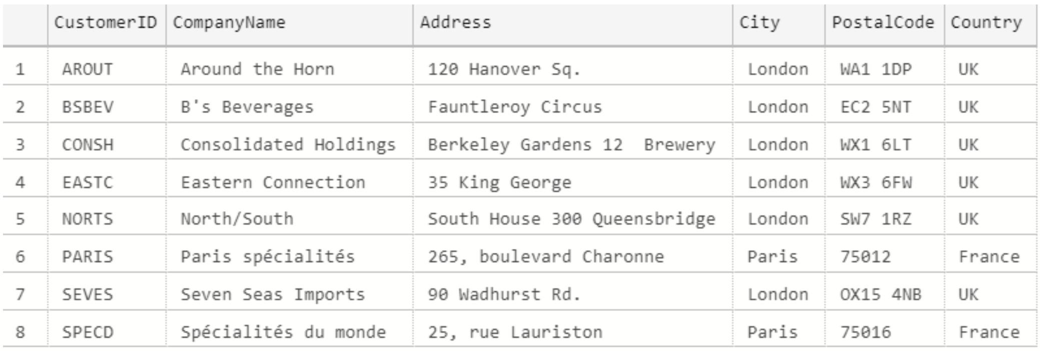
Exercise 1 – Northwind Queries (40 marks: 5 for each question)

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

Query:

```
-- Selecting CustomerID, CompanyName, Address, City, PostalCode and Country from the
Customer table
SELECT c.CustomerID,c.CompanyName,c.Address,c.City,c.PostalCode,c.Country FROM Customers c
-- Filtering to only return results where the city is Paris or London
WHERE c.City IN ('Paris','London')
```

Response:



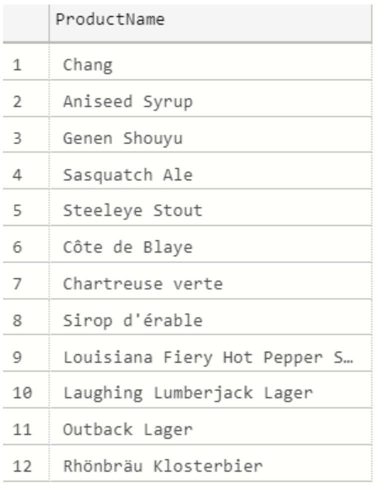
	CustomerID	CompanyName	Address	City	PostalCode	Country
1	AROUT	Around the Horn	120 Hanover Sq.	London	WA1 1DP	UK
2	BSBEV	B's Beverages	Fauntleroy Circus	London	EC2 5NT	UK
3	CONSH	Consolidated Holdings	Berkeley Gardens 12 Brewery	London	WX1 6LT	UK
4	EASTC	Eastern Connection	35 King George	London	WX3 6FW	UK
5	NORTS	North/South	South House 300 Queensbridge	London	SW7 1RZ	UK
6	PARIS	Paris spécialités	265, boulevard Charonne	Paris	75012	France
7	SEVES	Seven Seas Imports	90 Wadhurst Rd.	London	OX15 4NB	UK
8	SPECD	Spécialités du monde	25, rue Lauriston	Paris	75016	France

1.2 List all products stored in bottles.

Query:

```
-- Selecting Product Name from the Products table
SELECT ProductName FROM Products p
-- Filtering to only return results where the QuantityPerUnit field contains the word
‘bottle’
WHERE p.QuantityPerUnit LIKE '%Bottle%'
```

Response:



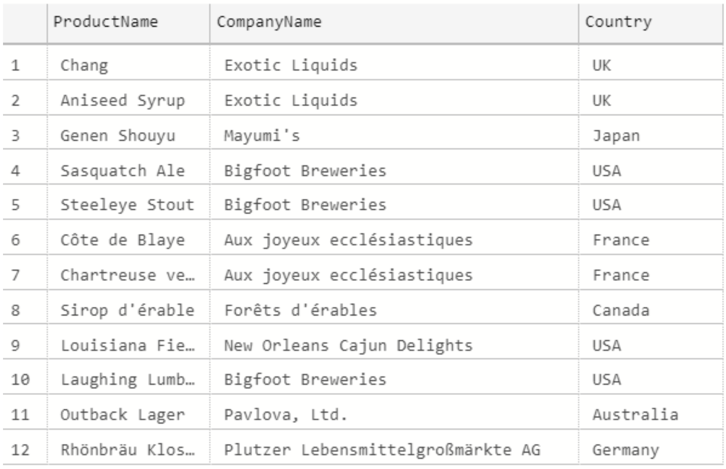
	ProductName
1	Chang
2	Aniseed Syrup
3	Genen Shouyu
4	Sasquatch Ale
5	Steeleye Stout
6	Côte de Blaye
7	Chartreuse verte
8	Sirop d'érable
9	Louisiana Fiery Hot Pepper S…
10	Laughing Lumberjack Lager
11	Outback Lager
12	Rhönbräu Klosterbier

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

Query:

```
-- Selecting Product Name, Company Name and Country from the Products table
SELECT p.ProductName, s.CompanyName, s.Country FROM Products p
-- Joining the suppliers table using the SupplierID
INNER JOIN Suppliers s ON p.SupplierID=s.SupplierID
-- Filtering to only return results where the QuantityPerUnit field contains the word
‘bottle’
WHERE p.QuantityPerUnit LIKE '%Bottle%'
```

Response:



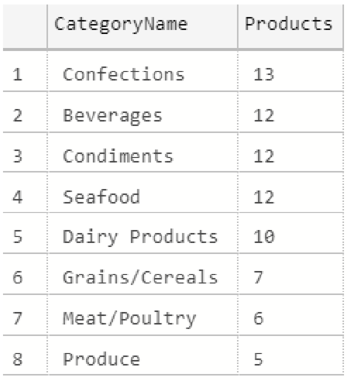
	ProductName	CompanyName	Country
1	Chang	Exotic Liquids	UK
2	Aniseed Syrup	Exotic Liquids	UK
3	Genen Shouyu	Mayumi's	Japan
4	Sasquatch Ale	Bigfoot Breweries	USA
5	Steeleye Stout	Bigfoot Breweries	USA
6	Côte de Blaye	Aux joyeux ecclésiastiques	France
7	Chartreuse ve…	Aux joyeux ecclésiastiques	France
8	Sirop d'érable	Forêts d'érables	Canada
9	Louisiana Fie…	New Orleans Cajun Delights	USA
10	Laughing Lumb…	Bigfoot Breweries	USA
11	Outback Lager	Pavlova, Ltd.	Australia
12	Rhönbräu Klos…	Plutzer Lebensmittelgroßmärkte AG	Germany

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.

Query:

```
-- Selecting Catagory Name and the product count from the Products table
SELECT c.CategoryName, COUNT(*) AS "Products" FROM Products p
JOIN Categories c ON p.CategoryID=c.CategoryID
-- Grouping the count by Category Name
GROUP BY c.CategoryName
-- Ordering the table in descending order based on the products column
ORDER BY "Products" DESC
```

Response:



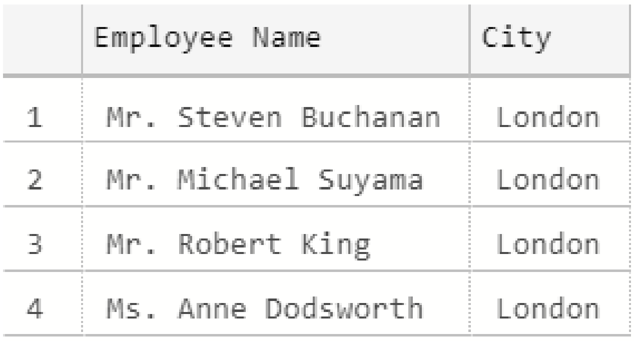
	CategoryName	Products
1	Confections	13
2	Beverages	12
3	Condiments	12
4	Seafood	12
5	Dairy Products	10
6	Grains/Cereals	7
7	Meat/Poultry	6
8	Produce	5

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.

