

Experiment 3

Student Name: Reyansh Arora
Branch: CSE - AIML
Semester: 4
Subject Name: DBMS

UID: 24BAI70273
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 'Reyansh' and Salary to 49000.
- 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

DECLARE

```
EMP_ID    NUMBER := 101;
EMP_NAME  VARCHAR2(20) := 'Reyansh';
EMP_SALARY NUMBER := 49000;
HRA       NUMBER;
```

BEGIN

-- Calculate HRA

```
HRA := EMP_SALARY * 0.40;
```

```
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('HRA (40%)    : Rs. ' || HRA);
```

IF EMP_SALARY > 60000 THEN

```
    DBMS_OUTPUT.PUT_LINE('Tax Status   : Tax applicable');
```

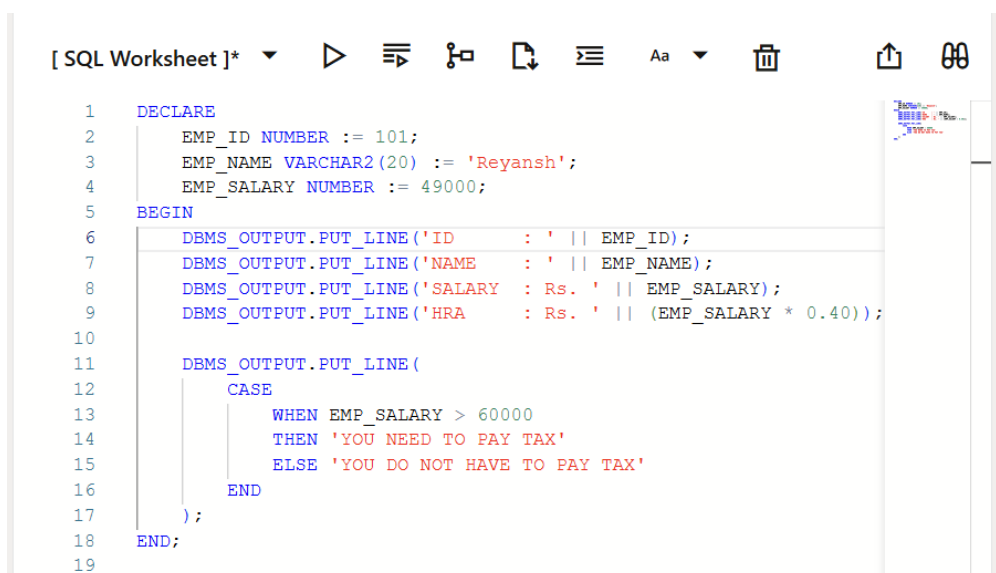
ELSE

```
    DBMS_OUTPUT.PUT_LINE('Tax Status   : No tax applicable');
```

END IF;

END;



Ouput




```
[ SQL Worksheet ]*  ▶  ⌵  🔑  📄  ⌵  Aa  🗑️  📶  🔍

1  DECLARE
2      EMP_ID NUMBER := 101;
3      EMP_NAME VARCHAR2(20) := 'Reyansh';
4      EMP_SALARY NUMBER := 49000;
5  BEGIN
6      DBMS_OUTPUT.PUT_LINE('ID       : ' || EMP_ID);
7      DBMS_OUTPUT.PUT_LINE('NAME     : ' || EMP_NAME);
8      DBMS_OUTPUT.PUT_LINE('SALARY  : Rs. ' || EMP_SALARY);
9      DBMS_OUTPUT.PUT_LINE('HRA    : Rs. ' || (EMP_SALARY * 0.40));
10
11     DBMS_OUTPUT.PUT_LINE(
12         CASE
13             WHEN EMP_SALARY > 60000
14             THEN 'YOU NEED TO PAY TAX'
15             ELSE 'YOU DO NOT HAVE TO PAY TAX'
16         END
17     );
18 END;
19
```

Query resultScript outputDBMS outputExplain PlanSQL history



ID : 101
NAME : Reyansh
SALARY : Rs. 49000
HRA : Rs. 19600
YOU DO NOT HAVE TO PAY TAX



PL/SQL procedure successfully completed.

Elapsed: 00:00:00.008

Learning Outcomes

- Gained knowledge of how a PL/SQL program is organised and executed step-by-step.
- Learned to create and manage variables using suitable data types inside a block.
- Used IF–ELSE conditions and calculations to control the flow of the program.
- Displayed the output using built-in functions to show results to the user.