Advanced Databases - Lab No. 4

Third Year of the "Computer Engineering" Program

PL/SQL control structures, cursors, functions, and procedures using Oracle's HR database

Part I: Basic Control Structures - Loops and Conditions

Exercise 1: Using Loops

Write an anonymous PL/SQL block that:
- Displays all numbers from 1 to 10 using a FOR loop.
- For each number, print whether it's even or odd.

Hint: use MOD(i,2) and DBMS_OUTPUT.PUT_LINE.

Exercise 2: Using IF / ELSIF

Write a PL/SQL block that:
- Declares a variable v_salary initialized to 3000.
- If salary < 2000 → print "Low salary".
- If between 2000 and 5000 → print "Average salary".
- Otherwise → print "High salary".

Exercise 3: Using CASE

Write a PL/SQL block that:
- Declares a variable v_job_id VARCHAR2(10) := 'IT_PROG'.
- Use a CASE statement to print:
  - 'Developer' if job_id = 'IT_PROG'
  - 'Manager' if job_id = 'ST_MAN'
  - 'Sales' if job_id = 'SA_REP'
  - 'Other' otherwise

Part II: Working with Implicit and Explicit Cursors

Exercise 4: Implicit Cursor

Display the total number of employees working in the HR department.

Question: What type of cursor is used automatically here?

```
DECLARE
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count
  FROM employees
```

```
  WHERE department_id = 40;  -- HR Department

  DBMS_OUTPUT.PUT_LINE('Number of HR employees: ' || v_count);
END;
```

Exercise 5: Explicit Cursor

Write a PL/SQL block that:
- Declares a cursor to select first_name, last_name, and salary from employees with salary > 10000.
- Fetches each record and displays the employee name and salary.

Hint: Use CURSOR, OPEN, FETCH, and CLOSE.

Part III: Procedures and Functions

Exercise 6: Procedures

Create a procedure show_employee_info that:
- Takes an input parameter p_emp_id (employee_id).
- Prints the employee's name, job, and salary.
- If the employee doesn't exist, print "Employee not found".
- Test it using:

```
BEGIN
  show_employee_info(101);
END;
```

Exercise 7: Functions

Create a function get_annual_salary that:
- Takes p_emp_id as input.
- Returns (salary + NVL(commission_pct,0)*salary)*12.
- Test it using:

```
SELECT first_name, get_annual_salary(employee_id) AS annual_salary
FROM employees
WHERE department_id = 90;
```

Part IV: Cursor with Conditions and Loops

Exercise 8: Cursor + IF condition

Write a PL/SQL block that:
- Declares a cursor for employees in department 60.
- For each employee:
  - If salary > 10000 → print "High salary".
  - Else → print "Normal salary".

Part V: Cursor FOR Loop and CASE Together

Exercise 9: Using Cursor FOR LOOP with CASE

Display each employee's name and a message depending on their job:

- 'SA_REP' → "Sales Representative"
- 'IT_PROG' → "Programmer"
- 'ST_MAN' → "Store Manager"
- Else → "Other position"

Hint: Use the following code format:

```
FOR rec IN (SELECT first_name, job_id FROM employees) LOOP
  CASE rec.job_id
    WHEN 'SA_REP' THEN ...
    WHEN 'IT_PROG' THEN ...
    ELSE ...
  END CASE;
END LOOP;
```

Part VI: Challenge - Procedure + Cursor

Exercise 10: Procedure with Cursor

Create a procedure increase_salary that:

- Increases the salary of all employees in a given department by 10%.
- Takes p_dept_id as a parameter.
- Uses an explicit cursor to update salaries one by one.
- Displays the total number of employees updated.

Test it using:

```
BEGIN
   increase_salary(50);
END;
```