PL/SQL Exercises

1. Basic & Variables

1. A company wants to calculate the **annual salary** of an employee. Write a PL/SQL block that takes basic_salary and bonus as variables and prints the annual salary.
2. A university stores a student's **marks in 3 subjects**. Write a PL/SQL block to calculate the **average marks** and display the result.

2. Conditional Statements

3. A bank system stores a customer's account balance.
    - If balance < 1000 → print "Low Balance"
    - If balance between 1000 and 5000 → print "Sufficient Balance"
    - If balance > 5000 → print "High Balance"
      ☞ Write a PL/SQL block using IF-ELSIF.
4. A grading system accepts a student's percentage.
    - 90–100 → "A Grade"
    - 75–89 → "B Grade"
    - 50–74 → "C Grade"
    - Below 50 → "Fail"
      ☞ Write using a CASE statement.
5. A shopping store gives discounts:
    - If the bill > 5000 → 20% discount
    - If the bill between 2000 and 5000 → 10% discount
    - Otherwise no discount
      ☞ Write a PL/SQL block to calculate **final bill after discount**.

3. Looping

6. Write a PL/SQL block that prints the **multiplication table of a number** entered by the user (example: table of 7).
7. A company wants to print **employee IDs from 100 to 120**. Use a FOR LOOP to print them.
8. Write a PL/SQL block to display the **factorial of a given number** using a WHILE loop.
9. A countdown timer should print numbers from 10 down to 1 using a REVERSE FOR loop.

4. Table-Based Scenarios (using employees table)

(Assume table employees(emp_id, emp_name, salary, dept_id) exists)

10. Print the names of all employees in the IT department using a FOR loop with a SELECT query.
11. Give a 10% salary increase to all employees whose salary < 3000. Use a loop to update salaries.
12. Display all employees whose salary is above the **average salary** of the company.
13. Write a PL/SQL block that prints:

- "High Earner" if salary > 8000
- "Mid Earner" if salary between 4000–8000
- "Low Earner" otherwise.

14. Write a PL/SQL program that prints the **total salary cost** of each department (group by dept_id).

5. Challenge Level 🚀

15. Write a PL/SQL block that accepts a number n and prints the **Fibonacci sequence** up to n terms.
16. A bank wants to process **100 transactions** stored in a table transactions(txn_id, amount, type) where type = 'CREDIT' or 'DEBIT'.
    ☞ Write a PL/SQL block that calculates **final account balance** after all transactions.
17. Write a PL/SQL procedure that takes an employee ID and prints:

- Employee Name
- Department Name
- Current Salary