Experiment 3

Student Name: Bhavik Pahuja	UID: 24BAI70522
Branch: CSE - AIML	Section/Group: 24AIT_KRG G1
Semester: 4	Date of Performance: 28/1/26
Subject Name: DBMS	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

The program is structured to handle the following operations:

- **Memory Allocation:** Initialize variables for Employee ID, Name, and Salary.
- **Arithmetic Processing:** Compute the House Rent Allowance (HRA) as 40% of the base salary:

  HRA = Salary \ 0.40
- **Conditional Evaluation:** Determine tax liability based on a salary threshold of **60,000**.

- **Data Output:** Stream results to the console using the DBMS_OUTPUT package.

Practical/Experiment Steps

1. **Environment Setup:** Connected to the database and enabled SERVEROUTPUT to ensure console visibility.
2. **Variable Definition:** Used the DECLARE section to allocate memory for numerical and character data types.
3. **Data Mapping:** Assigned static values (e.g., Name: 'THOMAS', Salary: 25,000) to simulate a single-record environment.
4. **Logic Execution:** * Initiated the BEGIN block for procedural processing.
5. Applied arithmetic formulas for allowance calculations.
6. Implemented an IF-ELSE structure to provide boolean-based tax status.
7. **Finalization:** Closed the block with the END; keyword and triggered the PL/SQL engine for execution.

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.

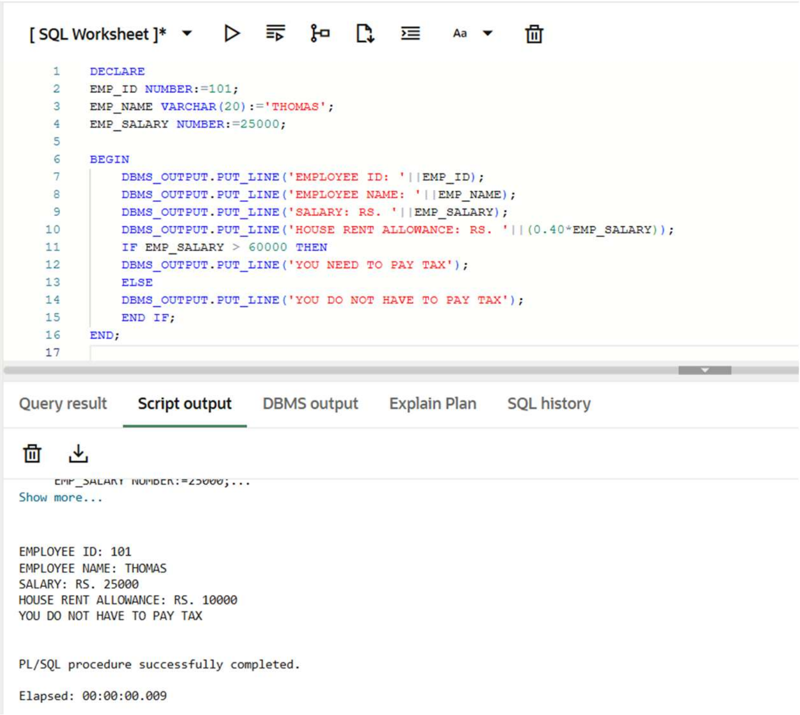
Input/Output Analysis

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



```
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;
```

```
    EMP_SALARY NUMBER:=25000;...
Show more...


EMPLOYEE ID: 101
EMPLOYEE NAME: THOMAS
SALARY: RS. 25000
HOUSE RENT ALLOWANCE: RS. 10000
YOU DO NOT HAVE TO PAY TAX


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.009
```

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.