



# University Institute of Engineering

## Department of Computer Science & Engineering

### Experiment: 4

**Student Name: Hunar**

**UID:24BDA70085**

**Branch: CSE**

**Section/Group: AIT-KRG-G2**

**Semester: 4th**

**Date of Performance:30/01/26**

**Subject Name: DBMS**

- 1. Aim of the practical:** To understand the basic structure of a PL/SQL program by creating and executing a simple PL/SQL block that includes declaration and execution sections, and to display output using built-in procedures.

### 2. Tool Used:

- **Database Management System:**
- **Oracle Database**

### 3. Objective:

Implement control structures in PL/SQL (IF-ELSE, ELSE-IF , ELSE-IF LADDER, CASE STATEMENTS in PL-SQL BLOCK)

### 4. Practical / Experimental Steps



# University Institute of Engineering

## Department of Computer Science & Engineering

Step 1: Open SQL\*Plus or Oracle SQL Developer and connect to the database.

Step 2: Enable output using SET SERVEROUTPUT ON;.

Step 3: Write and execute a PL/SQL program using IF–ELSE to check positive or non-positive number.

Step 4: Write and run a PL/SQL program using IF–ELSIF–ELSE to display student grade based on marks.

Step 5: Write and execute a PL/SQL program using an ELSIF ladder to determine performance status.

Step 6: Write and run a PL/SQL program using CASE statement to display day name from day number.

Step 7: Modify input values and re-execute programs to test different conditions.

Step 8: Observe and record the outputs for each conditional control statement program.

## 5. I / O Analysis

### A) IF-Else Statement

```
DECLARE
    num NUMBER := -5;
BEGIN
    IF num > 0 THEN
        DBMS_OUTPUT.PUT_LINE('The number is Positive');
    ELSE
        DBMS_OUTPUT.PUT_LINE('The number is Non-Positive');
    END IF;
END;
/
```

Output:

107 ms

The number is Non-Positive

### B) IF-ElseIf-ELSE Statement

```
DECLARE
    marks NUMBER := 85;
```



# University Institute of Engineering

## Department of Computer Science & Engineering

```
BEGIN
  IF marks >= 90 THEN
    DBMS_OUTPUT.PUT_LINE('Grade: A+');
  ELSIF marks >= 80 THEN
    DBMS_OUTPUT.PUT_LINE('Grade: A');
  ELSIF marks >= 70 THEN
    DBMS_OUTPUT.PUT_LINE('Grade: B');
  ELSIF marks >= 60 THEN
    DBMS_OUTPUT.PUT_LINE('Grade: C');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Grade: Fail');
  END IF;
END;
/
```

Output:

125 ms

Grade: A

### C) ELSIF Ladder

```
DECLARE
  marks NUMBER := 72;
BEGIN
  IF marks >= 90 THEN
    DBMS_OUTPUT.PUT_LINE('Performance: Excellent');
  ELSIF marks >= 75 THEN
    DBMS_OUTPUT.PUT_LINE('Performance: Very Good');
  ELSIF marks >= 60 THEN
    DBMS_OUTPUT.PUT_LINE('Performance: Good');
  ELSIF marks >= 40 THEN
    DBMS_OUTPUT.PUT_LINE('Performance: Average');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Performance: Poor');
  END IF;
END;
/
```

Output:

111 ms

Performance: Good



# University Institute of Engineering

## Department of Computer Science & Engineering

### D) CASE Statement

```
DECLARE
    day_num NUMBER := 4;
    day_name VARCHAR2(20);
BEGIN
    CASE day_num
        WHEN 1 THEN day_name := 'Sunday';
        WHEN 2 THEN day_name := 'Monday';
        WHEN 3 THEN day_name := 'Tuesday';
        WHEN 4 THEN day_name := 'Wednesday';
        WHEN 5 THEN day_name := 'Thursday';
        WHEN 6 THEN day_name := 'Friday';
        WHEN 7 THEN day_name := 'Saturday';
        ELSE day_name := 'Invalid Day Number';
    END CASE;

    DBMS_OUTPUT.PUT_LINE('Day: ' || day_name);
END;
/
```

Output:

116 ms

Day: Wednesday

### 6. Learning outcomes (What I have learnt):

Understood the basic structure of a PL/SQL block, including the **DECLARE** and **BEGIN...END** sections.

Learned how to declare and initialize variables for storing data values.

Gained knowledge of using `DBMS_OUTPUT.PUT_LINE` to display results during program execution.

Practiced performing mathematical operations on variables within PL/SQL programs.

Developed understanding of decision-making using **IF-ELSIF-ELSE** conditional statements.

Acquired practical insight into how PL/SQL can be used for simple database-related computations and logic implementation.