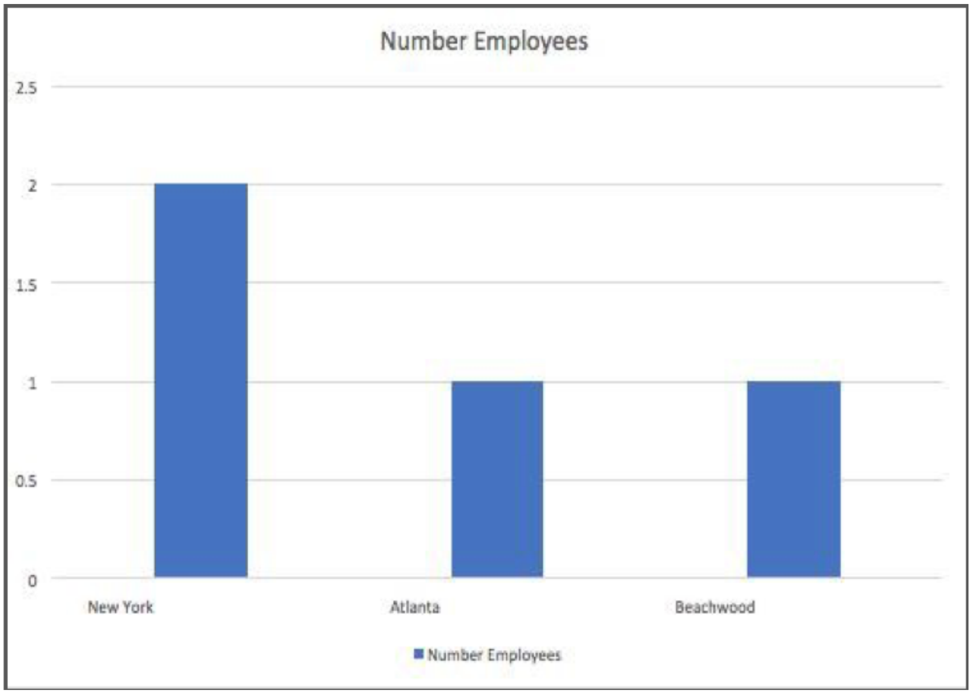
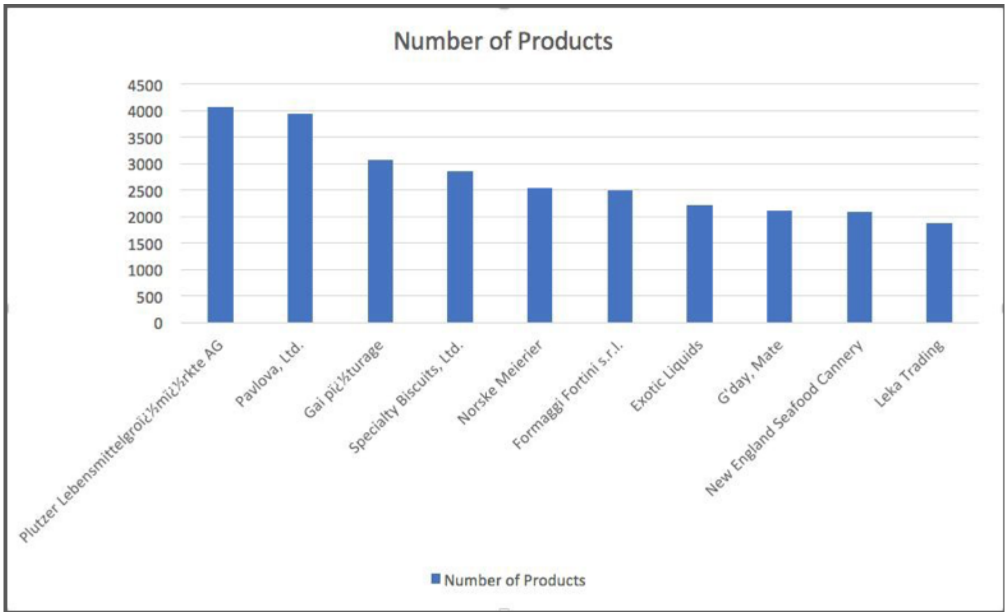
SQL For Territory Data



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:

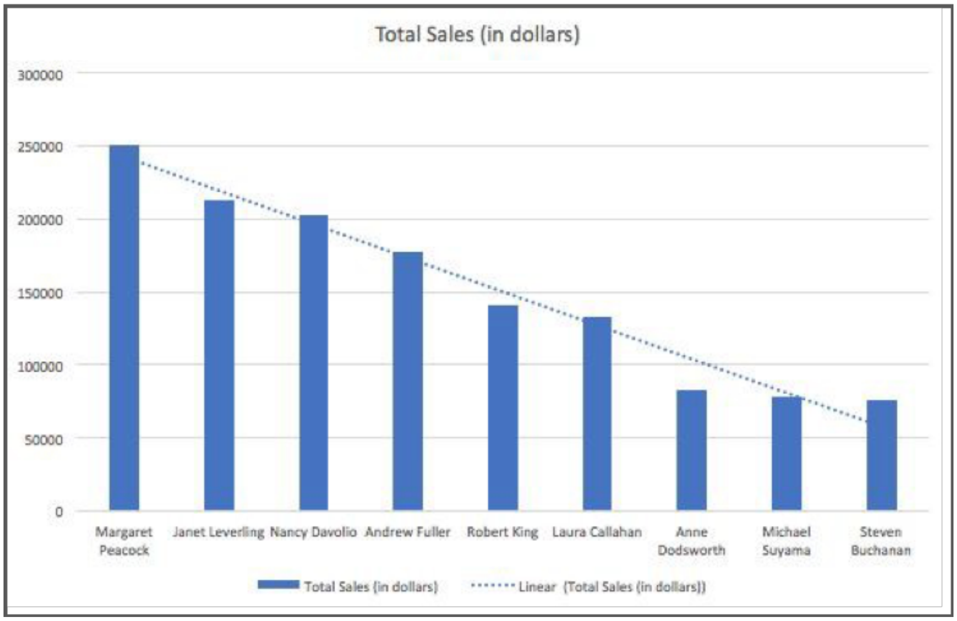
```
select Territories.TerritoryDescription, Count(*) as num from Employees
join EmployeeTerritories join Territories on Employees.EmployeeID =
EmployeeTerritories.EmployeeID and EmployeeTerritories.TerritoryID =
Territories.TerritoryID group by Territories.TerritoryDescription order by
