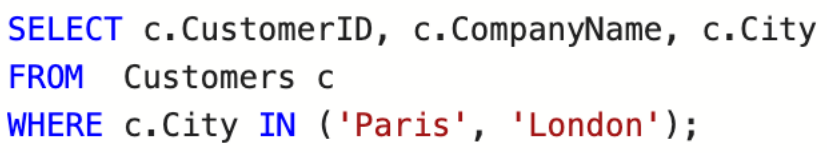
SQL MINI PROJECT

Asakar Hussain

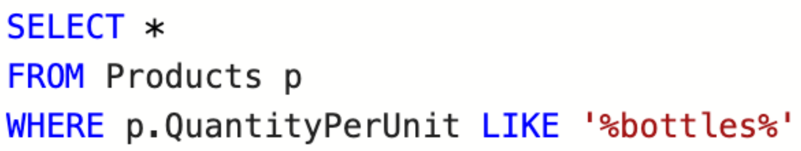
SPARTA GLOBAL | JAVA SDET TRAINEE

1.1 Write a query that lists all Customers in either Paris or London. Include Customer ID, Company Name and all address fields.



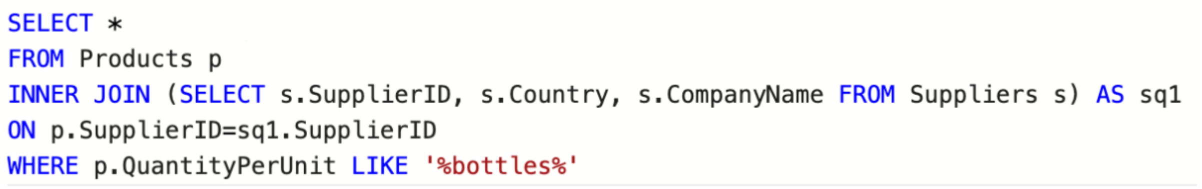
```
SELECT c.CustomerID, c.CompanyName, c.City
FROM  Customers c
WHERE c.City IN ('Paris', 'London');
```

1.2 List all products stored in bottles.



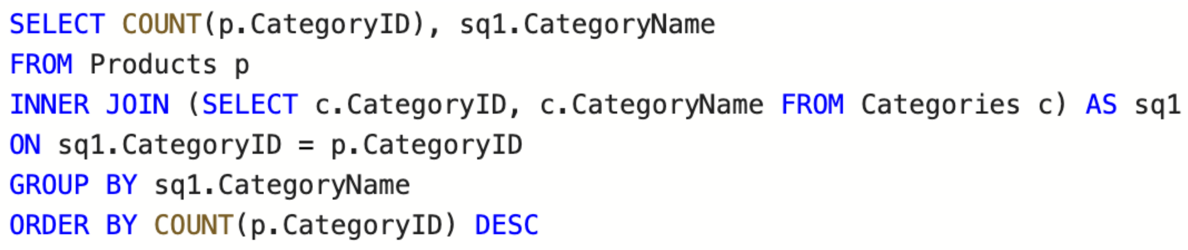
```
SELECT *
FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottles%'
```

1.3 Repeat question above, but add in the Supplier Name and Country.



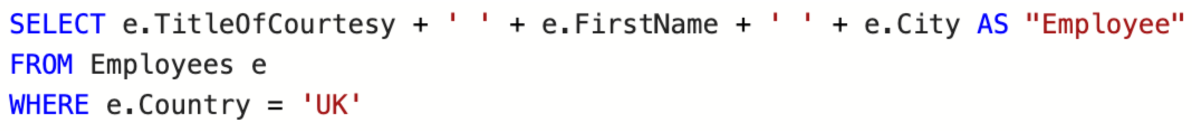
```
SELECT *
FROM Products p
INNER JOIN (SELECT s.SupplierID, s.Country, s.CompanyName FROM Suppliers s) AS sq1
ON p.SupplierID=sq1.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottles%'
```

1.4 Write an SQL Statement that shows how many products there are in each category. Include Category Name in result set and list the highest number first.



