Global Online Orders SQL Practice Set

Description:

This SQL practice set is designed to provide hands-on experience with real-world datasets and cover all key SQL topics. It contains **2000+ practice questions** based on a global online orders dataset consisting of **8 tables**:

- order_details
- orders
- shippers
- employees
- customers
- products
- suppliers
- categories

The questions range from **basic to advanced levels** and cover essential SQL concepts including **Filtering & Logical Operations, Aggregations, Joins, Subqueries, Window Functions, String Functions, Data Cleaning Functions, Views, Indexes, Temporary & Derived Tables, and Stored Procedures**. This practice set is ideal for learners preparing for **SQL interviews, database assignments, or real-world analytics tasks**.

Objective:

- To strengthen understanding of SQL concepts and their practical applications.
- To provide experience in writing complex queries on multiple tables.
- To enhance skills in **data extraction, transformation, analysis, and reporting**.
- To prepare learners for **real-world scenarios** involving large datasets and multi-table operations.
- To build confidence in using **advanced SQL features** such as window functions, subqueries, and stored procedures.

Objective:

- To strengthen understanding of SQL concepts and their practical applications.
- To provide experience in writing complex queries on multiple tables.
- To enhance skills in **data extraction, transformation, analysis, and reporting**.
- To prepare learners for **real-world scenarios** involving large datasets and multi-table operations.
- To build confidence in using **advanced SQL features** such as window functions, subqueries, and stored procedures.

Practice Set Design Table

Sr. No.	SQL Topic	Questions Count
1	Filtering & Logical Operations	120
2	Aggregations	120
3	Joins	380
4	Subqueries	170
5	Window Functions	360
6	String Functions	180
7	Data Cleaning Functions	240
8	Views	160
9	Indexes	150
10	Temporary & Derived Tables	80
11	Stored Procedures	200
		Total = 2160

Filtering & Logical Operations

Comparison operators (=, !=, >, <, >=, <=)

1. Find all products whose UnitPrice > 20.
2. List all customers who are not from Germany (Country != 'Germany').
3. Find employees who were born before 1960 (BirthDate < '1960-01-01').
4. Retrieve orders where the ShipperID = 3 (orders shipped by Federal Shipping).
5. Show all products where UnitsInStock = 0 (out-of-stock items).
6. Find suppliers where City = 'Tokyo'.
7. List products where the ReorderLevel <= 10.
8. Find customers whose PostalCode >= '5000' (string comparison works lexicographically).
9. Show orders placed by EmployeeID != 3.
10. Find all categories where CategoryID > 5.
11. Fetch order details where Quantity >= 20.
12. Retrieve products where Discontinued = 1.
13. Find suppliers not located in the USA.
14. List all products where CategoryID = 1 (Beverages).
15. Show employees where LastName < 'M' alphabetically.
16. Find customers where CustomerID > 50.
17. List all orders placed before '1996-07-10'.
18. Show order details of products where ProductID <= 10.
19. Find suppliers whose SupplierID >= 20.
20. Retrieve products where UnitsOnOrder != 0.

Logical operators (AND, OR, NOT)

1. Find all products where UnitPrice > 20 AND UnitsInStock > 30.
2. List customers who live in Germany OR France.
3. Retrieve orders where EmployeeID = 5 AND ShipperID = 3.
4. Find suppliers where Country = 'Japan' OR Country = 'USA'.
5. Show products that are NOT discontinued.
6. Get employees born before 1960 AND LastName LIKE 'D%'.
7. List customers whose Country = 'Mexico' AND City = 'México D.F.'.
8. Find orders where OrderDate >= '1996-07-15' AND OrderDate <= '1996-07-25'.
9. Retrieve products where CategoryID = 1 OR CategoryID = 2.
10. Show suppliers NOT located in the UK.
11. Find order details where Quantity < 10 OR Quantity > 30.
12. List products where UnitsInStock = 0 AND Discontinued = 1.

13. Retrieve customers where NOT (City = 'Berlin').
14. Show orders shipped by ShipperID = 1 OR ShipperID = 2.
15. Find employees where NOT (BirthDate > '1965-01-01').
16. Display products where UnitPrice > 15 AND UnitsOnOrder > 0.
17. Find customers where Country = 'Germany' OR Country = 'Sweden'.
18. Retrieve products where NOT (UnitsInStock < 20).
19. List order details where Quantity BETWEEN 10 AND 20 OR ProductID < 5.
20. Show orders where CustomerID = 20 AND EmployeeID != 3.

BETWEEN

1. Find all products whose UnitPrice is between 10 and 25.
2. List customers whose CustomerID is between 20 and 40.
3. Retrieve orders placed between '1996-07-10' and '1996-07-20'.
4. Show order details where Quantity is between 5 and 15.
5. Find employees born between '1955-01-01' and '1965-12-31'.
6. List products with UnitsInStock between 10 and 50.
7. Fetch suppliers whose SupplierID is between 5 and 15.
8. Get customers whose PostalCode is between '3000' and '6000' (lexical comparison).
9. List categories where CategoryID is between 3 and 7.
10. Retrieve products where ReorderLevel is between 5 and 20.
11. Show orders where EmployeeID is between 4 and 8.
12. Find order details where ProductID is between 1 and 10.
13. Get suppliers whose Phone number length is between 10 and 14 characters.
14. Find products where UnitsOnOrder is between 1 and 50.
15. Retrieve customers whose City name length is between 5 and 8 letters.
16. Show products where CategoryID is between 2 and 4.
17. List orders where CustomerID is between 60 and 90.
18. Get employees whose EmployeeID is between 3 and 7.
19. Find order details where OrderID is between 10260 and 10280.
20. Retrieve products where UnitPrice is between 5 and 15.

IN

1. Find all products where CategoryID is IN (1, 2, 3).
2. List customers who live in IN ('Germany', 'France', 'UK').
3. Retrieve orders shipped by shippers with ShipperID IN (1, 3).
4. Show products supplied by suppliers with SupplierID IN (5, 10, 15).
5. Fetch employees whose EmployeeID is IN (2, 4, 6).
6. Find suppliers located in countries IN ('Japan', 'Spain', 'USA').

7. List products where UnitsInStock is IN (0, 10, 20).
8. Get customers whose City is IN ('Berlin', 'London', 'México D.F.').
9. Show order details where Quantity is IN (5, 10, 15).
10. Retrieve categories where CategoryID is IN (6, 7, 8).
11. Find orders placed by employees IN (3, 5, 9).
12. Get products where ProductID is IN (1, 5, 10, 20).
13. List suppliers whose SupplierID is IN (20, 21, 22, 23, 24).
14. Show orders where OrderID is IN (10248, 10250, 10254).
15. Find customers where Country is NOT IN ('Germany', 'Sweden').
16. Retrieve products where Discontinued is IN (0, 1) (both active or discontinued).
17. Fetch order details where OrderID is IN (10260, 10261, 10262).
18. List employees whose LastName is IN ('King', 'Davolio', 'Fuller').
19. Show products where ReorderLevel is IN (0, 10, 25).
20. Find customers whose CustomerID is IN (5, 15, 25, 35, 45).

LIKE

1. Find all customers whose CustomerName **starts with 'A'** (CustomerName LIKE 'A%').
2. List employees whose LastName **ends with 'n'** (LastName LIKE '%n').
3. Retrieve products where ProductName **contains the word 'Mix'** (ProductName LIKE '%Mix%').
4. Show suppliers whose ContactName **starts with 'C'** (ContactName LIKE 'C%').
5. Find customers whose City **contains 'o'** (City LIKE '%o%').
6. List products where QuantityPerUnit **starts with a number** (QuantityPerUnit LIKE '[0-9]%').
7. Get employees whose FirstName **has exactly 5 letters** (FirstName LIKE '_____').
8. Retrieve suppliers where Phone **contains '555'** (Phone LIKE '%555%').
9. Show customers whose Address **ends with 'St.'** (Address LIKE '%St.%').
10. Find products whose ProductName **starts with 'Ch'** (ProductName LIKE 'Ch%').
11. List categories where CategoryName **contains 'ea'** (CategoryName LIKE '%ea%').
12. Retrieve customers whose ContactName **starts with 'M'** and **ends with 's'** (ContactName LIKE 'M%s').
13. Find employees whose LastName **contains 'er'** (LastName LIKE '%er%').
14. Show orders where OrderDate **starts with '1996-07'** (OrderDate LIKE '1996-07%').
15. List suppliers whose City **starts with 'T'** (City LIKE 'T%').
16. Retrieve products where UnitPrice **formatted text contains '.'** (UnitPrice LIKE '%.%').

17. Find customers whose PostalCode **begins with 'S'** (PostalCode LIKE 'S%').
18. Show employees whose FirstName **contains 'a'** (FirstName LIKE '%a%').
19. List products whose ProductName **ends with 'Tea'** (ProductName LIKE '%Tea').
20. Find suppliers whose Address **contains a space** (Address LIKE '% %').

IS NULL, IS NOT NULL

1. Find all products where UnitsOnOrder IS NULL.
2. List customers whose PostalCode IS NULL.
3. Retrieve employees whose BirthDate IS NULL.
4. Show orders where OrderDate IS NOT NULL.
5. Find products where ReorderLevel IS NULL.
6. List suppliers whose Phone IS NOT NULL.
7. Get customers whose ContactName IS NOT NULL.
8. Retrieve products where UnitPrice IS NOT NULL.
9. Show employees whose FirstName IS NOT NULL.
10. Find categories where DescriptionText IS NULL.
11. List orders where ShipperID IS NOT NULL.
12. Retrieve products where QuantityPerUnit IS NULL.
13. Find suppliers whose City IS NOT NULL.
14. Show order_details where Quantity IS NOT NULL.
15. List products where UnitsInStock IS NULL.
16. Retrieve customers whose Address IS NOT NULL.
17. Find employees whose LastName IS NULL.
18. Show products where Discontinued IS NOT NULL.
19. List categories where CategoryName IS NOT NULL.
20. Retrieve orders where CustomerID IS NOT NULL.

Aggregations

COUNT()

1. Count the total number of products in the database.
2. Count how many customers are from Germany.
3. Count the total number of orders placed.
4. Count the number of employees born before 1960.
5. Count how many products have UnitsInStock = 0.
6. Count the total number of suppliers.
7. Count the number of products in CategoryID = 2 (Condiments).
8. Count the total number of orders shipped by ShipperID = 1.
9. Count how many customers live in Mexico.
10. Count the number of order_details where Quantity >= 20.
11. Count the total number of discontinued products.
12. Count how many products have ReorderLevel <= 10.
13. Count the number of employees whose LastName starts with 'D'.
14. Count how many orders were placed by EmployeeID = 4.
15. Count the number of products supplied by SupplierID = 5.

SUM()

1. Find the total UnitsInStock of all products.
2. Calculate the total UnitsOnOrder across all products.
3. Find the total Quantity ordered in order_details.
4. Calculate the total UnitPrice × UnitsInStock for all products (total stock value).
5. Find the total UnitPrice × Quantity for order_details of OrderID = 10248.
6. Calculate the total UnitsInStock for products in CategoryID = 1 (Beverages).
7. Find the total Quantity of order_details where Quantity >= 10.
8. Calculate the total UnitPrice × UnitsOnOrder for discontinued products.
9. Find the total UnitsInStock for products supplied by SupplierID = 3.
10. Calculate the total Quantity of products in orders placed by EmployeeID = 4.
11. Find the total UnitsInStock for products where ReorderLevel <= 10.
12. Calculate the total Quantity in order_details for products in CategoryID = 2 (Condiments).
13. Find the total UnitsOnOrder for products where Discontinued = 0.
14. Calculate the total UnitPrice × Quantity for all orders shipped by ShipperID = 2.
15. Find the total Quantity of order_details where ProductID IN (1, 5, 10).

AVG()

1. Find the average **UnitPrice** of all products.
2. Calculate the average **UnitsInStock** of all products.
3. Find the average **UnitsOnOrder** for all products.
4. Calculate the average **Quantity** in order_details.
5. Find the average **UnitPrice** of products in CategoryID = 1 (Beverages).
6. Calculate the average **UnitsInStock** for products supplied by SupplierID = 3.
7. Find the average **Quantity** in order_details for OrderID = 10248.
8. Calculate the average **UnitPrice** of discontinued products.
9. Find the average **UnitsInStock** for products where ReorderLevel <= 10.
10. Calculate the average **Quantity** for order_details where Quantity >= 10.
11. Find the average **UnitPrice** of products in CategoryID = 2 (Condiments).
12. Calculate the average **UnitsOnOrder** for products that are not discontinued.
13. Find the average **Quantity** in order_details for products with ProductID IN (1,5,10).
14. Calculate the average **UnitPrice × UnitsInStock** for all products (average stock value).
15. Find the average **Quantity** of order_details for orders placed by EmployeeID = 4.

MIN(), MAX()

1. Find the **minimum UnitPrice** among all products.
2. Find the **maximum UnitPrice** among all products.
3. Find the **minimum UnitsInStock** of all products.
4. Find the **maximum UnitsInStock** of all products.
5. Find the **minimum Quantity** in order_details.
6. Find the **maximum Quantity** in order_details.
7. Find the **minimum UnitPrice** of products in CategoryID = 1 (Beverages).
8. Find the **maximum UnitPrice** of products supplied by SupplierID = 3.
9. Find the **minimum UnitsOnOrder** among all products.
10. Find the **maximum UnitsOnOrder** among discontinued products.
11. Find the **minimum ReorderLevel** of all products.
12. Find the **maximum Quantity** in order_details for orders placed by EmployeeID = 4.
13. Find the **minimum UnitPrice** among products where UnitsInStock > 0.
14. Find the **maximum UnitPrice** of products in CategoryID = 2 (Condiments).
15. Find the **minimum BirthDate** among all employees (oldest employee).

GROUP BY

1. Find the total number of products in each category.
2. Calculate the total UnitsInStock for products in each category.
3. Find the average UnitPrice of products grouped by CategoryID.
4. Count the number of customers in each country.
5. Count the number of employees born in each year.
6. Find the total Quantity ordered for each ProductID.
7. Calculate the total Quantity for each OrderID.
8. Find the maximum UnitPrice for products in each category.
9. Find the minimum UnitsInStock for products grouped by SupplierID.
10. Count the number of orders placed by each CustomerID.
11. Find the total UnitsOnOrder for products grouped by SupplierID.
12. Calculate the average Quantity for each ProductID in order_details.
13. Count the number of orders shipped by each ShipperID.
14. Find the total UnitPrice × UnitsInStock for each category.
15. Calculate the average UnitPrice of products for each SupplierID.
16. Count the number of discontinued products in each category.
17. Find the total Quantity for each OrderID placed by EmployeeID = 4.
18. Count the number of products supplied by each SupplierID.
19. Find the total UnitsInStock for products where ReorderLevel <= 10, grouped by CategoryID.
20. Calculate the maximum Quantity for each ProductID.
21. Count the number of orders for each month (GROUP BY MONTH(OrderDate)).
22. Find the total Quantity ordered for each CustomerID.
23. Calculate the average UnitsInStock for each CategoryID.
24. Count the number of products in stock for each SupplierID.
25. Find the maximum UnitPrice for discontinued products grouped by CategoryID.
26. Count the number of orders placed by each EmployeeID.
27. Find the total UnitsOnOrder for each CategoryID.
28. Calculate the average Quantity for products supplied by each SupplierID.
29. Count the number of products in each CategoryID that are discontinued.
30. Find the total Quantity in order_details for each ProductID where Quantity >= 10.

HAVING

1. Find categories having more than 5 products.
2. Retrieve suppliers having supplied more than 10 products.
3. Find products grouped by CategoryID having average UnitPrice > 20.
4. Show orders grouped by CustomerID having total Quantity > 50.

5. Find employees having placed more than 5 orders.
6. Retrieve products grouped by SupplierID having total UnitsInStock < 100.
7. Find categories having average UnitPrice > 15.
8. Show orders grouped by ShipperID having count(OrderID) > 20.
9. Find customers having placed more than 3 orders.
10. Retrieve products grouped by CategoryID having total UnitsOnOrder > 50.
11. Find suppliers having more than 5 discontinued products.
12. Show orders grouped by EmployeeID having total Quantity > 40.
13. Find products where maximum UnitPrice > 25 grouped by SupplierID.
14. Retrieve categories having total UnitsInStock < 200.
15. Find employees having average order Quantity > 10.
16. Show products grouped by CategoryID having minimum UnitsInStock = 0.
17. Find customers with total orders Quantity > 30.
18. Retrieve suppliers having average UnitPrice of products > 20.
19. Find products grouped by SupplierID having sum(UnitsOnOrder) > 50.
20. Show categories having more than 2 discontinued products.
21. Find orders grouped by CustomerID having maximum Quantity in order_details > 20.
22. Retrieve employees having total orders Quantity > 25.
23. Find products grouped by CategoryID having total UnitsInStock between 50 and 150.
24. Show suppliers having minimum UnitPrice of products < 10.
25. Find customers having orders count > 10 in July 1996 (grouped by CustomerID).
26. Retrieve products grouped by CategoryID having average UnitsInStock > 20.
27. Find orders grouped by ShipperID having total orders > 15 in July 1996.
28. Show employees having count of distinct orders > 5.
29. Find products grouped by SupplierID having sum(UnitPrice × UnitsInStock) > 500.
30. Retrieve categories having average UnitPrice < 20.

