

Here are sample answers to the five SQL-related questions mentioned:

1. Analyzing Order Data Using Window Functions

Question: Write a SQL query to calculate the total, average, and maximum order amount per country per month. Also, rank the countries by the total order amount for each month.

Answer:

```
WITH MonthlyData AS (  
    SELECT  
        Country,  
        DATE_FORMAT(OrderDate, '%Y-%m') AS Month,  
        SUM(OrderAmount) AS TotalAmount,  
        AVG(OrderAmount) AS AvgAmount,  
        MAX(OrderAmount) AS MaxAmount  
    FROM Orders  
    GROUP BY Country, DATE_FORMAT(OrderDate, '%Y-%m')  
)  
SELECT  
    Country,  
    Month,  
    TotalAmount,  
    AvgAmount,  
    MaxAmount,  
    RANK() OVER (PARTITION BY Month ORDER BY TotalAmount DESC) AS CountryRank  
FROM MonthlyData;
```

2. Analyzing Sales Data

Question: Write a SQL query that retrieves the total sales amount for each country for the last 12 months.

Answer:

```
SELECT  
    Country,  
    SUM(SalesAmount) AS TotalSales  
FROM Sales  
WHERE OrderDate >= DATE_ADD(CURDATE(), INTERVAL -12 MONTH)  
GROUP BY Country  
ORDER BY TotalSales DESC;
```

3. Filtering Customers Based on Purchase History

Question: Identify customers who made their last purchase more than 30 days ago and have total spending greater than \$1,000.

Answer:

```
WITH CustomerSpending AS (  
    SELECT
```

```

        CustomerID,
        MAX(OrderDate) AS LastPurchaseDate,
        SUM(OrderAmount) AS TotalSpending
    FROM Orders
    GROUP BY CustomerID
)
SELECT
    CustomerID,
    LastPurchaseDate,
    TotalSpending
FROM CustomerSpending
WHERE LastPurchaseDate < DATE_ADD(CURDATE(), INTERVAL -30 DAY)
    AND TotalSpending > 1000;

```

4. Calculating Product Performance Metrics

Question: Calculate the gross profit margin, average units sold, and net profit for each product based on sales and cost data.

Answer:

```

SELECT
    ProductID,
    SUM(SalesAmount - CostAmount) AS NetProfit,
    AVG(UnitsSold) AS AvgUnitsSold,
    SUM(SalesAmount - CostAmount) / SUM(SalesAmount) * 100 AS GrossProfitMargin
FROM ProductSales
GROUP BY ProductID
ORDER BY NetProfit DESC;

```

5. Understanding SQL Constraints

Question: Explain the purpose of the FOREIGN KEY and UNIQUE constraints in SQL.

Answer:

- **FOREIGN KEY:**

A foreign key enforces a relationship between two tables by ensuring that the value in the foreign key column in one table corresponds to a primary key value in another table. It ensures referential integrity. For example, an `Order` table may have a `CustomerID` column as a foreign key referencing the `Customer` table.

- **UNIQUE:**

A unique constraint ensures that all values in a column or a combination of columns are distinct. For example, in a `User` table, the `Email` column may have a unique constraint to prevent duplicate email addresses.

If you need further explanation or examples for any of these, feel free to ask!