROM Stud_Marks WHERE name = 'Kim Seon Ho')
) AS Grade;

```
+-------------+
| Grade       |
+-------------+
| First class |
+-------------+
```