CLASSMATE
Date :
Page :

EXPERIMENT-10

AIM: Study and implementation of PL/SQL in DBMS:

THEORY: PL/SQL is a block structured language that enables developers to combine the power of SQL with procedural statements. All the statements of a block are passed to oracle engine all at once which increases processing speed and decreases the traffic.

→ PL/SQL stands for Procedural Language extensions to the Structured Query Language (SQL).
→ PL/SQL is a combination of SQL along with the procedural features of programming languages.
→ Oracle uses a PL/SQL engine to process the PL/SQL statements.
→ PL/SQL includes procedural language elements like conditions and loops. It allows declaration of constants and variables, procedures and functions, types and variable of those types and triggers.

⇒ Features of PL/SQL:

1) Procedural language:
PL/SQL is a full-fledged procedural language, allowing you to write code with control structures

such as loops, conditional statements, and exception handling.
It supports feature like variables, constants, data types and subprograms (procedures and functions) to organize and structure your code.

2) Integration with SQL:
→ PL/SQL integrates seamlessly with SQL, allowing you to embed SQL statements within your code blocks.
→ You can use SQL statements like SELECT, INSERT, UPDATE, DELETE and more to manipulate data stored in the database
→ PL/SQL provides improved performance by reducing network traffic between the database and client applications.

3) Exception Handling:

→ PL/SQL offers robust exception handling mechanisms to catch and handle errors that occur during code execution.
→ You can define exception handlers to gracefully handle exceptional conditions and provide custom error messages or perform recovery.
→ PL/SQL provides predefined exceptions all allows you to define your own custom exceptions.

4) Modularization and Reusability :

→ PL/SQL allows you to encapsulate code into reusable program units such as stored procedures, functions, and packages.
→ Stored procedures and functions can be invoked from which PL/SQL or called from external applications.
→ Packages provide a way to organize related procedures, functions, variables and cursors into a single unit for easier management and better code recognization.

5) Performance Optimization:

→ PL/SQL supports bulk processing, which allows you to perform operations on multiple rows of data at once, reducing the number of context switches between the PL/SQL engine and the SQL engine
→ It provides cursor variables (REF CURSOR) for dynamic SQL operations
→ PL/SQL supports compiler optimizations, caching of compiled code, and efficient memory management to enhance performance.

6) Security and Priviliges:

→ PL/SQL programs execute with the priviliges of the invoking user, allowing fine-grained access control.
→ You can define PL/SQL code to run with definer's

Code (PL/SQL example):

```
DECLARE
  total_salary NUMBER := 0;
  employee_count NUMBER := 0;
  average_salary NUMBER;
BEGIN
  -- Calculate total salary and count of employees
  FOR emp IN (SELECT salary FROM employees) LOOP
    total_salary := total_salary + emp.salary;
    employee_count := employee_count + 1;
  END LOOP;

  -- Calculate average salary
  IF employee_count > 0 THEN
    average_salary := total_salary / employee_count;
    DBMS_OUTPUT.PUT_LINE('Average Salary: ' || average_salary);
  ELSE
    DBMS_OUTPUT.PUT_LINE('No employees found.');
  END IF;
END;
/
```

In this program:

We declare three variables: total_salary to store the sum of all employee salaries, employee_count to store the count of employees, and average_salary to calculate and store the average salary.

The program loops through all the records in the employees table using a cursor emp (implicitly defined in the FOR loop). For each record, it adds the salary to total_salary and increments employee_count.

After the loop, the program checks if any employees were found (employee_count > 0). If there are employees, it calculates the average salary by dividing total_salary by employee_count and displays the result using DBMS_OUTPUT.PUT_LINE.

If no employees are found (employee_count <= 0), it displays a message indicating the absence of employees.

rights (invoker's rights) providing controlled access to database objects.

CONCLUSION: PL/SPL has been studied and implemented successfully in DBMS