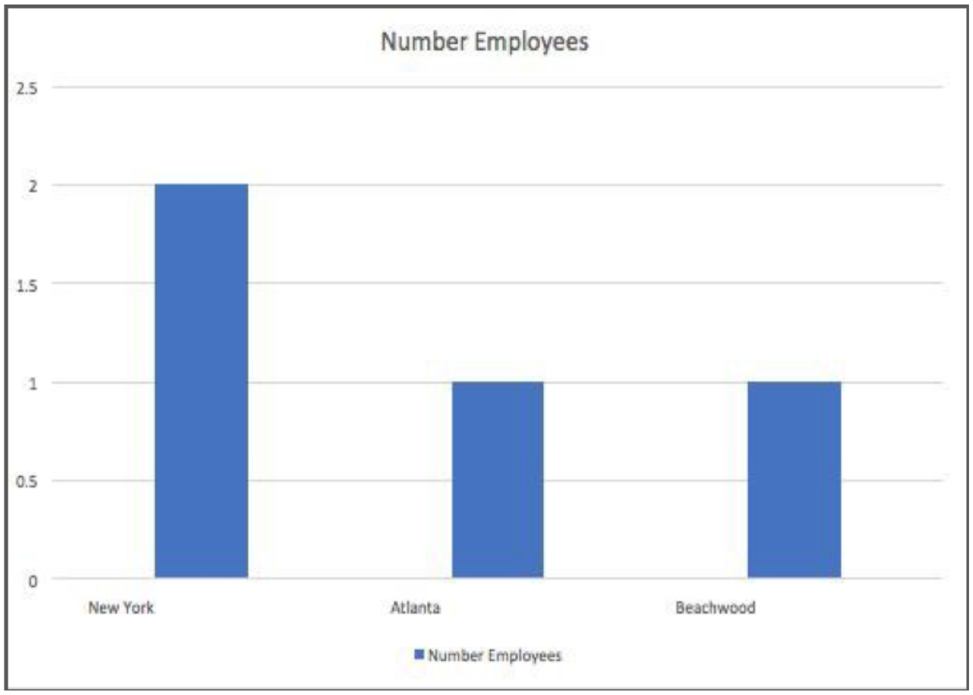
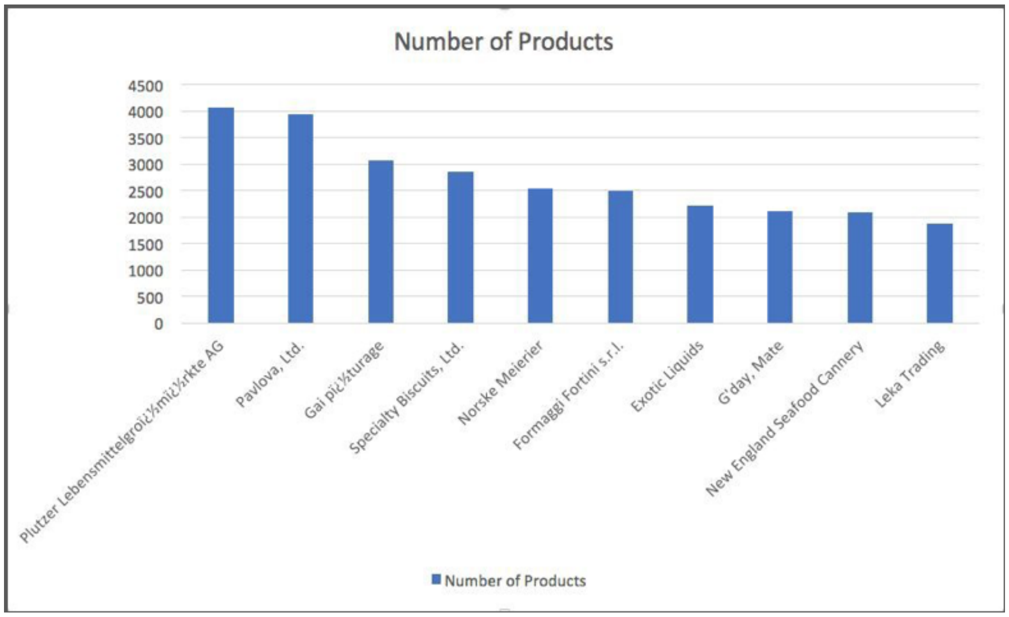
# SQL For Territory Data



This visualization displays on the X-axis the top 3 represented territories, and the Y-axis shows the number of employees supporting each territory. According to the visualization, New York is supported by 2 employees, whereas all others are supported by 1.

SQL Code:

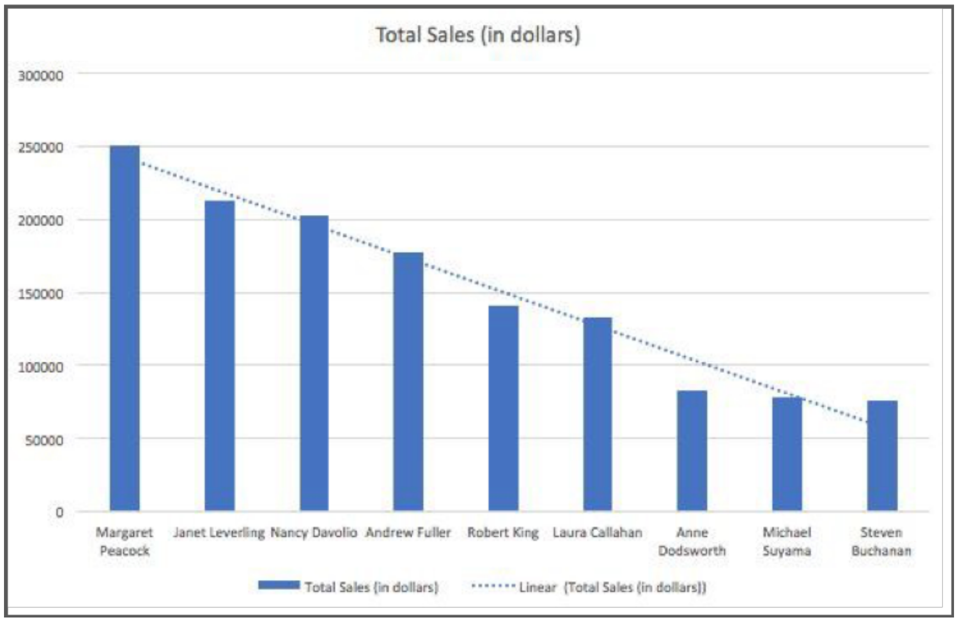
```
SELECT Territories.TerritoryDescription, Count(*) AS num
FROM Employees JOIN EmployeeTerritories JOIN Territories
ON Employees.EmployeeID = EmployeeTerritories.EmployeeID AND
EmployeeTerritories.TerritoryID = Territories.TerritoryID
GROUP BY Territories.TerritoryDescription
ORDER BY num DESC
LIMIT 3;
```



On the X-axis are our suppliers, and on the Y-axis is the number of products that have been purchased by our customers. According to this visualization, we sell the most products from Plutzer at around 4000 products

SQL Code:

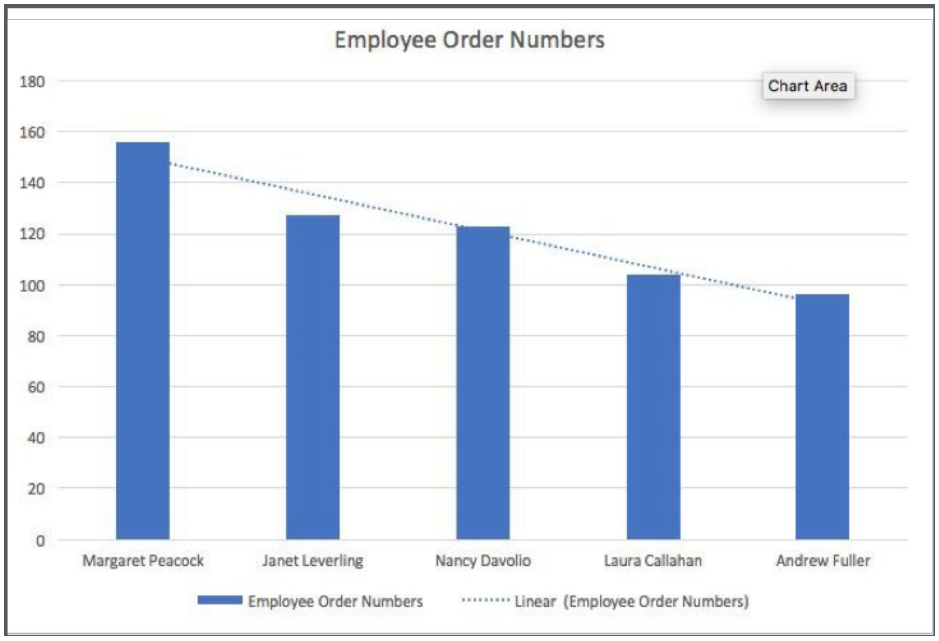
```
SELECT Suppliers.CompanyName, sum(OrderDetails.Quantity) AS NumProducts
FROM Suppliers JOIN Products JOIN OrderDetails
ON Suppliers.SupplierID = Products.SupplierID AND Products.ProductID =
OrderDetails.ProductID
GROUP BY Suppliers.CompanyName
ORDER BY NumProducts DESC
LIMIT 10;
```



This visualization displays on the X-axis each of our employees, and on the Y-axis the amount of sales in dollars each employee has sold. According to this visualization, Margaret Peacock has the most sales at around $250,000.

SQL Code:

```
SELECT Employees.FirstName, Employees.LastName,
sum(OrderDetails.Quantity*OrderDetails.UnitPrice) AS totalSales
FROM Employees JOIN Orders JOIN OrderDetails
ON Employees.EmployeeID = Orders.EmployeeID AND Orders.OrderId =
OrderDetails.OrderID
GROUP BY Employees.EmployeeID
ORDER BY totalSales DESC;
```



This visualization shows on the X-axis employees who have above average order numbers, and on the Y-axis is the amount oforders. According to this chart, Margaret Peacock has the most orders at around 156 orders.

SQL Code:

```
SELECT Employees.FirstName, Employees.LastName, count(Orders.OrderId) AS
EmployeeOrders
FROM Employees
JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID
JOIN (select (count(Orders.OrderId) / count(distinct Employees.EmployeeID))AS
average FROM Employees
JOIN Orders on Employees.EmployeeID = Orders.EmployeeID) AS subquery
GROUP BY Orders.EmployeeID HAVING EmployeeOrders > average
ORDER BY EmployeeOrders DESC;
```