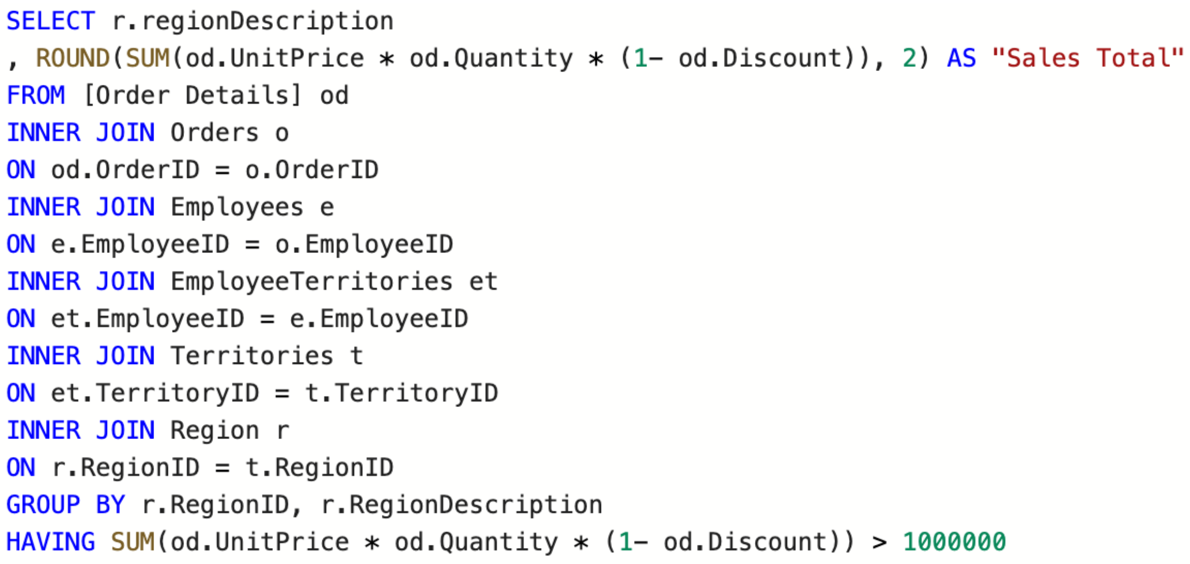
```
SELECT COUNT(p.CategoryID), sq1.CategoryName
FROM Products p
INNER JOIN (SELECT c.CategoryID, c.CategoryName FROM Categories c) AS sq1
ON sq1.CategoryID = p.CategoryID
GROUP BY sq1.CategoryName
ORDER BY COUNT(p.CategoryID) DESC
```

1.5 List all UK employees using concatenation to join their title of courtesy, first name and last name together. Also include their city of residence.



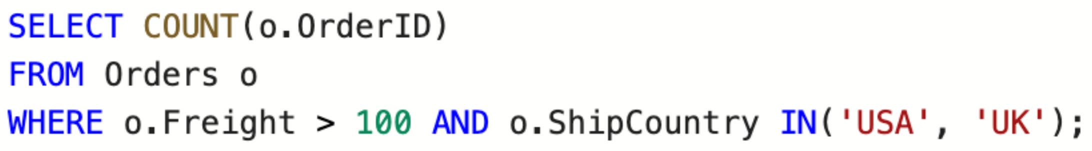
```
SELECT e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.City AS "Employee"
FROM Employees e
WHERE e.Country = 'UK'
```

1.6 List Sales Totals for all Sales Regions (via the Territories table using 4 joins) with a Sales Total greater than 1,000,000. Use rounding or FORMAT to present the numbers.



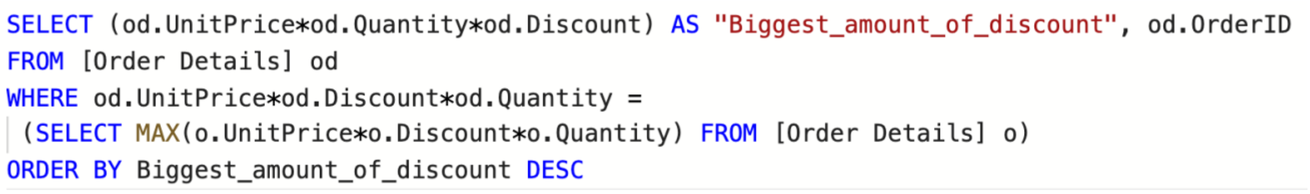
```
SELECT r.regionDescription
, ROUND(SUM(od.UnitPrice * od.Quantity * (1- od.Discount)), 2) AS "Sales Total"
FROM [Order Details] od
INNER JOIN Orders o
ON od.OrderID = o.OrderID
INNER JOIN Employees e
ON e.EmployeeID = o.EmployeeID
INNER JOIN EmployeeTerritories et
ON et.EmployeeID = e.EmployeeID
INNER JOIN Territories t
ON et.TerritoryID = t.TerritoryID
INNER JOIN Region r
ON r.RegionID = t.RegionID
GROUP BY r.RegionID, r.RegionDescription
HAVING SUM(od.UnitPrice * od.Quantity * (1- od.Discount)) > 1000000
```

1.7 Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.



