EXPERIMENT 3

Student Name: ARNAV SHARMA **UID: 24BAI70781**
Branch: AIT-CSE (AIML) **Section/Group: 24AIT_KRG2**
Semester: 4
Subject Name: Database Management System **Subject Code: 24CSH-298**

Experiment 3

Calculating employee salary and applying a bonus using PL/SQL. This experiment demonstrates variable declaration, arithmetic operations, and displaying output using DBMS_OUTPUT.PUT_LINE.

Aim

The aim of this experiment is to practice writing PL/SQL blocks that perform calculations on employee data, calculate bonuses, and display results in a structured format.

Objective

- To declare variables in PL/SQL.
- To perform arithmetic calculations on employee salary.
- To calculate a bonus amount based on a percentage of salary.
- To display employee details and salary information before and after applying the bonus.

Software Requirements

- Database: Oracle XE or Oracle Live SQL

Practical / Experiment Steps

1. Declare variables for employee ID, name, and salary.
2. Calculate a 10% bonus on the employee salary.
3. Calculate the total salary after adding the bonus.
4. Display employee details and salary information before and after bonus.
5. Take screenshots of outputs for documentation.

EXPERIMENT 3

Procedure of the Experiment

1. Start the system and log in.
2. Open Oracle XE or Live SQL.
3. Connect to the database.
4. Declare variables for employee details and bonus calculation.
5. Write the PL/SQL block to calculate bonus and total salary.
6. Execute the PL/SQL block and verify outputs in the console.
7. Take screenshots of the outputs (s1 and s2).

Input / Output Details

Input

- Employee details:
    - emp_id INTEGER := 101;
    - emp_name VARCHAR(50) := 'ADITYA RAJ BHAKTA';
    - emp_salary NUMERIC := 45000;
    - bonus_amount NUMERIC;
    - new_salary NUMERIC;
- Bonus calculation: 10% of employee salary

Output

- Step 1: Display employee details and original salary.
- Step 2: Display bonus amount and total salary after applying 10% bonus.
- Screenshots of outputs (s1 and s2) are attached.