Joins

INNER JOIN

1. List all orders along with customer names.
2. Retrieve order details with product names.
3. Show products with their supplier names.
4. List orders with employee names who placed them.
5. Find all orders with shipper names.
6. Display products along with category names.
7. Get order details with corresponding order dates.
8. Find customers and the total orders they have placed.
9. Show employees along with their orders.
10. List products and order quantities from order_details.
11. Retrieve suppliers along with the products they supply.
12. Show orders with customer and shipper names.
13. Find products supplied by suppliers in Japan.
14. List order_details with order and product information for orders > 10 quantity.
15. Retrieve customers along with the products they ordered.
16. Show orders along with customer and employee who handled them.
17. Find categories with products currently in stock.
18. List products with supplier and category names.
19. Show orders along with order_details and product names.
20. Find employees and the total number of orders they handled.
21. List customers and their orders placed in July 1996.
22. Show suppliers and products with UnitsInStock < 20.
23. Retrieve orders with customer, employee, and shipper info.
24. List order_details with product names and category names.
25. Find customers who ordered products from category "Beverages."
26. Show all orders with order_details where quantity > 15.
27. Retrieve products with suppliers and reorder levels.
28. List employees and orders for which they were responsible.
29. Show customers along with orders and total quantity ordered.
30. Find suppliers who provide products with UnitPrice > 20.
31. List order_details with orders placed by EmployeeID = 4.
32. Retrieve products and suppliers where products are discontinued.
33. Show orders and order_details with product info where quantity < 10.
34. Find customers and products they ordered in CategoryID = 2.
35. List orders and employee names who handled them and the shipper used.
36. Retrieve products with category and supplier info for UnitsInStock > 30.
37. Show orders along with total quantity per order.
38. Find employees and the customers they served.
39. List order_details and product names for orders shipped by ShipperID = 1.

40. Retrieve categories and the products supplied under each category.
41. Show orders and corresponding order_details with product UnitPrice.
42. Find suppliers and products where UnitsOnOrder > 0.
43. List customers and total distinct products they ordered.
44. Retrieve employees and orders where the total quantity exceeds 20.
45. Show products, suppliers, and categories where UnitPrice > 15.
46. Find orders along with order_details for products supplied by SupplierID = 5.
47. List customers and orders along with shipper names in July 1996.
48. Retrieve products with category names and reorder levels less than 10.
49. Show orders with employee names and products ordered with quantity > 5.
50. Find suppliers and the total number of products supplied in each category.

LEFT JOIN

1. List all customers and their orders (include customers with no orders).
2. Show all employees and the orders they handled (include employees with no orders).
3. Display all products and their order_details (include products never ordered).
4. List all categories and products (include categories with no products).
5. Show all shippers and orders shipped (include shippers with no orders).
6. Retrieve all suppliers and products supplied (include suppliers with no products).
7. Find all customers and orders placed in July 1996 (include customers with no orders).
8. Show all products and category names (include products with no category, if possible).
9. List all employees and total orders they handled (include employees with zero orders).
10. Display all orders and order_details (include orders with no details).
11. Find all categories and products in stock (include categories with no products).
12. Show all suppliers and products with UnitsInStock > 0 (include suppliers with no products).
13. List all customers and their shipped orders (include customers with no shipped orders).
14. Display all employees and orders with quantity > 10 (include employees with no such orders).
15. Find all products and suppliers (include products with no suppliers, if possible).
16. Show all orders and products in order_details (include orders with no products).
17. List all customers and products they ordered from CategoryID = 1 (include customers who ordered none).
18. Display all shippers and total orders shipped (include shippers with no

orders).

19. Find all suppliers and total units in stock of their products (include suppliers with zero stock).
20. Show all categories and average UnitPrice of products (include categories with no products).
21. List all employees and orders placed in July 1996 (include employees with no orders).
22. Display all customers and total quantity ordered (include customers with no orders).
23. Find all products and their order_details for quantity > 20 (include products with no matching orders).
24. Show all suppliers and products with ReorderLevel <= 10 (include suppliers with no products).
25. List all orders and corresponding customer names (include orders with no customer, if possible).
26. Display all products and suppliers where Discontinued = 0 (include products without suppliers).
27. Find all customers and orders where ShipperID = 1 (include customers with no such orders).
28. Show all employees and total quantity handled (include employees with no orders).
29. List all categories and number of products in stock (include categories with zero products).
30. Display all shippers and orders placed in July 1996 (include shippers with no orders).
31. Find all products and total quantity ordered (include products never ordered).
32. Show all suppliers and maximum UnitPrice of their products (include suppliers with no products).
33. List all customers and orders with total quantity > 15 (include customers with no such orders).
34. Display all products and categories (include products without category).
35. Find all employees and number of orders placed by them (include employees with zero orders).
36. Show all orders and order_details where Quantity >= 10 (include orders with no such details).
37. List all categories and total UnitsInStock of products (include categories with zero stock).
38. Display all suppliers and products in CategoryID = 2 (include suppliers with no such products).
39. Find all customers and orders shipped by ShipperID = 3 (include customers with no such orders).
40. Show all products and suppliers with UnitsOnOrder > 0 (include products never ordered).
41. List all employees and orders where UnitPrice > 20 (include employees with no such orders).

42. Display all orders and total quantity per order (include orders with no order_details).
43. Find all categories and average UnitsInStock (include categories with no products).
44. Show all shippers and total orders per shipper (include shippers with zero orders).
45. List all suppliers and total UnitsOnOrder (include suppliers with no products).
46. Display all customers and products ordered from SupplierID = 5 (include customers with no such orders).
47. Find all employees and average Quantity per order (include employees with no orders).
48. Show all products and categories with UnitPrice > 15 (include products without category).
49. List all orders and product names for ProductID IN (1,5,10) (include orders without these products).
50. Display all customers and total orders where Quantity >= 20 (include customers with no such orders).

RIGHT JOIN

1. List all orders and customers, including customers with no orders.
2. Show all order_details and products, including products never ordered.
3. Display all products and suppliers, including suppliers with no products.
4. List all categories and products, including products without a category (if possible).
5. Show all orders and employees, including employees with no orders.
6. Retrieve all shippers and orders, including shippers with no orders.
7. Find all products and categories, including categories with no products.
8. Show all order_details and orders, including orders with no order_details.
9. List all customers and orders shipped by ShipperID = 1, including customers with no such orders.
10. Display all employees and orders in July 1996, including employees with no orders.
11. Find all products and order_details where quantity > 10, including products never ordered.
12. Show all suppliers and products with UnitsInStock > 0, including suppliers with no products.
13. List all customers and products they ordered from CategoryID = 2, including customers who ordered none.
14. Display all categories and total products, including categories with zero products.
15. Find all employees and total quantity handled, including employees with no orders.

16. Show all orders and products in order_details, including orders with no products.
17. List all suppliers and average UnitPrice of products, including suppliers with no products.
18. Display all customers and total quantity ordered, including customers with no orders.
19. Find all products and suppliers where Discontinued = 0, including suppliers without products.
20. Show all categories and total UnitsInStock, including categories with no products.
21. List all orders and shippers, including shippers who shipped no orders.
22. Display all employees and order_details where Quantity >= 20, including employees with no orders.
23. Find all products and categories where UnitPrice > 20, including categories with no products.
24. Show all suppliers and products in CategoryID = 1, including suppliers with no products.
25. List all customers and orders where UnitPrice > 15, including customers with no such orders.
26. Display all order_details and orders where OrderID IN (10248, 10250), including orders with no matching details.
27. Find all employees and orders handled, including employees with zero orders.
28. Show all products and total UnitsOnOrder, including products never ordered.
29. List all categories and average UnitPrice of products, including categories with no products.
30. Display all customers and products ordered from SupplierID = 5, including customers with no such orders.
31. Find all orders and products in order_details where Quantity >= 5, including orders with no matching products.
32. Show all suppliers and maximum UnitPrice of their products, including suppliers with no products.
33. List all employees and number of orders placed, including employees with zero orders.
34. Display all orders and customer names, including orders with no customer (if possible).
35. Find all products and categories where UnitsInStock > 0, including categories with no products.
36. Show all customers and total orders placed in July 1996, including customers with zero orders.
37. List all shippers and total orders shipped, including shippers with no orders.
38. Display all products and suppliers where ReorderLevel <= 10, including suppliers with no products.
39. Find all employees and average Quantity per order, including employees with no orders.
40. Show all categories and number of discontinued products, including

categories with zero discontinued products.

41. List all order_details and products with ProductID IN (1,5,10), including products never ordered.
42. Display all suppliers and total UnitsInStock, including suppliers with no products.
43. Find all customers and total Quantity for orders with Quantity > 10, including customers with no such orders.
44. Show all products and total UnitsInStock per category, including categories with zero products.
45. List all orders and total Quantity per order, including orders with no order_details.
46. Display all employees and orders with Quantity > 15, including employees with zero orders.
47. Find all categories and total UnitsOnOrder, including categories with no products.
48. Show all suppliers and products in CategoryID = 2, including suppliers with no products.
49. List all customers and orders along with ShipperID = 3, including customers with no such orders.
50. Display all products and suppliers with UnitPrice > 15, including suppliers with no products.

FULL OUTER JOIN

1. List all customers and their orders, including customers with no orders and orders with no customers.
2. Show all products and order_details, including products never ordered and order_details with no matching product.
3. Display all suppliers and products, including suppliers with no products and products without suppliers.
4. List all categories and products, including categories with no products and products without categories.
5. Show all employees and orders, including employees with no orders and orders with no employees.
6. Retrieve all shippers and orders, including shippers with no orders and orders with no shippers.
7. Find all products and categories, including categories with no products and products without category.
8. Show all order_details and orders, including orders with no order_details and order_details with no orders.
9. List all customers and orders shipped by ShipperID = 1, including customers with no such orders and orders with no customers.
10. Display all employees and orders placed in July 1996, including employees with no orders and orders with no employees.

11. Find all products and order_details where quantity > 10, including products never ordered and order_details with no product.
12. Show all suppliers and products with UnitsInStock > 0, including suppliers with no products and products without suppliers.
13. List all customers and products they ordered from CategoryID = 2, including customers who ordered none and products not ordered.
14. Display all categories and total products, including categories with zero products and products not belonging to any category.
15. Find all employees and total quantity handled, including employees with no orders and order_details with no employees.
16. Show all orders and products in order_details, including orders with no products and products never ordered.
17. List all suppliers and average UnitPrice of products, including suppliers with no products and products without suppliers.
18. Display all customers and total quantity ordered, including customers with no orders and order_details with no customer.
19. Find all products and suppliers where Discontinued = 0, including suppliers without products and products without suppliers.
20. Show all categories and total UnitsInStock, including categories with no products and products without category.
21. List all orders and shippers, including shippers who shipped no orders and orders with no shippers.
22. Display all employees and order_details where Quantity >= 20, including employees with no orders and order_details with no employee.
23. Find all products and categories where UnitPrice > 20, including categories with no products and products without category.
24. Show all suppliers and products in CategoryID = 1, including suppliers with no products and products without suppliers.
25. List all customers and orders where UnitPrice > 15, including customers with no orders and orders without customers.
26. Display all order_details and orders where OrderID IN (10248, 10250), including orders with no matching details and order_details with no orders.
27. Find all employees and orders handled, including employees with zero orders and orders without employees.
28. Show all products and total UnitsOnOrder, including products never ordered and order_details with no product.
29. List all categories and average UnitPrice of products, including categories with no products and products without category.
30. Display all customers and products ordered from SupplierID = 5, including customers with no orders and products never ordered.
31. Find all orders and products in order_details where Quantity >= 5, including orders with no matching products and products never ordered.
32. Show all suppliers and maximum UnitPrice of their products, including suppliers with no products and products without suppliers.
33. List all employees and number of orders placed, including employees with zero orders and orders without employees.

34. Display all orders and customer names, including orders with no customer and customers with no orders.
35. Find all products and categories where UnitsInStock > 0, including categories with no products and products without category.
36. Show all customers and total orders placed in July 1996, including customers with zero orders and orders without customers.
37. List all shippers and total orders shipped, including shippers with no orders and orders without shippers.
38. Display all products and suppliers where ReorderLevel <= 10, including suppliers with no products and products without suppliers.
39. Find all employees and average Quantity per order, including employees with no orders and order_details without employees.
40. Show all categories and number of discontinued products, including categories with zero discontinued products and products without category.
41. List all order_details and products with ProductID IN (1,5,10), including products never ordered and order_details with no product.
42. Display all suppliers and total UnitsInStock, including suppliers with no products and products without suppliers.
43. Find all customers and total Quantity for orders with Quantity > 10, including customers with no such orders and order_details with no customer.
44. Show all products and total UnitsInStock per category, including categories with zero products and products without category.
45. List all orders and total Quantity per order, including orders with no order_details and order_details with no order.
46. Display all employees and orders with Quantity > 15, including employees with zero orders and order_details without employees.
47. Find all categories and total UnitsOnOrder, including categories with no products and products without category.
48. Show all suppliers and products in CategoryID = 2, including suppliers with no products and products without suppliers.
49. List all customers and orders along with ShipperID = 3, including customers with no such orders and orders without customers.
50. Display all products and suppliers with UnitPrice > 15, including suppliers with no products and products without suppliers.

CROSS JOIN

1. List all combinations of customers and products.
2. Show all combinations of employees and products.
3. List all combinations of orders and products.
4. Show all combinations of categories and suppliers.
5. List all combinations of products and suppliers.
6. Show all combinations of customers and shippers.
7. List all combinations of orders and shippers.

8. Show all combinations of employees and customers.
9. List all combinations of categories and products.
10. Show all combinations of products and orders.
11. List all combinations of customers and order_details.
12. Show all combinations of employees and shippers.
13. List all combinations of suppliers and orders.
14. Show all combinations of products and order_details.
15. List all combinations of categories and shippers.
16. Show all combinations of employees and categories.
17. List all combinations of customers and suppliers.
18. Show all combinations of orders and employees.
19. List all combinations of products and categories.
20. Show all combinations of shippers and suppliers.
21. List all combinations of employees and order_details.
22. Show all combinations of customers and categories.
23. List all combinations of suppliers and products with UnitsInStock > 0.
24. Show all combinations of orders and shippers for July 1996.
25. List all combinations of employees and orders with quantity > 10.
26. Show all combinations of products and suppliers for CategoryID = 1.
27. List all combinations of customers and orders with UnitPrice > 20.
28. Show all combinations of employees and products with UnitPrice < 15.
29. List all combinations of shippers and orders shipped by ShipperID = 1.
30. Show all combinations of categories and products with UnitsInStock > 30.
31. List all combinations of customers and order_details where Quantity >= 5.
32. Show all combinations of employees and customers from Germany.
33. List all combinations of suppliers and products where Discontinued = 1.
34. Show all combinations of orders and products in CategoryID = 2.
35. List all combinations of products and orders for ProductID IN (1,5,10).
36. Show all combinations of customers and suppliers in Japan.
37. List all combinations of employees and shippers with orders in July 1996.
38. Show all combinations of categories and suppliers supplying products in CategoryID = 3.
39. List all combinations of products and order_details with Quantity > 20.
40. Show all combinations of customers and products where UnitPrice > 25.
41. List all combinations of orders and suppliers with products supplied by SupplierID = 5.
42. Show all combinations of employees and categories with products in stock.
43. List all combinations of products and customers who ordered them.
44. Show all combinations of suppliers and customers in the same country.
45. List all combinations of categories and employees responsible for orders in that category.
46. Show all combinations of shippers and customers who used them.
47. List all combinations of order_details and products in CategoryID = 1.
48. Show all combinations of employees and orders with total quantity > 15.
49. List all combinations of products and suppliers with ReorderLevel <= 10.
50. Show all combinations of categories and orders containing products from that

category.

SELF JOIN

1. Find employees who share the same birth year.
2. List employees whose birth year is earlier than another employee's birth year.
3. Find employees whose last name starts with the same letter as another employee.
4. Show pairs of employees where one has handled more orders than the other.
5. List employees who are older than at least one other employee.
6. Find employees and other employees born in the same month.
7. Show employees whose first name matches another employee's first name.
8. Find customers living in the same city as another customer.
9. List customers whose postal code matches another customer.
10. Find suppliers located in the same country as another supplier.
11. Show products supplied by the same supplier.
12. Find products in the same category with higher UnitPrice than another product.
13. List products that have the same UnitsInStock as another product.
14. Find orders placed by different customers on the same date.
15. Show orders handled by different employees on the same date.
16. List customers and other customers from the same city but different countries.
17. Find suppliers with the same phone number as another supplier (if duplicates exist).
18. Show employees whose birth month and day match another employee.
19. List products where UnitsOnOrder matches another product's UnitsOnOrder.
20. Find orders that have the same CustomerID and ShipperID as another order.
21. Show employees whose LastName and FirstName match another employee.
22. Find customers in the same city with different CustomerIDs.
23. List suppliers in the same city and country as another supplier.
24. Show products in the same category where UnitPrice is less than another product.
25. Find order_details where Quantity matches another order_detail's Quantity.
26. List employees with fewer orders than another employee.
27. Show customers who share the same postal code with another customer.
28. Find suppliers whose products belong to the same category as another supplier's products.
29. List products supplied by different suppliers but belonging to the same category.
30. Show orders placed on the same date by different employees for the same customer.

