Experiment 4: Conditional Control Statements in PL/SQL

---

1. Aim of the Session

The aim of this practical is to design and implement PL/SQL programs using **conditional control statements** such as **IF–ELSE**, **IF–ELSIF–ELSE**, **ELSIF ladder**, and **CASE statements** in order to control the flow of execution based on logical conditions.

Purpose of the Practical:

- To understand decision-making constructs in PL/SQL
- To learn how conditional statements control program flow
- To analyze different types of conditional structures
- To implement real-world decision logic using PL/SQL

---

2. Objective of the Session

The specific objectives of this experiment are:

- To understand conditional control statements in PL/SQL
- To learn the syntax and usage of IF–ELSE and IF–ELSIF–ELSE
- To implement ELSIF ladder for multi-condition evaluation
- To use CASE statements for fixed-value comparisons

Upon completion, students will be able to:

- Write PL/SQL programs involving conditional logic
- Select appropriate control structures based on problem requirements
- Implement decision-making logic efficiently
- Execute and debug PL/SQL programs in Oracle environment

---

3. Practical / Experiment Description

This experiment demonstrates the use of **conditional control structures** in PL/SQL, which allow the execution of different code blocks depending on whether specified conditions evaluate to TRUE or FALSE.

The experiment includes the following conditional constructs:

- **IF–ELSE** – Used for binary decision making
- **IF–ELSIF–ELSE** – Used for multiple condition evaluation
- **ELSIF Ladder** – Used to evaluate a range of conditions sequentially

- **CASE Statement** – Used for fixed-value comparison

The program performs various checks such as:

- Determining whether a number is positive or non-positive
- Evaluating grades based on marks
- Determining performance status
- Displaying the day name based on day number

---

4. Procedure of the Practical

Follow the steps given below to execute the experiment:

(i) Start the System and Open DBMS

- Power on the system
- Open **Oracle Live SQL / Oracle SQL Developer**
- Connect using valid database credentials

(ii) Enable Output Display

```
SET SERVEROUTPUT ON;
```

This command enables the display of output generated by PL/SQL programs.

(iii) Write the PL/SQL Program

```
SET SERVEROUTPUT ON;


DECLARE
   num      NUMBER := 5;
   marks1   NUMBER := 82;
   marks2   NUMBER := 68;
   day_num   NUMBER := 4;
BEGIN
   IF num > 0 THEN
      DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is POSITIVE');
   ELSE
      DBMS_OUTPUT.PUT_LINE('The number ' || num || ' is NON-POSITIVE');
```

```
END IF;

IF marks1 >= 90 THEN
   DBMS_OUTPUT.PUT_LINE('Grade: A');
ELSIF marks1 >= 75 THEN
   DBMS_OUTPUT.PUT_LINE('Grade: B');
ELSIF marks1 >= 60 THEN
   DBMS_OUTPUT.PUT_LINE('Grade: C');
ELSE
   DBMS_OUTPUT.PUT_LINE('Grade: FAIL');
END IF;

IF marks2 >= 85 THEN
   DBMS_OUTPUT.PUT_LINE('Performance: Excellent');
ELSIF marks2 >= 70 THEN
   DBMS_OUTPUT.PUT_LINE('Performance: Very Good');
ELSIF marks2 >= 55 THEN
   DBMS_OUTPUT.PUT_LINE('Performance: Good');
ELSIF marks2 >= 40 THEN
   DBMS_OUTPUT.PUT_LINE('Performance: Average');
ELSE
   DBMS_OUTPUT.PUT_LINE('Performance: Poor');
END IF;

CASE day_num
   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;
/
```

(iv) Execute the Program

- Run the program using /
- Observe the output displayed in the output window

(v) Verify the Result

- Ensure correct messages are displayed for each condition
- Confirm that control statements execute as expected

---

5. Input / Output Analysis

Inputs Provided:

- num = 5
- marks1 = 82
- marks2 = 68
- day_num = 4

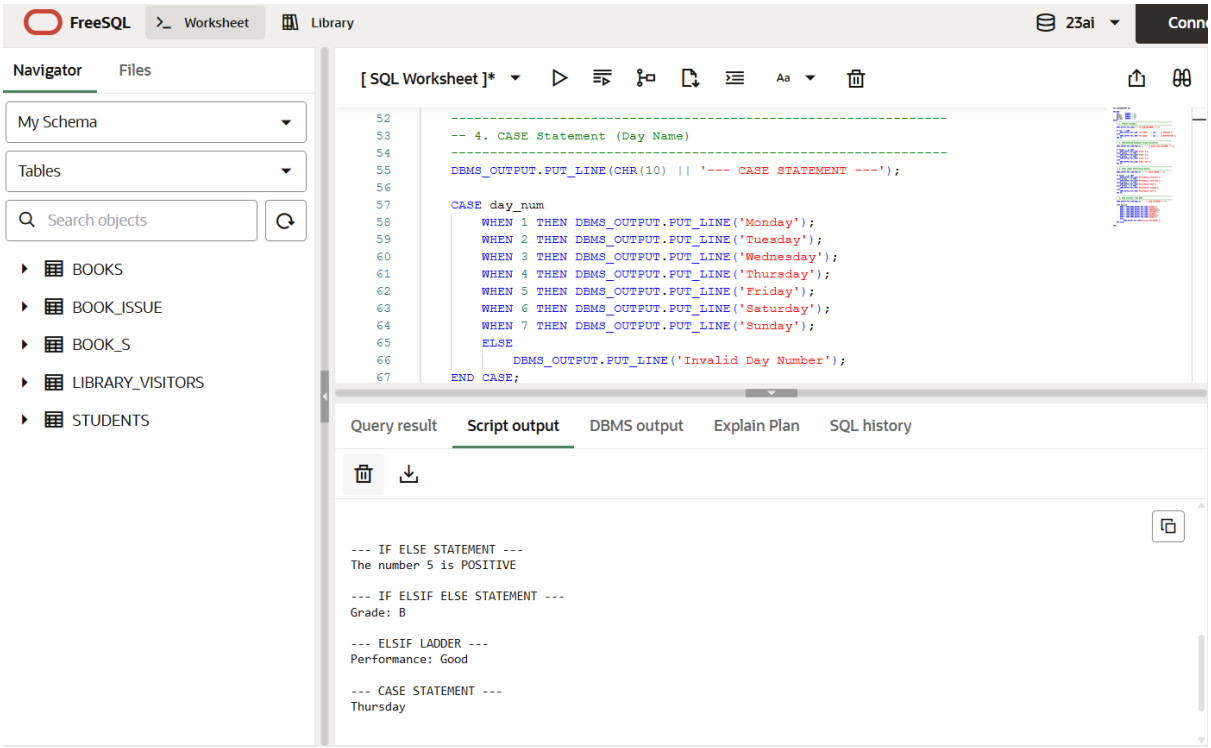
Operations Performed:

- Comparison of numeric values using IF–ELSE
- Grade evaluation using IF–ELSIF–ELSE
- Performance analysis using ELSIF ladder
- Day selection using CASE statement

Output Generated:

- Number is POSITIVE
- Grade: B
- Performance: Good
- Thursday

SCREENSHOTS OF THE OUTPUT ARE ATTACHED BELOW



---

6. Learning Outcomes

Concepts Understood:

- Conditional control statements in PL/SQL
- Logical comparison and branching
- Syntax and usage of IF, ELSIF, and CASE

Skills Developed:

- Writing structured PL/SQL programs
- Implementing decision-making logic
- Using DBMS_OUTPUT for displaying results
- Debugging conditional statements

Practical Exposure Gained:

- Hands-on experience with PL/SQL control structures
- Understanding execution flow in procedural SQL
- Working with Oracle PL/SQL environment

---

7. Result

Thus, PL/SQL programs demonstrating **IF–ELSE**, **IF–ELSIF–ELSE**, **ELSIF ladder**, and **CASE statements** were successfully designed, executed, and verified. The experiment effectively illustrated decision-making capabilities in PL/SQL.