1. Insert a new user into the Users table with username ‘bobsharma1’.
2. Fetch all users' Username and Email.
3. Update email by UserID of the user ‘bobsharma1’.
4. Delete a user where Username is 'bobsharma1'.
5. Retrieve all orders with the Status 'Completed'.
6. Count the total number of products in the Products table.
7. Display all orders sorted by OrderDate in descending order.
8. Fetch all products with a price greater than 80000.
9. Select all users born after the year 2000.
10. Add a new column PhoneNumber to the Users table.
11. Retrieve users whose Username starts with 'A'.
12. Fetch products with stock less than 20.
13. Calculate the total revenue generated from the Orders table.
14. Find the minimum price of a product in the Products table.
15. Retrieve orders placed in the last 30 days.
16. Update the stock of a product where ProductID is 100.
17. Delete a product with the Category ‘Levofloxacin’.
18. Display the first 5 users registered.
19. Retrieve the top 3 most expensive products.
20. Fetch orders where the Amount is between 1000 and 5000.
21. Find the average price of products in each Category.
22. List all distinct Categories of products.
23. Change the Status of all pending orders to 'Completed'.
24. Find the total number of orders placed by a user with UserID = 905.
25. Count the number of users born each month, sorted according to month names.
26. Show all users whose email contains 'forbes.com'.
27. Fetch orders placed on '2025-01-01'.
28. Update the Price of all products in the 'Alcohol' category by reducing 10%.
29. Delete orders where the Amount is less than 1500.
30. Fetch users(USERID, USERNAME) along with their age.
31. Retrieve orders(USERID, ORDERID) where the user has placed more than one order.

32. Display the total number of Orders placed by each user(USERID, USERNAME, NO OF ORDERS).
33. Find the name of the product with the highest price.
34. Insert a new product with Stock as 50 and Price as 150.
35. List all products where the stock is greater than 4500.
36. Retrieve orders along with user information (UserID, Username, Email, ORDERID, STATUS), SORTED BY USERID.
37. Create an index on the Email column in the Users table.
38. Select all products sorted by their Category and Price.
39. Add a default Stock value of 20 for new products.
40. Fetch all products where Category is 'Alcohol' or 'Paracetamol'.
41. Show the total revenue for each product BY PRODUCT ID AND NAME.
42. Add a new Discount column to the Products table WITH DEFAULT VALUE 5.00.
43. List all orders and their statuses.
44. Fetch the user details for the latest registered user.
45. Find the second highest price in the Products table.
46. Count the number of completed orders in the last week.
47. Retrieve all users who haven't placed any orders.
48. Calculate the average amount spent by users on orders.
49. Fetch products that have never been ordered.
50. Change the Status of an order with OrderID 10 to 'Cancelled'.
51. Fetch the details of users along with the total number of orders they've placed.
52. Show the top 5 users who have spent the most on orders.
53. Find the total revenue grouped by Status in the Orders table.
54. Create a view to display the user’s Username and their total orders.
55. Add a foreign key constraint to link Orders.ProductID with Products.ProductID.
56. Write a query to find the most popular product.
57. Create a stored procedure to insert a new product.
58. Write a query to find users who ARE BORN in the same month.
59. Create a trigger to update product stock when an order is placed.
60. List users who have placed orders worth more than 200,000.
61. Retrieve orders that include products from a specific category ‘PARACETAMOL’.

62. Find users who haven’t updated their profiles in the last 6 months.
63. Delete products that haven't been sold in the last year.
64. Create a query to show the total number of orders each day for the last 7 days.
65. Write a query to calculate the average stock of products across all categories.
66. Find users who share the same birth month. // REPEATED QUESTION
67. Generate a report showing revenue by product category with revenue >800,000.
68. Create a query to show orders grouped by their Status.
69. Write a query to find all products with stock less than the average stock.
70. Show the total revenue generated by each user.
71. Write a query to identify the user who has placed the maximum number of orders.
72. List all products along with the total quantity sold for each.
73. Find the order with the highest amount and the user who placed it.
74. Update all pending orders older than a month to 'Cancelled'.
75. Write a query to calculate the total stock value for each product.
76. List users who haven’t placed an order in the last 6 months.
77. Write a query to find products that are low in stock (less than 5 units).
78. Show the top 3 categories by total sales amount.
79. Find all users who have placed orders with an amount greater than the average order value.
80. Write a query to find the difference between the highest and lowest product price.
81. Calculate the average revenue generated per day for the last month.
82. Write a query to rank users based on their total order amount.