Multiple table JOINs

1. List all orders with customer name and employee name who handled them.
2. Show all orders with customer, employee, and shipper details.
3. Retrieve order_details with product name, order date, and customer name.
4. List products with supplier name and category name.
5. Show all orders with product names, customer names, and shipper names.
6. Find all order_details with product, order, and supplier details.
7. List all customers and total quantity of products ordered along with product names.
8. Show employees and orders they handled including products in each order.
9. Retrieve orders with customer name, employee name, and total quantity ordered.
10. List all products with supplier and category names where UnitsInStock > 0.
11. Show order_details along with product, supplier, and category information.
12. Find orders with customer, shipper, and total order quantity.
13. List products along with category and total UnitsOnOrder.
14. Show customers with orders, product names, and UnitsInStock.
15. Retrieve employees and orders they handled with product UnitPrice.
16. List suppliers and all products in their categories.
17. Show orders along with employee name, customer name, and product UnitPrice.
18. Find customers and products they ordered along with supplier name.
19. List products with UnitsInStock, supplier, and category name where UnitPrice > 20.
20. Show order_details along with product, order, customer, and employee details.
21. Retrieve orders with customer, shipper, and employee details where OrderDate in July 1996.
22. List products with supplier and order_details including Quantity ordered > 10.
23. Show customers and all products they ordered with supplier and category info.
24. Find employees and total quantity they handled grouped by product and customer.
25. List orders with product, customer, and total UnitPrice (UnitPrice × Quantity).
26. Show suppliers and products along with total UnitsInStock per category.
27. Retrieve orders with customer, product, and supplier where UnitPrice > 25.
28. List products, supplier, and category names where products are discontinued.
29. Show order_details along with order, customer, product, and shipper details.
30. Find employees and total UnitsInStock of products in orders they handled.
31. List customers and total quantity ordered grouped by product and supplier.
32. Show orders with customer, employee, and product details for CategoryID = 1.
33. Retrieve products and orders along with supplier, category, and

UnitsOnOrder.

34. List orders with employee, customer, and total UnitPrice per order.
35. Show suppliers and products along with orders they were included in.
36. Find customers and total number of products ordered per category.
37. List products along with supplier, category, and orders with quantity > 5.
38. Show order_details, orders, customers, and products where Quantity >= 10.
39. Retrieve employees and all products in the orders they handled with supplier info.
40. List products, category, and total UnitsOnOrder grouped by supplier.
41. Show orders along with customer, product, employee, and shipper details where Quantity > 15.
42. Find suppliers and products with UnitsInStock < 20 along with category name.
43. List customers with orders, products, and suppliers where UnitPrice > 20.
44. Show products along with supplier, category, and order details including order date.
45. Retrieve order_details, orders, products, and customers for ShipperID = 1.
46. List employees and all orders with product details grouped by category.
47. Show customers, orders, order_details, and products where Quantity > 10 and UnitPrice > 15.
48. Find products, suppliers, categories, and total quantity ordered per product.
49. List orders with customer, employee, product, and supplier details for July 1996.
50. Show all customers, orders, products, suppliers, and categories including UnitPrice and UnitsInStock.

JOIN with conditions

1. List orders with customer names where OrderDate > '1996-07-10'.
2. Show products and suppliers where UnitsInStock > 20.
3. Retrieve orders and employees where EmployeeID = 4.
4. List customers and orders where ShipperID = 3.
5. Show order_details with product and order info where Quantity >= 10.
6. Find products with category and supplier where UnitPrice > 15.
7. List orders with customer and shipper info where OrderDate BETWEEN '1996-07-01' AND '1996-07-15'.
8. Show customers and orders where Country = 'Germany'.
9. Retrieve employees and orders where BirthDate < '1960-01-01'.
10. List products and suppliers where Discontinued = 1.
11. Show orders with customer and employee where EmployeeID != 3.
12. Find products, category, and supplier where UnitsOnOrder > 0.
13. List order_details, products, and orders where ProductID IN (1,5,10).
14. Show orders with customer, employee, and shipper where ShipperID = 2.
15. Retrieve products and suppliers where CategoryID = 2.

16. List customers and orders where PostalCode LIKE 'S%'.
17. Show order_details, products, and orders where Quantity BETWEEN 5 AND 20.
18. Find products with supplier and category where UnitPrice <= 20.
19. List employees and orders where OrderDate < '1996-07-15'.
20. Show customers and orders where Country != 'USA'.
21. Retrieve products, suppliers, and categories where UnitsInStock = 0.
22. List order_details, products, and orders where Quantity > 10 AND UnitPrice > 15.
23. Show orders, customers, and shippers where OrderDate >= '1996-07-20'.
24. Find products and suppliers where SupplierID BETWEEN 5 AND 15.
25. List employees and orders where EmployeeID IN (1,2,4).
26. Show products, categories, and suppliers where UnitsOnOrder > ReorderLevel.
27. Retrieve customers and orders where City = 'Berlin'.
28. List orders, products, and customers where UnitPrice > 20.
29. Show order_details, orders, and products where Quantity <= 10.
30. Find suppliers and products where Country = 'Japan' AND UnitsInStock > 10.
31. List orders, customers, and employees where ShipperID != 1.
32. Show products and suppliers where Discontinued = 0 AND UnitsInStock > 0.
33. Retrieve customers, orders, and products where CategoryID = 1.
34. List order_details, products, and orders where Quantity >= 20 OR UnitPrice > 25.
35. Show employees and orders where OrderDate BETWEEN '1996-07-10' AND '1996-07-30'.
36. Find products, suppliers, and categories where UnitsInStock < 20 AND UnitPrice > 10.
37. List customers and orders where Country = 'Mexico' AND PostalCode >= '5000'.
38. Show orders, products, and order_details where ProductID <= 10.
39. Retrieve employees and orders where BirthDate >= '1960-01-01'.
40. List products and categories where CategoryID IN (1,2,3).
41. Show suppliers and products where UnitsOnOrder = 0 OR Discontinued = 1.
42. Find customers, orders, and shippers where ShipperID = 3 AND OrderDate < '1996-07-20'.
43. List order_details, orders, and products where Quantity > 5 AND UnitPrice <= 20.
44. Show products, categories, and suppliers where CategoryID = 5 AND UnitsInStock >= 10.
45. Retrieve orders, customers, and employees where CustomerID > 50.
46. List products and suppliers where UnitPrice BETWEEN 10 AND 20.
47. Show customers and orders where Country NOT IN ('USA','UK').
48. Find order_details, orders, and products where Quantity < 15 AND CategoryID = 2.
49. List employees and orders where EmployeeID <= 5 OR OrderDate >= '1996-07-25'.

50. Show products, suppliers, and categories where UnitsOnOrder > 0 AND CategoryID IN (1,3,5).

Subqueries

SINGLE-ROW SUBQUERY

1. Find the product whose UnitPrice is higher than the UnitPrice of ProductID = 1.
2. List customers whose CustomerID is greater than the CustomerID of the most recent order.
3. Find employees born earlier than the employee with EmployeeID = 5.
4. Show products with UnitsInStock greater than the UnitsInStock of 'Chai'.
5. List orders placed after the first order date in the Orders table.
6. Find suppliers located in the same country as SupplierID = 1.
7. Retrieve products whose UnitPrice is less than the minimum UnitPrice of CategoryID = 2.
8. Show customers living in the same city as the customer with CustomerID = 10.
9. Find orders whose Freight is greater than the freight of OrderID = 10248.
10. List products that have a UnitPrice equal to the average UnitPrice of all products.
11. Find categories with CategoryID greater than the maximum CategoryID of products in stock.
12. Show employees whose HireDate is after the HireDate of EmployeeID = 3.
13. Retrieve customers whose PostalCode matches the PostalCode of CustomerID = 5.
14. Find orders shipped using the same ShipperID as OrderID = 10255.
15. Show products in the same category as the cheapest product.
16. Retrieve suppliers whose City matches the City of SupplierID = 4.
17. List orders placed by the customer who placed the earliest order.
18. Find products with UnitsInStock less than the UnitsInStock of the most expensive product.
19. Show customers from the same Country as the company with highest CustomerID.
20. Retrieve categories whose CategoryID equals the category of the product with max UnitPrice.
21. List products whose ReorderLevel is higher than the reorder level of ProductID = 3.
22. Find orders handled by the employee with the lowest EmployeeID.
23. Show products priced lower than the average price of CategoryID = 1.
24. Find suppliers from the same country as the supplier offering the cheapest product.
25. List products with UnitsInStock equal to the maximum UnitsInStock of any discontinued product.
26. Find orders placed on the same date as OrderID = 10260.
27. Retrieve products whose UnitPrice is more than the UnitPrice of 'Chang'.

28. List employees whose LastName comes alphabetically after the LastName of EmployeeID = 2.
29. Find customers whose Country matches the Country of the customer with the earliest CustomerID.
30. Show order_details where Quantity is greater than the Quantity of OrderID = 10248 for ProductID = 11.

MULTI-ROW SUBQUER

1. Find products whose CategoryID is **IN** the list of categories that have more than 1 product.
2. List customers whose CustomerID is **IN** the set of customers who placed orders in July 1996.
3. Find employees whose EmployeeID is **IN** the list of employees who handled orders shipped by ShipperID = 3.
4. Show orders where CustomerID is **IN** the group of customers from Germany.
5. List products whose SupplierID is **IN** suppliers located in Japan.
6. Show categories whose CategoryID is **IN** the categories used by discontinued products.
7. Find suppliers whose country is **IN** the customer countries from which orders were placed.
8. Retrieve products whose ProductID is **IN** the list of ProductIDs in order_details with Quantity > 30.
9. Show employees whose EmployeeID is **IN** the list of employees who handled orders for CustomerID 20.
10. Find products whose UnitPrice is greater than **ANY** UnitPrice of products in CategoryID = 1.
11. Find products whose UnitPrice is greater than **ALL** UnitPrices in CategoryID = 2.
12. Show orders whose OrderID is **IN** the top 5 earliest order IDs.
13. Find customers whose CustomerID is **IN** the list of customers who placed orders shipped by 'Federal Shipping'.
14. List suppliers whose SupplierID is **IN** the SupplierIDs of products priced above 20.
15. Find orders whose ShipperID is **IN** the list of shippers used by EmployeeID = 4.
16. Show products whose ProductID is **NOT IN** the order_details table (never ordered products).
17. Retrieve employees whose EmployeeID is **IN** employees with birthdates earlier than **ANY** customer registration date (if added).
18. Find categories whose CategoryID is **IN** categories having products out of stock.
19. Show customers whose City is **IN** the cities of suppliers.

20. Find orders whose OrderDate is earlier than **ALL** order dates of CustomerID 34.
21. Retrieve products whose UnitPrice is less than **ANY** UnitPrice in CategoryID = 5.
22. Show suppliers whose SupplierID is **IN** the SupplierIDs of the cheapest products.
23. Find orders whose CustomerID is **IN** customers with PostalCode starting with '5'.
24. Show order_details whose Quantity is greater than **ALL** quantities of OrderID = 10248.
25. Retrieve products whose UnitsInStock is less than **ANY** UnitsInStock of discontinued products.
26. Find orders where EmployeeID is **IN** employees who were born in the 1950s.
27. Show categories whose CategoryID is **IN** categories of products with UnitsOnOrder > 20.
28. List customers whose Country is **IN** the countries of suppliers who supply CategoryID = 1 products.
29. Find products whose UnitPrice is greater than **ANY** UnitPrice of products ordered by CustomerID = 90.
30. Retrieve shippers whose ShipperID is **IN** orders placed after '1996-07-20'.

NESTED SUBQUERIES

Products

1. Find products whose UnitPrice is higher than the average UnitPrice of products in the most expensive category.
2. List products that belong to the category of the cheapest product supplied by suppliers in Japan.
3. Retrieve products whose UnitsInStock is greater than the maximum UnitsInStock of any product in categories that include a discontinued product.
4. Show products priced higher than the average price of products ordered by customers from Germany.
5. Find products whose SupplierID matches suppliers in countries that have more than 3 suppliers.

Customers / Orders

6. List customers who placed orders in the same month as the earliest order placed by EmployeeID = 4.
7. Find customers who ordered products from the category with the highest number of products.
8. Retrieve customers who placed the maximum number of orders among all customers.
9. List customers whose city matches the city of suppliers who supply products

to OrderID = 10248.

10. Show customers who ordered any product that has a UnitPrice greater than the maximum UnitPrice of CategoryID = 1.

Employees

11. Find employees who handled orders placed by customers in the country with the highest number of customers.
12. Show employees who processed orders containing products with quantities higher than the average quantity of all order_details.
13. Retrieve employees whose birthdates are earlier than the minimum birthdate of employees who handled more than 5 orders.
14. List employees who worked on orders shipped by the shipper that delivered the fewest orders.
15. Find employees who handled orders for customers who belong to the city that appears most frequently in the customers table.

Orders

16. Retrieve orders whose OrderDate is earlier than the minimum OrderDate of orders shipped by ShipperID = 3.
17. Show orders placed by customers who also ordered products supplied by suppliers in the same country as SupplierID = 2.
18. Find orders that include products whose UnitPrice is greater than the average UnitPrice of products supplied by SupplierID = 1.
19. Show orders placed by the customers who placed orders with the maximum total quantity in order_details.
20. Retrieve orders where the employee who processed them has a birthdate earlier than the birthdate of EmployeeID = 1.

Suppliers / Categories

21. Find suppliers who supply products in categories where the average UnitPrice is above 25.
22. List suppliers whose country matches the country of customers who placed orders with OrderID > 10300.
23. Show suppliers who supply products more expensive than the maximum UnitPrice in CategoryID = 4.
24. Retrieve suppliers who are located in countries that have customers who ordered the cheapest product.
25. List suppliers who supply products ordered in quantities greater than the maximum quantity ordered in July 1996.

Order Details

26. Find order_details where quantity is higher than the average quantity of all order_details for orders placed by CustomerID = 20.
27. List order_details for products more expensive than the minimum UnitPrice of products ordered by EmployeeID = 4.
28. Show order_details whose Quantity is greater than the maximum Quantity for products in CategoryID = 3.
29. Retrieve order_details that belong to orders placed by customers in the country with the earliest customer record.
30. List order_details where ProductID exists in products supplied by suppliers

located in the country with the most suppliers.

CORRELATED SUBQUERIES

Products

1. Find products whose UnitPrice is higher than the average UnitPrice of other products in the *same category*.
2. List products whose UnitsInStock is lower than the average UnitsInStock of all products from the same supplier.
3. Retrieve products that are the most expensive product within their category.
4. Find products that have a UnitPrice greater than every other product from the same supplier.
5. List products where UnitsOnOrder is greater than the total UnitsOnOrder of all other products in the same category.

Suppliers

6. Find suppliers who supply more products than the average number of products supplied by all suppliers.
7. Retrieve suppliers who supply at least one product that is priced higher than the average UnitPrice of that product's category.
8. List suppliers who supply products with UnitsInStock below the average UnitsInStock within each category.
9. Show suppliers whose average product price is higher than the average price of products for all suppliers.
10. Find suppliers that supply the cheapest product in their country.

Customers

11. Find customers who have placed more orders than the average order count of customers in their country.
12. List customers whose total quantity ordered is greater than the total quantity ordered by any other customer in the same city.
13. Retrieve customers who placed at least one order containing a product whose UnitPrice is greater than the average UnitPrice purchased by that customer.
14. Find customers whose latest order date is earlier than the latest order date of other customers in the same country.
15. Show customers who ordered a product more times than any other customer ordered that same product.

Employees

16. List employees whose age is lower than the average age of employees who processed orders for the same customers they served.
17. Find employees who handled more orders than the average orders handled by employees with the same birth decade.
18. Retrieve employees whose total quantity processed in orders is greater than the total quantity processed by all other employees.
19. Show employees who shipped orders faster than the average shipping speed of other employees (requires shipping date if available).

20. List employees who processed orders for customers that no other employee processed.

Orders

21. Find orders whose OrderDate is earlier than the average OrderDate of all orders by the same customer.
22. List orders where the freight (if exists) is higher than the average freight charged by the same shipper.
23. Retrieve orders whose total quantity (summed in order_details) is greater than the total quantity of all other orders placed on the same date.
24. Show orders that include at least one product whose price is higher than the average price of products in that order.
25. Find orders whose employee handled *fewer* total orders than the average orders handled by employees in that same month.

Order Details

26. Find order_details where Quantity is greater than the average quantity for products in the same category.
27. List order_details for which the product was ordered more frequently by this customer than any other customer.
28. Retrieve order_details where the price of the product is higher than the average price of all products in other order_details of the same OrderID.
29. Show order_details where quantity exceeds the average quantity ordered by the same employee across all their orders.
30. Find order_details where the product belongs to a category that the ordering customer has purchased more than any other category.

SUBQUERY WITH JOIN

1. Find customers who placed orders for products supplied by suppliers located in the same country as the customer (use subquery + join).
2. List employees whose orders include any product from the 'Beverages' category (join with subquery).
3. Find the customers whose last order amount is greater than the average order amount of their country (subquery + join).
4. List all products whose total sales quantity is higher than the average quantity sold across all products (subquery + join).
5. Find suppliers who supply products that appear in more than 10 different orders (subquery + join).
6. List customers who ordered items shipped by the shipper with the highest number of orders (subquery + join).
7. Find employees whose total sales are above the overall employee average (subquery + join).
8. Display products ordered by customers from the same city as the employee who handled the order (subquery + join).
9. Find categories that include products purchased by customers from more

than 5 countries (subquery + join).

10. List suppliers who have never supplied any product ordered by customers from Germany (subquery + join).
11. List customers who placed orders for the most expensive product (subquery + join).
12. Find employees who processed orders for all shippers used in the last 6 months (subquery + join).
13. Show products whose suppliers are located in a country where no customer has placed an order (subquery + join).
14. List employees who handled orders placed by customers who have above-average purchase frequency (subquery + join).
15. Find customers who placed orders for products from suppliers with a total product count above 5 (subquery + join).
16. List customers whose total order amount is above the average order amount of their region (subquery + join).
17. List employees who processed orders containing products from the category with the highest number of products (subquery + join).
18. Find products supplied by suppliers who have the minimum number of supplied products (subquery + join).
19. Show orders that include products more expensive than the average product price of their category (subquery + join).
20. Find customers who placed orders for products that appear in the top 3 most ordered items (subquery + join).
21. List products purchased only by customers from a specific set of countries (subquery + join).
22. Find employees who processed orders shipped by the shipper with the lowest average shipping delay (subquery + join).
23. List suppliers whose products have never been ordered by customers from the USA (subquery + join).
24. Show products that were ordered by customers but never shipped by “Speedy Express” (subquery + join).
25. Find customers who placed orders for products supplied by suppliers in the same region as the employee processing the order (subquery + join).
26. Display orders whose order amount is above the average of orders for the same customer (subquery + join).
27. List products that belong to categories where total sales exceed the average sales of all categories (subquery + join).
28. Find employees whose sales are greater than the sales of at least two other employees (subquery + join).
29. List customers who ordered products that are also ordered by customers from Canada (subquery + join).
30. Find products supplied by suppliers whose total sales count is higher than average supplier sales (subquery + join).
31. List orders that include products from the supplier supplying the cheapest product (subquery + join).
32. Find customers who ordered products from the supplier with the highest

number of orders (subquery + join).

