Name: Tejas Rajesh Machkar
Roll No: 23
Class: TE2 Comp
PRN: F18112025

```
CREATE OR REPLACE PROCEDURE proc_Grade(roll NUMBER,name VARCHAR,totmarks
NUMBER) IS
BEGIN
    IF(totmarks >= 990 and totmarks <= 1500) THEN
        INSERT INTO Result1 VALUES(roll,name,'DISTINCTION');
        INSERT INTO Stud_marks VALUES(name,totmarks);
    ELSIF(totmarks >= 900 and totmarks <= 989) THEN
        INSERT INTO Result1 VALUES(roll,name,'FIRST CLASS');
        INSERT INTO Stud_marks VALUES(name,totmarks);
    ELSIF(totmarks >= 825 and totmarks <=899) THEN
        INSERT INTO Result1 VALUES(roll,name,'HIGHER SECOND CLASS');
        INSERT INTO Stud_marks VALUES(name,totmarks);
    END IF;
END proc_Grade;
/
-------------------------------------------------------------------
DECLARE
  Name1 Stud_marks.Name%type;
  roll1 Result1.RollNo%type;
  totmarks1 Stud_marks.TotalMarks%type;
BEGIN
  roll1:=&roll1;
  Name1:=&Name1;
  totmarks1:=&totmarks1;
  proc_Grade(roll1,Name1,totmarks1);
  END LOOP;
END;
/
```

Name: Tejas. Rajesh. Machkar
RollNo: 23 PRN: F18112025
Batch: D Class: TE-2 COMP

DBMSL-Assignment-7
Group A

• Questions:
Q1) What's a stored procedure?
A.1) A procedure is a subprogram unit that comprises a group of PL/SQL statements. Each procedure has its own unique name by which it can be refered.
• It's stored as database object.
• Call to these procedures might be made by referring to their name to execute PL/SQL statements.
• The values can be passed into the procedure or fetched from it by using parameters.
• Syntax:

```
CREATE OR REPLACE PROCEDURE procname (<parameters>)
 [IS AS]
       <declaration part>
BEGIN
 :
 :

END;
```

Q2) Explain the use of %ROWTYPE and %TYPE in SQL.
A.2) 1) %ROWTYPE: This attribute is used to provide a record type that represents a row in a table (or view). Columns in a row and corresponding fields in record have same names and datatypes. However, fields in %ROWTYPE record don't inherit the NOT NULL column constraint.
2) %TYPE: This attribute provides the datatype of a variable or database column. It's particularly useful. When declaring variables that refer to database columns Variables %TYPE, don't inherit NOT NULL constraints

Q3) Explain IN, OUT and IN-OUT mode in stored procedure.
A.3) 1) IN parameter: It's used for giving input to the subprograms. It's a read only variable inside the subprograms. In the calling statement, these

parameters can be a variable or a literal value or an expression.
- By default, parameters are of IN Type.

2) OUT parameter: It's used for ~~four~~ getting output from a subprogram. It's a read-write variable. In calling statement, these parameters should always be a variable.

3) IN-OUT parameter: Used for giving input to and getting output from a subprogram. It's a read-write variable in a sub-program. In the calling statement these parameters should always be a variable to hold value from the subprogram.

Q4) What is a stored function?

A.4) A function is a standalone PL/SQL subprogram. Functions have a unique name by which the function can be referred to. These are stored as PL/SQL database object.
- It uses a RETURN keyword to return any value. A function must always return value or raise an exception.
- Function can also return the value through OUT parameters other than using RETURN.
- Syntax:

```
CREATE OR REPLACE FUNCTION <func_name> (<parameters>)
RETURN <datatype> [IS|AS] <declaration>
BEGIN
  :
  :
  .
END;
```

Q5) What's the difference between stored function and stored procedures?

A5)

PROCEDURE	FUNCTION
1) Use mainly to execute certain process.	1) Used mainly to perform some calculations.
2) Cannot be Called in SELECT statement.	2) A function that contains no DML, can be called in a SELECT statement.

3) OUT parameter is used to return a value	3) RETURN is used to return a value.
4) Return will simply exit the control from the subprogram.	4) MUST compulsarily return a value. Return will exit control and also return a value.