Practical No 06

Problem Statement :- Write a stored procedure namely proc_grade for the categorization of the student if marks scored by the student in examination is &lt;=1500 and marks&gt;=990 then student will be placed in distinction category. If marks scored are between 989 and 900 then category is First class,if marks are between 899 and 825 then category is Higher 2nd class else in Pass Class

Write Create a function which will return total students in a given class. Create the table Student marks(Roll_No,Name,Total marks) Result(Roll_No,Name,Class)

Submitted By :- 1. Bhushan Asati
                2. Asha Mandale
                3. Sayaji Dhandge

```
SQL*Plus: Release 10.2.0.1.0 - Production on Sat Nov 20 13:59:43 2021

Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> CREATE TABLE STUD_MARK(
  2  ROLL_NO NUMBER(2),
  3  NAME VARCHAR(20),
  4 TOTAL_MARK NUMBER(4));

Table created.

SQL> CREATE TABLE STUD_RESULT(
  2  ROLL_NO NUMBER(2),
  3  NAME VARCHAR(20),
  4 CLASS VARCHAR(20));

Table created.

SQL>
SQL> INSERT INTO STUD_MARK VALUES(1,'NILESH',1200);
```

```
1 row created.
```

```
SQL> CREATE OR REPLACE PROCEDURE PROC_GRADE(ROLL IN NUMBER)
  2  IS
  3  MARK NUMBER(4);
  4  NM VARCHAR(20);
  5  CLASS VARCHAR(20);
  6  BEGIN
  7     SELECT TOTAL_MARK,NAME INTO MARK,NM FROM STUD_MARK WHERE
  ROLL_NO=ROLL; 8
  9  IF MARK<=1500 AND MARK>=990 THEN
 10  DBMS_OUTPUT.PUT_LINE('DISTINCTION');
 11  CLASS:='DISTINCTION';
 12    INSERT INTO STUD_RESULT
 VALUES(ROLL,NM,CLASS); 13
 14  ELSIF MARK<=989 AND MARK>=900 THEN
 15  DBMS_OUTPUT.PUT_LINE('FIRST CLASS');
 16  CLASS:='FIRST CLASS';
 17    INSERT INTO STUD_RESULT
 VALUES(ROLL,NM,CLASS); 18
 19  ELSIF MARK<=899 AND MARK>=825 THEN
 20  DBMS_OUTPUT.PUT_LINE('SECOND CLASS');
 21  CLASS:='SECOND CLASS';
 22    INSERT INTO STUD_RESULT
 VALUES(ROLL,NM,CLASS); 23
 24  ELSE
 25  DBMS_OUTPUT.PUT_LINE('PASS CLASS');
 26  CLASS:='PASS CLASS';
 27    INSERT INTO STUD_RESULT
 VALUES(ROLL,NM,CLASS); 28
 29  END IF;
 30  END;
 31 /
Procedure created.
SQL> DECLARE
  2  ROLL_NO NUMBER(10) :=&R;
  3  BEGIN
  4  PROC_GRADE(ROLL_NO);
  5  END;
  6 /
Enter value for r: 1
old   2: ROLL_NO NUMBER(10)
:=&R; new 2: ROLL_NO NUMBER(10)
:=1;

PL/SQL procedure successfully completed.
```

```
SQL> CREATE OR REPLACE FUNCTION FUN_GRADE
  2  RETURN NUMBER
  3  IS
  4  RECORD NUMBER :=0;
  5
  6  BEGIN
  7
  8  SELECT COUNT(*) INTO RECORD FROM STUD_MARK;
  9  RETURN RECORD;
 10  END;
 11  /

Function created.

SQL> DECLARE
  2  R NUMBER;
  3     BE
  GIN 4
  5  R:=FUN_GRADE();
  6  DBMS_OUTPUT.PUT_LINE(R);
  7  END;
  8 /

PL/SQL procedure successfully completed.

SQL> select * from STUD_MARK;

   ROLL_NO NAME                 TOTAL_MARK
 ---------- -------------------- ----------
         1 NILESH                     1200

SQL> select * from STUD_RESULT;

   ROLL_NO NAME                 CLASS
            --------------------
         1 NILESH               DISTINCTION
SQL>
```