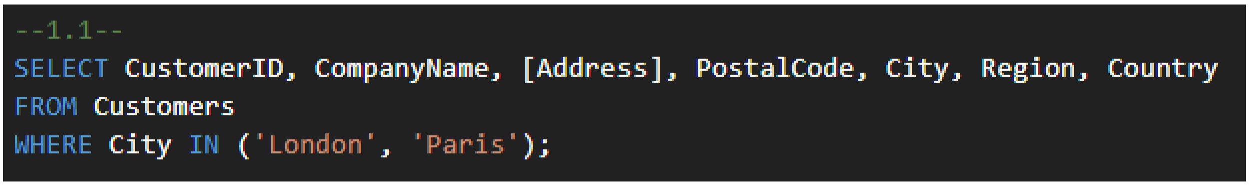
BENJAMIN RANSON
3/18/21

SQL Practical Exercise

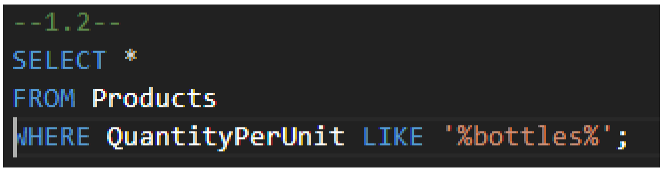
Exercise 1 – Northwind Queries

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



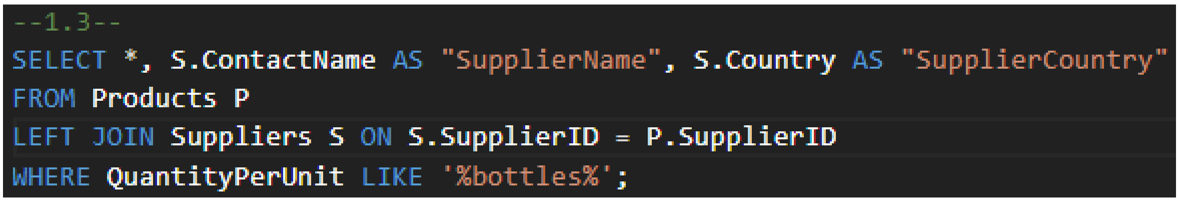
```sql
--1.1--
SELECT CustomerID, CompanyName, [Address], PostalCode, City, Region, Country
FROM Customers
WHERE City IN ('London', 'Paris');
```

1.2 List all products stored in bottles.



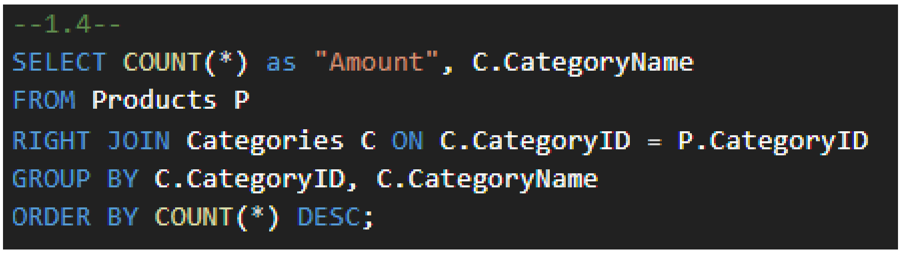
```sql
--1.2--
SELECT *
FROM Products
WHERE QuantityPerUnit LIKE '%bottles%';
```

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



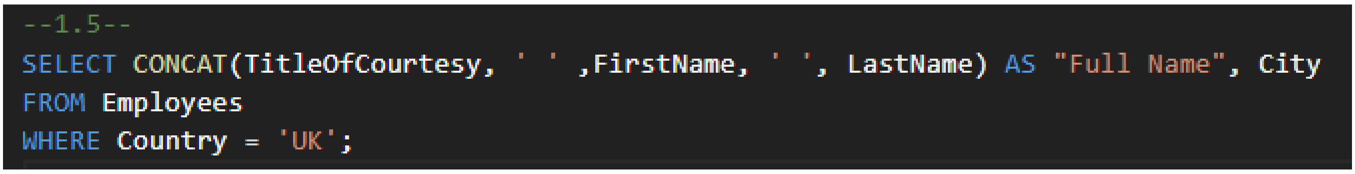
```sql
--1.3--
SELECT *, S.ContactName AS "SupplierName", S.Country AS "SupplierCountry"
FROM Products P
LEFT JOIN Suppliers S ON S.SupplierID = P.SupplierID
WHERE QuantityPerUnit LIKE '%bottles%';
```