Query:

```
-- Concatenating the required fields to get every employees full name and title from the
employees table
SELECT CONCAT(e.TitleOfCourtesy , ' ' + e.FirstName , ' ' , e.LastName) AS "Employee Name",
 e.City FROM Employees e
-- Only selecting employees from the UK
WHERE e.Country = 'UK'
```

Response



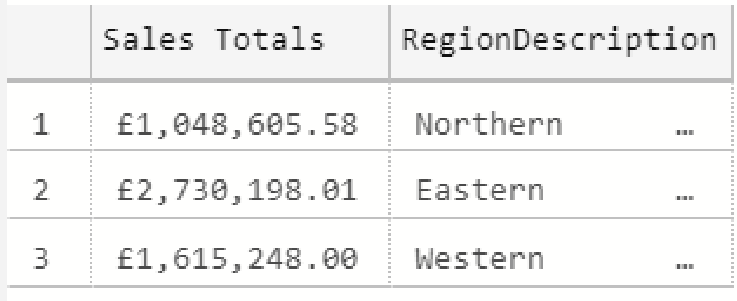
	Employee Name	City
1	Mr. Steven Buchanan	London
2	Mr. Michael Suyama	London
3	Mr. Robert King	London
4	Ms. Anne Dodsworth	London

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.

Query:

```
-- Selecting total sales, formatted as GBP and region names
SELECT FORMAT(SUM(od.Quantity*od.UnitPrice*(1-od.Discount)), 'C', 'en-
gb') AS "Sales Totals", r.RegionDescription FROM [Order Details] od
-- Joining the Order Details table to the Regions table with via 3 other tables
INNER JOIN Orders o ON od.OrderID=o.OrderID
INNER JOIN EmployeeTerritories et ON o.EmployeeID=et.EmployeeID
INNER JOIN Territories t ON t.TerritoryID=et.TerritoryID
INNER JOIN Region r ON r.RegionID=t.RegionID
-- Grouping sales by the Region Description
GROUP BY r.RegionDescription
-- Only selecting rows where sales exceed £1000000
HAVING SUM(od.Quantity*(od.UnitPrice-od.Discount)) > 1000000
```

Response:



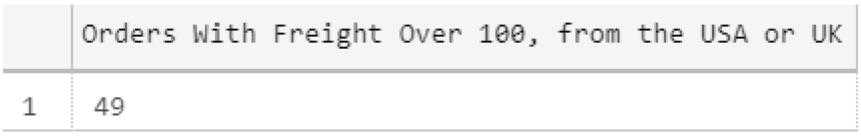
	Sales Totals	RegionDescription
1	£1,048,605.58	Northern …
2	£2,730,198.01	Eastern …
3	£1,615,248.00	Western …

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

Query:

```
-- Counting all orders
SELECT COUNT(*) AS "Orders With Freight Over 100, from the USA or UK" FROM Orders o
-- Filtering to only include orders where the freight is over 100 and the ship country is
USA or UK
WHERE o.Freight > 100 AND ShipCountry IN ('USA','UK')
```

Response:



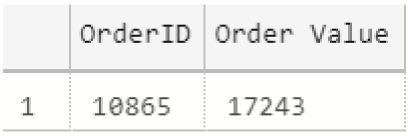
	Orders With Freight Over 100, from the USA or UK
1	49

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.

Query:

```
-- Selecting only the top value for OrderID and Sale Value
SELECT TOP 1 od.OrderID, ROUND(SUM(od.Quantity*(od.UnitPrice-
od.Discount)),2) AS "Order Value"FROM [Order Details] od
-- Grouping by order
GROUP BY od.OrderID
-- Ordering the list from highest to lowest
ORDER BY "Order Value" DESC
```

Response:



	OrderID	Order Value
1	10865	17243

Exercise 2 – Create Spartans Table (20 marks – 10 each)

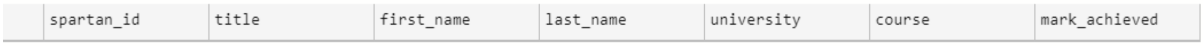
2.1 Write the correct SQL statement to create the following table:

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.