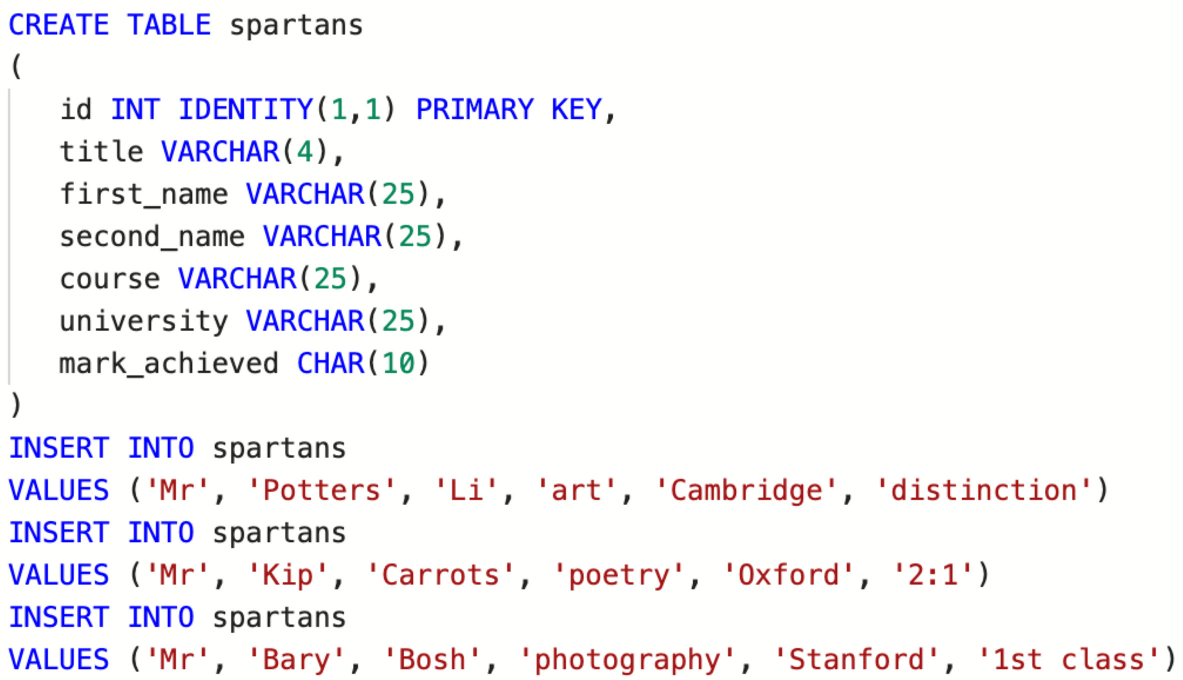
```
SELECT COUNT(o.OrderID)
FROM Orders o
WHERE o.Freight > 100 AND o.ShipCountry IN('USA', 'UK');
```

1.8 Write an SQL Statement to identify the Order Number of the Order with the highest amount(value) of discount applied to that order.



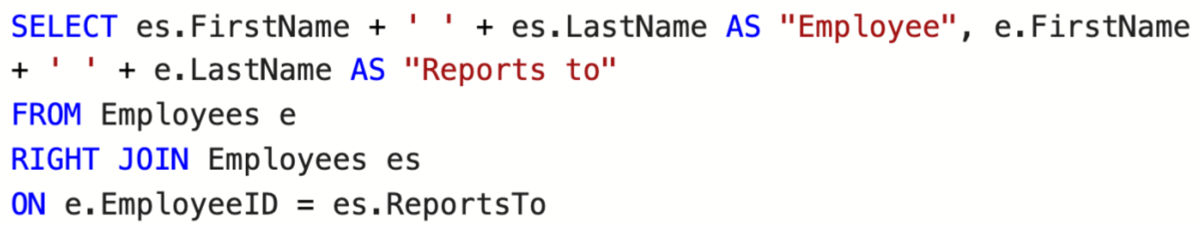
```
SELECT (od.UnitPrice*od.Quantity*od.Discount) AS "Biggest_amount_of_discount", od.OrderID
FROM [Order Details] od
WHERE od.UnitPrice*od.Discount*od.Quantity =
 (SELECT MAX(o.UnitPrice*o.Discount*o.Quantity) FROM [Order Details] o)
ORDER BY Biggest_amount_of_discount DESC
```