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.



```sql
--1.4--
SELECT COUNT(*) as "Amount", C.CategoryName
FROM Products P
RIGHT JOIN Categories C ON C.CategoryID = P.CategoryID
GROUP BY C.CategoryID, C.CategoryName
ORDER BY COUNT(*) DESC;
```

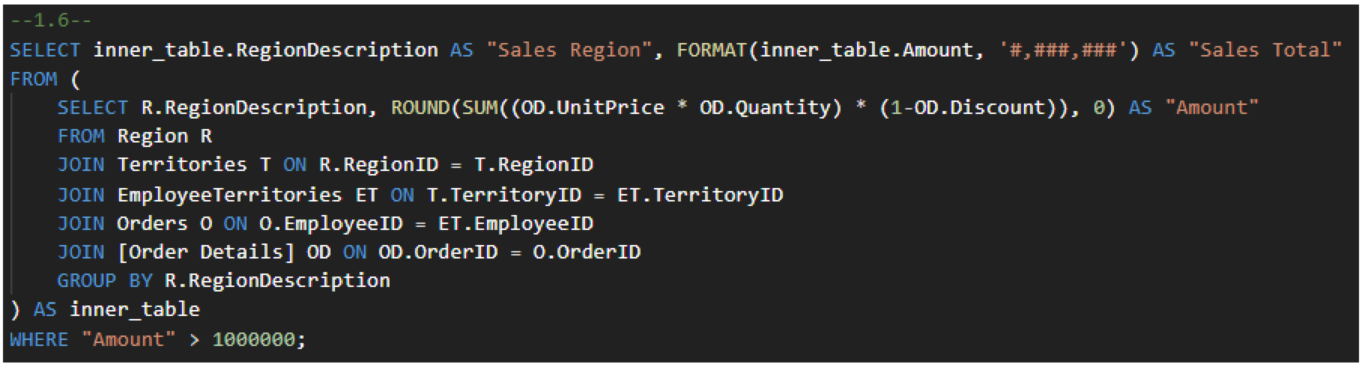
1.5 List all UK employees using concatenation to join their title of courtesy, first name and last name together. Also include their city of residence.



```sql
--1.5--
SELECT CONCAT(TitleOfCourtesy, ' ' ,FirstName, ' ', LastName) AS "Full Name", City
FROM Employees
WHERE Country = 'UK';
```

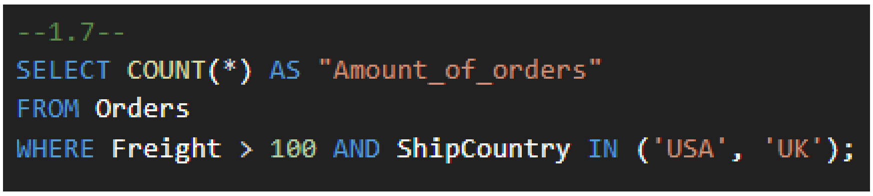
SQL Practical Exercise

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.



