

**Student Name:** Ansh Rana  
**Branch:** AIT-CSE- AIML  
**Semester:** 4  
**Subject Name:** DBMS

**UID:** 24BAI70892  
**Section/Group:** 24AIT\_KRG-G2

## WORKSHEET 4

**AIM:** To design and implement PL/SQL programs utilizing conditional control statements such as IF-ELSE, IF-ELSIF-ELSE, ELSIF ladder, and CASE constructs in order to control the flow of execution based on logical conditions and to analyze decision-making capabilities in PL/SQL blocks.

**S/W Requirement:**

- Database Management System: PostgreSQL / Oracle Database Express Edition
- Database Administration Tool: pgAdmin

### **OBJECTIVES:**

- To understand and implement conditional control statements in PL/SQL
- To analyze decision-making using IF-ELSE, ELSIF ladder, and CASE statements
- To enhance logical thinking using PL/SQL blocks

### **PROBLEM STATEMENT:**

Develop and execute PL/SQL programs that demonstrate the use of conditional control statements. The programs should employ IF-ELSE, IF-ELSIF-ELSE, ELSIF ladder, and CASE statements to evaluate given conditions and control the flow of execution accordingly.

#### **1. PROBLEM STATEMENT – IF-ELSE STATEMENT**

---

**Write a PL/SQL program to check whether a given number is positive or non-positive using the IF-ELSE conditional control statement and display an appropriate message.**

#### **PROGRAM:**

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;

## 2. PROBLEM STATEMENT – IF–ELSIF–ELSE STATEMENT

---

**Write a PL/SQL program to evaluate the grade of a student based on obtained marks and display the corresponding grade.**

**PROGRAM:**

**DECLARE**

    marks NUMBER := 78;

BEGIN

    IF marks >= 90 THEN

        DBMS\_OUTPUT.PUT\_LINE('Grade: A');

    ELSIF marks >= 75 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;

## 3. PROBLEM STATEMENT – ELSIF LADDER

---

**Write a PL/SQL program to determine the performance status of a student based on marks using an ELSIF ladder.**

**PROGRAM:**

**DECLARE**



marks NUMBER := 82;

BEGIN

IF marks >= 85 THEN

DBMS\_OUTPUT.PUT\_LINE('Performance: Excellent');

ELSIF marks >= 70 THEN

DBMS\_OUTPUT.PUT\_LINE('Performance: Very Good');

ELSIF marks >= 55 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;

#### 4. PROBLEM STATEMENT – CASE STATEMENT

---

**Write a PL/SQL program to display the name of the day based on a given day number using the CASE statement.**

**PROGRAM:**

DECLARE

day\_num NUMBER := 3;

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 is: ' || day_name);
END;

```

### **LEARNING OUTCOMES:**

1. Understood the use of conditional control statements in PL/SQL.
2. Learned to apply IF–ELSE and IF–ELSIF–ELSE statements for decision-making.
3. Implemented ELSIF ladder for evaluating multiple conditions.
4. Used CASE statements to simplify complex conditional logic.
5. Improved logical reasoning and procedural programming skills in PL/SQL.

### **OUTPUT :**

```

10
11  DECLARE
12  |  num NUMBER := -5;
13  BEGIN
14  |  IF num > 0 THEN
15  |  |  DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is Positive');
16  |  ELSE
17  |  |  DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is Non-Positive');
18  |  END IF;
19  END;
20 /
21

```

Query result    Script output    DBMS output    Explain Plan    SQL history



The number -5 is Non-Positive

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007

```
32
33  DECLARE
34    marks NUMBER := 82;
35  BEGIN
36    IF marks >= 90 THEN
37      DBMS_OUTPUT.PUT_LINE('Grade: A');
38    ELSIF marks >= 75 THEN
39      DBMS_OUTPUT.PUT_LINE('Grade: B');
40    ELSIF marks >= 60 THEN
41      DBMS_OUTPUT.PUT_LINE('Grade: C');
42    ELSE
43      DBMS_OUTPUT.PUT_LINE('Grade: Fail');
44    END IF;
45  END;
46 /
47
48
```

Query result    Script output    DBMS output    Explain Plan    SQL history



Grade: B

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007

```
-- 48
49  DECLARE
50    marks NUMBER := 68;
51  BEGIN
52    IF marks >= 85 THEN
53      DBMS_OUTPUT.PUT_LINE('Performance: Excellent');
54    ELSIF marks >= 70 THEN
55      DBMS_OUTPUT.PUT_LINE('Performance: Very Good');
56    ELSIF marks >= 55 THEN
57      DBMS_OUTPUT.PUT_LINE('Performance: Good');
58    ELSIF marks >= 40 THEN
59      DBMS_OUTPUT.PUT_LINE('Performance: Average');
60    ELSE
61      DBMS_OUTPUT.PUT_LINE('Performance: Poor');
62    END IF;
63  END;
64 --
```



Query result    Script output    DBMS output    Explain Plan    SQL history



Performance: Good

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007

```

66
67  <| DECLARE
68      |   day_no NUMBER := 3;
69      |   day_name VARCHAR2(20);
70  <| BEGIN
71      |   day_name := CASE day_no
72      |       WHEN 1 THEN 'Sunday'
73      |       WHEN 2 THEN 'Monday'
74      |       WHEN 3 THEN 'Tuesday'
75      |       WHEN 4 THEN 'Wednesday'
76      |       WHEN 5 THEN 'Thursday'
77      |       WHEN 6 THEN 'Friday'
78      |       WHEN 7 THEN 'Saturday'
79      |       ELSE 'Invalid Day Number'
80   END;
81
82   DBMS_OUTPUT.PUT_LINE('Day: ' || day_name);
83 END;
84 /
85

```

Query result    **Script output**    DBMS output    Explain Plan    SQL history



Day: Tuesday



PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007

## **CONCLUSION:**

This experiment provided hands-on experience with conditional control statements in PL/SQL. The use of IF-ELSE, ELSIF ladder, and CASE statements helped in understanding decision-making mechanisms and control flow within PL/SQL programs.