In PL/SQL (Procedural Language/Structured Query Language), a package is a database object that groups related functions, procedures, variables, and other PL/SQL constructs into a single, reusable unit. Packages are used to encapsulate and modularize code, making it easier to manage, maintain, and share code within a database. A package consists of two main parts: a **package specification** and a **package body**.

package pl/sql programme -1

```
CREATE OR REPLACE PACKAGE my_package AS
 -- Public variables/constants
 pi CONSTANT NUMBER := 3.14159;
 g_counter NUMBER := 10;
 -- Public procedure declaration
 PROCEDURE squre_number(n integer);

 -- Public function declaration
 FUNCTION calculate_area(radius IN NUMBER) RETURN NUMBER;
END my_package;
```

```
CREATE OR REPLACE PACKAGE BODY my_package AS
 -- Implementation of the reset_counter procedure
 PROCEDURE squre_number( n integer) IS
 BEGIN
  g_counter := 0;
  dbms_output.put_line(n*n||'  '||g_counter);
  dbms_output.put_line('procedure reset calling vai package');
 END squre_number;

 -- Implementation of the calculate_area function
 FUNCTION calculate_area(radius IN NUMBER) RETURN NUMBER IS
 BEGIN
  RETURN pi * radius * radius;
 END calculate_area;

END my_package;
```

```
set SERVEROUTPUT on
DECLARE
  radius NUMBER := 5;
BEGIN
  my_package.squre_number(8);
  DBMS_OUTPUT.PUT_LINE('Area: ' || my_package.calculate_area(radius));
END;
```

package pl/sql programme -2

employee table

id	*name*	*salary*
1	John Doe	55100
2	Jane Smith	64100
3	Bob Johnson	60100
4	vikas	65100
8	jatin	13500
9	lakshaya	3000
10	bhumi	6000
12	null	null
7	rohit	12500
12	vikas	null
12	23	23
12	23	23
13	23	23

```
-- Create a package specification
CREATE OR REPLACE PACKAGE employee_pkg AS
  -- Public cursor declaration
--  CURSOR employee_cursor RETURN SYS_REFCURSOR;
CURSOR employee_cursor IS
    SELECT *  FROM employee;

  -- Public procedure to retrieve employee information
  PROCEDURE get_employee_info(employee_id IN NUMBER);
END employee_pkg;



-- Create a package body
CREATE OR REPLACE PACKAGE BODY employee_pkg AS
  -- Implementation of the cursor
--  CURSOR employee_cursor IS
--    SELECT * FROM employee;

  -- Implementation of the procedure to retrieve employee information
  PROCEDURE get_employee_info(employee_id IN NUMBER) IS
    emp_record employee%ROWTYPE;
  BEGIN
    -- Open the cursor
    OPEN employee_cursor;

    -- Loop through the cursor to find the employee by ID
    LOOP
      FETCH employee_cursor INTO emp_record;
      EXIT WHEN employee_cursor%NOTFOUND;

      -- Check if the current record matches the requested employee ID
      IF emp_record.id = employee_id THEN
        DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.id);
        DBMS_OUTPUT.PUT_LINE('Name: ' || emp_record.name);
        DBMS_OUTPUT.PUT_LINE('Salary: ' || emp_record.salary);
```

```
      EXIT;
      END IF;
    END LOOP;

    -- Close the cursor
    CLOSE employee_cursor;
  END get_employee_info;
END employee_pkg;


set SERVEROUTPUT on
DECLARE
  emp_id NUMBER := 12; -- Replace with the desired employee ID
BEGIN
  employee_pkg.get_employee_info(emp_id);
END;
```