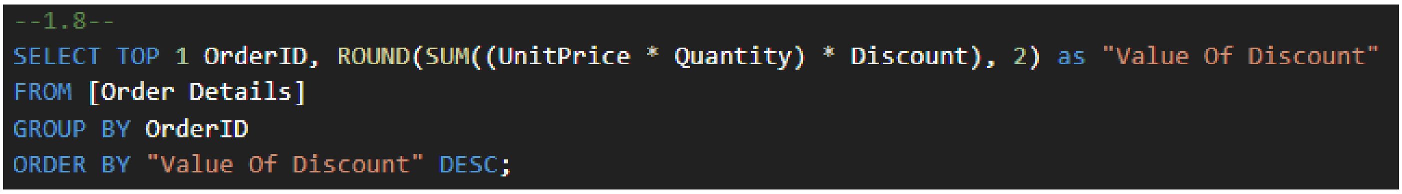
```
--1.6--
SELECT inner_table.RegionDescription AS "Sales Region", FORMAT(inner_table.Amount, '#,###,###') AS "Sales Total"
FROM (
    SELECT R.RegionDescription, ROUND(SUM((OD.UnitPrice * OD.Quantity) * (1-OD.Discount)), 0) AS "Amount"
    FROM Region R
    JOIN Territories T ON R.RegionID = T.RegionID
    JOIN EmployeeTerritories ET ON T.TerritoryID = ET.TerritoryID
    JOIN Orders O ON O.EmployeeID = ET.EmployeeID
    JOIN [Order Details] OD ON OD.OrderID = O.OrderID
    GROUP BY R.RegionDescription
) AS inner_table
WHERE "Amount" > 1000000;
```

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



```
--1.7--
SELECT COUNT(*) AS "Amount_of_orders"
FROM Orders
WHERE Freight > 100 AND ShipCountry IN ('USA', 'UK');
```

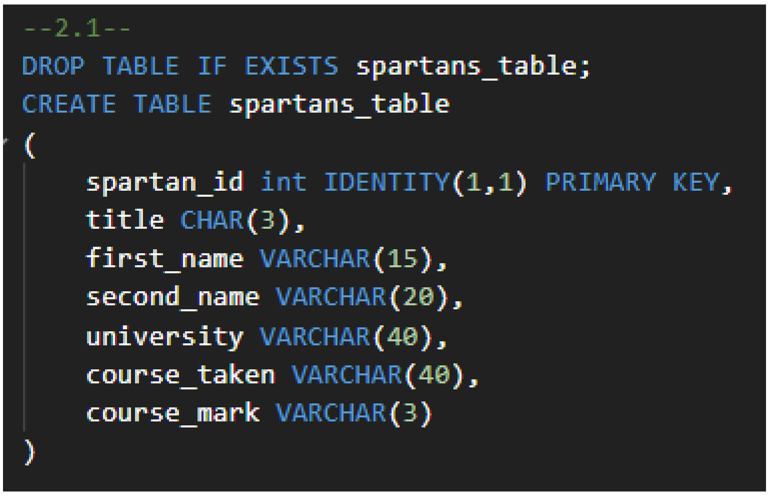
1.8 Write an SQL Statement to identify the Order Number of the Order with the highest amount(value) of discount applied to that order.



```
--1.8--
SELECT TOP 1 OrderID, ROUND(SUM((UnitPrice * Quantity) * Discount), 2) as "Value Of Discount"
FROM [Order Details]
GROUP BY OrderID
ORDER BY "Value Of Discount" DESC;
```

