Question 1.

Refer to https://nbviewer.org/github/EdwardKHKim/shopify_ds/blob/main/edwardkim_shopify.ipynb for the python script to run the data analysis and reasoning.

(a) The problem with the calculation was that the total **`order_amount`** across all stores was divided by the total count of **`total_items`** which is equivalent to 5,000. However, some rows contain a value of > 1 for **`total_items`**. The AOV value after re-calculation with proper denominator is **`357.92`**. However, looking at std for **`order_amount`** in IQR, it appeared that there could be an outlier assuming that the **`total_items`** value for that **`shop_id`** and/or **`order_id`** is not high. After further analysis, it was found that **`shop_id=78`** has a large **`order_amount`** per single item. Therefore, all data corresponding to **`shop_id=78`** was removed. The output after such removal was **`307.01`**.

(b) Although the previous calculation provides us with aN AOV metric for all shoes sold in the store, a more proper metric could be to determine the AOV of individual stores. This would remove the discrepancy dependent on the **`total_items`** sold by each **`shop_id`** and would peg the AOV value to individual shops. The metric will assume that each shop sells a single pair of shoes. Then I find the average value of shoes across shops.

(c) The value based on the (b) metric is **`152.48`**.

Question 2.

(a) **Answer:** 54

```
SELECT COUNT(*)
FROM Orders o JOIN Shippers s ON o.ShipperID = s.ShipperID
WHERE ShipperName = "Speedy Express"
```

(b) **Answer:** Peacock

```
SELECT LastName
FROM Orders o JOIN Employees e ON o.EmployeeID = e.EmployeeID
GROUP BY o.EmployeeID
ORDER BY COUNT(*) DESC
```

(c) **Answer:** Steeleye Stoute

```sql
SELECT ProductName
FROM Orders o JOIN Customers c ON o.CustomerID = c.CustomerID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON p.ProductID = od.ProductID
WHERE Country='Germany'
ORDER BY Quantity DESC
```