33. List categories that include products ordered by customers who never ordered from category “Confections” (subquery + join).
34. Show employees whose orders include any product purchased by customers from Brazil (subquery + join).
35. Find shippers who shipped products that belong to categories with total quantity > 500 (subquery + join).
36. List customers who ordered products supplied only by suppliers from Europe (subquery + join).
37. Find suppliers with at least one product purchased by customers who placed above-average number of orders (subquery + join).
38. List products purchased in orders handled by employees who have total sales below average (subquery + join).
39. Show customers who ordered products that are more expensive than all products from category ‘Seafood’ (subquery + join).
40. List employees who processed orders that include the product with the highest total sales quantity (subquery + join).
41. List customers whose orders contain any product supplied by suppliers with the minimum average product price (subquery + join).
42. Find products ordered by customers who belong to countries where average order size is highest (subquery + join).
43. List suppliers whose products are ordered only by customers from cities where employee sales are highest (subquery + join).
44. Find customers who placed orders that include products from categories with the fewest products (subquery + join).
45. List employees who handled orders shipped by the shipper with the highest delivery count (subquery + join).
46. Show products that were shipped more times than the average shipping count for all products (subquery + join).
47. Find customers whose total spending is higher than the highest spending of customers in Germany (subquery + join).
48. List categories whose products are ordered by customers who never placed more than 5 orders (subquery + join).
49. Show employees who processed orders for customers from more than 10 cities (subquery + join).
50. List products ordered in orders placed by customers who ordered from all shippers (subquery + join).

Window Functions

OVER()

1. Display each order with its total order amount and also show the running total of order amounts sorted by order date (using SUM() OVER()).
2. For each employee, show each order they handled and also display the total sales per employee (using SUM() OVER(PARTITION BY)).
3. Show each product with its total ordered quantity and its rank based on total quantity sold (using RANK() OVER()).
4. Display customers with each of their orders along with the average order amount per customer (using AVG() OVER(PARTITION BY)).
5. For each category, list all products with their price and also show the maximum price in that category using a window function.
6. Show each order with order amount and calculate the difference between that order amount and the customer's average order amount (using window AVG + arithmetic).
7. Display the top 3 best-selling products in each category (using ROW_NUMBER() OVER(PARTITION BY ORDER BY)).
8. Show customers and their orders with a field showing the total number of orders by each customer (using COUNT() OVER(PARTITION BY)).
9. For each employee, show each order and the employee's highest order amount (using MAX() OVER(PARTITION BY)).
10. List products along with the cumulative quantity sold ordered by product ID (using running SUM).
11. For each supplier, list their products with average product unit price for that supplier (window AVG).
12. Show each order along with total number of orders placed before that order (using ROWS BETWEEN).
13. For each customer, show the orders with their order rank (1st, 2nd, 3rd…) based on order date.
14. Display categories with product prices and show how much each product price differs from the category minimum (window MIN + arithmetic).
15. For each shipper, list the orders they shipped and the percentage contribution of each order to total shipped revenue (window SUM + division).
16. Show each product with its total sales and display the dense rank of the product across all categories.
17. For each country, list customers and their total orders and show the country-wise customer rank based on number of orders.
18. Display all orders sorted by amount and show lag value (previous order amount) using LAG() OVER().
19. Display all orders and show the lead value (next order amount) using LEAD() OVER().
20. For each product, calculate total sales and display the moving average of sales

across products ordered by product ID.

21. For each supplier, show products and calculate the running count of products (window COUNT).
22. List employees and show each order with a column showing the employee's minimum order amount (window MIN).
23. Show customers and their orders with the running order count (1,2,3…) per customer sorted by date.
24. For each category, display products and show the category revenue rank using RANK() OVER(PARTITION BY).
25. Show suppliers with their products and calculate the difference between product price and supplier average price.
26. For each product, show total sales and compute the ratio of its sales to the highest product sales using MAX() OVER().
27. Show each order with the highest order amount recorded so far (running MAX).
28. For each customer, show orders and compute the order streak using row numbers (continuously increasing order count).
29. For each employee, show orders and the rolling average of order amounts ordered by date.
30. For each category, list products and show the percentile rank of each product price within that category (using window functions with NTILE()).

PARTITION BY

1. Show each order along with the total order amount per customer using SUM() OVER(PARTITION BY CustomerID).
2. For each employee, list all orders handled and also show the total number of orders handled by that employee.
3. Display each product with its total ordered quantity and show the total quantity sold per category.
4. For each supplier, list the products and show the average UnitPrice for that supplier.
5. For each category, list products and display the maximum UnitPrice inside the category.
6. Show each customer’s orders and display the latest order date per customer.
7. For each product, show all its order_details and the total revenue generated by each product.
8. For each shipper, list the orders shipped and display the count of orders shipped by each shipper.
9. For each employee, show all orders and the sum of all orders' amounts handled by that employee.
10. For each category, display products and show the minimum UnitsInStock for that category.
11. List suppliers and their products showing the supplier-wise total number of

products supplied.

12. For each customer, list orders and show the customer's average order amount.
13. For each product, show all orders and display the maximum quantity ordered for that product.
14. For each order, list order_details and show the total items ordered within that order.
15. For each customer, show orders and the customer-wise order count.
16. For each category, show all products and the sum of UnitsOnOrder in that category.
17. For each employee, list orders and show the minimum order value handled by that employee.
18. For each supplier, show products and the total UnitsInStock per supplier.
19. For each product, list its order details and show the average quantity ordered for that product.
20. For each country, show customers and the total number of customers in that country.
21. For each category, show products and the average product price in that category.
22. For each customer, show orders and the highest order amount they ever placed.
23. For each shipper, list orders and display the total freight cost shipped by that shipper.
24. For each order, show order_details and the maximum discount applied in the order (if discount column exists).
25. For each employee, display orders and the employee-wise average order amount.
26. For each supplier, list products and show the maximum UnitsOnOrder for products supplied by that supplier.
27. For each category, show products and display the total number of products in that category.
28. For each customer, list orders and show the count of distinct products ordered using window COUNT + PARTITION BY.
29. For each employee, show orders and the employee-wise last order date.
30. For each customer, show every order and show the customer's total quantity ordered across all orders.

ORDER BY Inside Window

1. For each customer, list their orders and assign order numbers (1st, 2nd, 3rd) based on OrderDate using ROW_NUMBER() OVER(PARTITION BY CustomerID ORDER BY OrderDate).
2. For each employee, show orders ranked by highest order amount using RANK() OVER(PARTITION BY EmployeeID ORDER BY OrderAmount

DESC).

3. For each product, list all order_details and sort by quantity using DENSE_RANK() OVER(PARTITION BY ProductID ORDER BY Quantity DESC).
4. For each customer, show orders along with a running total of order amounts using SUM() OVER(PARTITION BY CustomerID ORDER BY OrderDate).
5. For each employee, show a running count of orders sorted by OrderDate.
6. For each product, list orders and show the running total quantity sold ordered by OrderID.
7. For each category, show products and assign a price ranking (highest to lowest).
8. For each supplier, list products and assign a stock rank based on UnitsInStock.
9. For each customer, show their orders with the difference from the previous order amount using LAG().
10. For each customer, show their orders with the next order date using LEAD().
11. For each shipper, list orders and rank them by freight cost.
12. For each employee, show their orders and compute moving average order amount using window ORDER BY.
13. For each category, show products and compute a running minimum price sorted by UnitPrice.
14. For each product, show total sales and calculate the percentage of running total sales.
15. For each customer, list orders and compute the gap between two consecutive orders (date difference).
16. For each supplier, sort products by price and calculate price difference from previous product.
17. For each category, sort products by UnitsOnOrder and calculate a running max.
18. For each customer, sort orders by amount and find the top 3 order amounts.
19. For each employee, list orders and calculate the cumulative revenue handled by the employee.
20. For each shipper, list orders sorted by freight and calculate a running average freight cost.
21. For each category, sort products by UnitPrice and assign quartile groups using NTILE(4).
22. For each product, list all orders and calculate running revenue sorted by OrderID.
23. For each supplier, sort products by UnitsInStock and calculate difference from category average using window avg ordered.
24. For each employee, list their orders and show how much each order contributes to running revenue (percentage).
25. For each customer, sort orders by OrderDate and mark the first order of each year.
26. For each order, sort order_details by Quantity and assign row numbers.
27. For each category, sort products by price and get the second highest priced

product (using row_number).

28. For each product, sort order_details by quantity and compute running max quantity.
29. For each supplier, sort products by UnitPrice and compute the ratio of price to previous product’s price.
30. For each customer, sort orders by amount and categorize them into NTILE(5) bucket groups (low → high spenders).

ROW_NUMBER()

1. Assign a row number to each order based on OrderDate (earliest first).
2. For each customer, assign row numbers to their orders sorted by OrderDate.
3. For each employee, number their orders by OrderAmount (highest first).
4. For each category, assign row numbers to products sorted by UnitPrice (highest to lowest).
5. For each product, assign row numbers to order_details sorted by Quantity.
6. For each supplier, number products by UnitsInStock (highest first).
7. Assign row numbers to customers sorted by Country and then by CustomerName.
8. For each shipper, number orders sorted by Freight.
9. For each category, number products by UnitsOnOrder.
10. Number employees based on their BirthDate (youngest first).
11. For each city, number customers sorted alphabetically by CustomerName.
12. For each order, assign row numbers to order_details sorted by Quantity (largest first).
13. Number suppliers by CompanyName alphabetically.
14. For each customer, assign row numbers to orders sorted by OrderAmount (highest first).
15. For each employee, number orders by OrderDate (oldest first).
16. For each product, number order_details by discount (if discount exists).
17. For each category, number products by UnitsInStock (lowest first).
18. Number employees by number of orders they handled (requires join + window).
19. For each country, number customers by total orders they placed (requires join).
20. For each supplier, number products by UnitPrice from lowest to highest.
21. For each product, number order_details by OrderID.
22. For each shipper, number orders by OrderDate.
23. Assign row numbers to all categories sorted by CategoryName.
24. For each employee, assign row numbers to orders sorted by total items (sum of quantities).
25. For each customer, assign row numbers to products they ordered ordered by total quantity purchased.
26. For each country, number suppliers alphabetically by supplier name.

27. For each category, number products by value of stock (UnitsInStock * UnitPrice).
28. For customers in the same city, number them by CustomerID.
29. For each employee, number orders by freight cost of the shipped orders.
30. For each supplier, assign row numbers to products based on UnitsOnOrder (highest first).

RANK()

1. Rank orders by total order amount, highest amount first.
2. For each customer, rank their orders by OrderDate (earliest first).
3. For each employee, rank their orders by OrderAmount (largest first).
4. Rank products within each category by UnitPrice (highest first).
5. For each product, rank order_details by Quantity (highest first).
6. Rank suppliers by the number of products they supply (most first).
7. Rank customers by total number of orders placed.
8. For each shipper, rank their orders by Freight (highest first).
9. Rank categories by total products sold.
10. For each employee, rank orders by Quantity of items in the order.
11. Rank customers by total spending (sum of all orders).
12. For each supplier, rank products by UnitsInStock (highest first).
13. Rank employees by total number of orders handled.
14. For each product, rank orders by UnitPrice.
15. Rank customers in each country by total order amount.
16. For each category, rank products by UnitsOnOrder.
17. Rank suppliers by total sales value of their products.
18. For each shipper, rank orders by order date (oldest first).
19. Rank products across all categories by total quantity sold.
20. For each customer, rank orders by total quantity purchased.
21. Rank employees by total revenue handled from orders.
22. For each category, rank products by total sales value (UnitsInStock * UnitPrice).
23. Rank suppliers by total UnitsOnOrder across their products.
24. For each customer, rank products ordered by total quantity purchased.
25. Rank orders by the number of items in the order.
26. For each employee, rank orders by Freight cost (highest first).
27. Rank customers by the number of distinct products purchased.
28. For each category, rank products by total number of times ordered.
29. Rank shippers by total freight revenue.
30. For each supplier, rank products by UnitPrice within the supplier.

DENSE_RANK()

1. Rank orders by total amount using DENSE_RANK() (highest first).
2. For each customer, rank their orders by OrderDate (earliest first).
3. For each employee, rank their orders by OrderAmount (largest first).
4. Rank products within each category by UnitPrice (highest first).
5. For each product, rank order_details by Quantity (highest first).
6. Rank suppliers by number of products supplied (most first).
7. Rank customers by total number of orders placed.
8. For each shipper, rank their orders by Freight (highest first).
9. Rank categories by total products sold.
10. For each employee, rank orders by Quantity of items in the order.
11. Rank customers by total spending (sum of all orders).
12. For each supplier, rank products by UnitsInStock (highest first).
13. Rank employees by total number of orders handled.
14. For each product, rank orders by UnitPrice.
15. Rank customers in each country by total order amount.
16. For each category, rank products by UnitsOnOrder.
17. Rank suppliers by total sales value of their products.
18. For each shipper, rank orders by order date (oldest first).
19. Rank products across all categories by total quantity sold.
20. For each customer, rank orders by total quantity purchased.
21. Rank employees by total revenue handled from orders.
22. For each category, rank products by total sales value (UnitsInStock * UnitPrice).
23. Rank suppliers by total UnitsOnOrder across their products.
24. For each customer, rank products ordered by total quantity purchased.
25. Rank orders by the number of items in the order.
26. For each employee, rank orders by Freight cost (highest first).
27. Rank customers by the number of distinct products purchased.
28. For each category, rank products by total number of times ordered.
29. Rank shippers by total freight revenue.
30. For each supplier, rank products by UnitPrice within the supplier.

LAG()

1. For each customer, show their orders along with the previous order date.
2. For each employee, list orders and show the previous order amount handled by the employee.
3. For each product, list order_details and show the previous quantity ordered.
4. For each category, show products and the previous product’s UnitPrice (sorted by UnitPrice).
5. For each supplier, list products and show the previous UnitsInStock.
6. For each customer, display orders and calculate the difference between current and previous order amount.
7. For each employee, show orders and show the previous order’s Freight cost.

8. For each product, show orders and show the previous order’s UnitPrice.
9. For each customer, display orders with the previous order’s total quantity.
10. For each order, show order_details and the previous quantity ordered for the same product.
11. For each category, list products and show the difference between UnitPrice and previous UnitPrice.
12. For each employee, list orders and show the previous order date.
13. For each customer, display orders and show the previous order’s ShipperID.
14. For each product, show all order_details and show the previous OrderID for that product.
15. For each supplier, list products and show the previous UnitPrice.
16. For each customer, show orders sorted by OrderDate and show previous order amount for comparison.
17. For each category, show products sorted by UnitsInStock and show the previous stock value.
18. For each employee, show orders sorted by OrderDate and calculate time difference from previous order.
19. For each product, show order_details sorted by OrderID and show previous quantity ordered.
20. For each customer, show orders and display previous product purchased in that order.
21. For each shipper, show orders sorted by OrderDate and display previous order Freight.
22. For each product, show order_details sorted by Quantity and display previous order’s Quantity.
23. For each supplier, show products sorted by UnitPrice and display previous price for comparison.
24. For each category, show products sorted by UnitPrice and calculate difference from previous product price.
25. For each employee, show orders and display the previous customer handled.
26. For each customer, show orders sorted by OrderID and display the previous total order quantity.
27. For each product, list orders and show previous order’s UnitsOnOrder.
28. For each customer, display orders and show the previous ShipperID used.
29. For each employee, show orders and display the previous OrderID they handled.
30. For each customer, show their orders and calculate the difference between current order amount and previous order amount

---

LEAD()

1. For each customer, show their orders along with the next order date.
2. For each employee, list orders and show the next order amount handled by the employee.

3. For each product, list order_details and show the next quantity ordered.
4. For each category, show products and the next product's UnitPrice (sorted by UnitPrice).
5. For each supplier, list products and show the next UnitsInStock.
6. For each customer, display orders and calculate the difference between next order amount and current order amount.
7. For each employee, show orders and show the next order's Freight cost.
8. For each product, show orders and show the next order's UnitPrice.
9. For each customer, display orders with the next order's total quantity.
10. For each order, show order_details and the next quantity ordered for the same product.
11. For each category, list products and show the difference between next UnitPrice and current UnitPrice.
12. For each employee, list orders and show the next order date.
13. For each customer, display orders and show the next order's ShipperID.
14. For each product, show all order_details and show the next OrderID for that product.
15. For each supplier, list products and show the next UnitPrice.
16. For each customer, show orders sorted by OrderDate and show next order amount for comparison.
17. For each category, show products sorted by UnitsInStock and show the next stock value.
18. For each employee, show orders sorted by OrderDate and calculate time difference to next order.
19. For each product, show order_details sorted by OrderID and show next quantity ordered.
20. For each customer, show orders and display next product purchased in that order.
21. For each shipper, show orders sorted by OrderDate and display next order Freight.
22. For each product, show order_details sorted by Quantity and display next order's Quantity.
23. For each supplier, show products sorted by UnitPrice and display next price for comparison.
24. For each category, show products sorted by UnitPrice and calculate difference from next product price.
25. For each employee, show orders and display the next customer handled.
26. For each customer, show orders sorted by OrderID and display the next total order quantity.
27. For each product, list orders and show next order's UnitsOnOrder.
28. For each customer, display orders and show the next ShipperID used.
29. For each employee, show orders and display the next OrderID they handled.
30. For each customer, show their orders and calculate the difference between next order amount and current order amount.