Exercise 2 – Create Spartans Table

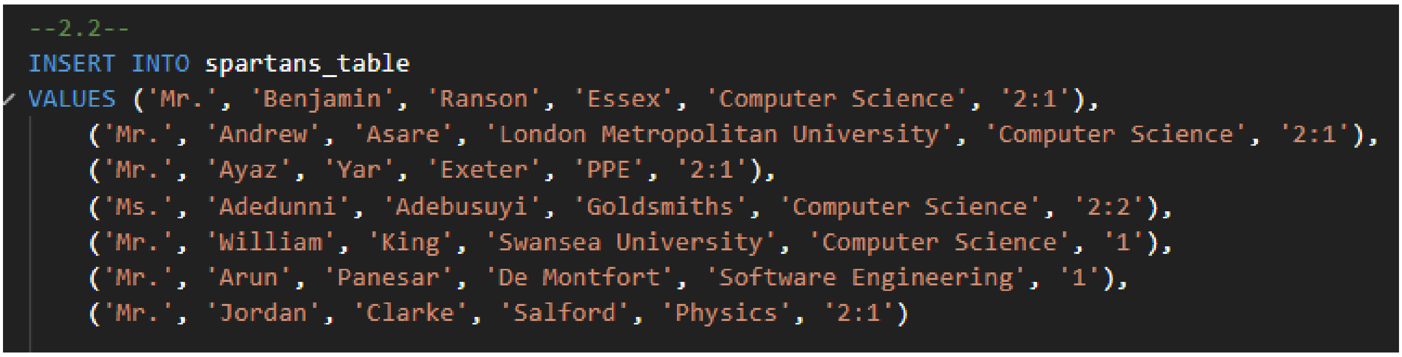
2.1 Write the correct SQL statement to create the following table:



```
--2.1--
DROP TABLE IF EXISTS spartans_table;
CREATE TABLE spartans_table
(
    spartan_id int IDENTITY(1,1) PRIMARY KEY,
    title CHAR(3),
    first_name VARCHAR(15),
    second_name VARCHAR(20),
    university VARCHAR(40),
    course_taken VARCHAR(40),
    course_mark VARCHAR(3)
)
```

SQL Practical Exercise

2.2 Write SQL statements to add the details of the Spartans in your course to the table you have created.

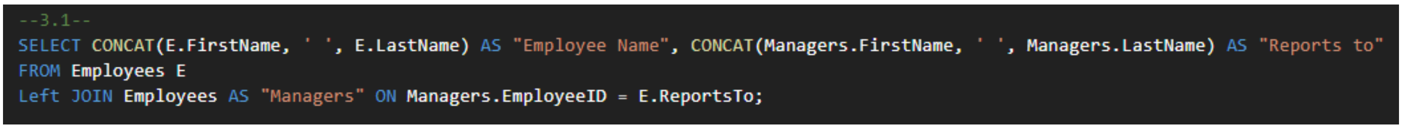


```
--2.2--
INSERT INTO spartans_table
VALUES ('Mr.', 'Benjamin', 'Ranson', 'Essex', 'Computer Science', '2:1'),
    ('Mr.', 'Andrew', 'Asare', 'London Metropolitan University', 'Computer Science', '2:1'),
    ('Mr.', 'Ayaz', 'Yar', 'Exeter', 'PPE', '2:1'),
    ('Ms.', 'Adedunni', 'Adebusuyi', 'Goldsmiths', 'Computer Science', '2:2'),
    ('Mr.', 'William', 'King', 'Swansea University', 'Computer Science', '1'),
    ('Mr.', 'Arun', 'Panesar', 'De Montfort', 'Software Engineering', '1'),
    ('Mr.', 'Jordan', 'Clarke', 'Salford', 'Physics', '2:1')
```

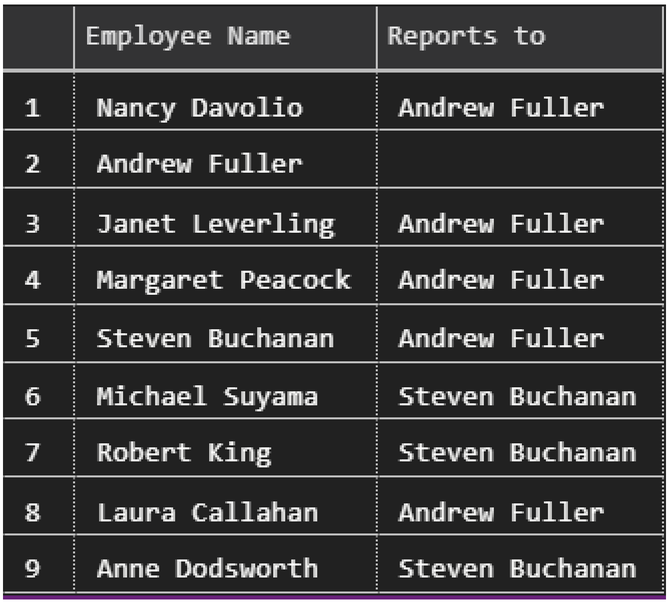
Exercise 3 – Northwind Data Analysis linked to Excel (30 marks)

Write SQL statements to extract the data required for the following charts (create these in Excel):

3.1 List all Employees from the Employees table and who they report to. No Excel required. Please mention the Employee Names and the ReportTo names.



