Lab 19.1: Creating Nested Procedures

Welcome, PL/SQL enthusiasts, to Lab 19.1! Today, we're diving into the world of nested procedures, a powerful concept that's often overlooked. Think of nested procedures as specialized tools within a larger toolbox, ready to tackle specific tasks in your PL/SQL code.

por Mayko Silva

Objectives: Master Nested Procedures

Create & Implement

Learn how to define and utilize nested procedures within your PL/SQL code, adding another layer of organization and efficiency.

Parameter Modes

Explore the different parameter modes (IN, OUT, IN OUT) and understand their role in data transfer between procedures.

Forward Declaration

Grasp the concept of forward declaration, a technique that allows you to use a procedure before it's fully defined.

Scope & Accessibility

Gain a deep understanding of the scope and accessibility of nested procedures, ensuring you use them correctly.

Nested Procedure: A Practical Example

```
DECLARE
  -- Variables for the main block
  v_emp_count NUMBER;
  v_dept_name VARCHAR2(30);

  -- Nested procedure definition
  PROCEDURE count_dept_employees(
    p_dept_id IN NUMBER,
    p_dept_name OUT VARCHAR2,
    p_emp_count OUT NUMBER
  ) IS
  BEGIN
    -- Get department name and employee count
    SELECT d.department_name, COUNT(e.employee_id)
    INTO p_dept_name, p_emp_count
    FROM departments d
    LEFT JOIN employees e ON (d.department_id = e.department_id)
    WHERE d.department_id = p_dept_id
    GROUP BY d.department_name;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      p_dept_name := 'Department not found';
      p_emp_count := 0;
  END count_dept_employees;
BEGIN
  -- Main block execution
  count_dept_employees(60, v_dept_name, v_emp_count);
  DBMS_OUTPUT.PUT_LINE('Department: ' || v_dept_name);
  DBMS_OUTPUT.PUT_LINE('Number of Employees: ' || v_emp_count);
END;
```

Deconstructing the Nested Procedure

1 Main Block Variables

These variables, declared in the main block, are accessible within the nested procedure. Think of them as global variables for this specific block of code.

2 Nested Procedure Definition

Here, we define the nested procedure `count_dept_employees`, which is designed to calculate the number of employees in a specific department.

3 Parameters

The parameters (IN, OUT, IN OUT) control how data is passed between the main block and the nested procedure. We'll dive deeper into their roles later.

4 SELECT Statement

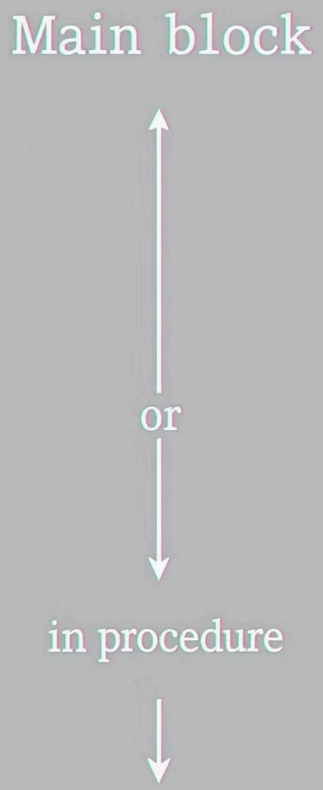
This statement efficiently retrieves the department name and counts employees based on the provided department ID.

5 EXCEPTION Handling

This block gracefully handles scenarios where the department is not found, preventing the procedure from crashing and instead providing appropriate default values.

Calling the Nested Procedure

In the main block, we call the nested procedure `count_dept_employees`, passing in the department ID (60) as a parameter. The procedure then executes, retrieves the data, and returns the results. Finally, the main block displays the retrieved information using `DBMS_OUTPUT.PUT_LINE`.



Benefits of Nested Procedures

Local Definition

Defined within the declaration section of a PL/SQL block, nested procedures are easily accessible within their parent block.

Limited Scope

The scope of nested procedures is confined to the containing block, ensuring they are not accessible outside of their designated context.

Parent Block Access

Nested procedures can access variables declared in the containing block, allowing them to leverage existing data.

Local Variables

Nested procedures can declare their own local variables, maintaining their own internal state and minimizing interference with other procedures.

Expdicte's to offee can

necsue with Nested Toool box

The redllictiools s you nesset and nex on the free fursh besore and iantormation only uere, to cyond our eody and comple in colorration ourr deficiuing prevellyly.

Parameter Modes: The Next Step

Now that you've got a good grasp of nested procedures, we're ready to explore the exciting world of parameter modes. We'll discuss how IN, OUT, and IN OUT parameters work and how they impact data transfer between procedures. Buckle up, PL/SQL wizards, it's going to be a thrilling ride!