Statement:

```
-- Creating a new table containing the required fields, ensuring the correct datatype and
not accepting NULL values
CREATE TABLE spartans
(
    spartan_id INT IDENTITY PRIMARY KEY,
    title VARCHAR(4) NOT NULL,
    first_name VARCHAR(16) NOT NULL,
    last_name VARCHAR(16) NOT NULL,
    university VARCHAR(50) NOT NULL,
    course VARCHAR(50) NOT NULL,
    mark_achieved VARCHAR(50) NOT NULL,
)
```

Result:

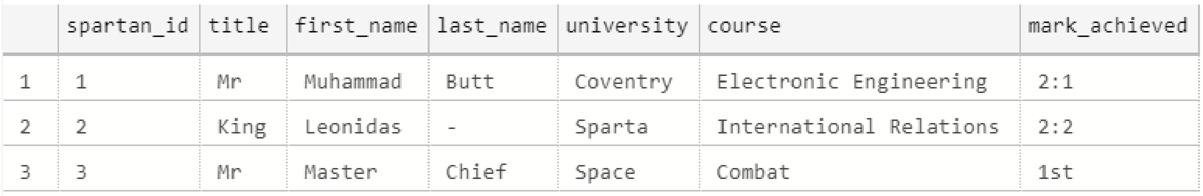


	spartan_id	title	first_name	last_name	university	course	mark_achieved

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

```
-- Inserting rows. Inserting data in the correct order
INSERT INTO spartans (title, first_name, last_name, university, course, mark_achieved)
VALUES
    ('Mr', 'Muhammad', 'Butt', 'Coventry', 'Electronic Engineering', '2:1'),
    ('King', 'Leonidas', '-', 'Sparta', 'International Relations', '2:2'),
    ('Mr', 'Master', 'Chief', 'Space', 'Combat', '1st')
```

Result:



	spartan_id	title	first_name	last_name	university	course	mark_achieved
1	1	Mr	Muhammad	Butt	Coventry	Electronic Engineering	2:1
2	2	King	Leonidas	-	Sparta	International Relations	2:2
3	3	Mr	Master	Chief	Space	Combat	1st

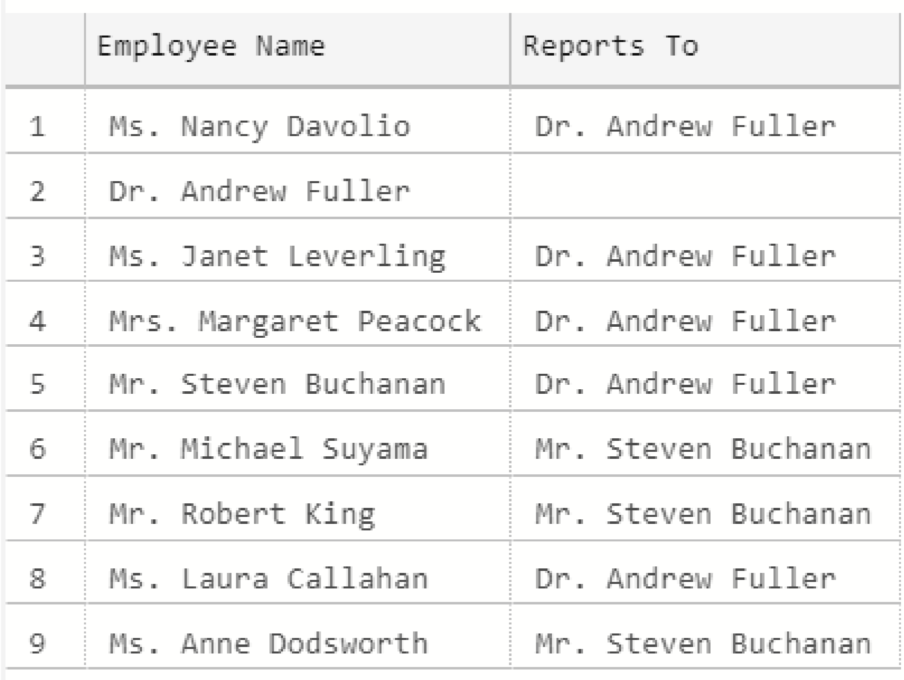
Exercise 3 – Northwind Data Analysis linked to Excel (30 marks)