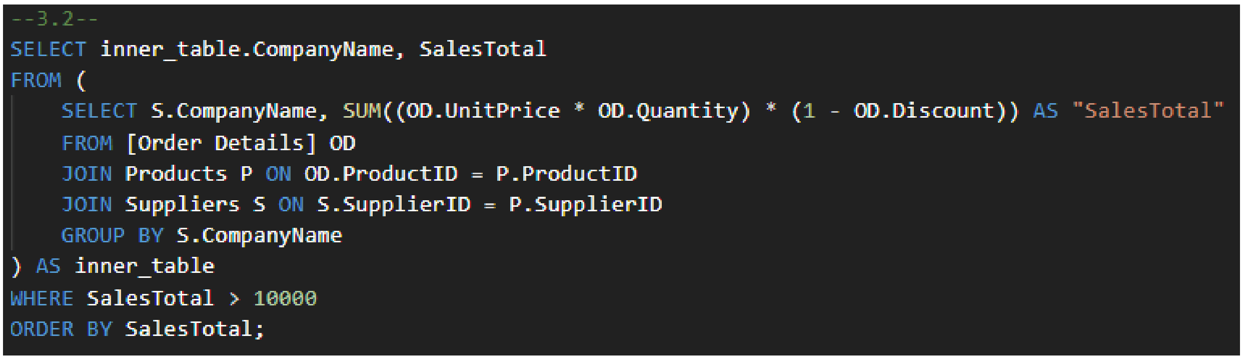
```
--3.1--
SELECT CONCAT(E.FirstName, ' ', E.LastName) AS "Employee Name", CONCAT(Managers.FirstName, ' ', Managers.LastName) AS "Reports to"
FROM Employees E
Left JOIN Employees AS "Managers" ON Managers.EmployeeID = E.ReportsTo;
```



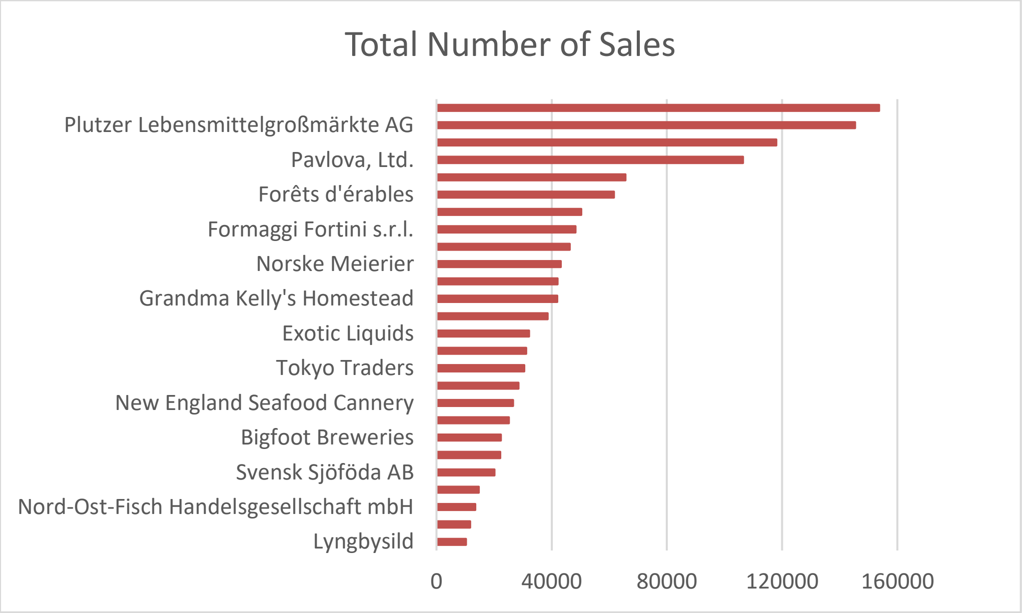
	Employee Name	Reports to
1	Nancy Davolio	Andrew Fuller
2	Andrew Fuller	
3	Janet Leverling	Andrew Fuller
4	Margaret Peacock	Andrew Fuller
5	Steven Buchanan	Andrew Fuller
6	Michael Suyama	Steven Buchanan
7	Robert King	Steven Buchanan
8	Laura Callahan	Andrew Fuller
9	Anne Dodsworth	Steven Buchanan

