Winter 2022 Data Science Intern Challenge

Question 1

On Shopify, we have exactly 100 sneaker shops, and each of these shops sells only one model of shoe. We want to do some analysis of the average order value (AOV). When we look at orders data over a 30 day window, we naively calculate an AOV of $3145.13. Given that we know these shops are selling sneakers, a relatively affordable item, something seems wrong with our analysis.

a. **Think about what could be going wrong with our calculation. Think about a better way to evaluate this data.**

The issue with the calculation is that it's skewed by two types of very high cost orders: orders that have a large amount of relatively average priced sneakers, and orders consisting of very expensive sneakers. For the first type, see for example row 17: the order amount is $704,000 but there are 2000 total items, meaning each pair of sneakers cost about $352. For the second type, see for example row 162: there is only one total item but it costs $25,725.
Since the majority of the orders have order value less than $1000 (only 71 out of 5000 orders, or 1.42%, have order value greater than or equal to $1000), the average order value is more than 3 times higher than the cost of most (98.59%) of the orders. This shows that the high cost orders mentioned above skew the average higher than the order value of most orders. In particular, 63 orders have order value between $25,000 and $705,000. This means that 63 orders have order value more than 25 to 705 times higher than the majority of the orders (4929 out of 5000), which really shows us why the average is much higher than what was expected.

b. **What metric would you report for this dataset?**

Instead of calculating the average order value for this dataset, I would use a different metric in order to find the value that is closest to the centre of the dataset. I would instead calculate the median of the dataset, which will better represent the majority of the orders. If we only consider orders with order value less than $1000, we find that the average order value is $301.06, which we will see is much closer to the median than the original average order value of $3145.13.

c. **What is its value?**

The median of the given dataset is $284.

Question 2

a. **How many orders were shipped by Speedy Express in total?**

My query was as follows:

```
SELECT COUNT(OrderID)
FROM Orders
WHERE EXISTS (SELECT ShipperName FROM Shippers WHERE Orders.ShipperID =
Shippers.ShipperID AND ShipperName = 'Speedy Express');
```

My final answer was 54.

For this question, I wanted to select the OrderID's that were shipped by Speedy Express, but OrderID and ShipperName don't appear in the same table. Therefore, I used the WHERE EXISTS command to make sure I only selected the OrderID's which had the ShipperID corresponding to Speedy Express, since the Orders and Shippers tables share the ShipperID column. I then used COUNT to see how many OrderID's fit this criteria.

b. **What is the last name of the employee with the most orders?**

My query was as follows:

```
SELECT LastName
FROM(SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders FROM
Orders
LEFT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
GROUP BY LastName
ORDER BY NumberOfOrders DESC)
LIMIT 1;
```

My final answer was Peacock.

For this question, the employee's LastName doesn't appear anywhere except the Employees table, and that table doesn't contain any information about the orders they sold. To get around this, I used LEFT JOIN on the tables Employees and Orders since they share the EmployeeID column. The table I ended up with from my query had two columns: the employee's LastName, and NumberOfOrders, which was the total count of OrderID's associated with each employee. I achieved this by using GROUP BY to group each number of orders with the associated employee. I also ordered this table by NumberOfOrders descending so that the employee with the most orders would be at the top of the table. Finally, I selected LastName with a LIMIT of 1 so that I would get only the last name of the employee with the most orders.

c. **What product was ordered the most by customers in Germany?**

My query was as follows:

```
SELECT ProductName
FROM(SELECT Customers.Country, Products.ProductName, Products.ProductID,
COUNT(Products.ProductID)
FROM (((Customers
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
INNER JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID)
INNER JOIN Products ON OrderDetails.ProductID = Products.ProductID)
WHERE Country = 'Germany'
GROUP BY Products.ProductID
ORDER BY COUNT(Products.ProductID) DESC)
LIMIT 1;
```

My final answer was Gorgonzola Telino.

In this question, ProductName and Country do not appear in the same table, and they also do not share columns with a single other table. Instead, we have to link the two via two other tables. To do this, I used INNER JOIN on the four tables Customers, Orders, OrderDetails, and Products, where the first two share the column CustomerID, the middle two share the column OrderId, and the last two share the column ProductID. The query gives the INNER JOIN of those four tables, but only the rows which have Germany as the Country; the table's contents are also grouped by ProductID and ordered so that the product ordered most at the top. Then I selected the ProductName LIMIT 1 so that I would only get the ProductName with the most orders.