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. (5 Marks)

Query:

```
-- Selecting and concatenating Employee name and Reports to names with aliasing
SELECT CONCAT(e.TitleOfCourtesy , ' ' , e.FirstName , ' '  , e.LastName) AS "Employee Name"
 , CONCAT(en.TitleOfCourtesy , ' '  , en.FirstName , ' ' , en.LastName) AS "Reports To"
FROM Employees e
-- Self join connecting ReportsTo to EmployeeID to get a name for ReportsTo
LEFT JOIN Employees en ON e.ReportsTo=en.EmployeeID
```

Response:



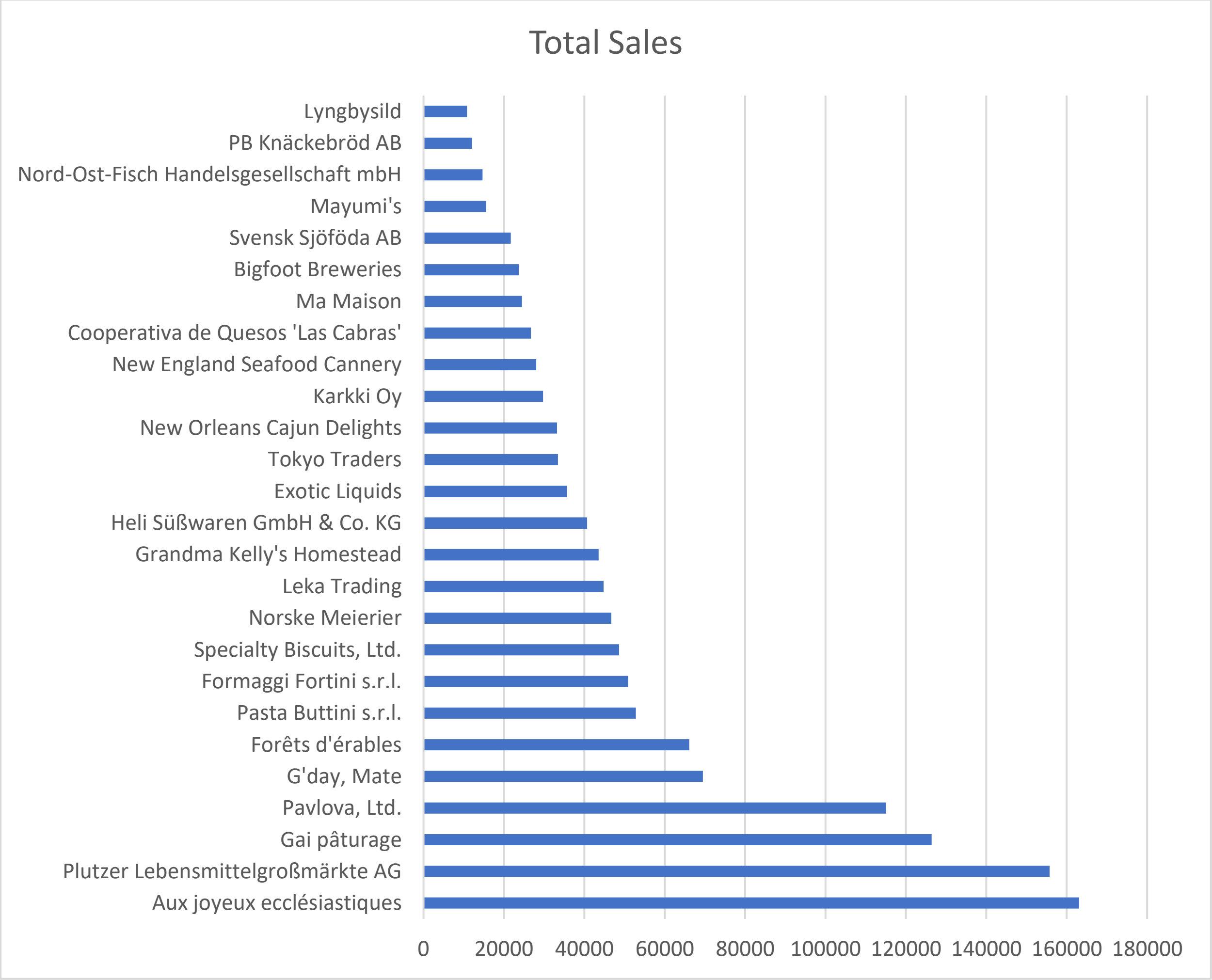
	Employee Name	Reports To
1	Ms. Nancy Davolio	Dr. Andrew Fuller
2	Dr. Andrew Fuller	
3	Ms. Janet Leverling	Dr. Andrew Fuller
4	Mrs. Margaret Peacock	Dr. Andrew Fuller
5	Mr. Steven Buchanan	Dr. Andrew Fuller
6	Mr. Michael Suyama	Mr. Steven Buchanan
7	Mr. Robert King	Mr. Steven Buchanan
8	Ms. Laura Callahan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

