Dr. D.Y. PATIL
Academic Education Excellence
Federation

D.Y. Patil Academic Education Excellence Federation's
**Dr.D.Y. Patil Technical Campus**
(Engineering and MCA)
(Approved by AICTE, DTE-Govt of Maharashtra, Affiliated to Savitribai Phule Pune University, Pune)
Sr.No.32/1/A/7, Near Talegaon Railway Station, A/P Varale, Talegaon Dabhade
Tal-Maval, Dist-Pune 410507
Ph. No.9920141406,9309228311,7666829653,9307909501
Website: www.dypatiltcs.com

## Lab Manual – Named PL/SQL Block – Stored Procedure and Function

## Experiment No. 5

### Title:

Named PL/SQL Block – Stored Procedure and Function

---

### Objectives:

- To understand the concept of **stored procedures** and **stored functions** in PL/SQL.
- To implement categorization of students into grade categories based on total marks.
- To demonstrate passing parameters to stored procedures and inserting results into a table.
- To invoke a stored procedure inside a PL/SQL block.

### Problem Statement:

Write a **stored procedure** named proc_Grade that categorizes students based on marks:

- If marks ≥ 990 and ≤ 1500 → **Distinction**
- If marks between 900 and 989 → **First Class**
- If marks between 825 and 899 → **Higher Second Class**
- Otherwise → **Fail**

Tables:

- **Stud_Marks(name, total_marks)**
- **Result(roll, name, class)**

Write a PL/SQL block that uses the procedure proc_Grade to insert categorized results into the **Result** table.

## Software and Hardware Requirements:

- **Software:** Oracle Database 10g/11g/12c or higher, SQL*Plus / Oracle SQL Developer
- **Hardware:** Intel i3/i5 Processor, 4GB+ RAM, 500GB HDD, Windows/Linux OS

---

## Theory – Concept in Brief:

- **Stored Procedure:** A precompiled PL/SQL block stored in the database, invoked by name.
- **Stored Function:** Similar to procedure but returns a single value.
- **Advantages:** Reusability, modularity, better performance, security.
- **This experiment demonstrates categorization logic using conditions (IF-ELSE) in stored procedures.**

---

## Algorithm:

1. Create the Stud_Marks and Result tables.
2. Write the stored procedure proc_Grade with inputs (Roll_no, Name, Marks).
3. Inside the procedure:
    - Apply IF-ELSIF logic to categorize marks.
    - Insert the result into Result table.
4. Write an anonymous PL/SQL block to call proc_Grade for each student.
5. Execute and verify the output.

---

## Flowchart:

---

## PL/SQL Code:

### Step 1: Table Creation

```
CREATE TABLE Stud_Marks (
   Roll_no NUMBER PRIMARY KEY,
   Name VARCHAR2(50),
   Total_Marks NUMBER
);

CREATE TABLE Result (
   Roll_no NUMBER,
   Name VARCHAR2(50),
   Class VARCHAR2(30)
);
```

### Step 2: Stored Procedure

```
CREATE OR REPLACE PROCEDURE proc_Grade (
   p_roll Stud_Marks.Roll_no%TYPE,
   p_name Stud_Marks.Name%TYPE,
   p_marks Stud_Marks.Total_Marks%TYPE
) AS
   v_class VARCHAR2(30);
BEGIN
   IF p_marks BETWEEN 990 AND 1500 THEN
      v_class := 'Distinction';
   ELSIF p_marks BETWEEN 900 AND 989 THEN
      v_class := 'First Class';
   ELSIF p_marks BETWEEN 825 AND 899 THEN
      v_class := 'Higher Second Class';
   ELSE
      v_class := 'Fail';
   END IF;

   INSERT INTO Result(Roll_no, Name, Class)
   VALUES (p_roll, p_name, v_class);

   DBMS_OUTPUT.PUT_LINE('Student ' || p_name || ' categorized as: ' || v_class);
END;
/
```

### Step 3: Anonymous PL/SQL Block to Call Procedure

```
BEGIN
   FOR rec IN (SELECT * FROM Stud_Marks) LOOP
      proc_Grade(rec.Roll_no, rec.Name, rec.Total_Marks);
```

```
  END LOOP;
END;
/
```

**Test Cases:**

| Roll_no | Name | Total_Marks | Expected Class |
|---|---|---|---|
| 1 | Amit | 1200 | Distinction |
| 2 | Priya | 950 | First Class |
| 3 | Rahul | 850 | Higher Second Class |
| 4 | Sneha | 700 | Fail |

**Test Data Set:**

```
INSERT INTO Stud_Marks VALUES (1, 'Amit', 1200);

INSERT INTO Stud_Marks VALUES (2, 'Priya', 950);

INSERT INTO Stud_Marks VALUES (3, 'Rahul', 850);

INSERT INTO Stud_Marks VALUES (4, 'Sneha', 700);
```

**Conclusion / Analysis:**

- A **stored procedure** was successfully created to categorize students based on marks.
- The procedure was invoked using a PL/SQL block and results were inserted into the **Result** table.
- Control structures (IF-ELSIF-ELSE) implemented classification logic correctly.
- The experiment demonstrates modular programming and reusability of PL/SQL stored procedures.