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

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

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

SELECT \* FROM product WHERE IsDiscontinued = 0;

1. **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;

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

SELECT \* FROM customer WHERE country = 'Mexico';

1. **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;

1. **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;

1. **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;

1. **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;

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

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

1. **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;

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

SELECT YEAR(OrderDate) AS year, COUNT(\*) AS order\_count

FROM orders

GROUP BY year;

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

SELECT YEAR(OrderDate) AS year, SUM(TotalAmount) AS total\_revenue

FROM orders

GROUP BY year;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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;

1. **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 );

1. **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;

1. **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');