1. Create a database named sales_db.
2. Create a table customers with columns: customer_id, name, email, phone.
3. Add a PRIMARY KEY to customer_id.
4. Add a UNIQUE constraint to email.
5. Add a column created_at with DEFAULT CURRENT_TIMESTAMP.
6. Create a table orders with FOREIGN KEY referencing customers.
7. Create a table employees with CHECK constraint: salary > 0.
8. Add AUTO_INCREMENT to customer_id.
9. Drop the column phone from customers.
10. Truncate the orders table.
11. Drop the table orders.
12. Rename table customers to client_master.
13. Create a table products with price having DEFAULT 0.
14. Add CASCADE DELETE rule between orders and customers.
15. Show all tables in sales_db.
16. Insert 5 records into customers.
17. Insert multiple rows into products in one query.
18. Update salary of employee id = 101.
19. Increase product price by 10%.
20. Delete a customer whose id = 10.
21. Delete all inactive customers.
22. Insert a new order for customer id = 3.
23. Update order status to SHIPPED.
24. Delete orders older than 2022.
25. Copy data from customers to backup_customers.
26. Display all customers.
27. Display only name and email of customers.
28. Find all employees with salary > 50000.
29. List products ordered by price DESC.
30. Show distinct cities from customers.
31. Count total number of customers.
32. Find average salary of employees.
33. Find maximum product price.
34. Find minimum order amount.

35. Calculate total sales amount.
36. Show department-wise employee count.
37. Show department-wise average salary.
38. Display only departments having more than 5 employees.
39. Find total sales per customer.
40. Find total revenue per month.
41. List employees who joined after 2021.
42. Show orders between two dates.
43. Find customers from Delhi.
44. Sort employees by salary ASC.
45. Display top 5 highest paid employees.
46. Show products priced between 1000 and 5000.
47. Display customers whose name starts with A.
48. Find employees whose email contains hr.
49. List orders where status is NOT CANCELLED.
50. Show customers with NULL phone numbers.
51. Display customer names with their orders (INNER JOIN).
52. Show all customers even if they have no orders (LEFT JOIN).
53. Show all orders even if customer is missing (RIGHT JOIN).
54. Show employee names with department names.
55. List products with their category names.
56. Display orders with customer and product details.
57. Find employees working in IT department.
58. Show customers who never placed an order.
59. Display departments with no employees.
60. List all possible combinations of customers and products (CROSS JOIN).
61. Find employees earning more than average salary.
62. Find customers who placed the highest order.
63. Find products costing more than average product price.
64. Find department with maximum employees.
65. Find employees who earn more than their manager.
66. Find customers who never placed any order.
67. Find orders greater than average order value.
68. Find the second highest salary using subquery.

69. Find products that were never sold.
70. Find employees working in the same department as John.
71. Create a view vw_active_customers.
72. Create a view showing employee name, department, salary.
73. Update data using a view.
74. Drop a view.
75. Create a view for monthly sales summary.
76. Create an index on email in customers table.
77. Create a composite index on orders(order_date, customer_id).
78. Drop an index.
79. Show all indexes on employees.
80. Explain how an index improves query performance using EXPLAIN.
81. Create a procedure to insert a new customer.
82. Create a procedure to get employee details by id.
83. Create a procedure to calculate total sales of a customer.
84. Create a procedure to update employee salary.
85. Call a stored procedure.
86. Create a function to calculate GST for a given amount.
87. Create a function to return employee grade based on salary.
88. Use a function in a SELECT query.
89. Drop a function.
90. Compare procedure vs function with examples.
91. Create a trigger to log every insert on orders.
92. Create a trigger to prevent salary update below 30000.
93. Create a trigger to auto-update stock after sale.
94. Create a BEFORE DELETE trigger on customers.
95. Drop a trigger.
96. Insert records using START TRANSACTION.
97. Use COMMIT after successful order insertion.
98. Use ROLLBACK when error occurs.
99. Demonstrate SAVEPOINT usage.
100. Simulate money transfer between two accounts using transaction control.