2.1 Write the correct SQL statement to create the following table:
Spartans Table – include details about all the Spartans on this course. Separate Title, First Name and Last Name into separate columns, and include University attended, course taken and mark achieved. Add any other columns you feel would be appropriate.
IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.
2.2 Write SQL statements to add the details of the Spartans in your course to the table you have created.



```
CREATE TABLE spartans
(
    id INT IDENTITY(1,1) PRIMARY KEY,
    title VARCHAR(4),
    first_name VARCHAR(25),
    second_name VARCHAR(25),
    course VARCHAR(25),
    university VARCHAR(25),
    mark_achieved CHAR(10)
)
INSERT INTO spartans
VALUES ('Mr', 'Potters', 'Li', 'art', 'Cambridge', 'distinction')
INSERT INTO spartans
VALUES ('Mr', 'Kip', 'Carrots', 'poetry', 'Oxford', '2:1')
INSERT INTO spartans
VALUES ('Mr', 'Bary', 'Bosh', 'photography', 'Stanford', '1st class')
```

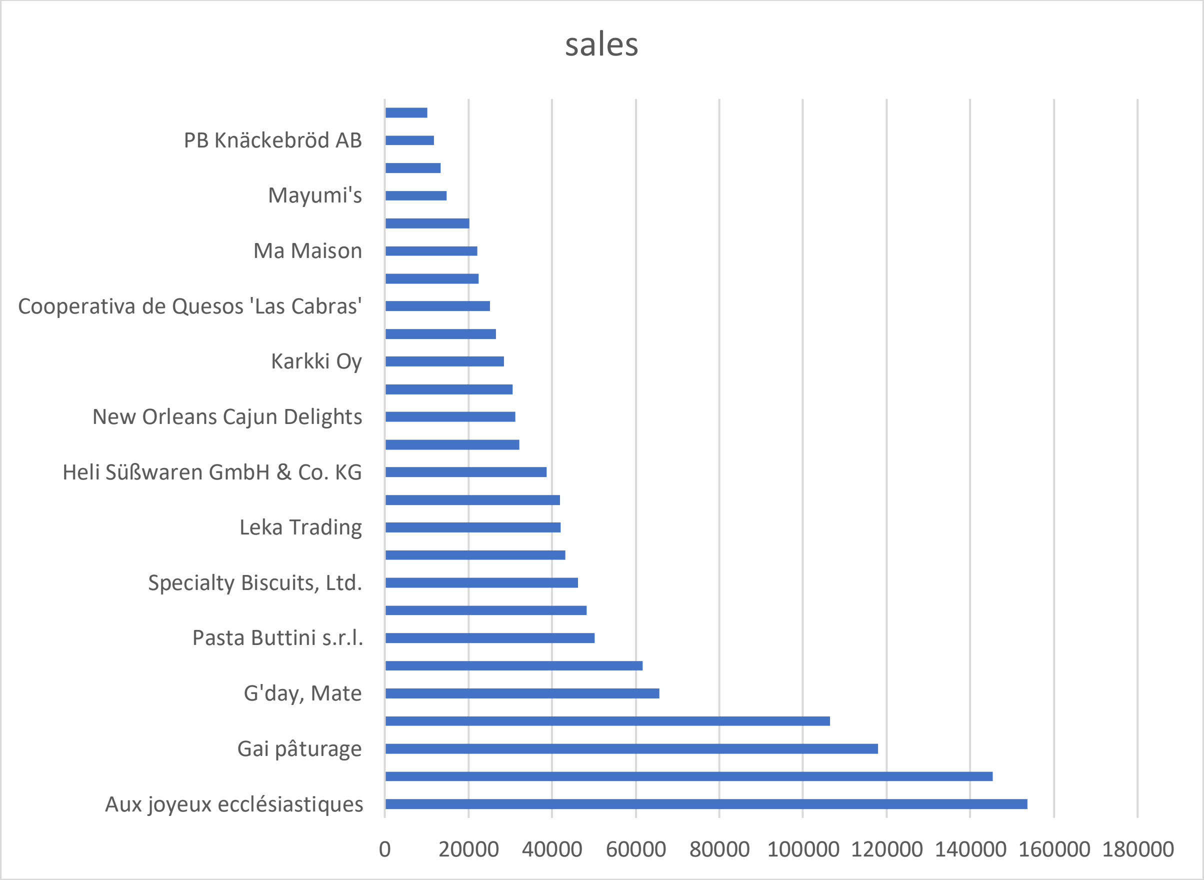
3.1 List all Employees from the Employees table and who they report to.



