1. **Find the country-wise count of customers.**

SELECT country, COUNT(*) AS customer_count FROM customer GROUP BY country;

2. **Display the products which are not discontinued.**

SELECT * FROM product WHERE IsDiscontinued = 0;

3. **Display the list of companies along with the product name that they are supplying.**

SELECT supplier.CompanyName, product.ProductName FROM supplier JOIN product ON supplier.Id = product.SupplierId;

4. **Display customer information about who stays in 'Mexico'**

SELECT * FROM customer WHERE country = 'Mexico';

5. **Display the costliest item that is ordered by the customer.**

SELECT orderitem.ProductId, product.ProductName, orderitem.UnitPrice FROM orderitem JOIN product ON orderitem.ProductId = product.Id ORDER BY orderitem.UnitPrice DESC LIMIT 1;

6. **Display supplier id who owns the highest number of products.**

SELECT SupplierId, COUNT(*) AS product_count FROM product GROUP BY SupplierId ORDER BY product_count DESC LIMIT 1;

7. **Display month-wise and year-wise counts of the orders placed.**

SELECT YEAR(OrderDate) AS order_year, MONTH(OrderDate) AS order_month, COUNT(*) AS order_count FROM orders GROUP BY order_year, order_month;

8. **Which country has the maximum number of suppliers?**

SELECT country, COUNT(*) AS supplier_count FROM supplier GROUP BY country ORDER BY supplier_count DESC LIMIT 1;

9. **Which customers did not place any orders.**

SELECT * FROM customer WHERE Id NOT IN (SELECT DISTINCT CustomerId FROM orders);

10. **Arrange the product id, product name based on high demand by the customer.**

```
SELECT product.Id, product.ProductName, SUM(orderitem.Quantity) AS total_quantity
FROM product
JOIN orderitem ON product.Id = orderitem.ProductId
GROUP BY product.Id
ORDER BY total_quantity DESC;
```

11. **Display the number of orders delivered every year.**

```
SELECT YEAR(OrderDate) AS year, COUNT(*) AS order_count
FROM orders
GROUP BY year;
```

12. **Calculate year-wise total revenue.**

```
SELECT YEAR(OrderDate) AS year, SUM(TotalAmount) AS total_revenue
FROM orders
GROUP BY year;
```

13. **Display the customer details whose order amount is maximum including his past orders.**

```
SELECT customer.*, orders.TotalAmount
FROM customer
JOIN orders ON customer.Id = orders.CustomerId
ORDER BY orders.TotalAmount DESC
LIMIT 1;
```

14. **Display total amount ordered by each customer from high to low.**

```
SELECT customer.FirstName, customer.LastName, SUM(orders.TotalAmount) AS total_amount
FROM customer
JOIN orders ON customer.Id = orders.CustomerId
GROUP BY customer.Id
ORDER BY total_amount DESC;
```

15. **A sales and marketing department of this company wants to find out how frequently customer have business with them.**

```
WITH CustomerOrderDates AS (
    SELECT CustomerId, OrderDate,
      LAG(OrderDate) OVER (PARTITION BY CustomerId ORDER BY OrderDate) AS PreviousOrderDate
  FROM orders)
SELECT CustomerId,
    COUNT(OrderDate) AS total_orders,
    AVG(DATEDIFF(OrderDate, PreviousOrderDate)) AS avg_days_between_orders,
    MIN(DATEDIFF(OrderDate, PreviousOrderDate)) AS min_days_between_orders,
    MAX(DATEDIFF(OrderDate, PreviousOrderDate)) AS max_days_between_orders
FROM CustomerOrderDates
WHERE PreviousOrderDate IS NOT NULL
GROUP BY CustomerId
ORDER BY avg_days_between_orders;
```

16. **Find out top 3 suppliers in terms of revenue generated by their products.**

```
SELECT supplier.CompanyName, SUM(orderitem.UnitPrice * orderitem.Quantity) AS total_revenue
FROM supplier
JOIN product ON supplier.Id = product.SupplierId
JOIN orderitem ON product.Id = orderitem.ProductId
GROUP BY supplier.Id
ORDER BY total_revenue DESC
LIMIT 3;
```

