Q1.

**a.** A naïve view of the data set explains that the Average Order Value is 3145.13. This is obviously skewed by the multiple large orders (amounting to $704,000 each) and some erroneous entries (shop\_id 78 values 1 sneaker @ $25,725 {which should not be true :o}).

The most important aspect is that each store sells exactly 1 type of sneaker and hence we have a constant parameter.

My code leverages this assumption. Acknowledging the fact that the stores might sell sneakers at different price points, outliers are calculated and removed from the df using IQR on the selling cost per pair of sneakers (which is also calculated in the code using the order amount and number of items).

Generally, outliers are either corrected or removed, since not much is known about the actual correct costs, and it is unlikely that shop 78 is selling sneakers at that exorbitant price, I removed all outliers.

Once outliers are removed, the new Average Order Value is calculated.

**b.** The metric for the data set must ignore the outliers on the selling cost of pair of sneakers, and hence will be the median of this cost.

**c.** The value of this metric (median of selling price of sneakers) is approx. $153.

Q2.

1.

SELECT COUNT(o.OrderID) AS 'Number of Speedy Express Orders'

FROM Orders AS o

INNER JOIN Shippers AS s ON (s.ShipperID = o.ShipperID AND s.ShipperName = 'Speedy Express');

**Numerical Value:** 54

2.

SELECT e.LastName AS 'Last Name'

FROM Employees AS e

INNER JOIN Orders AS o ON

(e.EmployeeID = o.EmployeeID)

GROUP BY o.EmployeeID

ORDER BY COUNT(o.EmployeeID) DESC LIMIT 1;

**Solution:** Peacock

(Continued on Next Page)

3.

SELECT p.ProductName AS 'Product'

FROM Products AS p

INNER JOIN OrderDetails AS o ON

(p.ProductID = o.ProductID)

INNER JOIN Orders AS q ON

(o.OrderID = q.OrderID)

INNER JOIN Customers AS c ON

(q.CustomerID = c.CustomerID)

WHERE c.Country = 'Germany'

GROUP BY o.ProductID

ORDER BY COUNT(o.ProductID) DESC LIMIT 1;

**Solution:** Gorgonzola Telino