

# Experiment 4

Student Name: SEJAL  
Branch: CSE - AIML  
Semester: 4  
Subject Name: DBMS

UID: 24BAI70050  
Section/Group: 24ait\_Krg G1  
Subject Code: 24CSH-298

## Aim

To design and implement PL/SQL programs utilizing conditional control statements such as IF–ELSE, ELSIF, 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

## Software Requirements

- PostgreSQL
- pgAdmin

## Objectives

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

## Problem Statement

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

## Practical/Experiment Steps

- Start SQL\*Plus / Oracle Live SQL.
- Enter SET SERVEROUTPUT ON.
- Copy and run the above PL/SQL program.
- Enter the required values when prompted.
- Observe the displayed output.

### **Procedure**

- Open Oracle SQL\*Plus / Oracle Live SQL.
- Enable output using SET SERVEROUTPUT ON.
- Declare required variables in the DECLARE section.
- Use conditional control statements (IF-ELSE, IF-ELSIF-ELSE, ELSIF ladder).
- Display results using DBMS\_OUTPUT.PUT\_LINE.
- Execute the program and observe the output.

### **Input/Output Analysis**

## **PROBLEM STATEMENT 1**

```

DECLARE
    num NUMBER := 4;
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**

```
SQL> DECLARE
      num NUMBER := 4;
    BEGIN
      IF num > 0 THEN...
Show more...

The number is POSITIVE

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.005
```

## **PROBLEM STATEMENT 2**

```
DECLARE
  marks NUMBER := &marks;
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');
  ELSIF marks >= 40 THEN
    DBMS_OUTPUT.PUT_LINE('Grade: D');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Grade: F (Fail)');
  END IF;
END;
/
```

### **Output**

```
SQL> DECLARE
      marks NUMBER := 13;
    BEGIN
      IF marks >= 90 THEN...
Show more...

Grade: F (Fail)

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.013
```

## **PROBLEM STATEMENT 3**

```
DECLARE
  marks NUMBER := &marks;
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;
/
```

## Output

```
SQL> DECLARE
      marks NUMBER := 55;
    BEGIN
        IF marks >= 85 THEN...
Show more...
```

```
Performance: Good
```

```
PL/SQL procedure successfully completed.
```

```
Elapsed: 00:00:00.010
```

## Learning Outcomes

- Understand the structure of a **PL/SQL block** (DECLARE, BEGIN, END).
- Use **conditional control statements** such as IF-ELSE, IF-ELSIF-ELSE, and ELSIF ladder in PL/SQL.
- Accept **user input** using substitution variables.
- Implement **decision-making logic** in database programs.
- Display output using the DBMS\_OUTPUT.PUT\_LINE statement.
- Analyze and classify data based on given conditions.
- Develop basic problem-solving skills using PL/SQL.