# Apex Institute of Technology Computer Science & Engineering

## Experiment 3

**Name: Aman Raj Ojha** **UID: 24BAI70520**

**Branch: B.E. CSE (AIML)** **Section: 24AIT_KRG-G1**

**Semester: 4** **Date of Performance: 28.01.2026**

**Subject Name: Database Management System** **Subject Code: 24CSH-298**

**AIM:** To understand the basic structure of a PL/SQL program by creating and executing a simple PL/SQL block that includes declaration and execution sections, and to display output using built-in procedures.

### OBJECTIVES:

- To create a simple PL/SQL program demonstrating **Declaration Section** and **Execution Section**.
- To understand the use of conditional statements such as **IF–ELSIF–ELSE** in PL/SQL for decision-based operations.

### SOFTWARE REQUIREMENTS:

- Oracle FreeSQL

### PRACTICAL/EXPERIMENT STEPS:

# Apex Institute of Technology Computer Science & Engineering

1. The PL/SQL block is started using the DECLARE keyword to define required variables.
2. Variables such as employee ID, employee name, and employee salary are declared with appropriate data types and initialized with values.
3. Employee details like ID, name, and salary are displayed using the DBMS_OUTPUT.PUT_LINE procedure.
4. An IF–ELSIF–ELSE conditional structure is used to classify salary status based on different salary ranges.
5. If the salary is greater than or equal to 60000, the employee is considered a high salary employee.
6. If the salary is between 40000 and 59999, the employee is considered a medium salary employee.
7. Otherwise, the employee is considered a low salary employee.
8. The salary status is displayed as output.
9. The PL/SQL block ends successfully after execution.

## PROCEDURE:

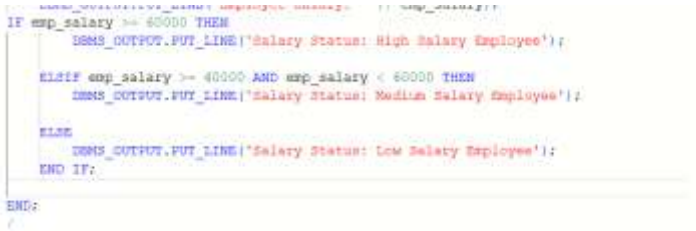
1. Start the Oracle FreeSQL environment.
2. Write the following code to declare the variables.

```
DECLARE
    emp_id     NUMBER := 101;
    emp_name   VARCHAR2(50) := 'Aman Raj Ojha';
    emp_salary NUMBER := 45000;
```

3. Inside the BEGIN block, write the following code to display the employee data usingDBMS_OUTPUT.PUT_LINE.

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Employee Details');
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_name);
    DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || emp_salary);
```

4. Use IF-ELSIF-ELSE conditional statement to determine the category based on the employee salary.



5. End the PL/SQL block using the END; statement and execute the program.


6. Verify the output displayed on the screen.

## CODE:

```
DECLARE
   emp_id     NUMBER := 101;
   emp_name   VARCHAR2(50) := 'Aman Raj Ojha';
   emp_salary NUMBER := 45000;

BEGIN
   DBMS_OUTPUT.PUT_LINE('Employee Details');
   DBMS_OUTPUT.PUT_LINE('----------------------');
```

```
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_id);
    DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp_name);
    DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || emp_salary);

    IF emp_salary >= 60000 THEN
        DBMS_OUTPUT.PUT_LINE('Salary Status: High Salary Employee');

    ELSIF emp_salary >= 40000 AND emp_salary < 60000 THEN
        DBMS_OUTPUT.PUT_LINE('Salary Status: Medium Salary Employee');

    ELSE
        DBMS_OUTPUT.PUT_LINE('Salary Status: Low Salary Employee');
    END IF;

END;
/
```

## I/O ANALYSIS:

This PL/SQL program demonstrates the basic structure of a PL/SQL block. It declares variables for employee details under the DECLARE block and uses an IF–ELSIF–ELSE conditional statement to determine the salary status of the employee. The program displays employee ID,

name, salary, and salary

```
Employee Details
Employee ID: 101
Employee Name: Aman Raj Ojha
Employee Salary: 45000
Salary Status: Medium Salary Employee


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.009
```

**LEARNING OUTCOMES:**

1. Understand the basic structure of a PL/SQL block, including the Declaration section and Execution section.
2. Gain hands-on experience in declaring variables and performing decision-based operations.
3. Learn to use conditional statements like IF–ELSIF–ELSE in PL/SQL.
4. Learn to display output using built-in PL/SQL procedures such as DBMS_OUTPUT.PUT_LINE.