ASSIGNMENT A6

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)**

```
-- Create the Stud_Marks table
CREATE TABLE Stud_Marks (
    Roll NUMBER(14),
    name VARCHAR(255),
    total_marks NUMBER(14),
    PRIMARY KEY (Roll)
);

-- Insert sample data into Stud_Marks
INSERT INTO Stud_Marks VALUES (16, 'Rahul', 1350);
INSERT INTO Stud_Marks VALUES (21, 'Yash', 979);
INSERT INTO Stud_Marks VALUES (3, 'Tejas', 834);
INSERT INTO Stud_Marks VALUES (44, 'Dinesh', 667);
INSERT INTO Stud_Marks VALUES (35, 'Simran', 1400);

-- Create the Result table
CREATE TABLE Result (
    Roll NUMBER(14),
    name VARCHAR(255),
    Class VARCHAR(255),
    FOREIGN KEY (Roll) REFERENCES Stud_Marks (Roll)
);

-- Create the procedure to determine grades
CREATE OR REPLACE PROCEDURE proc_grade (
    rollno IN NUMBER,
    p_roll_no OUT Stud_Marks.Roll%TYPE,
    p_name OUT Stud_Marks.name%TYPE,
    p_total OUT Stud_Marks.total_marks%TYPE
) AS
BEGIN
    SELECT Roll, name, total_marks
    INTO p_roll_no, p_name, p_total
    FROM Stud_Marks
    WHERE Roll = rollno;

    IF p_total <= 1500 AND p_total >= 990 THEN
        INSERT INTO Result VALUES (p_roll_no, p_name, 'Distinction');
    ELSIF p_total <= 989 AND p_total >= 900 THEN
        INSERT INTO Result VALUES (p_roll_no, p_name, 'First Class');
    ELSIF p_total <= 899 AND p_total >= 825 THEN
        INSERT INTO Result VALUES (p_roll_no, p_name, 'Higher Second Class');
    ELSE
        INSERT INTO Result VALUES (p_roll_no, p_name, 'FAIL');
    END IF;

EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Rollno ' || rollno || ' is not present!!');
END;
/

-- Display contents of Stud_Marks and Result tables before calling the procedure
SELECT * FROM Stud_Marks;

SELECT * FROM Result;

-- Block to execute the procedure
DECLARE
    rollno_a NUMBER(14) := 16;
    rollno_b NUMBER(14) := 3;
    rollno_c NUMBER(14) := 57;
    p_roll_no Stud_Marks.Roll%TYPE;
    p_name Stud_Marks.name%TYPE;
    p_total_marks Stud_Marks.total_marks%TYPE;
BEGIN
    -- Call the procedure with the assigned roll number
    proc_grade(rollno_a, p_roll_no, p_name, p_total_marks);
    proc_grade(rollno_b, p_roll_no, p_name, p_total_marks);
    proc_grade(rollno_c, p_roll_no, p_name, p_total_marks);
END;
/

-- Display contents of Stud_Marks and Result tables after calling the procedure
SELECT * FROM Stud_Marks;

SELECT * FROM Result;
```

OUTPUT:

```
Table created.

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

Table created.

Procedure created.
```

ROLL	NAME	TOTAL_MARKS
16	Rahul	1350
21	Yash	979
3	Tejas	834
44	Dinesh	667
35	Simran	1400

```
5 rows selected.

no data found

Statement processed.
Rollno 57 is not present!!
```

ROLL	NAME	TOTAL_MARKS
16	Rahul	1350
21	Yash	979
3	Tejas	834
44	Dinesh	667
35	Simran	1400

```
5 rows selected.
```

ROLL	NAME	CLASS
16	Rahul	Distinction
3	Tejas	Higher Second Class