# Experiment 3

Student Name: Nandesh T

Branch: CSE - AIML

Semester: 4

Subject Name: DBMS

UID: 24BAI70121

Section/Group: 24AIT_KRG G1

Date of Performance: 28/1/26

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.

## Software Requirements

- Database Management System:
    - PostgreSQL
- Database Administration Tool:
    - pgAdmin

## Objectives

- To create a simple PL/SQL program demonstrating Declaration Section and Execution Section.

## Problem Statement

Design and implement a simple PL/SQL program that demonstrates the basic structure of a PL/SQL block. The program should include a declaration section to define variables and an execution section to perform operations using those variables and display the results using appropriate output statements.

Declaration Section (DECLARE)

Variables are declared and initialized:

- emp_id → Employee ID
- emp_name → Employee Name
- emp_salary → Employee Salary

Execution Section (BEGIN ... END)
DBMS_OUTPUT.PUT_LINE is used to display output.

## Practical/Experiment Steps

- Block Structuring: Designed a foundational PL/SQL block consisting of a Declaration section for memory allocation and an Execution section for logic processing.
- Variable Initialization: Defined and assigned static values to organizational variables, including Employee ID, Name, and Salary, to simulate a single-record data environment.
- Computational Logic: Integrated an arithmetic operation within the block to calculate the House Rent Allowance (HRA) at 40% of the base salary.
- Conditional Processing: Implemented a selection structure using an IF-ELSE statement to evaluate tax liability based on the defined salary threshold.
- Output Orchestration: Utilized the DBMS_OUTPUT.PUT_LINE procedure to format and display the processed information in the console.

## Procedure

- Established a connection to the database environment.
- Initiated the DECLARE section to reserve memory for numerical and character data types.
- Mapped real-world data to the defined variables, such as setting the Employee Name to 'THOMAS' and Salary to 25,000.
- Constructed the BEGIN block to initiate the procedural execution of the code.
- Coded a series of output commands to display the primary employee details and the computed HRA value.
- Applied a conditional logic check to compare the employee's salary against the 60,000 threshold.
- Defined the alternative output paths to inform the user of their tax status based on the boolean result of the salary check.
- Concluded the block with the END; keyword and executed the script to trigger the PL/SQL engine.
- Verified the console output against the manual calculations to ensure the logic and variables were handled correctly.
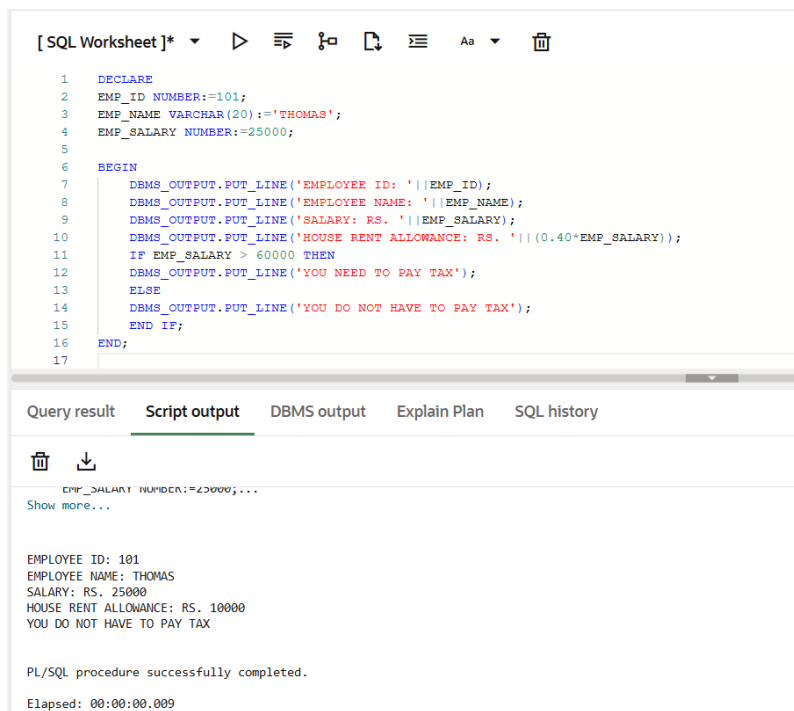
# Input/Output Analysis

## SQL Input Queries

```
DECLARE
EMP_ID NUMBER:=101;
EMP_NAME VARCHAR(20):='THOMAS';
EMP_SALARY NUMBER:=25000;

BEGIN
   DBMS_OUTPUT.PUT_LINE('EMPLOYEE ID: '||EMP_ID);
   DBMS_OUTPUT.PUT_LINE('EMPLOYEE NAME: '||EMP_NAME);
   DBMS_OUTPUT.PUT_LINE('SALARY: RS. '||EMP_SALARY);
   DBMS_OUTPUT.PUT_LINE('HOUSE     RENT     ALLOWANCE:     RS.
'||(0.40*EMP_SALARY));
   IF EMP_SALARY > 60000 THEN
   DBMS_OUTPUT.PUT_LINE('YOU NEED TO PAY TAX');
   ELSE
   DBMS_OUTPUT.PUT_LINE('YOU DO NOT HAVE TO PAY TAX');
   END IF;
END;
```

# Output

# Learning Outcomes

- Understanding the fundamental organisation of PL/SQL declaration and execution sections.
- Learnt to declare, initialise, and use different data types within a procedural block.
- Implementing conditional branching and basic arithmetic operations to process data dynamically.
- Utilising built-in procedures to format and display calculated results to the user.