SQL Practical Exercise

3.2 List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table and present as a bar chart as below:

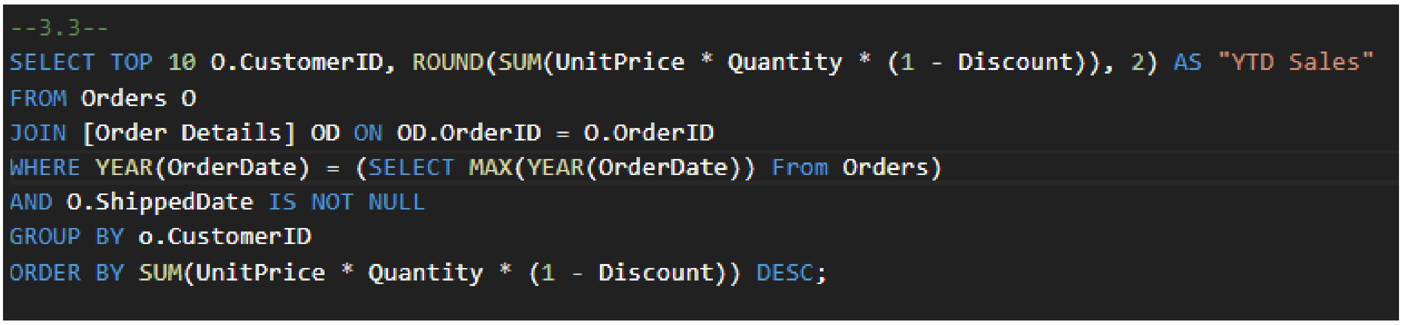


```
--3.2--
SELECT inner_table.CompanyName, SalesTotal
FROM (
    SELECT S.CompanyName, SUM((OD.UnitPrice * OD.Quantity) * (1 - OD.Discount)) AS "SalesTotal"
    FROM [Order Details] OD
    JOIN Products P ON OD.ProductID = P.ProductID
    JOIN Suppliers S ON S.SupplierID = P.SupplierID
    GROUP BY S.CompanyName
) AS inner_table
WHERE SalesTotal > 10000
ORDER BY SalesTotal;
```

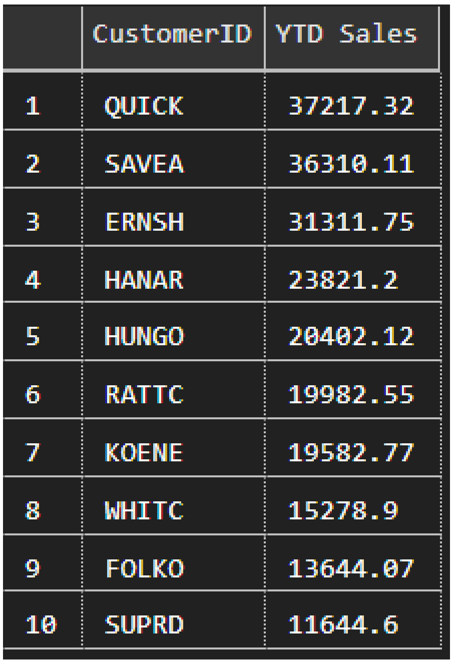


SQL Practical Exercise

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.