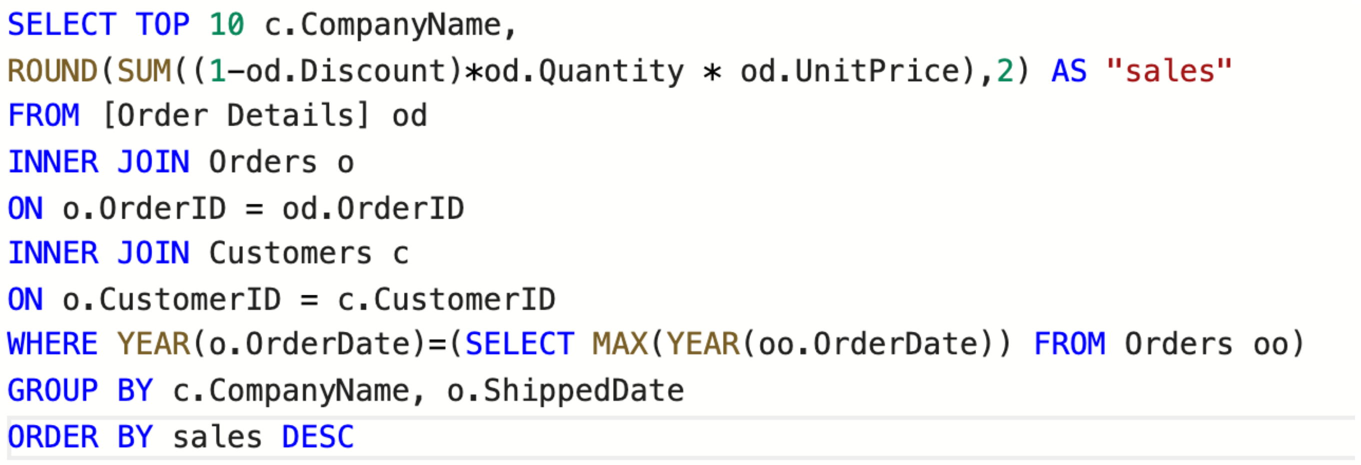
```
SELECT es.FirstName + ' ' + es.LastName AS "Employee", e.FirstName
+ ' ' + e.LastName AS "Reports to"
FROM Employees e
RIGHT JOIN Employees es
ON e.EmployeeID = es.ReportsTo
```

3.2 List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table

```sql
SELECT ROUND(SUM((1-od.Discount)*od.Quantity * od.UnitPrice),2) AS "sales", s.CompanyName
FROM Orders o
INNER JOIN [Order Details] od
ON od.OrderID = o.OrderID
INNER JOIN Products p
ON p.ProductID = od.ProductID
INNER JOIN Suppliers s
ON s.SupplierID = p.SupplierID
GROUP BY s.CompanyName
HAVING ROUND(SUM((1-od.Discount)*od.Quantity * od.UnitPrice),2)  > 10000.00
ORDER BY sales
```