FIRST_VALUE()

1. For each customer, show their orders and the first order date.
2. For each employee, list orders and display the first order amount they handled.
3. For each product, list order_details and show the first quantity ordered.
4. For each category, display products and show the first UnitPrice in that category.
5. For each supplier, list products and display the first product added (lowest ProductID).
6. For each customer, show orders and display the first ShipperID used.
7. For each employee, show orders and display the first OrderDate sorted by OrderDate.
8. For each product, show orders and display the first order's UnitPrice.
9. For each category, show products and display the first product name alphabetically.
10. For each customer, display orders and show the first total quantity ordered.
11. For each supplier, list products and show the first UnitsInStock.
12. For each employee, show orders and display the first customer handled.
13. For each product, list order_details and display the first OrderID for that product.
14. For each category, show products and display the first product with UnitsOnOrder > 0.
15. For each customer, show orders and display the first order amount above 100.
16. For each employee, list orders and show the first order with Freight > 0.
17. For each product, show order_details and display the first order by quantity descending.
18. For each supplier, list products and display the first product with UnitPrice > 20.
19. For each category, show products and display the first discontinued product.
20. For each customer, show orders and display the first order placed in 1996.
21. For each employee, show orders and display the first order in each month.
22. For each product, display order_details and show the first order per product.
23. For each category, show products and display the first product sorted by UnitsInStock descending.
24. For each customer, show orders and display the first order by ShipperID.
25. For each employee, show orders and display the first high-value order (amount > 500).
26. For each product, show order_details and display the first order with Quantity > 10.
27. For each supplier, show products and display the first product alphabetically.
28. For each category, show products and display the first product with UnitsOnOrder > 0.
29. For each customer, show orders and display the first order using ShipperID = 1.
30. For each employee, show orders and display the first order with total

quantity > 50.

LAST_VALUE()

1. For each customer, show their orders and the last order date.
2. For each employee, list orders and display the last order amount they handled.
3. For each product, list order_details and show the last quantity ordered.
4. For each category, display products and show the last UnitPrice in that category.
5. For each supplier, list products and display the last product added (highest ProductID).
6. For each customer, show orders and display the last ShipperID used.
7. For each employee, show orders and display the last OrderDate sorted by OrderDate.
8. For each product, show orders and display the last order's UnitPrice.
9. For each category, show products and display the last product name alphabetically.
10. For each customer, display orders and show the last total quantity ordered.
11. For each supplier, list products and show the last UnitsInStock.
12. For each employee, show orders and display the last customer handled.
13. For each product, list order_details and display the last OrderID for that product.
14. For each category, show products and display the last product with UnitsOnOrder > 0.
15. For each customer, show orders and display the last order amount above 100.
16. For each employee, list orders and show the last order with Freight > 0.
17. For each product, show order_details and display the last order by quantity descending.
18. For each supplier, list products and display the last product with UnitPrice > 20.
19. For each category, show products and display the last discontinued product.
20. For each customer, show orders and display the last order placed in 1996.
21. For each employee, show orders and display the last order in each month.
22. For each product, display order_details and show the last order per product.
23. For each category, show products and display the last product sorted by UnitsInStock descending.
24. For each customer, show orders and display the last order by ShipperID.
25. For each employee, show orders and display the last high-value order (amount > 500).
26. For each product, show order_details and display the last order with Quantity > 10.
27. For each supplier, show products and display the last product alphabetically.
28. For each category, show products and display the last product with

UnitsOnOrder > 0.

29. For each customer, show orders and display the last order using ShipperID = 1.
30. For each employee, show orders and display the last order with total quantity > 50.

NTILE()

1. Divide all products into 4 price quartiles using NTILE(4) based on UnitPrice.
2. Divide all customers into 3 groups based on total orders placed.
3. For each category, divide products into 5 groups by UnitsInStock.
4. For each employee, divide orders into 4 groups by OrderAmount.
5. For all orders, divide them into 10 buckets by OrderDate.
6. For each supplier, divide products into 3 groups based on UnitPrice.
7. Divide all customers into 5 groups by total quantity purchased.
8. For each shipper, divide orders into 3 buckets based on Freight.
9. For each product, divide order_details into 4 groups based on Quantity.
10. Divide employees into 4 groups by total orders handled.
11. For each customer, divide orders into 4 buckets based on OrderAmount.
12. Divide categories into 3 groups based on number of products.
13. For each supplier, divide products into 5 buckets by UnitsOnOrder.
14. Divide all orders into 4 buckets based on number of items in the order.
15. For each category, divide products into 4 price quartiles.
16. For each customer, divide orders into 5 buckets based on OrderDate.
17. Divide all products into 3 groups by UnitsInStock.
18. For each employee, divide orders into 3 buckets based on Freight.
19. For each supplier, divide products into 4 groups by UnitPrice descending.
20. Divide all customers into 4 groups by total spending.
21. For each product, divide order_details into 3 buckets based on Quantity.
22. For each shipper, divide orders into 5 groups based on OrderDate.
23. Divide categories into 4 buckets based on total UnitsOnOrder.
24. For each employee, divide orders into 4 buckets based on total order quantity.
25. Divide all suppliers into 3 groups by number of products supplied.
26. For each customer, divide orders into 4 buckets by total quantity purchased.
27. For each category, divide products into 5 groups by total stock value (UnitsInStock * UnitPrice).
28. Divide all products into 3 buckets based on UnitsOnOrder.
29. For each shipper, divide orders into 4 buckets based on total Freight amount.
30. For each supplier, divide products into 5 buckets based on total stock value (UnitPrice * UnitsInStock).

Moving averages, Running total, Rolling sums

1. Calculate the running total of order amounts for each customer ordered by OrderDate.
2. Compute the running total of quantities for each product across all orders.
3. For each employee, calculate the running total of Freight for orders handled.
4. For each customer, calculate the moving average of order amounts for the last 3 orders.
5. For each product, calculate the moving average quantity ordered over the last 5 orders.
6. Compute the running total UnitsInStock for products in each category ordered by ProductID.
7. For each supplier, calculate the rolling sum of UnitsOnOrder sorted by ProductID.
8. For each customer, calculate the cumulative total quantity purchased over time.
9. For each employee, calculate the moving average of total order amount over the last 3 orders.
10. For each product, calculate the running sum of revenue (UnitPrice * Quantity) across all order_details.
11. For each customer, calculate the cumulative average order amount.
12. For each category, calculate the running total UnitsOnOrder for products sorted by UnitPrice.
13. For each employee, calculate the rolling sum of quantities ordered for orders handled.
14. Compute the running total of Freight per shipper over time.
15. For each customer, calculate the moving average order quantity for the last 3 orders.
16. For each product, calculate the cumulative sum of UnitsOnOrder.
17. For each employee, calculate the running total revenue from the orders handled.
18. For each customer, calculate the rolling sum of order amounts for 3 consecutive orders.
19. For each category, calculate the cumulative revenue of all products sorted by ProductID.
20. For each product, calculate the moving average revenue over last 5 orders.
21. For each supplier, calculate the running total of UnitsInStock.
22. For each customer, calculate the rolling sum of quantities ordered over last 4 orders.
23. For each employee, calculate the cumulative sum of Freight handled.
24. For each shipper, calculate the running total of orders shipped.
25. For each category, calculate the moving average UnitPrice over products sorted by ProductID.
26. For each product, calculate the rolling sum of quantities over last 3 order_details.
27. For each customer, calculate the cumulative sum of total revenue from orders.
28. For each employee, calculate the moving average order amount over last 5

orders.

29. For each shipper, calculate the cumulative sum of Freight per month.
30. For each product, calculate the running total UnitsInStock + UnitsOnOrder as rolling inventory.

String Functions

CONCAT

1. Concatenate CustomerName and City for all customers.
2. Concatenate FirstName and LastName for all employees.
3. Concatenate ProductName and CategoryName for all products.
4. Concatenate ShipperName and Phone for all shippers.
5. Concatenate SupplierName and City for all suppliers.
6. Concatenate CustomerName, City, and Country for all customers.
7. Concatenate FirstName, LastName, and BirthDate for all employees.
8. Concatenate ProductName and QuantityPerUnit for all products.
9. Concatenate OrderID and CustomerID for all orders.
10. Concatenate EmployeeID and OrderID for all orders.
11. Concatenate SupplierName and ContactName for all suppliers.
12. Concatenate CategoryName and DescriptionText for all categories.
13. Concatenate ProductName, UnitPrice, and UnitsInStock for all products.
14. Concatenate CustomerName and PostalCode for all customers.
15. Concatenate FirstName, LastName, and City for all employees.
16. Concatenate ShipperName, Phone, and ShipperID for all shippers.
17. Concatenate OrderID, ProductID, and Quantity for order_details.
18. Concatenate CategoryName, ProductName, and SupplierName for products.
19. Concatenate CustomerName, ContactName, and Address for all customers.
20. Concatenate Employee FirstName, LastName, and BirthDate formatted as a string.

CONCAT_WS

1. Concatenate FirstName and LastName of employees with a space: CONCAT_WS(' ', FirstName, LastName).
2. Concatenate CustomerName, City, and Country with commas.
3. Concatenate ProductName and CategoryName with a dash (-).
4. Concatenate SupplierName, City, and Country with slashes (/).
5. Concatenate ShipperName and Phone with a colon (:).
6. Concatenate CustomerName, PostalCode, and Address with a pipe (|).
7. Concatenate FirstName, LastName, and BirthDate with hyphens (-).
8. Concatenate ProductName, UnitPrice, and UnitsInStock with commas.
9. Concatenate OrderID, CustomerID, and EmployeeID with underscores (_).
10. Concatenate CategoryName and DescriptionText with a colon (:).
11. Concatenate ProductName, SupplierName, and CategoryName with ->.
12. Concatenate CustomerName, City, and Country with , for mailing label

format.

13. Concatenate FirstName, LastName, and City with a space.
14. Concatenate ShipperName, Phone, and ShipperID with hyphens (-).
15. Concatenate OrderID, ProductID, and Quantity for order_details with commas.
16. Concatenate CustomerName, ContactName, and Address with semicolons (;).
17. Concatenate ProductName, UnitPrice, and ReorderLevel with a dash (-).
18. Concatenate Employee FirstName, LastName, and BirthDate with slashes (/).
19. Concatenate CategoryName, ProductName, and SupplierName with |.
20. Concatenate CustomerName, City, PostalCode, and Country with commas.

LENGTH

1. Find the length of CustomerName for all customers.
2. Find the length of ContactName for all customers.
3. Find the length of ProductName for all products.
4. Find the length of CategoryName for all categories.
5. Find the length of SupplierName for all suppliers.
6. Find the length of ShipperName for all shippers.
7. Find the length of Address for all customers.
8. Find the length of City for all customers.
9. Find the length of PostalCode for all customers.
10. Find the length of Phone for all shippers.
11. Find the length of FirstName for all employees.
12. Find the length of LastName for all employees.
13. Find the length of DescriptionText for all categories.
14. Find the length of QuantityPerUnit for all products.
15. Find the length of Country for all customers.
16. Find the length of Supplier ContactName for all suppliers.
17. Find the length of ProductName + CategoryName concatenated.
18. Find the length of CustomerName + City + Country concatenated.
19. Find the length of ShipperName + Phone concatenated.
20. Find the length of Address + PostalCode concatenated for all customers.

SUBSTRING

1. Extract the first 5 characters of CustomerName.
2. Extract the first 3 characters of ProductName.
3. Extract the last 4 characters of PostalCode.
4. Extract the first 2 letters of Country.
5. Extract the first 4 characters of City for all customers.
6. Extract characters 2 to 6 of ContactName.

7. Extract the first 5 characters of SupplierName.
8. Extract characters 3 to 7 of Employee LastName.
9. Extract the first 10 characters of Address.
10. Extract the first 6 characters of ShipperName.
11. Extract the first 8 characters of CategoryName.
12. Extract characters 5 to 10 of ProductName.
13. Extract the first 3 characters of QuantityPerUnit.
14. Extract the first 2 characters of Country for all suppliers.
15. Extract characters 4 to 8 of CustomerName.
16. Extract the first 5 characters of City for employees' addresses.
17. Extract characters 3 to 5 of PostalCode.
18. Extract the first 6 characters of ContactName for suppliers.
19. Extract characters 2 to 6 of DescriptionText for categories.
20. Extract the first 7 characters of ProductName + CategoryName concatenated.

LEFT / RIGHT

1. Get the first 5 characters of CustomerName using LEFT().
2. Get the last 3 characters of PostalCode using RIGHT().
3. Get the first 4 characters of ProductName.
4. Get the last 2 characters of Country for all customers.
5. Get the first 3 characters of City using LEFT().
6. Get the last 4 characters of ContactName using RIGHT().
7. Get the first 6 characters of SupplierName.
8. Get the last 5 characters of Phone for shippers.
9. Get the first 2 characters of CategoryName.
10. Get the last 3 characters of QuantityPerUnit for products.
11. Get the first 5 characters of Employee FirstName.
12. Get the last 4 characters of Employee LastName.
13. Get the first 8 characters of Address for customers.
14. Get the last 6 characters of ShipperName.
15. Get the first 3 characters of DescriptionText for categories.
16. Get the last 3 characters of ProductName.
17. Get the first 7 characters of CustomerName + City concatenated.
18. Get the last 5 characters of SupplierName + City concatenated.
19. Get the first 4 characters of OrderID converted to string.
20. Get the last 3 characters of PostalCode + Country concatenated.

INSTR

1. Find the position of 'a' in CustomerName.
2. Find the position of 'Street' in Address for all customers.

3. Find the position of 'Ch' in ProductName.
4. Find the position of 'Berlin' in City for all customers.
5. Find the position of 'Co' in SupplierName.
6. Find the position of 'Express' in ShipperName.
7. Find the position of 'Ltd' in CompanyName (if applicable).
8. Find the position of 'Box' in QuantityPerUnit for all products.
9. Find the position of 'Germany' in Country.
10. Find the position of 'Cheese' in DescriptionText for categories.
11. Find the position of 'Maria' in ContactName.
12. Find the position of 'Beer' in ProductName.
13. Find the position of 'USA' in Country for suppliers.
14. Find the position of 'Street' in Address for employees.
15. Find the position of 'Ale' in ProductName.
16. Find the position of 'Tokyo' in City for suppliers.
17. Find the position of 'Co' in CustomerName.
18. Find the position of 'Delivery' in ShipperName.
19. Find the position of 'Condiments' in CategoryName.
20. Find the position of 'Box' in QuantityPerUnit for products supplied by SupplierID = 1.

REPLACE

1. Replace 'Street' with 'St.' in all customer Address fields.
2. Replace 'Road' with 'Rd.' in all supplier Address fields.
3. Replace 'Box' with 'Pack' in QuantityPerUnit for products.
4. Replace 'Co.' with 'Company' in SupplierName.
5. Replace 'Ltd' with 'Limited' in SupplierName.
6. Replace 'Express' with 'Fast' in ShipperName.
7. Replace 'Soft drink' with 'Beverage' in CategoryName.
8. Replace 'Ave' with 'Avenue' in customer Address.
9. Replace 'x' with '*' in QuantityPerUnit of products.
10. Replace 'Germany' with 'DE' in customer Country.
11. Replace 'Tokyo Traders' with 'Tokyo Trade Co.' in SupplierName.
12. Replace 'Sweet' with 'Dessert' in CategoryName.
13. Replace 'Coffee' with 'Coffees' in ProductName.
14. Replace 'Ana Trujillo' with 'Ana T.' in customer ContactName.
15. Replace 'Express' with 'Delivery' in ShipperName.
16. Replace 'Beverages' with 'Drinks' in CategoryName.
17. Replace 'Obere' with 'Upper' in customer Address.
18. Replace 'Chai' with 'Chai Tea' in ProductName.
19. Replace 'Ale' with 'Beer' in ProductName.
20. Replace 'Soft drinks, coffees' with 'Drinks, Coffees' in DescriptionText for categories.

LOWER / UPPER

1. Convert all CustomerName to uppercase.
2. Convert all CustomerName to lowercase.
3. Convert all ContactName of customers to uppercase.
4. Convert all ContactName of customers to lowercase.
5. Convert all ProductName to uppercase.
6. Convert all ProductName to lowercase.
7. Convert all CategoryName to uppercase.
8. Convert all CategoryName to lowercase.
9. Convert all SupplierName to uppercase.
10. Convert all SupplierName to lowercase.
11. Convert all ShipperName to uppercase.
12. Convert all ShipperName to lowercase.
13. Convert all City of customers to uppercase.
14. Convert all City of customers to lowercase.
15. Convert all Country of suppliers to uppercase.
16. Convert all Country of suppliers to lowercase.
17. Convert FirstName and LastName of employees to uppercase using UPPER().
18. Convert FirstName and LastName of employees to lowercase using LOWER().
19. Convert concatenated CustomerName + City + Country to uppercase.
20. Convert concatenated ProductName + CategoryName + SupplierName to lowercase.

