

1.

- a. The calculation evaluates the average of all total order amounts, regardless of price, item count, etc. This is not a useful metric; the data has a large range with a minimum value of \$90 and a maximum of \$704000. I would evaluate the average order amount in a way that accounts for this discrepancy. I also considered metrics like average price of shoe per sale, but the problem explicitly asks for average order value. This also lays down the assumption that people are not purchasing large quantities of shoes at a time because the problem statement asks for average order value while it disregards the current calculation as naïve because of its high price when compared to cost per shoe.
- b. I would report the median order value, as opposed to the mean, to eliminate the effect of outliers like the possibly bulk buyer who bought 2000 sneakers at \$704000 from shop 42. It would be a better representation of the “average” buyer from such stores.
- c. The median order value is \$284.

2.

- a. `SELECT COUNT(ShipperID) FROM Orders WHERE ShipperID=1;`

Numerical result: 54

Speedy Express has the ID of 1, so you can simply count how many orders have a ShipperID of 1

- b. `SELECT LastName
FROM Employees
WHERE EmployeeID=(SELECT EmployeeID
FROM (SELECT EmployeeID, MAX(EmployeeOrderCount)
FROM (SELECT EmployeeID, COUNT(EmployeeID) AS EmployeeOrderCount
FROM Orders
GROUP BY EmployeeID))));`

Result: Peacock

First, I tallied up each employee ID from the order table, then found the one with the maximum count. I then looked at the employees table and filtered for the one with the ID that matches the highest tally from before.

c. `SELECT ProductName
FROM Products
WHERE ProductID=(SELECT ProductID
FROM (SELECT ProductID, MAX(TotalBoughtGermany)
FROM (SELECT ProductID, SUM(Quantity) AS TotalBoughtGermany
FROM (SELECT OrderDetails.ProductID, OrderDetails.Quantity, Customers.Country
FROM OrderDetails
LEFT JOIN Orders
ON OrderDetails.OrderID=Orders.OrderID
LEFT JOIN Customers
ON Customers.CustomerID=Orders.CustomerID
WHERE Country='Germany')
GROUP BY ProductID))));`

Result: Boston Crab Meat

I first isolated product ID and quantity ordered from the OrderDetails table. I then merged the Orders and OrderDetails table using LEFT JOIN by their order ids to match up order product and quantity, with the customer. I also included a column for the customers' country and merged the tables again using LEFT JOIN based on entries with the same CustomerID and filtered out only the ones who are from Germany. Now that the product ID and quantities have been isolated for only German buyers, I added all the quantities by their product ID and found the productid with the maximum quantity sum within that. Finally, I matched that product ID with its associated product name.