1. List all records from the Customers table.
2. Show all products with UnitPrice greater than 50 from the Products table.
3. Find Orders placed after January 1, 1998 (filter on OrderDate).
4. List customers from India (filter on Customers.Country).
5. List products sorted by descending UnitPrice.
6. Display the customers’ names as customer\_name (alias for ContactName in Customers).
7. List all unique countries where your customers come from (distinct Country in Customers).
8. What is the total revenue earned this month?  
   (Calculate sum of Order Details.Quantity \* UnitPrice for orders placed this month.)
9. What is the average order value for each customer?  
   (Average of total order amounts grouped by CustomerID.)
10. How many orders has each customer placed?
11. What are the total sales for each product category?  
    (Sum sales grouped by CategoryName from joined Products and Categories.)
12. List customers who have made more than 5 orders.
13. List all orders with the corresponding customer names.
14. Show all customers along with their most recent order, if any.
15. Find employees who are also managers.  
    (Employees where EmployeeID = some other employee's ReportsTo.)
16. List orders whose value is above the average order value.
17. Show customers who have never placed any order.
18. Show all customer email addresses in lowercase.  
    (Note: Northwind Customers table doesn't store email by default, so if extended, use LOWER function on email field.)
19. List all orders placed in the last 30 days.
20. Find products that have no description listed.
21. Show a running total of sales per day.
22. List the top 3 products by revenue for each month.
23. Identify “repeat customers” (those who have ordered more than once).
24. Rank all customers by the number of orders placed.
25. Present monthly sales data as columns for each year.
26. List an entire management hierarchy or employee org chart.  
    (Use recursive queries or self-join on Employees.EmployeeID and Employees.ReportsTo.)
27. Group users into cohorts based on when they made their first purchase.
28. Compare this year’s sales to last year’s, month by month.
29. Identify orders that reference nonexistent customers.
30. Retrieve the 5 most expensive products.
31. List customers who’ve never placed an order.
32. Find the customer with the largest single order.
33. Show the top-selling product for each month.
34. Calculate month-over-month growth in total orders.
35. For every customer, show their first and last purchase date.
36. For every product, compute a rolling 3-month average of sales.
37. List all products that have never been sold.
38. Find the average time between a customer’s consecutive orders.
39. For each order, calculate what percentage of total revenue that order represents.
40. Identify which customers increased their order value compared to their last order.

* Writing DDL queries (CREATE/ALTER/DROP table)
* Adding/modifying constraints (primary/foreign keys, indexes)
* Writing simple stored procedures and triggers
* Transaction control (BEGIN/COMMIT/ROLLBACK)
* Managing user and security privileges
* Identifying and preventing SQL injection vulnerabilities