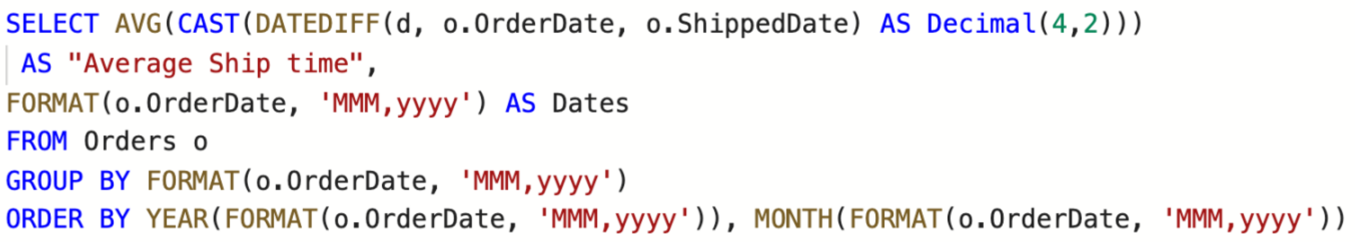


3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped. No Excel required

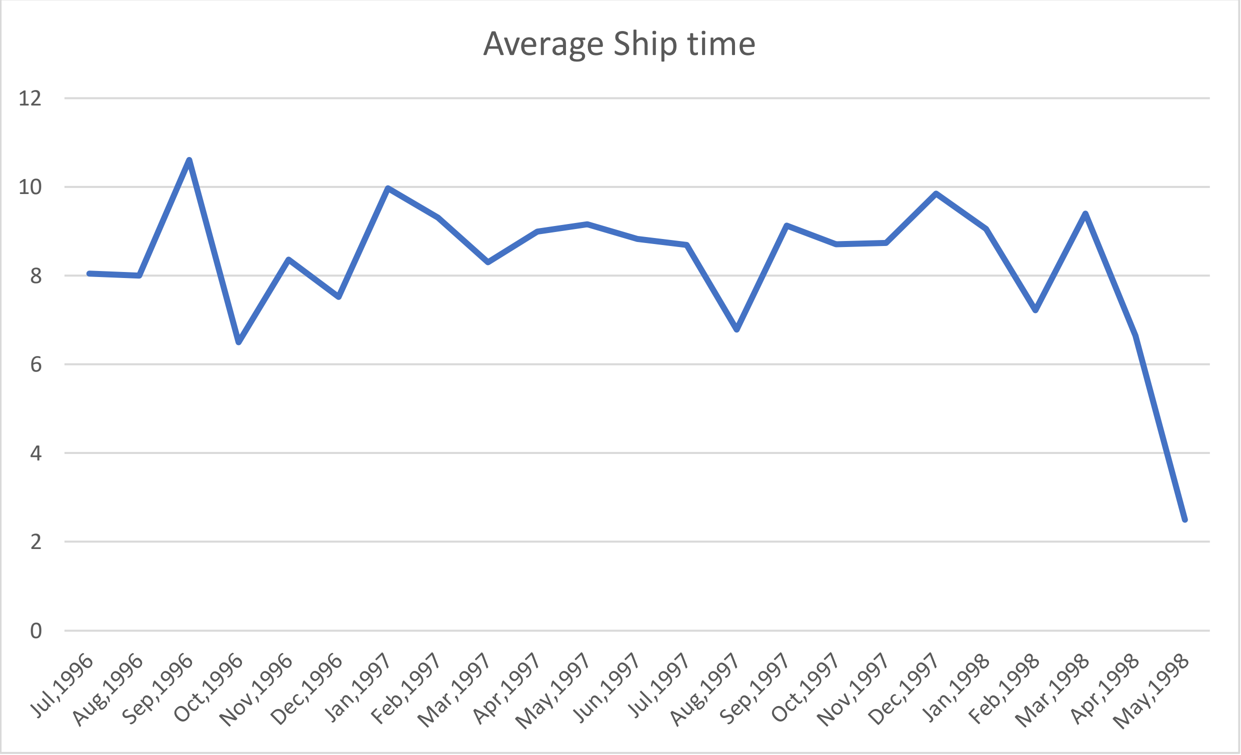


```sql
SELECT TOP 10 c.CompanyName,
ROUND(SUM((1-od.Discount)*od.Quantity * od.UnitPrice),2) AS "sales"
FROM [Order Details] od
INNER JOIN Orders o
ON o.OrderID = od.OrderID
INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
WHERE YEAR(o.OrderDate)=(SELECT MAX(YEAR(oo.OrderDate)) FROM Orders oo)
GROUP BY c.CompanyName, o.ShippedDate
ORDER BY sales DESC
```

3.4 Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below.



```sql
SELECT AVG(CAST(DATEDIFF(d, o.OrderDate, o.ShippedDate) AS Decimal(4,2)))
 AS "Average Ship time",
FORMAT(o.OrderDate, 'MMM,yyyy') AS Dates
FROM Orders o
GROUP BY FORMAT(o.OrderDate, 'MMM,yyyy')
ORDER BY YEAR(FORMAT(o.OrderDate, 'MMM,yyyy')), MONTH(FORMAT(o.OrderDate, 'MMM,yyyy'))
```


Average Ship time
Jul,1996
Aug,1996
Sep,1996
Oct,1996
Nov,1996
Dec,1996
Jan,1997
Feb,1997
Mar,1997
Apr,1997
May,1997
Jun,1997
Jul,1997
Aug,1997
Sep,1997
Oct,1997
Nov,1997
Dec,1997
Jan,1998
Feb,1998
Mar,1998
Apr,1998
May,1998