Experiment 4

Student Name: Diwansh
UID: 24BAI70294
Branch: CSE - AIML
Section/Group: 24AIT_KRG G1
Semester: 4
Date of Performance: 4/2/26
Subject Name: DBMS
Subject Code: 24CSH-298

Aim

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

Software Requirements

Database Management System: PostgreSQL
Database Administration Tool: pgAdmin

Objectives

- Implement control structures in PL/SQL blocks.
- Understand how conditional statements guide execution flow.

Problem Statement

Develop and execute PL/SQL programs demonstrating IF–ELSE, IF–ELSIF–ELSE, ELSIF ladder, and CASE statements to illustrate decision-making.

1. Problem Statement – IF–ELSE Statement

Write a PL/SQL program to determine whether a given day number represents a **weekday or weekend** using the IF–ELSE statement and display an appropriate message.

---

2. Problem Statement – IF–ELSIF–ELSE Statement

Write a PL/SQL program to check whether a number is **positive, negative, or zero** using IF–ELSIF–ELSE and display the result.

---

3. Problem Statement – ELSIF Ladder

Write a PL/SQL program to determine the **category of a person based on age** using an ELSIF ladder and display the appropriate category.

---

4. Problem Statement – CASE Statement

Write a PL/SQL program to determine the **season based on a given month number** using the CASE conditional statement.

Practical / Experiment Steps

- Designed multiple PL/SQL blocks to explore diverse conditional logic formats, including simple branching and multi-path evaluation.
- Utilised IF–ELSE to classify days as weekday or weekend.
- Applied IF–ELSIF–ELSE to categorize numbers as positive, negative, or zero.
- Used an ELSIF ladder to classify age groups into meaningful categories.
- Implemented the CASE statement to map month numbers to seasons efficiently.

• Integrated variable-driven output strings to provide real-time feedback.
• Validated the decision-making capabilities by testing different input values.

Procedure

• Enabled the server output environment to ensure results were visible.
• Constructed an IF–ELSE block to check if a day number represents weekend or weekday.
• Developed an IF–ELSIF–ELSE block to identify whether a number is positive, negative, or zero.
• Expanded the logic into an ELSIF ladder to categorize age into Child, Teenager, Adult, and Senior Citizen.
• Implemented a CASE statement to convert month numbers into seasons.
• Initialised various test values for validation.
• Executed each block and observed the DBMS output console.
• Verified outputs with expected logical results.

---

1. Problem Statement – IF–ELSE Statement

Write a PL/SQL program to determine whether a given day is a **weekday or weekend**.

```
DECLARE
    day_no NUMBER := 6;
BEGIN
    IF day_no = 6 OR day_no = 7 THEN
        DBMS_OUTPUT.PUT_LINE('It is Weekend.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('It is a Weekday.');
    END IF;
END;
/
```

Output:

```
SQL> DECLARE
         day_no NUMBER := 6;
     BEGIN
         IF day_no = 6 OR day_no = 7 THEN...
Show more...


It is Weekend.


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.005
```

---

2. Problem Statement – IF–ELSIF–ELSE Statement

Write a PL/SQL program to check whether a number is **positive, negative, or zero**.

```
DECLARE
    n NUMBER := 0;
BEGIN
    IF n > 0 THEN
        DBMS_OUTPUT.PUT_LINE('The number ' || n || ' is Positive.');
    ELSIF n < 0 THEN
        DBMS_OUTPUT.PUT_LINE('The number ' || n || ' is Negative.');
    ELSE
        DBMS_OUTPUT.PUT_LINE('The number is Zero.');
    END IF;
END;
/
```

Output:

```
SQL> DECLARE
         n NUMBER := 0;
     BEGIN
         IF n > 0 THEN...
Show more...


The number is Zero.
```

---

3. Problem Statement – ELSIF Ladder

Write a PL/SQL program to determine **age category**.

```
DECLARE
    age NUMBER := 22;
BEGIN
    IF age >= 60 THEN
        DBMS_OUTPUT.PUT_LINE('Category: Senior Citizen');
    ELSIF age >= 20 THEN
        DBMS_OUTPUT.PUT_LINE('Category: Adult');
    ELSIF age >= 13 THEN
        DBMS_OUTPUT.PUT_LINE('Category: Teenager');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Category: Child');
    END IF;
END;
```

Output:

```
SQL> DECLARE
         age NUMBER := 22;
     BEGIN
         IF age >= 60 THEN...
Show more...


Category: Adult


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007
```

---

4. Problem Statement – CASE Statement

Write a PL/SQL program to determine **season based on month number**.

```
DECLARE
    m NUMBER := 12;
    season VARCHAR2(20);
BEGIN
    season := CASE
                WHEN m IN (12, 1, 2) THEN 'Winter'
                WHEN m IN (3, 4, 5) THEN 'Summer'
                WHEN m IN (6, 7, 8) THEN 'Monsoon'
                WHEN m IN (9, 10, 11) THEN 'Autumn'
                ELSE 'Invalid Month'
              END;

    DBMS_OUTPUT.PUT_LINE('Season: ' || season);
END;
```

Output:

```
SQL> DECLARE
         m NUMBER := 12;
         season VARCHAR2(20);
     BEGIN...
Show more...


Season: Winter


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007
```

Learning Outcomes

- Learned to use IF–ELSE for binary decisions.
- Used IF–ELSIF–ELSE for multi-condition checks.
- Understood ELSIF ladder for range-based classification.
- Applied CASE for clean value mapping.