

Student Name: KUSHAGRA VASHISTHA

UID:24BAI70915

Branch: Computer Science & Engineering (AIML)

Section/Group:24AIT-KRG1/G2

Semester:4th

Subject Name: DBMS

1. Aim

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

2. Objective of the Session

1. To understand decision-making constructs in PL/SQL.
2. To implement **IF-ELSE**, **IF-ELSIF-ELSE**, and **ELSIF ladder** statements.
3. To apply **CASE statements** for multi-way branching.
4. To analyze program flow based on conditional evaluation.
5. To gain hands-on experience with **PL/pgSQL blocks in PostgreSQL**.

3. Theory

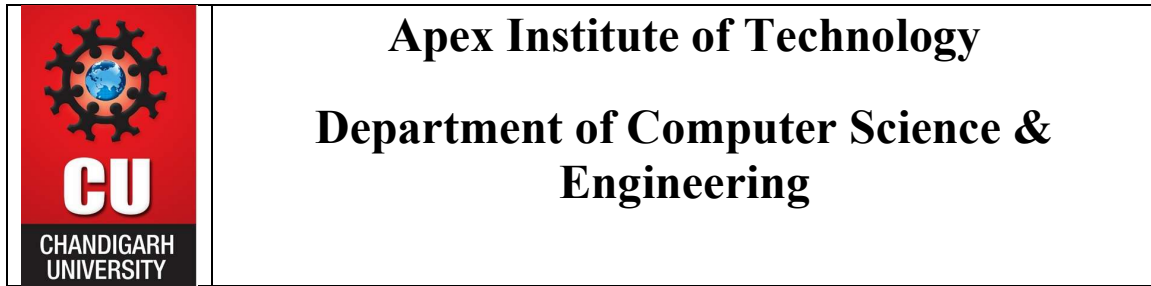
Conditional control statements allow a PL/SQL program to **make decisions** and execute different blocks of code depending on logical conditions.

1. IF-ELSE Statement

Used when there are two possible outcomes.

IF condition THEN

statements;



ELSE

statements;

END IF;

2. IF-ELSIF-ELSE Statement

Used when multiple mutually exclusive conditions are evaluated sequentially.

IF condition1 THEN

statements;

ELSIF condition2 THEN

statements;

ELSE

statements;

END IF;

3. ELSIF Ladder

An extension of IF-ELSIF-ELSE with multiple conditions checked in order. Execution stops at the first true condition.

4. CASE Statement

Provides a cleaner alternative to long ELSIF ladders.

CASE expression

WHEN value1 THEN statements;

WHEN value2 THEN statements;

ELSE statements;

END CASE;

4. Problem Statement

Develop and execute PL/SQL programs that demonstrate the use of **conditional control statements**. The programs must utilize **IF–ELSE**, **IF–ELSIF–ELSE**, **ELSIF ladder**, and **CASE** statements to evaluate conditions and control execution flow accordingly.

5. Procedure of the Practical

1. Start the system and open **Oracle SQL*Plus / Oracle SQL Developer**.
2. Connect to the Oracle database using valid **username** and **password**.
3. Enable output display by executing the command:
4. SET SERVEROUTPUT ON;
5. Open a new SQL worksheet.
6. Write the required **PL/SQL block** using the structure:

```
DECLARE  
  
variable declarations  
  
BEGIN  
  
executable statements  
  
END;
```

7. Declare necessary variables using appropriate **data types**.
8. Apply conditional control statements such as:
 - IF–ELSE
 - IF–ELSIF–ELSE
 - ELSIF ladder
 - CASE statement
9. Use DBMS_OUTPUT.PUT_LINE to display the output.
10. Execute the program by pressing **Run** or typing /.
11. Observe the output displayed in the **Output / DBMS Output window**.
12. Verify the correctness of results based on given conditions.
13. Repeat the steps for all problem statements.

6. I/O Analysis (Input / Output Analysis)

Input Output table:

Program Number	Input	Output
Program 1	num = 5	The number 5 is Positive
Program 2	marks = 82	Grade: B
Program 3	marks = 68	Performance: Good
Program 4	day_no = 3	Wednesday

Analysis

- IF-ELSE provides binary decision control.
- IF-ELSIF-ELSE enables multi-condition evaluation.
- ELSIF ladder executes only the first true condition, improving efficiency.
- CASE statements improve readability and reduce logical complexity.
- PL/pgSQL control structures closely resemble structured programming constructs.

