CREATE TABLE Stud\_Marks (

Roll NUMBER PRIMARY KEY,

Name VARCHAR2(255) NOT NULL,

Total\_Marks NUMBER CHECK (Total\_Marks >= 0)

);

CREATE TABLE Result (

Roll NUMBER PRIMARY KEY,

Name VARCHAR2(255) NOT NULL,

Class VARCHAR2(50) NOT NULL

);

CREATE OR REPLACE PROCEDURE proc\_Grade(rollNo IN NUMBER, Name IN VARCHAR2, Marks IN NUMBER) AS

BEGIN

IF (Marks <= 1500 AND Marks >= 990) THEN

DBMS\_OUTPUT.PUT\_LINE('DISTINCTION');

INSERT INTO Result VALUES (rollNo, Name, 'DISTINCTION');

ELSIF (Marks < 990 AND Marks >= 900) THEN

DBMS\_OUTPUT.PUT\_LINE('FIRST CLASS');

INSERT INTO Result VALUES (rollNo, Name, 'FIRST CLASS');

ELSIF (Marks < 900 AND Marks >= 825) THEN

DBMS\_OUTPUT.PUT\_LINE('HIGHER SECOND CLASS');

INSERT INTO Result VALUES (rollNo, Name, 'HIGHER SECOND CLASS');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Fail');

INSERT INTO Result VALUES (rollNo, Name, 'Fail');

END IF;

INSERT INTO Stud\_Marks (Roll, Name, Total\_Marks) VALUES (rollNo, Name, Marks);

END proc\_Grade;

/

CREATE OR REPLACE FUNCTION fun1(r IN NUMBER, n IN VARCHAR2, m IN NUMBER) RETURN VARCHAR2 AS

BEGIN

proc\_Grade(r, n, m);

RETURN 'SUCCESSFUL';

END fun1;

/

DECLARE

name VARCHAR2(50);

rollNo NUMBER;

marks NUMBER;

class VARCHAR2(50);

BEGIN

rollNo := &rollNo;

name := '&name';

marks := &marks;

class := fun1(rollNo, name, marks);

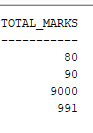
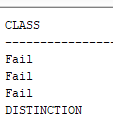
DBMS\_OUTPUT.PUT\_LINE(class);

END;

/

SELECT \* FROM Result;

SELECT \* FROM Stud\_Marks;

**OutPut**:-

