MySQL Practical Exam

Section A – Joins (30 Marks)

1. Inner Join Task (10 Marks)
- You have two tables:
  - employees(emp_id, emp_name, dept_id, salary)
  - departments(dept_id, dept_name)
- Write a query to display all employee names, their department names, and salaries.

2. Left Join Task (10 Marks)
- Some employees may not be assigned to a department.
- Write a query to display all employees along with department names (if available). Show 'No Department' if none.

3. Self Join Task (10 Marks)
- Assume employees also has a column manager_id referring to another emp_id.
- Write a query to display each employee's name along with their manager's name.

Section B – Stored Procedures (20 Marks)

4. Procedure Creation (10 Marks)
- Write a stored procedure GetEmployeeDetails(IN deptName VARCHAR(50)) that returns all employee names, salaries, and department names for the given department.

5. Procedure with Condition (10 Marks)
- Write a stored procedure IncreaseSalary(IN deptName VARCHAR(50), IN percentage DECIMAL(5,2)) that increases the salary of all employees in the given department by the specified percentage.

Section C – Indexing (20 Marks)

6. Index Usage (10 Marks)
- A table orders(order_id, customer_id, order_date, amount) has millions of rows.
- Write a query to create an index to improve performance when searching by customer_id and order_date.

7. Index Impact (10 Marks)
- Explain (with SQL queries):
  1. How you would check if an index is being used for a query.

2. Why too many indexes can reduce performance.

Section D – Advanced Select Queries (30 Marks)

8. Aggregation with Group By (10 Marks)
- From the orders table, write a query to display each customer's total order amount, average order amount, and number of orders.

9. Subquery with EXISTS (10 Marks)
- Write a query to list all customers who have placed at least one order above ₹10,000.

10. Window Functions (10 Marks)
- From the employees table, write a query to display each employee's name, department, salary, and their rank within the department based on salary (highest salary = rank 1).