



सी डैक  
CDAC

प्रगत संगणन विकास केंद्र  
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

# Post Graduate Diploma in Advance Computing

## Database Technologies Assignment – 6

Prepared By –

Md Farazul Haque

PRN - 210980420078

**1. Create a PL/SQL block that computes the commission amount for a given employee based on the employee's salary.**

**a. Use the DEFINE command to provide the employee ID. Pass the value to the PL/SQL block through a iSQL\*Plus substitution variable.**

**DEFINE p\_empno = 100**

**b. If the employee's salary is less than 2,000, display the bonus amount for the employee as 10% of the salary.**

**c. If the employee's salary is between 2,000 and 3,000, display the bonus amount for the employee as 15% of the salary.**

**d. If the employee's salary exceeds 3,000, display the bonus amount for the employee as 20% of the salary.**

**f. Test the PL/SQL block for each case using the following test cases, and check each bonus amount.**

```
DECLARE
    empno employees.employee_id%type := &p_empno;
    name employees.last_name%type;
    sal employees.salary%type;
    bonus employees.salary%type;
BEGIN
    SELECT last_name, salary INTO name, sal FROM employees WHERE
    employee_id=empno ;
    IF sal < 2000 THEN bonus:=sal*0.10;
    ELSIF sal < 3000 THEN bonus:=sal*0.15;
    ELSIF sal > 3000 THEN bonus:=sal*0.20;
    END IF;
    DBMS_OUTPUT.PUTLINE(name || ' ' || sal || ' ' || bonus);
END;
/
```

**2. Retrieve the first five employees with a job history using cursor.**

```
DECLARE
    CURSOR jobhistory_cur IS SELECT * FROM job_history ORDER BY employee_id
    ASC;
    emprec job_history%rowtype;
BEGIN
    OPEN jobhistory_cur;
    LOOP
        FETCH jobhistory_cur INTO emprec;
        EXIT WHEN jobhistory_cur%rowcount>5;
        DBMS_OUTPUT.PUTLINE(emprec.employee_id || ' ' || emprec.start_date || '
        ' || emprec.end_date || ' ' || emprec.job_id || ' ' ||
        emprec.department_id);
    END LOOP;
    CLOSE jobhistory_cur;
END;
/
```

**3. Accept a number n from the user where n represents the number of top n earners from the EMPLOYEES table. For example, to view the top five earners, enter 5.**

**Note: Use the DEFINE command to provide the value for n. Pass the value to the PL/SQL block through a iSQL\*Plus substitution variable.**

**DEFINE p\_num = 5**

**b. In a loop use the iSQL\*Plus substitution parameter created in step 1 and gather the salaries of the top n people from the EMPLOYEES table. There should be no duplication in the salaries. If two employees earn the same salary, the salary should be picked up only once.**

```
DECLARE
    CURSOR salary_cur IS SELECT DISTINCT salary FROM employees ORDER BY salary
    DESC;
    sal salary_cur%rowtype;
BEGIN
    OPEN salary_cur;
    LOOP
        FETCH salary_cur into sal;
        EXIT when salary_cur%rowcount>&p_num;
        DBMS_OUTPUT.PUTLINE(sal.salary);
    END LOOP;
    CLOSE salary_cur;
END;
/
```

**4. 1. Write a PL/SQL block to select the name of the employee with a given salary value.**

**a. Use the DEFINE command to provide the salary.**

**SET VERIFY OFF**

**DEFINE p\_sal = 6000**

**b. Pass the value to the PL/SQL block through a iSQL\*Plus substitution variable. If the salary entered returns more than one row, handle the exception with an appropriate exception handler and insert into the MESSAGES table the message "More than one employee with a salary of <salary>."**

**c. If the salary entered does not return any rows, handle the exception with an appropriate exception handler and insert into the MESSAGES table the message "No employee with a salary of <salary>."**

**d. If the salary entered returns only one row, insert into the MESSAGES table the employee's name and the salary amount.**

**e. Handle any other exception with an appropriate exception handler and insert into the MESSAGES table the message "Some other error occurred."**

**f. Test the block for a variety of test cases. Display the rows from the MESSAGES table to check whether the PL/SQL block has executed successfully**

```
DECLARE
    v_emp employees%ROWTYPE;
BEGIN
    SELECT * INTO v_emp FROM employees WHERE salary = &p_sal;
    INSERT INTO messages(results) VALUES(v_emp.first_name || ' ' ||
    v_emp.last_name || ' is the only employee with a salary of ' || &p_sal );
EXCEPTION
    WHEN TOO_MANY_ROWS THEN
        INSERT INTO messages(results)
        VALUES('More than one employee with a salary of ' || &p_sal);

    WHEN NO_DATA_FOUND
    THEN INSERT INTO messages(results)
    VALUES('No employee with a salary of ' || &p_sal);

    WHEN OTHERS THEN
        INSERT INTO messages(results) VALUES('Some other error occurred');
END;
/
```