8.

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. Stud\_Marks (RollNo, name, total\_marks) Result (RollNo, Name, Class)

create or replace procedure proc\_grade

(temp in number,

p\_roll\_no out stud\_marks.roll\_no%type,

p\_name out stud\_marks.name%type,

p\_total out stud\_marks.total\_marks%type)

as

begin

select name,total\_marks,roll\_no into p\_name,p\_total,p\_roll\_no from stud\_marks where roll\_no=temp;

if p\_total <=1500 and p\_total >= 990 then

insert into result values(p\_roll\_no,p\_name,'distinction');

else if p\_total <=989 and p\_total >= 900 then

insert into result values(p\_roll\_no,p\_name,'first class');

else if p\_total <=899 and p\_total >= 825 then

insert into result values(p\_roll\_no,p\_name,'HSC');

else

insert into result values(p\_roll\_no,p\_name,'fail');

end if;

end if;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('Roll no ' || temp ||' not found');

end;

/

Declare

temp number(20);

p\_roll\_no stud\_marks.roll\_no%type;

p\_name stud\_marks.name%type;

p\_total stud\_marks.total\_marks%type;

Begin

temp:=&temp;

Proc\_grade(temp,p\_roll\_no,p\_name,p\_total);

End;

/

Alternate code

-- Create Stud\_Marks table to store student roll number, name, and total marks

CREATE TABLE Stud\_Marks (

RollNo INT PRIMARY KEY,

name VARCHAR2(100),

total\_marks INT

);

drop table Result;

-- Create Result table to store the result for each student

CREATE TABLE Result (

RollNo INT PRIMARY KEY,

Name VARCHAR2(100),

Class VARCHAR2(20),

CONSTRAINT fk\_rollno FOREIGN KEY (RollNo) REFERENCES Stud\_Marks(RollNo)

);

-- Insert sample student marks

INSERT INTO Stud\_Marks (RollNo, name, total\_marks) VALUES (1, 'Alice', 1200);

INSERT INTO Stud\_Marks (RollNo, name, total\_marks) VALUES (2, 'Bob', 950);

INSERT INTO Stud\_Marks (RollNo, name, total\_marks) VALUES (3, 'Charlie', 870);

INSERT INTO Stud\_Marks (RollNo, name, total\_marks) VALUES (4, 'David', 780);

-- Call the stored procedure to categorize the students

EXEC proc\_Grade;

-- Check the results

SELECT \* FROM Result;

-- Create the stored procedure for grade categorization

CREATE OR REPLACE PROCEDURE proc\_Grade IS

-- Declare variables to hold student data

v\_rollno Stud\_Marks.RollNo%TYPE;

v\_name Stud\_Marks.name%TYPE;

v\_marks Stud\_Marks.total\_marks%TYPE;

v\_class Result.Class%TYPE;

BEGIN

-- Loop through each student in Stud\_Marks

FOR student IN (SELECT RollNo, name, total\_marks FROM Stud\_Marks) LOOP

v\_rollno := student.RollNo;

v\_name := student.name;

v\_marks := student.total\_marks;

-- Categorize based on marks

IF v\_marks >= 990 AND v\_marks <= 1500 THEN

v\_class := 'Distinction';

ELSIF v\_marks >= 900 AND v\_marks <= 989 THEN

v\_class := 'First Class';

ELSIF v\_marks >= 825 AND v\_marks <= 899 THEN

v\_class := 'Higher Second Class';

ELSE

v\_class := 'Not Classified'; -- In case marks are below 825

END IF;

-- Insert the result into the Result table

INSERT INTO Result (RollNo, Name, Class) VALUES (v\_rollno, v\_name, v\_class);

END LOOP;

-- Commit the changes after processing all students

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Handle any errors

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error occurred: ' || SQLERRM);

END proc\_Grade;

/