num desc limit 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:

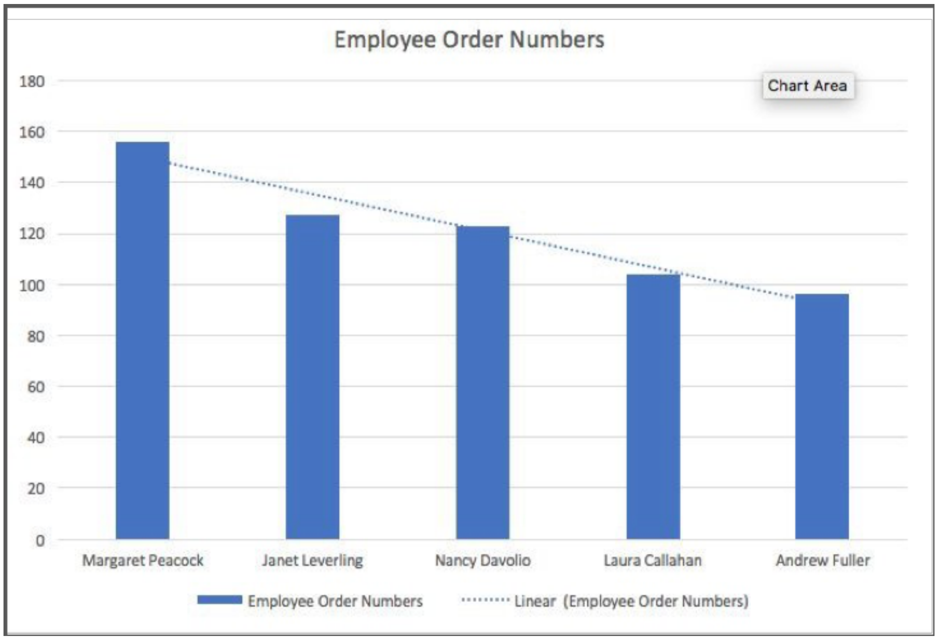
```
select Suppliers.CompanyName, sum(OrderDetails.Quantity) as NumProducts from
Suppliers join Products join OrderDetails on Suppliers.SupplierID =
Products.SupplierID and Products.ProductID = OrderDetails.ProductID group by
Suppliers.CompanyName order by NumProducts desc limit 10;
```



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;
```