TRIM, LTRIM, RTRIM

1. Remove leading and trailing spaces from CustomerName using TRIM().
2. Remove leading spaces from CustomerName using LTRIM().
3. Remove trailing spaces from CustomerName using RTRIM().
4. Remove spaces from ContactName using TRIM().
5. Remove leading spaces from ContactName using LTRIM().
6. Remove trailing spaces from ContactName using RTRIM().
7. Remove leading and trailing spaces from ProductName.
8. Remove leading spaces from ProductName.
9. Remove trailing spaces from ProductName.
10. Remove spaces from SupplierName using TRIM().
11. Remove leading spaces from SupplierName using LTRIM().
12. Remove trailing spaces from SupplierName using RTRIM().
13. Remove leading and trailing spaces from City for all customers.
14. Remove leading spaces from City for all customers.
15. Remove trailing spaces from City for all customers.
16. Remove leading and trailing spaces from ShipperName.
17. Remove leading spaces from CategoryName.

18. Remove trailing spaces from CategoryName.
19. Remove spaces from concatenated CustomerName + City + Country using TRIM().
20. Remove leading and trailing spaces from Address for all customers.

Data Cleaning Functions

COALESCE()

1. Show customer City, but if it’s NULL, display 'Unknown City'.
2. Show customer PostalCode, but if it’s NULL, display '00000'.
3. Show ContactName, but if NULL, display 'No Contact'.
4. Show product UnitPrice, but if NULL, display 0.
5. Show product UnitsInStock, but if NULL, display 0.
6. Show product UnitsOnOrder, but if NULL, display 0.
7. Show supplier Phone, but if NULL, display 'No Phone'.
8. Show supplier City, but if NULL, display 'Unknown City'.
9. Show employee BirthDate, but if NULL, display '1900-01-01'.
10. Show shipper Phone, but if NULL, display 'No Phone'.
11. Show Address for customers, if NULL display 'No Address'.
12. Show CategoryName, if NULL display 'Unknown Category'.
13. Show DescriptionText, if NULL display 'No Description'.
14. Show Discontinued for products, if NULL display 0.
15. Show ReorderLevel for products, if NULL display 0.
16. Show QuantityPerUnit for products, if NULL display 'Not Specified'.
17. Show customer Country, if NULL display 'Unknown'.
18. Show supplier Country, if NULL display 'Unknown'.
19. Show order ShipperID, if NULL display 1 (default shipper).
20. Show employee LastName, if NULL display 'Unknown LastName'.
21. Show employee FirstName, if NULL display 'Unknown FirstName'.
22. Show OrderDate, if NULL display '1900-01-01'.
23. Show OrderID in order_details, if NULL display 0.
24. Show ProductID in order_details, if NULL display 0.
25. Show Quantity in order_details, if NULL display 1.
26. Show concatenated CustomerName + City, but replace NULLs with 'Unknown'.
27. Show concatenated ProductName + SupplierName, but replace NULLs with 'Unknown'.
28. Show total stock UnitsInStock + UnitsOnOrder, replace NULL with 0.
29. Show UnitPrice, but if NULL replace with average UnitPrice of the category.
30. Show ContactName, if NULL replace with 'Customer Not Available'.

NULLIF()

1. Return UnitPrice, but NULL if it equals 0.
2. Return UnitsInStock, but NULL if it equals 0.

3. Return UnitsOnOrder, but NULL if it equals 0.
4. Return ReorderLevel, but NULL if it equals 0.
5. Return Discontinued, but NULL if it equals 0.
6. Return PostalCode, but NULL if it equals '00000'.
7. Return Phone, but NULL if it equals 'No Phone'.
8. Return Country, but NULL if it equals 'Unknown'.
9. Return City, but NULL if it equals 'Unknown'.
10. Return CustomerName, but NULL if it equals 'Unknown'.
11. Return ContactName, but NULL if it equals 'No Contact'.
12. Return ProductName, but NULL if it equals 'Unknown Product'.
13. Return CategoryName, but NULL if it equals 'Unknown Category'.
14. Return SupplierName, but NULL if it equals 'Unknown Supplier'.
15. Return ShipperName, but NULL if it equals 'Unknown Shipper'.
16. Return Employee FirstName, but NULL if it equals 'Unknown'.
17. Return Employee LastName, but NULL if it equals 'Unknown'.
18. Return Address, but NULL if it equals 'No Address'.
19. Return OrderDate, but NULL if it equals '1900-01-01'.
20. Return Quantity, but NULL if it equals 0 in order_details.
21. Return CustomerID, but NULL if it equals 0.
22. Return EmployeeID, but NULL if it equals 0.
23. Return ProductID, but NULL if it equals 0.
24. Return SupplierID, but NULL if it equals 0.
25. Return ShipperID, but NULL if it equals 0.
26. Return UnitPrice, but NULL if it equals UnitPrice of another product (self-comparison).
27. Return UnitsInStock, but NULL if it equals UnitsOnOrder.
28. Return concatenated CustomerName + City, but NULL if City = 'Unknown'.
29. Return QuantityPerUnit, but NULL if it equals 'Not Specified'.
30. Return DescriptionText, but NULL if it equals 'No Description'.

IFNULL()

1. Replace NULL UnitPrice with 0.
2. Replace NULL UnitsInStock with 0.
3. Replace NULL UnitsOnOrder with 0.
4. Replace NULL ReorderLevel with 0.
5. Replace NULL Discontinued with 0.
6. Replace NULL PostalCode with '00000'.
7. Replace NULL Phone with 'No Phone'.
8. Replace NULL Country with 'Unknown'.
9. Replace NULL City with 'Unknown City'.
10. Replace NULL CustomerName with 'Unknown Customer'.
11. Replace NULL ContactName with 'No Contact'.
12. Replace NULL ProductName with 'Unknown Product'.

13. Replace NULL CategoryName with 'Unknown Category'.
14. Replace NULL SupplierName with 'Unknown Supplier'.
15. Replace NULL ShipperName with 'Unknown Shipper'.
16. Replace NULL Employee FirstName with 'Unknown'.
17. Replace NULL Employee LastName with 'Unknown'.
18. Replace NULL Address with 'No Address'.
19. Replace NULL OrderDate with '1900-01-01'.
20. Replace NULL Quantity in order_details with 1.
21. Replace NULL CustomerID with 0.
22. Replace NULL EmployeeID with 0.
23. Replace NULL ProductID with 0.
24. Replace NULL SupplierID with 0.
25. Replace NULL ShipperID with 1.
26. Replace NULL UnitPrice with average price of category.
27. Replace NULL UnitsInStock with UnitsOnOrder.
28. Replace NULL QuantityPerUnit with 'Not Specified'.
29. Replace NULL DescriptionText with 'No Description'.
30. Replace NULL CustomerName + City concatenated with 'Unknown'.

CAST()

1. Cast OrderID to CHAR.
2. Cast CustomerID to CHAR.
3. Cast EmployeeID to CHAR.
4. Cast ShipperID to CHAR.
5. Cast SupplierID to CHAR.
6. Cast ProductID to CHAR.
7. Cast Quantity to DECIMAL.
8. Cast UnitPrice to INT.
9. Cast UnitsInStock to DECIMAL.
10. Cast UnitsOnOrder to DECIMAL.
11. Cast ReorderLevel to DECIMAL.
12. Cast Discontinued to CHAR.
13. Cast OrderDate to CHAR.
14. Cast BirthDate of employees to CHAR.
15. Cast PostalCode to INT.
16. Cast Phone to CHAR.
17. Cast UnitPrice to FLOAT.
18. Cast UnitsInStock + UnitsOnOrder to DECIMAL.
19. Cast QuantityPerUnit to CHAR.
20. Cast CustomerID + OrderID to CHAR.
21. Cast EmployeeID + OrderID to CHAR.
22. Cast ProductID + SupplierID to CHAR.
23. Cast Freight to DECIMAL.

24. Cast UnitPrice * Quantity to DECIMAL.
25. Cast OrderID to VARCHAR(10).
26. Cast CategoryID to CHAR.
27. Cast Quantity to VARCHAR.
28. Cast UnitsInStock to CHAR.
29. Cast UnitPrice to VARCHAR(10).
30. Cast CustomerName to CHAR.

CONVERT()

1. Convert OrderID to CHAR.
2. Convert CustomerID to CHAR.
3. Convert EmployeeID to CHAR.
4. Convert ShipperID to CHAR.
5. Convert SupplierID to CHAR.
6. Convert ProductID to CHAR.
7. Convert Quantity to DECIMAL.
8. Convert UnitPrice to INT.
9. Convert UnitsInStock to DECIMAL.
10. Convert UnitsOnOrder to DECIMAL.
11. Convert ReorderLevel to DECIMAL.
12. Convert Discontinued to CHAR.
13. Convert OrderDate to CHAR.
14. Convert BirthDate to CHAR.
15. Convert PostalCode to INT.
16. Convert Phone to CHAR.
17. Convert UnitPrice to FLOAT.
18. Convert UnitsInStock + UnitsOnOrder to DECIMAL.
19. Convert QuantityPerUnit to CHAR.
20. Convert CustomerID + OrderID to CHAR.
21. Convert EmployeeID + OrderID to CHAR.
22. Convert ProductID + SupplierID to CHAR.
23. Convert Freight to DECIMAL.
24. Convert UnitPrice * Quantity to DECIMAL.
25. Convert OrderID to VARCHAR(10).
26. Convert CategoryID to CHAR.
27. Convert Quantity to VARCHAR.
28. Convert UnitsInStock to CHAR.
29. Convert UnitPrice to VARCHAR(10).
30. Convert CustomerName to CHAR.

CASE IF

1. Show ProductName and categorize UnitPrice as 'Cheap' (< 10), 'Moderate' (10–20), 'Expensive' (> 20).
2. Show CustomerName and label 'Domestic' if Country = 'USA', else 'International'.
3. Show UnitsInStock and label 'Out of Stock' if 0, else 'In Stock'.
4. Show ProductName and label 'Discontinued' if Discontinued = 1, else 'Available'.
5. Show Employee FirstName and label 'Senior' if BirthDate < '1960-01-01', else 'Junior'.
6. Show OrderID and label 'Shipped by Speedy Express' if ShipperID = 1, else 'Other Shipper'.
7. Show Quantity and label 'High Quantity' if >= 50, else 'Normal Quantity'.
8. Show UnitPrice and label 'Low' if < 15, 'Medium' if 15–25, 'High' if > 25.
9. Show CustomerName and label 'European' if Country IN ('Germany', 'France', 'UK', 'Sweden'), else 'Non-European'.
10. Show UnitsOnOrder and label 'Pending Orders' if > 0, else 'No Pending Orders'.
11. Show CategoryName and label 'Beverage Category' if CategoryID = 1, else 'Other Category'.
12. Show City and label 'Capital City' if City IN ('Berlin', 'London', 'Tokyo'), else 'Other City'.
13. Show ProductName and label 'Special Offer' if UnitsInStock < 20, else 'Regular'.
14. Show EmployeeID and label 'Experienced' if EmployeeID <= 5, else 'New Employee'.
15. Show SupplierName and label 'Local Supplier' if Country = 'USA', else 'International Supplier'.
16. Show CustomerName and label 'VIP' if CustomerID <= 10, else 'Regular'.
17. Show OrderDate and label 'Old Order' if < '1996-07-10', else 'Recent Order'.
18. Show ShipperName and label 'Fast Delivery' if ShipperID = 1, else 'Standard Delivery'.
19. Show UnitPrice and label 'Discount Eligible' if > 20, else 'No Discount'.
20. Show QuantityPerUnit and label 'Bulk Pack' if QuantityPerUnit LIKE '%boxes%', else 'Single Pack'.
21. Show CustomerName and label 'High PostalCode' if PostalCode >= '5000', else 'Low PostalCode'.
22. Show UnitsInStock and label 'Low Stock' if UnitsInStock < 10, 'Medium Stock' if 10–50, 'High Stock' if > 50.
23. Show ProductName and label 'Top Supplier' if SupplierID IN (1, 2, 3), else 'Other Supplier'.
24. Show Discontinued and label 'Yes' if 1, else 'No'.
25. Show OrderID and label 'Weekend Order' if DAYOFWEEK(OrderDate) IN (1, 7), else 'Weekday Order'.
26. Show CustomerName and label 'Short Name' if LENGTH(CustomerName) <= 10, else 'Long Name'.

27. Show ProductName and label 'Reorder Needed' if UnitsInStock <= ReorderLevel, else 'Stock OK'.
28. Show EmployeeID and label 'Eligible for Bonus' if EmployeeID IN (1,3,5,7), else 'Not Eligible'.
29. Show CategoryName and label 'Popular Category' if CategoryID IN (1,2,3), else 'Other Category'.
30. Show UnitPrice and label 'High Value Product' if UnitPrice > 25, 'Medium Value Product' if 15–25, 'Low Value Product' if < 15.

REMOVING DUPLICATES → DISTINCT

1. List all distinct countries from the customers table.
2. List all distinct cities where suppliers are located.
3. List all distinct category names from categories.
4. List all distinct product names from products.
5. List all distinct employee last names.
6. List all distinct shippers from the shippers table.
7. List all distinct customer postal codes.
8. List all distinct suppliers' phone numbers.
9. List all distinct countries from suppliers.
10. List all distinct unit prices from products.
11. List all distinct quantities per unit from products.
12. List all distinct cities of customers.
13. List all distinct categories assigned to products.
14. List all distinct orders' shipper IDs.
15. List all distinct employee first names.
16. List all distinct contact names from customers.
17. List all distinct product suppliers (SupplierID).
18. List all distinct orders' customer IDs.
19. List all distinct units in stock from products.
20. List all distinct units on order from products.
21. List all distinct reorder levels from products.
22. List all distinct discontinued statuses from products.
23. List all distinct addresses of customers.
24. List all distinct order dates from orders.
25. List all distinct birth dates of employees.
26. List all distinct cities where shippers are based.
27. List all distinct supplier cities.
28. List all distinct product categories used in order_details.
29. List all distinct quantity values from order_details.
30. List all distinct combinations of CustomerName and City.

DATA TRANSFORMATION

1. Convert OrderID from INT to CHAR.
2. Convert UnitPrice from DECIMAL to INT.
3. Cast BirthDate of employees to CHAR.
4. Concatenate CustomerName and City into a single column.
5. Concatenate ProductName and CategoryName.
6. Convert PostalCode to INT.
7. Convert UnitsInStock + UnitsOnOrder to DECIMAL.
8. Replace NULL Phone in suppliers with 'No Phone'.
9. Replace NULL ContactName in customers with 'Unknown'.
10. Convert UnitPrice to string and add ' USD' suffix.
11. Use CASE WHEN to categorize products as 'Cheap', 'Moderate', 'Expensive'.
12. Convert OrderDate to YYYY-MM-DD format string.
13. Uppercase all ProductName values.
14. Lowercase all CustomerName values.
15. Trim leading and trailing spaces from Address.
16. Replace 'Street' with 'St.' in customer addresses.
17. Format UnitPrice to 2 decimal places.
18. Calculate TotalValue as UnitPrice * Quantity in order_details.
19. Add 10% tax to UnitPrice for products.
20. Transform Discontinued column to display 'Yes' or 'No'.
21. Combine FirstName and LastName of employees into FullName.
22. Replace NULL DescriptionText with 'No Description'.
23. Transform UnitsInStock to show 'Low Stock', 'Medium Stock', 'High Stock' using CASE WHEN.
24. Convert QuantityPerUnit text to uppercase.
25. Calculate remaining stock as UnitsInStock - UnitsOnOrder.
26. Convert OrderID to a string and prefix with 'ORD-'.
27. Format OrderDate to show month and year only.
28. Transform CustomerName to show first 5 characters only.
29. Replace NULL City with 'Unknown City'.

Views

CREATE VIEW

1. Create a view showing all customers from Germany.
2. Create a view showing products that are out of stock (UnitsInStock = 0).
3. Create a view showing discontinued products.
4. Create a view showing all employees born before 1960.
5. Create a view showing orders shipped by Federal Shipping.
6. Create a view showing high-priced products (UnitPrice > 20).
7. Create a view showing customers from Mexico.
8. Create a view showing products in the Beverages category.
9. Create a view showing suppliers in Japan.
10. Create a view showing orders placed before '1996-07-10'.
11. Create a view showing orders with quantity > 20 in order_details.
12. Create a view showing total units in stock per category.
13. Create a view showing products along with their supplier names.
14. Create a view showing orders along with customer names.
15. Create a view showing orders along with employee names.
16. Create a view showing orders along with shipper names.
17. Create a view showing order details with product names.
18. Create a view showing customers and their order count.
19. Create a view showing products that need to be reordered (UnitsInStock <= ReorderLevel).
20. Create a view showing suppliers along with the number of products supplied.
21. Create a view showing employees along with the number of orders handled.
22. Create a view showing products grouped by category with average price.
23. Create a view showing customers and their country.
24. Create a view showing all categories and their product count.
25. Create a view showing orders along with total quantity per order.
26. Create a view showing customers who have not placed any orders.
27. Create a view showing products with stock value (UnitPrice * UnitsInStock).
28. Create a view showing top 5 most expensive products.
29. Create a view showing orders placed by a specific employee (e.g., EmployeeID = 4).
30. Create a view showing suppliers who supply products in category 'Condiments'.
31. Create a view showing customers along with their city and country.
32. Create a view showing orders along with product names and quantities.
33. Create a view showing products with UnitPrice rounded to 2 decimals.
34. Create a view showing employees with age greater than 50.
35. Create a view showing orders with total number of products per order.
36. Create a view showing products in stock with low units (<10).
37. Create a view showing customers from Germany or France.

38. Create a view showing shippers and the number of orders handled by each.
39. Create a view showing products and suppliers where products are discontinued.
40. Create a view showing orders and the corresponding total price per order.
41. Create a view showing products whose name contains 'Chai'.
42. Create a view showing orders with quantity per product greater than 30.
43. Create a view showing employees who handled more than 5 orders.
44. Create a view showing categories and the average price of products in each.
45. Create a view showing customers along with the number of orders in July 1996.
46. Create a view showing suppliers who supply more than 3 products.
47. Create a view showing products and category names where UnitPrice > 15.
48. Create a view showing orders along with customer name, employee name, and shipper name.
49. Create a view showing products grouped by supplier with total units in stock.
50. Create a view showing orders along with product names, quantity, and total price.