7.SQL Implementation (PgAdmin / PostgreSQL)

- **IF-ELSE Statement**

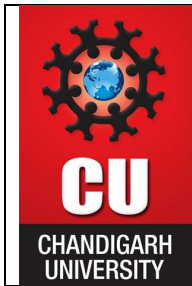
```

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

- **IF-ELSIF-ELSE Statement**

```

DECLARE
```



Apex Institute of Technology

Department of Computer Science & Engineering

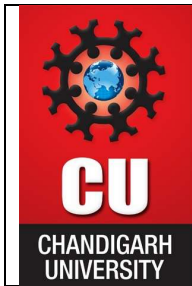
```
marks NUMBER := 82;
BEGIN
  IF marks >= 90 THEN
    DBMS_OUTPUT.PUT_LINE('Grade : A');
  ELSIF marks >= 75 THEN
    DBMS_OUTPUT.PUT_LINE('Grade : B');
  ELSIF marks >= 60 THEN
    DBMS_OUTPUT.PUT_LINE('Grade : C');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Grade : D');
  END IF;
END;
```

- **ELSIF Ladder**

```
DECLARE
  marks NUMBER := 68;
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 : Fail');
  END IF;
END;
```

- **CASE Statement**

```
DECLARE
  day_no NUMBER := 3;
```



Apex Institute of Technology

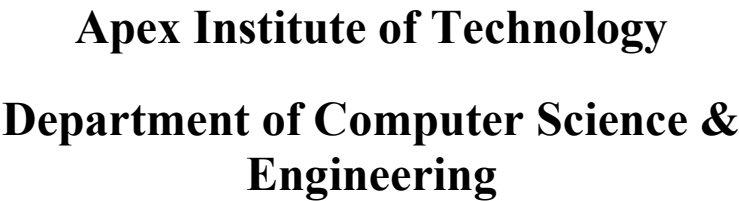
Department of Computer Science & Engineering

```
BEGIN
CASE day_no
  WHEN 1 THEN DBMS_OUTPUT.PUT_LINE('Monday');
  WHEN 2 THEN DBMS_OUTPUT.PUT_LINE('Tuesday');
  WHEN 3 THEN DBMS_OUTPUT.PUT_LINE('Wednesday');
  WHEN 4 THEN DBMS_OUTPUT.PUT_LINE('Thursday');
  WHEN 5 THEN DBMS_OUTPUT.PUT_LINE('Friday');
  WHEN 6 THEN DBMS_OUTPUT.PUT_LINE('Saturday');
  WHEN 7 THEN DBMS_OUTPUT.PUT_LINE('Sunday');
  ELSE
    DBMS_OUTPUT.PUT_LINE('Invalid Day Number');
  END CASE;
END;
```

8. Learning Outcomes

After completing this experiment, the student is able to:

- Implement conditional logic in PL/SQL programs.
- Select appropriate control statements for decision-making.
- Understand execution flow in PL/pgSQL blocks.
- Write structured, readable, and efficient PL/SQL code.
- Apply decision-making constructs to real-world database problems.



[SQL Worksheet]*

```

11
12 DECLARE
13     marks NUMBER := 82;
14 BEGIN
15     IF marks >= 90 THEN
16         DBMS_OUTPUT.PUT_LINE('Grade : A');
17     ELSIF marks >= 75 THEN
18         DBMS_OUTPUT.PUT_LINE('Grade : B');
19     ELSIF marks >= 60 THEN
20         DBMS_OUTPUT.PUT_LINE('Grade : C');
21     ELSE
22         DBMS_OUTPUT.PUT_LINE('Grade : D');
23     END IF;
24 END;
```

DBMS output

Grade : B



Apex Institute of Technology

Department of Computer Science & Engineering

```
26 DECLARE
27     marks NUMBER := 68;
28 BEGIN
29     IF marks >= 85 THEN
30         DBMS_OUTPUT.PUT_LINE('Performance : Excellent');
31     ELSIF marks >= 70 THEN
32         DBMS_OUTPUT.PUT_LINE('Performance : Very Good');
33     ELSIF marks >= 55 THEN
34         DBMS_OUTPUT.PUT_LINE('Performance : Good');
35     ELSIF marks >= 40 THEN
36         DBMS_OUTPUT.PUT_LINE('Performance : Average');
37     ELSE
38         DBMS_OUTPUT.PUT_LINE('Performance : Fail');
39     END IF;
40 END;
```

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



Performance : Good









Apex Institute of Technology

Department of Computer Science & Engineering

```
✓ DECLARE
    day_no NUMBER := 3;
✓ BEGIN
    CASE day_no
    WHEN 1 THEN DBMS_OUTPUT.PUT_LINE('Monday');
    WHEN 2 THEN DBMS_OUTPUT.PUT_LINE('Tuesday');
    WHEN 3 THEN DBMS_OUTPUT.PUT_LINE('Wednesday');
    WHEN 4 THEN DBMS_OUTPUT.PUT_LINE('Thursday');
    WHEN 5 THEN DBMS_OUTPUT.PUT_LINE('Friday');
    WHEN 6 THEN DBMS_OUTPUT.PUT_LINE('Saturday');
    WHEN 7 THEN DBMS_OUTPUT.PUT_LINE('Sunday');
    ELSE
    DBMS_OUTPUT.PUT_LINE('Invalid Day Number');
    END CASE;
END;
```

Query result	Script output	DBMS output	Explain Plan	SQL history
<div> </div> <div>Wednesday</div> <div> </div>				