Experiment 4

Student Name: Srijan Tripathi
UID: 24BAI70091
Branch: CSE - AIML
Section/Group: 24AIT_KRG G1
Semester: 4
Date of Performance: 4/2/26
Subject Name: DBMS
Subject Code: 24CSH-298

Aim

This practical focuses on building decision-making logic in procedural database code using conditional branching techniques. The goal is to create programs that evaluate conditions and execute specific code paths based on variable values and establish problem-solving skills with complex nested decision trees.

Software Requirements

- Database Management System:

Oracle SQL

Objectives

- Build working PL/SQL routines with multiple branching scenarios (IF-ELSE, ELSIF chains, CASE).

Problem Statement

Problem 1: Sign Detection
Write a PL/SQL program to check whether a given number is positive, zero, or negative and display the appropriate message.

Problem 2: Score-Based Grading System
Write a PL/SQL program to assign letter grades (A, B, C, D, E) based on numerical scores using nested IF-ELSIF conditions.

Problem 3: Performance Classification

Write a PL/SQL program to classify performance levels (Honours, Merit, Average, Satisfactory, Unsuccessful) based on numerical marks using multi-level IF-ELSIF-ELSE ladder.

Problem 4: Weekday Lookup
Write a PL/SQL program to convert numeric day identifiers (1-7) into corresponding weekday names using a CASE statement with error handling.

Practical/Experiment Steps

- Initialize the Oracle database environment and configure DBMS output display.
- Create a simple IF-ELSE block to test numeric values for sign classification.
- Develop an IF-ELSIF-ELSE decision tree for multi-threshold score evaluation.
- Build a comprehensive ELSIF ladder with multiple boundary conditions.
- Implement a CASE expression for discrete value-to-text mapping.
- Set up test data covering normal cases, boundary values, and edge cases.
- Wrap each logical routine within standard BEGIN...END executable blocks.
- Execute all programs and capture console output results.
- Verify output accuracy against manual trace-through of logic paths.
- Document the decision flow and branch coverage for each program.

Procedure

- Enable the DBMS output console to display procedural execution results.
- Construct the first IF-ELSE block to validate a numeric value for sign classification (positive vs non-positive).
- Create the second program using IF-ELSIF-ELSE chains to evaluate scores within defined threshold ranges.
- Develop the third program with expanded ELSIF logic to categorize marks into performance tiers.
- Write the fourth program using CASE expressions as a streamlined alternative for enumeration mapping.
- Initialize test variables with specific values to exercise each decision branch.
- Structure each routine within a PL/SQL block (BEGIN...END).
- Run each program sequentially and monitor output for correctness.

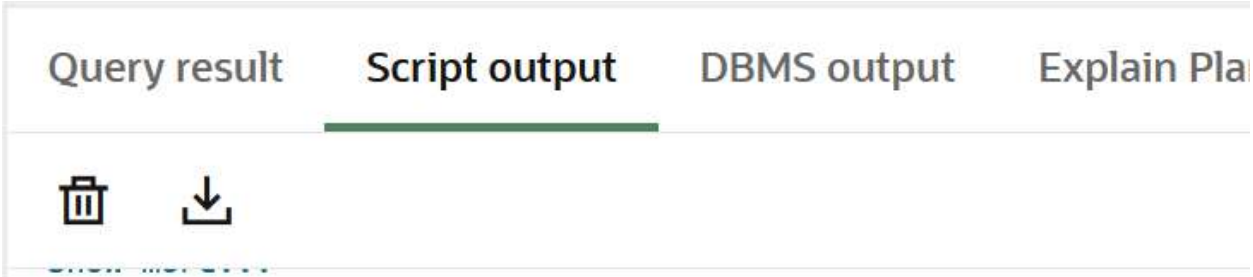
- Cross-validate output against expected results from manual condition evaluation.
- Verify that all logical paths were correctly executed based on input values.

Input/Output Analysis

```
DECLARE
   val NUMBER := -21;
BEGIN
   IF val < 1 THEN
      DBMS_OUTPUT.PUT_LINE('Value is zero or negative');
   ELSE
      DBMS_OUTPUT.PUT_LINE('Value is greater than zero');
   END IF;
END;
DECLARE
   score NUMBER := 68;
   letter CHAR(1);
BEGIN
   CASE
      WHEN score >= 90 THEN letter := 'A';
      WHEN score >= 80 THEN letter := 'B';
      WHEN score >= 70 THEN letter := 'C';
      WHEN score >= 60 THEN letter := 'D';
      ELSE letter := 'E';
   END CASE;
   DBMS_OUTPUT.PUT_LINE('Score: ' || score || ' | Grade: ' || letter);
END;
DECLARE
   result NUMBER := 38;
   category VARCHAR2(25);
BEGIN
   IF result >= 75 THEN
      category := 'Honours';
   ELSIF result >= 60 THEN
      category := 'Merit';
   ELSIF result >= 50 THEN
      category := 'Average';
```

```
    ELSIF result >= 35 THEN
        category := 'Satisfactory';
    ELSE
        category := 'Unsuccessful';
    END IF;
    DBMS_OUTPUT.PUT_LINE('Result: ' || result || ' - Category: ' || category);
END;
DECLARE
    weekday_num NUMBER := 3;
    weekday_name VARCHAR2(15);
BEGIN
    weekday_name := CASE weekday_num
        WHEN 1 THEN 'Sunday'
        WHEN 2 THEN 'Monday'
        WHEN 3 THEN 'Tuesday'
        WHEN 4 THEN 'Wednesday'
        WHEN 5 THEN 'Thursday'
        WHEN 6 THEN 'Friday'
        WHEN 7 THEN 'Saturday'
        ELSE 'Not a valid weekday'
    END;
    DBMS_OUTPUT.PUT_LINE('Day: ' || weekday_name);
END;
```

Output




```
Day: Tuesday

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.007
```

```
Value is zero or negative

PL/SQL procedure successfully completed.
```

```
Show more...

Score: 68 | Grade: D

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.008
```

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

```
        category VARCHAR2(25);
    BEGIN...
Show more...

Result: 38 - Category: Satisfactory

PL/SQL procedure successfully completed.
```

Learning Outcomes

- Conditional Control Mastery: Proficiency in constructing IF-ELSE, IF-ELSIF-ELSE ladders, and CASE statements to direct program execution based on logical conditions.
- Multi-Path Logic Design: Understanding how to structure complex decision trees with multiple evaluation thresholds and boundary conditions.
- Value Mapping Efficiency: Recognition that CASE statements provide cleaner, more readable syntax for enumeration mapping compared to long IF-ELSIF chains.

- Threshold-Based Classification: Skills in defining and implementing logical thresholds for categorizing continuous numerical values into discrete classifications.
- Variable Scope and State Management: Understanding how variable initialization, assignment, and scope work within PL/SQL executable blocks.