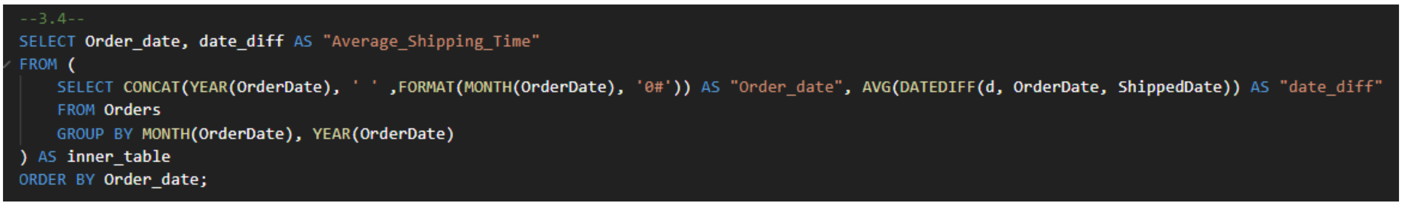
```
--3.3--
SELECT TOP 10 O.CustomerID, ROUND(SUM(UnitPrice * Quantity * (1 - Discount)), 2) AS "YTD Sales"
FROM Orders O
JOIN [Order Details] OD ON OD.OrderID = O.OrderID
WHERE YEAR(OrderDate) = (SELECT MAX(YEAR(OrderDate)) From Orders)
AND O.ShippedDate IS NOT NULL
GROUP BY o.CustomerID
ORDER BY SUM(UnitPrice * Quantity * (1 - Discount)) DESC;
```



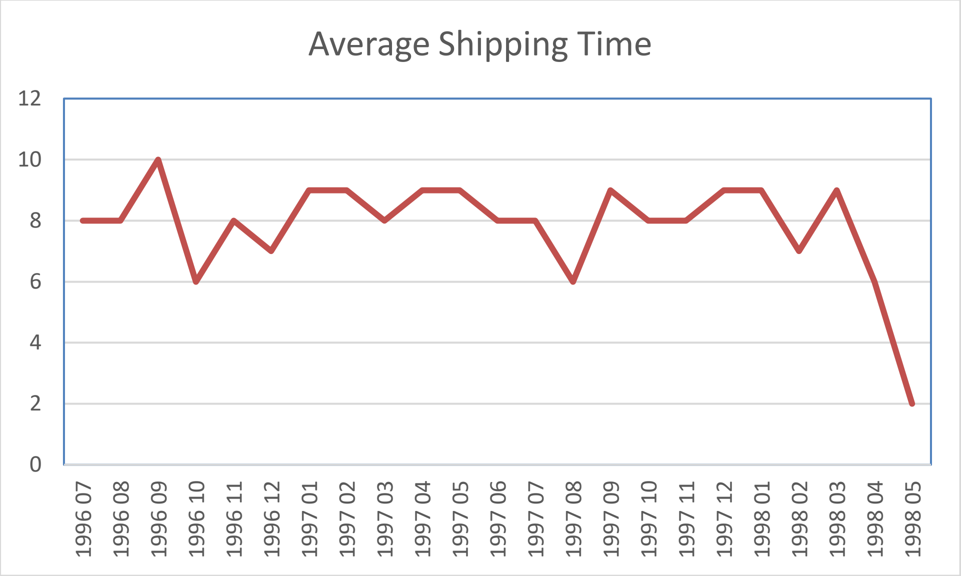
	CustomerID	YTD Sales
1	QUICK	37217.32
2	SAVEA	36310.11
3	ERNSH	31311.75
4	HANAR	23821.2
5	HUNGO	20402.12
6	RATTC	19982.55
7	KOENE	19582.77
8	WHITC	15278.9
9	FOLKO	13644.07
10	SUPRD	11644.6

SQL Practical Exercise

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



```
--3.4--
SELECT Order_date, date_diff AS "Average_Shipping_Time"
FROM (
    SELECT CONCAT(YEAR(OrderDate), ' ' ,FORMAT(MONTH(OrderDate), '0#')) AS "Order_date", AVG(DATEDIFF(d, OrderDate, ShippedDate)) AS "date_diff"
    FROM Orders
    GROUP BY MONTH(OrderDate), YEAR(OrderDate)
) AS inner_table
ORDER BY Order_date;
```


Average Shipping Time