17. **Display latest order date (should not be same as first order date) of all the customers with customer details.**

```
SELECT customer.*, MAX(orders.OrderDate) AS latest_order
FROM customer
JOIN orders ON customer.Id = orders.CustomerId
WHERE orders.OrderDate != (SELECT MIN(OrderDate)
FROM orders WHERE CustomerId = customer.Id) GROUP BY customer.Id;
```

18. **Display the product name and supplier name for each order**

```
SELECT orders.OrderNumber, product.ProductName, supplier.CompanyName
FROM orders
JOIN orderitem ON orders.Id = orderitem.OrderId
JOIN product ON orderitem.ProductId = product.Id
JOIN supplier ON product.SupplierId = supplier.Id;
```

19. **Fetch the records to display the customer details who ordered more than 10 product quantity in the single order**

```
SELECT customer.*, orderitem.Quantity
FROM customer
JOIN orders ON customer.Id = orders.CustomerId
JOIN orderitem ON orders.Id = orderitem.OrderId
WHERE orderitem.Quantity > 10;
```

20. **Display all the product details with the ordered quantity size as 1.**

```
SELECT product.*, orderitem.Quantity
FROM product
JOIN orderitem ON product.Id = orderitem.ProductId
WHERE orderitem.Quantity = 1;
```

21. **Display the compan(y)ies which supplies products whose cost is above 100.**

```
SELECT supplier.CompanyName, product.ProductName, product.UnitPrice
FROM supplier
JOIN product ON supplier.Id = product.SupplierId
WHERE product.UnitPrice > 100;
```

22. **Company sells the product at different discounted rates. Refer actual product price in product table and selling price in the order item table. Write a query to find out total amount saved in each order then display the orders from highest to lowest amount saved.**

```
SELECT orders.OrderNumber,
    SUM((product.UnitPrice - orderitem.UnitPrice) * orderitem.Quantity) AS total_saved
FROM orders
JOIN orderitem ON orders.Id = orderitem.OrderId
JOIN product ON orderitem.ProductId = product.Id
GROUP BY orders.OrderNumber
ORDER BY total_saved DESC;
```

23. **Mr. Kavin want to become a supplier. He got the database of "Richard's Supply" for reference. Help him to pick:**

    List few products that he should choose based on demand.

```
SELECT product.ProductName, SUM(orderitem.Quantity) AS total_quantity_sold
FROM product
JOIN orderitem ON product.Id = orderitem.ProductId
GROUP BY product.ProductName
ORDER BY total_quantity_sold DESC
LIMIT 5;
```

Who will be the competitors for him for the products suggested in above questions.

```
SELECT supplier.CompanyName, product.ProductName
FROM product
JOIN supplier ON product.SupplierId = supplier.Id
WHERE product.ProductName IN (
  SELECT product.ProductName
  FROM product
  JOIN orderitem ON product.Id = orderitem.ProductId
  GROUP BY product.ProductName
  ORDER BY SUM(orderitem.Quantity) DESC LIMIT 5 );
```

24. **Every supplier supplies specific products to the customers. Create a view of suppliers and total sales made by their products and write a query on this view to find out top 2 suppliers (using windows function RANK() in each country by total sales done by the products.**

```
CREATE VIEW SupplierSales AS
SELECT supplier.Id AS SupplierId, supplier.CompanyName, SUM(orderitem.UnitPrice *
orderitem.Quantity) AS total_sales
FROM supplier
JOIN product ON supplier.Id = product.SupplierId
JOIN orderitem ON product.Id = orderitem.ProductId
GROUP BY supplier.Id, supplier.CompanyName;

SELECT SupplierId, CompanyName, total_sales,
    RANK() OVER (PARTITION BY supplier.Country ORDER BY total_sales DESC) AS sales_rank
FROM SupplierSales
JOIN supplier ON SupplierSales.SupplierId = supplier.Id
WHERE sales_rank <= 2;
```

25. **Find out for which products, UK is dependent on other countries for the supply. List the countries which are supplying these products in the same list.**

```
SELECT product.ProductName, supplier.Country FROM product
JOIN supplier ON product.SupplierId = supplier.Id
WHERE supplier.Country != 'UK'
 AND product.Id IN (
    SELECT product.Id
    FROM product
    JOIN orderitem ON product.Id = orderitem.ProductId
    JOIN orders ON orderitem.OrderId = orders.Id
    JOIN customer ON orders.CustomerId = customer.Id
    WHERE customer.Country = 'UK');
```