UPDATE VIEW

1. Update a view to increase UnitPrice by 10% for all products.
2. Update a view to set UnitsInStock = 0 for discontinued products.
3. Update a view to set ReorderLevel = 20 where UnitsInStock < 10.
4. Update a view to mark Discontinued = 1 for products with UnitPrice = 0.
5. Update a view to change QuantityPerUnit to '24 bottles' for product 'Chang'.
6. Update a view to increase UnitsOnOrder by 5 for a specific product.
7. Update a view to set UnitPrice = 15 for products in category 'Beverages'.
8. Update a view to set UnitsInStock = UnitsInStock + 10 for all products.
9. Update a view to set Discontinued = 0 for product 'Chef Anton''s Gumbo Mix'.
10. Update a view to round UnitPrice to nearest integer for all products.
11. Customers Table / View Updates
12. Update a view to change Country to 'USA' for customers in Mexico.
13. Update a view to set PostalCode = '00000' for missing postal codes.
14. Update a view to append ' - VIP' to CustomerName of top 10 customers.
15. Update a view to replace Address with 'Not Provided' if NULL.
16. Update a view to set City = 'Unknown' for customers with missing city.
17. Update a view to update ContactName to 'TBD' where blank.
18. Update a view to change Country = 'Germany' for all customers in Berlin.
19. Update a view to prepend 'Mr./Ms.' to ContactName.
20. Update a view to capitalize all CustomerName.
21. Update a view to set PostalCode to numeric only using CAST/CONVERT.
22. Employees Table / View Updates
23. Update a view to set LastName = 'Doe' for EmployeeID = 1.

24. Update a view to change FirstName to 'Updated' where LastName = 'Davolio'.
25. Update a view to adjust BirthDate by adding 1 year.
26. Update a view to update FirstName to uppercase.
27. Update a view to update LastName to lowercase.
28. Update a view to add 5 days to BirthDate for all employees.
29. Update a view to change FirstName to 'Senior' for employees born before 1960.
30. Update a view to set LastName = 'Smith' where FirstName = 'Andrew'.
31. Update a view to swap FirstName and LastName in a view.
32. Update a view to truncate BirthDate to year only.
33. Suppliers Table / View Updates
34. Update a view to change Country to 'UK' for suppliers in London.
35. Update a view to set Phone = '(000) 000-0000' for missing numbers.
36. Update a view to replace 'St.' with 'Street' in Address.
37. Update a view to capitalize SupplierName.
38. Update a view to append ' (Preferred)' to SupplierName.
39. Update a view to set City = 'Unknown' where NULL.
40. Update a view to change ContactName = 'TBD' if NULL.
41. Update a view to update PostalCode to '99999' for missing codes.
42. Update a view to prepend 'Supplier-' to SupplierName.
43. Update a view to set Address = CONCAT(Address, ', Updated').
44. Orders / Order_Details Table / View Updates
45. Update a view to change ShipperID = 1 for all orders in July 1996.
46. Update a view to set OrderDate = '1996-08-01' for OrderID = 10250.
47. Update a view to increase Quantity by 5 in order_details for ProductID = 11.
48. Update a view to decrease Quantity by 2 in order_details for orders by CustomerID = 34.
49. Update a view to set EmployeeID = 3 for all orders in July 1996.
50. Update a view to change ProductID = 5 for all OrderDetailID > 50.
51. Update a view to increase UnitsOnOrder by 10 via order_details view.
52. Update a view to mark Discontinued = 1 for products in orders older than '1996-07-10'.
53. Update a view to change ShipperID = 2 for orders shipped by Federal Shipping.
54. Update a view to update Quantity to 1 where Quantity is NULL in order_details.

DROP VIEW

1. Drop a view showing all customers from Germany.
2. Drop a view showing employees born before 1960.
3. Drop a view showing suppliers in Japan.
4. Drop a view showing orders before '1996-07-10'.

5. Drop a view showing products along with supplier names.
6. Drop a view showing orders along with customer names.
7. Drop a view showing orders along with employee names.
8. Drop a view showing orders along with shipper names.
9. Drop a view showing customers and their order count.
10. Drop a view showing products grouped by category with average price.

MATERIALIZED VIEW (CONCEPT)

1. Explain what a materialized view is and how it differs from a normal view.
2. Create a materialized view showing total quantity of products per category.
3. Create a materialized view showing total sales per customer.
4. Create a materialized view showing the average UnitPrice per supplier.
5. Explain when it is better to use a materialized view instead of a normal view.
6. Create a materialized view showing the top 10 most expensive products.
7. Create a materialized view showing employees along with total orders handled.
8. Explain how to refresh a materialized view after the products table is updated.
9. Create a materialized view showing customers along with the total number of orders.
10. Create a materialized view showing total stock value per category (UnitPrice * UnitsInStock).
11. Explain the difference between FAST REFRESH and COMPLETE REFRESH in materialized views.
12. Create a materialized view showing orders along with product details and total quantity.
13. Create a materialized view showing discontinued products grouped by category.
14. Explain how a materialized view can improve performance in reporting dashboards.
15. Create a materialized view showing suppliers who supply more than 5 products.
16. Create a materialized view showing monthly order totals per customer.
17. Explain how indexing a materialized view can improve query performance.
18. Create a materialized view showing orders with high quantity products (Quantity > 30).
19. Explain why materialized views may require more storage space than normal views.
20. Create a materialized view showing top 5 customers with the most orders.
21. Explain how to schedule automatic refresh of a materialized view.
22. Create a materialized view showing average order quantity per employee.
23. Explain what happens if the underlying products table changes but the materialized view is not refreshed.

24. Create a materialized view showing suppliers along with total units supplied.
25. Create a materialized view showing orders shipped by each shipper along with count.
26. Explain the use of WITH DATA and WITH NO DATA in creating materialized views.
27. Create a materialized view showing customers who have placed orders worth more than 1000 USD.
28. Explain the difference between read-only and updatable materialized views.
29. Create a materialized view showing product categories along with average UnitsInStock.
30. Explain how partitioning a materialized view can help with performance.
31. Create a materialized view showing total orders per month along with order details.
32. Explain how to manually refresh a materialized view in Oracle or PostgreSQL.
33. Create a materialized view showing orders along with customer country and order count.
34. Explain why materialized views are useful in analytics and reporting.
35. Create a materialized view showing products with stock below reorder level.
36. Explain how materialized views can be used in Power BI or Tableau to improve query speed.
37. Create a materialized view showing total sales per product category.
38. Explain the potential drawbacks of using materialized views in OLTP systems.
39. Create a materialized view showing top suppliers by number of products supplied.
40. Explain how refresh frequency affects data accuracy in a materialized view.
41. Create a materialized view showing average UnitPrice by category.
42. Explain how a materialized view can help reduce CPU usage for complex joins.
43. Create a materialized view showing employees who handled more than 10 orders.
44. Explain the difference between a materialized view and a summary table.
45. Create a materialized view showing customers who have not placed any orders.
46. Explain the use of ON COMMIT refresh in materialized views.
47. Create a materialized view showing monthly revenue per customer.
48. Explain the use of query rewrite feature with materialized views.
49. Create a materialized view showing products grouped by supplier and category.
50. Explain how materialized views can be combined with indexes for faster queries.

Indexes

CREATE INDEX

1. Create an index on CustomerName in the customers table.
2. Create an index on City in the customers table.
3. Create an index on PostalCode in the customers table.
4. Create an index on Country in the customers table.
5. Create an index on ProductName in the products table.
6. Create an index on UnitPrice in the products table.
7. Create an index on CategoryID in the products table.
8. Create an index on SupplierID in the products table.
9. Create an index on UnitsInStock in the products table.
10. Create an index on UnitsOnOrder in the products table.
11. Create an index on ReorderLevel in the products table.
12. Create an index on Discontinued in the products table.
13. Create an index on OrderDate in the orders table.
14. Create an index on CustomerID in the orders table.
15. Create an index on EmployeeID in the orders table.
16. Create an index on ShipperID in the orders table.
17. Create an index on OrderID in the order_details table.
18. Create an index on ProductID in the order_details table.
19. Create an index on Quantity in the order_details table.
20. Create an index on LastName in the employees table.
21. Create an index on FirstName in the employees table.
22. Create an index on BirthDate in the employees table.
23. Create an index on ShipperName in the shippers table.
24. Create an index on Phone in the shippers table.
25. Create an index on SupplierName in the suppliers table.
26. Create an index on City in the suppliers table.
27. Create an index on Country in the suppliers table.
28. Create an index on ContactName in the suppliers table.
29. Create an index on Address in the suppliers table.
30. Create an index on PostalCode in the suppliers table.
31. Create an index on CategoryName in the categories table.
32. Create an index on DescriptionText in the categories table.
33. Create an index on CustomerName and City in customers (for single-column queries).
34. Create an index on UnitPrice in products for faster range queries.
35. Create an index on OrderDate in orders for monthly reports.
36. Create an index on Quantity in order_details to optimize SUM queries.
37. Create an index on Country in customers to improve filter queries.
38. Create an index on Discontinued in products for filtering discontinued items.
39. Create an index on SupplierName in suppliers for searching suppliers.

40. Create an index on BirthDate in employees to improve age-related queries.
41. Create an index on CategoryID in products to optimize JOIN with categories.
42. Create an index on CustomerID in orders for faster JOIN with customers.
43. Create an index on EmployeeID in orders for faster JOIN with employees.
44. Create an index on ShipperID in orders for faster JOIN with shippers.
45. Create an index on OrderID in order_details for faster JOIN with orders.
46. Create an index on ProductID in order_details for faster JOIN with products.
47. Create an index on UnitsInStock for low-stock reporting.
48. Create an index on UnitPrice for price range queries.
49. Create an index on City in customers for regional reporting.
50. Create an index on Country in suppliers for supplier location filtering.

Composite Index

1. Create a composite index on CustomerName, City in customers.
2. Create a composite index on Country, City in customers.
3. Create a composite index on CategoryID, SupplierID in products.
4. Create a composite index on UnitsInStock, UnitsOnOrder in products.
5. Create a composite index on UnitPrice, UnitsInStock in products.
6. Create a composite index on OrderDate, CustomerID in orders.
7. Create a composite index on OrderDate, EmployeeID in orders.
8. Create a composite index on CustomerID, EmployeeID in orders.
9. Create a composite index on OrderID, ProductID in order_details.
10. Create a composite index on Quantity, ProductID in order_details.
11. Create a composite index on LastName, FirstName in employees.
12. Create a composite index on City, Country in customers.
13. Create a composite index on SupplierName, City in suppliers.
14. Create a composite index on PostalCode, City in customers.
15. Create a composite index on CategoryName, DescriptionText in categories.
16. Create a composite index on Discontinued, UnitsInStock in products.
17. Create a composite index on OrderDate, ShipperID in orders.
18. Create a composite index on CustomerID, OrderDate in orders.
19. Create a composite index on UnitPrice, ReorderLevel in products.
20. Create a composite index on SupplierID, ProductID in products.
21. Create a composite index on City, ContactName in customers.
22. Create a composite index on Country, CustomerName in customers.
23. Create a composite index on BirthDate, LastName in employees.
24. Create a composite index on Quantity, OrderID in order_details.
25. Create a composite index on UnitsInStock, ReorderLevel in products.
26. Create a composite index on CategoryID, UnitPrice in products.
27. Create a composite index on ShipperName, Phone in shippers.
28. Create a composite index on SupplierName, Country in suppliers.
29. Create a composite index on CustomerID, City in customers.
30. Create a composite index on EmployeeID, LastName in employees.

31. Create a composite index on OrderDate, CustomerID, EmployeeID in orders.
32. Create a composite index on ProductID, CategoryID, SupplierID in products.
33. Create a composite index on UnitsOnOrder, UnitsInStock, ReorderLevel in products.
34. Create a composite index on CustomerID, Country, City in customers.
35. Create a composite index on OrderID, ProductID, Quantity in order_details.
36. Create a composite index on BirthDate, FirstName, LastName in employees.
37. Create a composite index on City, PostalCode, Country in customers.
38. Create a composite index on UnitPrice, UnitsInStock, Discontinued in products.
39. Create a composite index on SupplierID, Country, City in suppliers.
40. Create a composite index on ShipperID, ShipperName in shippers.
41. Create a composite index on CustomerID, OrderDate, ShipperID in orders.
42. Create a composite index on CategoryID, ProductName in products.
43. Create a composite index on City, CustomerName, Country in customers.
44. Create a composite index on OrderID, EmployeeID, CustomerID in orders.
45. Create a composite index on UnitsInStock, UnitsOnOrder, UnitPrice in products.
46. Create a composite index on CustomerID, ContactName, City in customers.
47. Create a composite index on OrderDate, Quantity, ProductID in order_details.
48. Create a composite index on CategoryID, SupplierID, UnitPrice in products.
49. Create a composite index on SupplierID, ProductID, UnitsInStock in products.
50. Create a composite index on CustomerID, Country, PostalCode in customers.

Clustered/Nonclustered indexes

1. Create a clustered index on CustomerID in customers.
2. Create a clustered index on ProductID in products.
3. Create a clustered index on OrderID in orders.
4. Create a clustered index on EmployeeID in employees.
5. Create a clustered index on ShipperID in shippers.
6. Create a clustered index on SupplierID in suppliers.
7. Create a clustered index on CategoryID in categories.
8. Create a nonclustered index on CustomerName in customers.
9. Create a nonclustered index on City in customers.
10. Create a nonclustered index on Country in customers.
11. Create a nonclustered index on UnitPrice in products.
12. Create a nonclustered index on UnitsInStock in products.
13. Create a nonclustered index on Discontinued in products.
14. Create a nonclustered index on OrderDate in orders.
15. Create a nonclustered index on ShipperID in orders.
16. Create a nonclustered index on Quantity in order_details.

17. Create a nonclustered index on ProductID in order_details.
18. Create a clustered index on OrderDetailID in order_details.
19. Create a clustered index on BirthDate in employees.
20. Create a nonclustered index on LastName in employees.
21. Create a nonclustered index on FirstName in employees.
22. Create a clustered index on SupplierID in suppliers.
23. Create a nonclustered index on SupplierName in suppliers.
24. Create a nonclustered index on City in suppliers.
25. Create a clustered index on CategoryID in products.
26. Create a nonclustered index on CategoryName in categories.
27. Create a nonclustered index on DescriptionText in categories.
28. Create a clustered index on CustomerID, OrderID in orders.
29. Create a nonclustered index on UnitPrice, UnitsInStock in products.
30. Create a clustered index on OrderDate, CustomerID in orders.
31. Create a nonclustered index on CustomerName, City in customers.
32. Create a clustered index on OrderDetailID, ProductID in order_details.
33. Create a nonclustered index on SupplierName, City in suppliers.
34. Create a clustered index on ProductID, CategoryID in products.
35. Create a nonclustered index on CustomerID, Country in customers.
36. Create a clustered index on EmployeeID, LastName in employees.
37. Create a nonclustered index on City, Country in customers.
38. Create a clustered index on OrderID, OrderDate in orders.
39. Create a nonclustered index on ProductName, SupplierID in products.
40. Create a clustered index on SupplierID, ProductID in products.
41. Create a nonclustered index on UnitsOnOrder, UnitsInStock in products.
42. Create a clustered index on OrderDate, ShipperID in orders.
43. Create a nonclustered index on CategoryID, SupplierID in products.
44. Create a clustered index on CustomerID, OrderDate, EmployeeID in orders.
45. Create a nonclustered index on UnitPrice, Discontinued in products.
46. Create a clustered index on ProductID, UnitsInStock in products.
47. Create a nonclustered index on PostalCode, City in customers.
48. Create a clustered index on SupplierID, Country in suppliers.
49. Create a nonclustered index on CustomerName, PostalCode in customers.
50. Create a clustered index on OrderID, CustomerID, EmployeeID in orders.

Temporary & Derived Tables

TEMP tables

1. Create a temporary table to store products with UnitsInStock < 10.
2. Create a temporary table for customers from Germany.
3. Create a temporary table showing orders placed in July 1996.
4. Create a temporary table showing employees born before 1960.
5. Create a temporary table for suppliers in Japan.
6. Create a temporary table for discontinued products.
7. Create a temporary table for orders shipped by “Speedy Express”.
8. Create a temporary table for products with UnitPrice > 20.
9. Create a temporary table for customers in Mexico or USA.
10. Create a temporary table for orders with total quantity > 50.
11. Create a temporary table for top 5 expensive products.
12. Create a temporary table for suppliers who supply more than 3 products.
13. Create a temporary table for products grouped by category.
14. Create a temporary table for employees and total orders handled.
15. Create a temporary table for customers who have placed no orders.
16. Create a temporary table for order details with Quantity > 20.
17. Create a temporary table for products with UnitsOnOrder != 0.
18. Create a temporary table for orders along with customer names.
19. Create a temporary table for orders along with employee names.
20. Create a temporary table for monthly order count per customer.
21. Create a temporary table for suppliers along with total products supplied.
22. Create a temporary table for categories with total products per category.
23. Create a temporary table for products in “Beverages” and “Condiments” categories.
24. Create a temporary table for orders with ShipperID = 3.
25. Create a temporary table for customers whose postal code starts with ‘5’.
26. Create a temporary table for products with stock value (UnitPrice * UnitsInStock).
27. Create a temporary table for orders with quantity per product > 10.
28. Create a temporary table for employees with first names starting with “A”.
29. Create a temporary table for suppliers in UK and France.
30. Create a temporary table for customers along with number of orders placed.