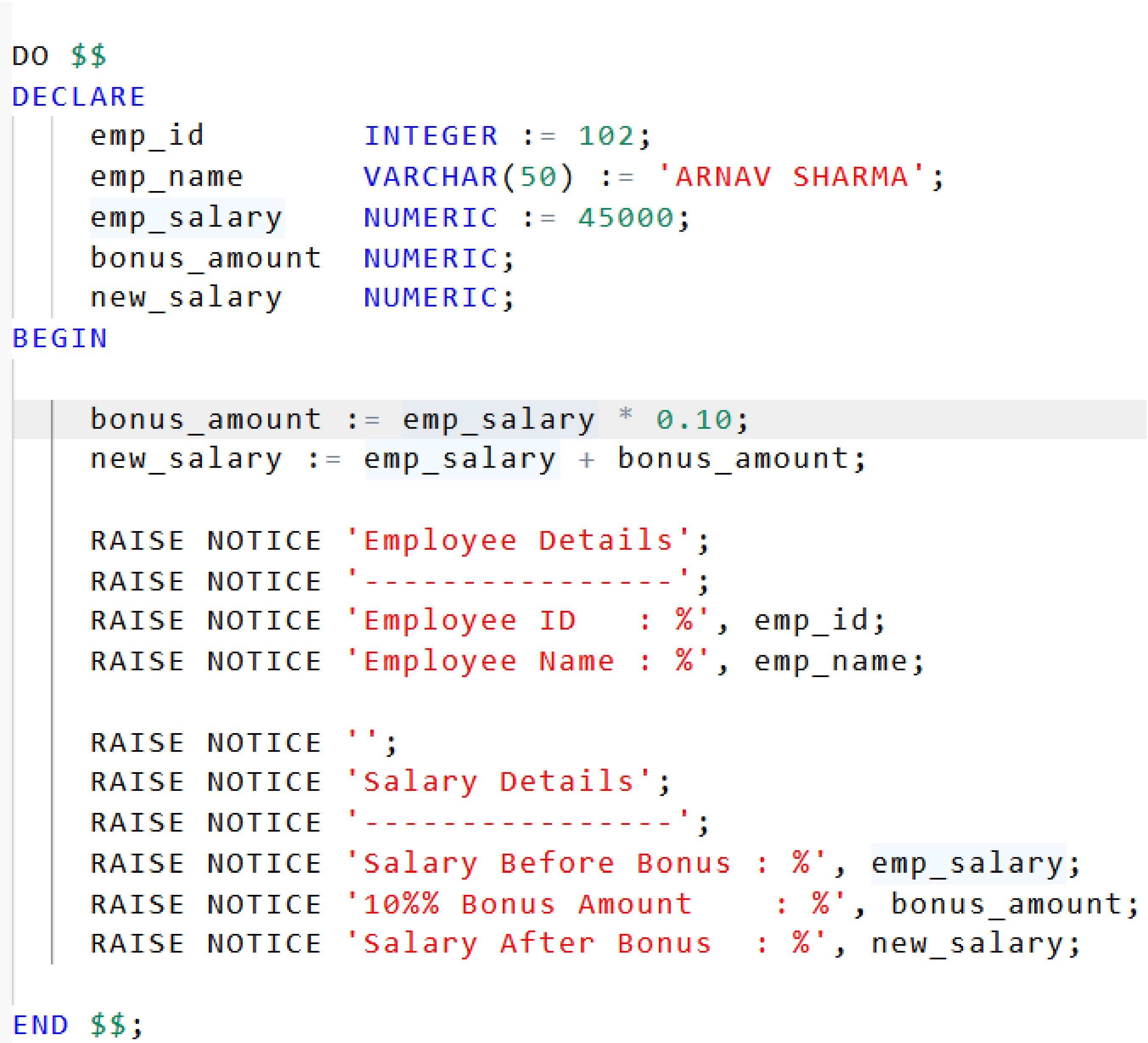
EXPERIMENT 3

Code :

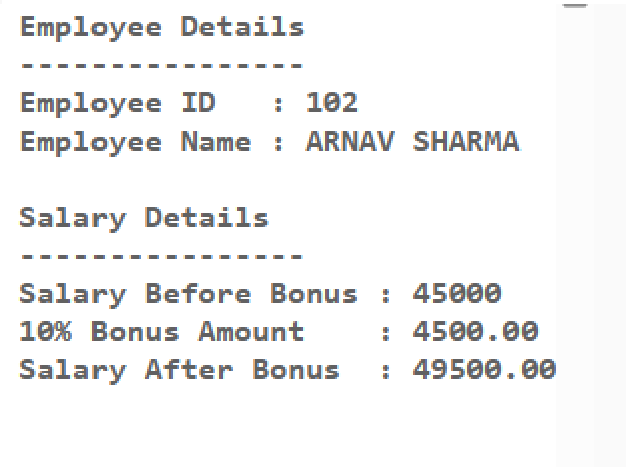
```
DO $$
DECLARE
    emp_id        INTEGER := 102;
    emp_name      VARCHAR(50) := ‘ARNAV SHARMA’;
    emp_salary    NUMERIC := 45000;
    bonus_amount  NUMERIC;
    new_salary    NUMERIC;
BEGIN

    bonus_amount := emp_salary * 0.10;
    new_salary := emp_salary + bonus_amount;

    RAISE NOTICE 'Employee Details';
    RAISE NOTICE '----------------';
    RAISE NOTICE 'Employee ID   : %', emp_id;
    RAISE NOTICE 'Employee Name : %', emp_name;

    RAISE NOTICE '';
    RAISE NOTICE 'Salary Details';
    RAISE NOTICE '----------------';
    RAISE NOTICE 'Salary Before Bonus : %', emp_salary;
    RAISE NOTICE '10%% Bonus Amount    : %', bonus_amount;
    RAISE NOTICE 'Salary After Bonus  : %', new_salary;

END $$;
```

Step 1 Output

EXPERIMENT 3



```
DO $$
DECLARE
    emp_id          INTEGER := 102;
    emp_name        VARCHAR(50) := 'ARNAV SHARMA';
    emp_salary      NUMERIC := 45000;
    bonus_amount    NUMERIC;
    new_salary      NUMERIC;
BEGIN

    bonus_amount := emp_salary * 0.10;
    new_salary := emp_salary + bonus_amount;

    RAISE NOTICE 'Employee Details';
    RAISE NOTICE '-----------------';
    RAISE NOTICE 'Employee ID    : %', emp_id;
    RAISE NOTICE 'Employee Name : %', emp_name;

    RAISE NOTICE '';
    RAISE NOTICE 'Salary Details';
    RAISE NOTICE '-----------------';
    RAISE NOTICE 'Salary Before Bonus : %', emp_salary;
    RAISE NOTICE '10%% Bonus Amount     : %', bonus_amount;
    RAISE NOTICE 'Salary After Bonus  : %', new_salary;

END $$;
```



```
Employee Details
----------------
Employee ID   : 102
Employee Name : ARNAV SHARMA

Salary Details
----------------
Salary Before Bonus : 45000
10% Bonus Amount    : 4500.00
Salary After Bonus  : 49500.00
```

EXPERIMENT 3

Learning Outcome

After completing this experiment, the student will be able to:

- Declare and use variables in PL/SQL.
- Perform arithmetic operations for salary calculations.
- Calculate bonuses based on a percentage of salary.
- Display structured outputs using DBMS_OUTPUT.PUT_LINE.
- Understand the workflow of PL/SQL blocks for practical data operations.