Common Table Expressions (CTE)

1. Create a CTE showing products with UnitPrice > 20.
2. Create a CTE for customers in Germany.
3. Create a CTE showing orders in July 1996.

4. Create a CTE for employees born before 1960.
5. Create a CTE showing suppliers in Japan.
6. Create a CTE for discontinued products.
7. Create a CTE for orders shipped by “Federal Shipping”.
8. Create a CTE for products in “Beverages” category.
9. Create a CTE showing customers with more than 3 orders.
10. Create a CTE for total units in stock per category.
11. Create a CTE for products along with supplier names.
12. Create a CTE for order details with quantity > 20.
13. Create a CTE for employees and total orders handled.
14. Create a CTE for customers who have placed no orders.
15. Create a CTE showing products grouped by category with average UnitPrice.
16. Create a CTE showing orders along with customer names.
17. Create a CTE showing orders along with employee names.
18. Create a CTE showing products with UnitsInStock < ReorderLevel.
19. Create a CTE showing top 5 customers with most orders.
20. Create a CTE showing suppliers who supply more than 3 products.
21. Create a CTE showing products and suppliers where products are discontinued.
22. Create a CTE showing total orders per customer.
23. Create a CTE showing products along with category names.
24. Create a CTE showing orders along with product names and quantities.
25. Create a CTE showing customers along with country.
26. Create a CTE showing employees who handled more than 5 orders.
27. Create a CTE for average UnitPrice per category.
28. Create a CTE showing customers with orders in July 1996.
29. Create a CTE showing suppliers along with total units supplied.
30. Create a CTE showing products with stock value (UnitPrice * UnitsInStock).

Recursive CTE

1. Use a recursive CTE to generate a sequence of numbers from 1 to 100.
2. Use a recursive CTE to generate the last 12 months for monthly reports.
3. Use a recursive CTE to calculate cumulative stock value per product category.
4. Use a recursive CTE to calculate running total of orders per customer.
5. Use a recursive CTE to generate all order dates between '1996-07-01' and '1996-07-31'.
6. Use a recursive CTE to calculate cumulative quantity ordered per product.
7. Use a recursive CTE to generate hierarchy levels of employees (if a ManagerID column existed).
8. Use a recursive CTE to calculate cumulative sales per month.
9. Use a recursive CTE to generate a list of 10 consecutive ProductIDs.
10. Use a recursive CTE to calculate running total of UnitPrice per category.
11. Use a recursive CTE to generate the next 5 business days from a given date.

12. Use a recursive CTE to show cumulative orders per customer per month.
13. Use a recursive CTE to calculate total stock for top 10 products recursively.
14. Use a recursive CTE to generate all multiples of 5 up to 100.
15. Use a recursive CTE to calculate cumulative number of orders per employee.
16. Use a recursive CTE to generate a sequence of CustomerIDs from 1 to 91.
17. Use a recursive CTE to calculate running total of UnitsInStock per category.
18. Use a recursive CTE to list all order dates recursively for each customer.
19. Use a recursive CTE to calculate cumulative revenue per product category.
20. Use a recursive CTE to generate a list of all months in 1996 for reporting.

Stored Procedures

CREATE PROCEDURE

1. Create a procedure to fetch all products from the products table.
2. Create a procedure to fetch all customers from the customers table.
3. Create a procedure to fetch all employees from the employees table.
4. Create a procedure to fetch all orders from the orders table.
5. Create a procedure to fetch all suppliers from the suppliers table.
6. Create a procedure to fetch all categories from the categories table.
7. Create a procedure to fetch all products in a specific category (use a parameter CategoryID).
8. Create a procedure to fetch all orders for a specific customer (parameter CustomerID).
9. Create a procedure to fetch all orders handled by a specific employee (parameter EmployeeID).
10. Create a procedure to fetch all products from a specific supplier (parameter SupplierID).
11. Create a procedure to fetch all products where UnitsInStock < 10.
12. Create a procedure to fetch all products where UnitPrice > 20.
13. Create a procedure to fetch all discontinued products.
14. Create a procedure to fetch total orders per customer.
15. Create a procedure to fetch total quantity sold per product.
16. Create a procedure to fetch orders placed within a specific date range (parameters StartDate and EndDate).
17. Create a procedure to fetch the top 5 most expensive products.
18. Create a procedure to fetch products along with their supplier names (JOIN with suppliers).
19. Create a procedure to fetch orders along with customer names (JOIN with customers).
20. Create a procedure to fetch orders along with employee names (JOIN with employees).
21. Create a procedure to insert a new product into the products table (parameters for all columns except ProductID).
22. Create a procedure to insert a new customer into the customers table.
23. Create a procedure to insert a new order into the orders table.
24. Create a procedure to insert a new supplier into the suppliers table.
25. Create a procedure to update UnitsInStock for a specific product (parameters ProductID and NewStock).
26. Create a procedure to update UnitPrice for a specific product.
27. Create a procedure to update ContactName for a specific customer.
28. Create a procedure to delete a product by ProductID.
29. Create a procedure to delete a customer by CustomerID.
30. Create a procedure to delete an order by OrderID.

31. Create a procedure to fetch customers from a specific country (parameter Country).
32. Create a procedure to fetch products with stock value (UnitPrice * UnitsInStock) greater than a given amount (parameter MinValue).
33. Create a procedure to fetch suppliers who supply more than N products (parameter MinProducts).
34. Create a procedure to fetch orders handled by employees with a specific last name (parameter LastName).
35. Create a procedure to fetch products grouped by category with average UnitPrice.
36. Create a procedure to fetch total units ordered for a specific product.
37. Create a procedure to fetch customers with no orders.
38. Create a procedure to fetch products by partial name match (parameter ProductNamePattern using LIKE).
39. Create a procedure to fetch the top N customers by total orders (parameter TopN).
40. Create a procedure to fetch orders for a specific shipper (parameter ShipperID).
41. Create a procedure to fetch the total stock value per category.
42. Create a procedure to fetch orders with quantity greater than a specific value (parameter MinQuantity).
43. Create a procedure to fetch employees born before a specific date (parameter BirthDate).
44. Create a procedure to fetch orders and their details for a specific customer.
45. Create a procedure to fetch suppliers along with total products supplied.
46. Create a procedure to fetch customers by city and country (parameters City and Country).
47. Create a procedure to fetch products with UnitsOnOrder != 0.
48. Create a procedure to fetch products by multiple categories (parameterized list of CategoryID).
49. Create a procedure to fetch all products with quantity per unit containing a specific keyword (parameter Keyword).
50. Create a procedure to fetch total orders per month for a given year (parameter Year).

Parameters

Simple Scalar Functions (20 Questions)

1. Create a function to return the total stock value of a product (UnitPrice * UnitsInStock) given ProductID.
2. Create a function to return the total orders placed by a customer (CustomerID).
3. Create a function to return the total quantity of a product ordered (ProductID).

4. Create a function to return the remaining stock after orders for a product.
5. Create a function to return the average unit price of products in a category (CategoryID).
6. Create a function to return the maximum unit price in a category.
7. Create a function to return the minimum unit price in a category.
8. Create a function to return total discontinued products in a category.
9. Create a function to return the total number of suppliers in a country.
10. Create a function to return the number of customers in a city.
11. Create a function to return total products supplied by a supplier (SupplierID).
12. Create a function to return total units on order for a product.
13. Create a function to return the stock status (IN STOCK/OUT OF STOCK) for a product.
14. Create a function to return the total quantity of orders handled by an employee (EmployeeID).
15. Create a function to return the number of orders shipped by a specific shipper (ShipperID).
16. Create a function to return total stock value in a category.
17. Create a function to return the total orders in a date range (StartDate, EndDate).
18. Create a function to return the full name of an employee (concatenate FirstName + LastName) given EmployeeID.
19. Create a function to return the country of a customer given CustomerID.
20. Create a function to return total products in stock for a supplier.

String Functions with Parameters (10 Questions)

21. Create a function to return the first N characters of a product name (ProductID, N).
22. Create a function to return the last N characters of a product name.
23. Create a function to convert the product name to UPPERCASE given ProductID.
24. Create a function to convert the product name to lowercase.
25. Create a function to return the length of a product name.
26. Create a function to return a product name with a keyword replaced (parameters ProductID, OldText, NewText).
27. Create a function to return a concatenated string of product name and category name.
28. Create a function to return the first N characters of customer's city.
29. Create a function to return the last N characters of employee's last name.
30. Create a function to return the country and city concatenated for a customer.

Aggregation & Calculation Functions (10 Questions)

31. Create a function to return total quantity ordered for a product category.
32. Create a function to return total revenue generated by a product (UnitPrice * Quantity).
33. Create a function to return the average order quantity for a product.
34. Create a function to return total quantity of products ordered by a customer.

35. Create a function to return total stock value per supplier.
36. Create a function to return the highest quantity ordered in an order.
37. Create a function to return the lowest quantity ordered in an order.
38. Create a function to return total quantity of products on order across all products.
39. Create a function to return total stock value for discontinued products.
40. Create a function to return average UnitsInStock for a category.

Conditional & Parameterized Functions (10 Questions)

41. Create a function to return 'High Stock' if UnitsInStock > 50 else 'Low Stock' for a product.
42. Create a function to return 'Expensive' if UnitPrice > 20 else 'Affordable'.
43. Create a function to return 1 if a customer has placed any orders, else 0.
44. Create a function to return 1 if a product is discontinued, else 0.
45. Create a function to return 'Available' if UnitsInStock > ReorderLevel, else 'Reorder Needed'.
46. Create a function to return 'Active Supplier' if supplier supplies more than 5 products.
47. Create a function to return a category name given CategoryID.
48. Create a function to return the shipper name given ShipperID.
49. Create a function to return the total orders in a given month and year (parameters Month, Year).
50. Create a function to return the number of products whose unit price falls between two values (MinPrice, MaxPrice).

IF…ELSE logic

Conditional SELECT / Queries in Procedures (20 Questions)

1. Create a procedure to check stock of a product (ProductID) and return 'High Stock' if UnitsInStock > 50, else 'Low Stock'.
2. Create a procedure to check if a product is discontinued and return 'Discontinued' or 'Available'.
3. Create a procedure to classify a product as 'Expensive' if UnitPrice > 20, else 'Affordable'.
4. Create a procedure to check supplier's total products and return 'Top Supplier' if > 5, else 'Normal Supplier'.
5. Create a procedure to classify a customer as 'Active' if they have > 3 orders, else 'New'.
6. Create a procedure to classify a category as 'Popular' if it has > 5 products.
7. Create a procedure to classify an order as 'Complete' if total quantity = 0, else 'Pending'.
8. Create a procedure to classify an employee as 'Veteran' if BirthDate < '1960-01-01', else 'Junior'.
9. Create a procedure to classify a shipper as 'Busy' if they shipped > 10 orders.

10. Create a procedure to check if UnitsOnOrder > 50 and return 'High Demand' or 'Normal Demand'.
11. Create a procedure to classify a customer as 'VIP' if total orders > 10, else 'Regular'.
12. Create a procedure to classify stock value as 'Low Revenue' if < 500, else 'High Revenue'.
13. Create a procedure to check if a product needs restocking (UnitsInStock < ReorderLevel).
14. Create a procedure to classify an order as 'Big Order' if Quantity > 20.
15. Create a procedure to classify a product as 'New' if ProductID > 20.
16. Create a procedure to classify a supplier as 'Prime' if Country = 'USA'.
17. Create a procedure to classify a customer as 'Local' if Country = 'Germany', else 'International'.
18. Create a procedure to classify a product as 'High Price' if UnitPrice > 50.
19. Create a procedure to classify an employee as 'Few Orders' if total handled orders < 5.
20. Create a procedure to classify stock as 'Full Stock' if UnitsInStock >= 100, else 'Partial Stock'.

Conditional INSERT / UPDATE in Procedures (20 Questions)

21. Create a procedure to insert a product only if UnitsInStock > 0, else print a message.
22. Create a procedure to update UnitPrice to 20 if UnitPrice < 20.
23. Create a procedure to update UnitsInStock to 50 if it is less than 50.
24. Create a procedure to insert a customer only if Country = 'Germany'.
25. Create a procedure to insert an order only if CustomerID exists.
26. Create a procedure to update Discontinued to 1 if UnitsInStock = 0.
27. Create a procedure to update ReorderLevel to 10 if UnitsInStock < 10.
28. Create a procedure to insert a supplier only if SupplierName is unique.
29. Create a procedure to update UnitPrice to 0 for discontinued products.
30. Create a procedure to update CustomerName to uppercase if Country = 'USA'.
31. Create a procedure to update UnitsOnOrder to 0 if UnitsInStock > 100.
32. Create a procedure to insert a product only if UnitPrice > 0.
33. Create a procedure to update QuantityPerUnit to ‘Not Specified’ if NULL.
34. Create a procedure to insert a new employee only if BirthDate < '2000-01-01'.
35. Create a procedure to update UnitsInStock to 0 if Discontinued = 1.
36. Create a procedure to update CategoryName to ‘Misc’ if NULL.
37. Create a procedure to insert an order only if ShipperID exists.
38. Create a procedure to update PostalCode to ‘00000’ if NULL.
39. Create a procedure to update UnitsOnOrder to UnitsInStock if NULL.
40. Create a procedure to insert a new product with Discontinued = 0 only if UnitPrice > 10.

Conditional DELETE / Misc in Procedures (10 Questions)

41. Create a procedure to delete a product if UnitsInStock = 0.

42. Create a procedure to delete orders if OrderDate < '1996-07-01'.
43. Create a procedure to delete customers if they have no orders.
44. Create a procedure to delete suppliers if they supply no products.
45. Create a procedure to delete order_details if Quantity = 0.
46. Create a procedure to delete employees who are retired (BirthDate < '1950-01-01').
47. Create a procedure to delete products if Discontinued = 1 and UnitsInStock = 0.
48. Create a procedure to delete orders if total quantity < 5.
49. Create a procedure to delete categories if they have no products.
50. Create a procedure to delete shippers if they have shipped no orders.

Loops (concept)

WHILE Loop (20 Questions)

1. Create a procedure to print ProductName for all products while ProductID <= 10.
2. Create a procedure to update UnitsInStock by adding 5 for products while UnitsInStock < 50.
3. Create a procedure to print all customer names while CustomerID <= 20.
4. Create a procedure to calculate total stock value for products with ProductID <= 15.
5. Create a procedure to print orders while OrderID <= 10260.
6. Create a procedure to mark products as discontinued while UnitsInStock = 0.
7. Create a procedure to increase UnitsOnOrder by 10 while UnitsOnOrder < 50.
8. Create a procedure to print suppliers' names while SupplierID <= 10.
9. Create a procedure to list employees born before 1960 using a WHILE loop.
10. Create a procedure to update ReorderLevel to 10 while UnitsInStock < 10.
11. Create a procedure to calculate total UnitPrice for the first 20 products.
12. Create a procedure to count customers per country while iterating through CustomerID.
13. Create a procedure to print top 5 products in stock using WHILE loop.
14. Create a procedure to set Discontinued = 1 for products with UnitsOnOrder = 0.
15. Create a procedure to print all order details while OrderDetailID <= 50.
16. Create a procedure to update UnitPrice to 20 if less than 20 while looping through products.
17. Create a procedure to print orders for EmployeeID = 3 using WHILE loop.
18. Create a procedure to calculate total orders for each customer while iterating through CustomerID.
19. Create a procedure to print product names in category CategoryID = 1.
20. Create a procedure to print all shippers using WHILE loop.

LOOP / LEAVE / ITERATE (15 Questions)

21. Create a procedure to loop through products and leave the loop if a discontinued product is found.
22. Create a procedure to loop through orders and iterate to next if quantity < 5.
23. Create a procedure to loop through customers and leave if total orders > 10.
24. Create a procedure to loop through employees and mark as veteran if BirthDate < 1960.
25. Create a procedure to loop through suppliers and skip those in Japan.
26. Create a procedure to loop through products and set UnitsInStock = 50 if less than 50.
27. Create a procedure to loop through order_details and calculate total quantity.
28. Create a procedure to loop through categories and print their names.
29. Create a procedure to loop through products and leave when UnitPrice > 50.
30. Create a procedure to loop through orders and skip if ShipperID = 2.
31. Create a procedure to loop through customers and leave if country = 'Germany'.
32. Create a procedure to loop through products and iterate if UnitsOnOrder = 0.
33. Create a procedure to loop through suppliers and print names.
34. Create a procedure to loop through orders and leave if OrderDate < 1996-07-01.
35. Create a procedure to loop through employees and print full name using CONCAT.

REPEAT / UNTIL (15 Questions)

36. Create a procedure to calculate cumulative stock value using REPEAT until ProductID > 20.
37. Create a procedure to increase UnitsOnOrder by 5 for products until UnitsOnOrder >= 50.
38. Create a procedure to print customer names until CustomerID = 30.
39. Create a procedure to sum UnitPrice until UnitsInStock = 0.
40. Create a procedure to update Discontinued = 1 until all products with UnitsInStock = 0 are processed.
41. Create a procedure to print order details until OrderID > 10270.
42. Create a procedure to iterate suppliers until all suppliers in USA are listed.
43. Create a procedure to calculate total orders per customer until last CustomerID.
44. Create a procedure to print employees born before a certain date until all are processed.
45. Create a procedure to increase UnitsInStock by 10 until UnitsInStock >= ReorderLevel.
46. Create a procedure to print products with UnitPrice > 20 until last product.
47. Create a procedure to iterate categories and calculate total products until last CategoryID.
48. Create a procedure to iterate orders and leave REPEAT when quantity > 20.
49. Create a procedure to print all shippers until last ShipperID.
50. Create a procedure to calculate cumulative quantity in order_details until OrderDetailID > 100.