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: (5 Marks)

Query:

```
-- Selecting Company name and total sales
SELECT s.CompanyName, SUM((od.UnitPrice -
 od.Discount)*od.Quantity) AS "Total Sales" FROM Suppliers s
-- Joining Order Details table via Products table
INNER JOIN Products p ON s.SupplierID = p.SupplierID
INNER JOIN [Order Details] od ON p.ProductID = od.ProductID
-- Grouping by company name to see how many sales they have
GROUP BY s.CompanyName
-- Filtering out results with less than 10000 in sales
HAVING SUM((od.UnitPrice - od.Discount)*od.Quantity) > 10000
-- Ordering the table from highest sales value to lowest
ORDER BY "Total Sales" DESC
```

Result:

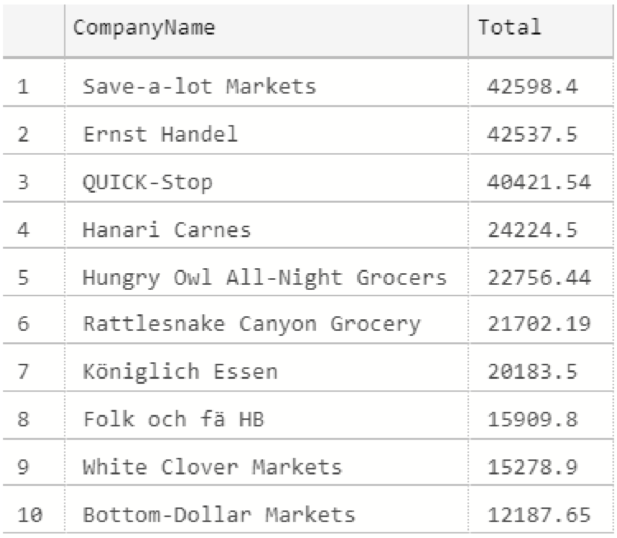


3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped. No Excel required. (10 Marks)

Query:

```
-- Selecting top 10 results
SELECT TOP 10 c.CompanyName, ROUND(SUM(od.Quantity*(od.UnitPrice-
od.Discount)),2) AS "Total" FROM Orders o
-- Joining the order details and Customers table to the Orders table
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
-- Filtering to results only in the latest year
-- Finding latest year with a subquery
WHERE YEAR(o.OrderDate)=(SELECT MAX(YEAR(o.OrderDate)) FROM Orders o)
-- Grouping by Company Name
GROUP BY c.CompanyName
-- Ordering to have the highest total first and the rest descending
ORDER BY "Total" DESC
```

Response:



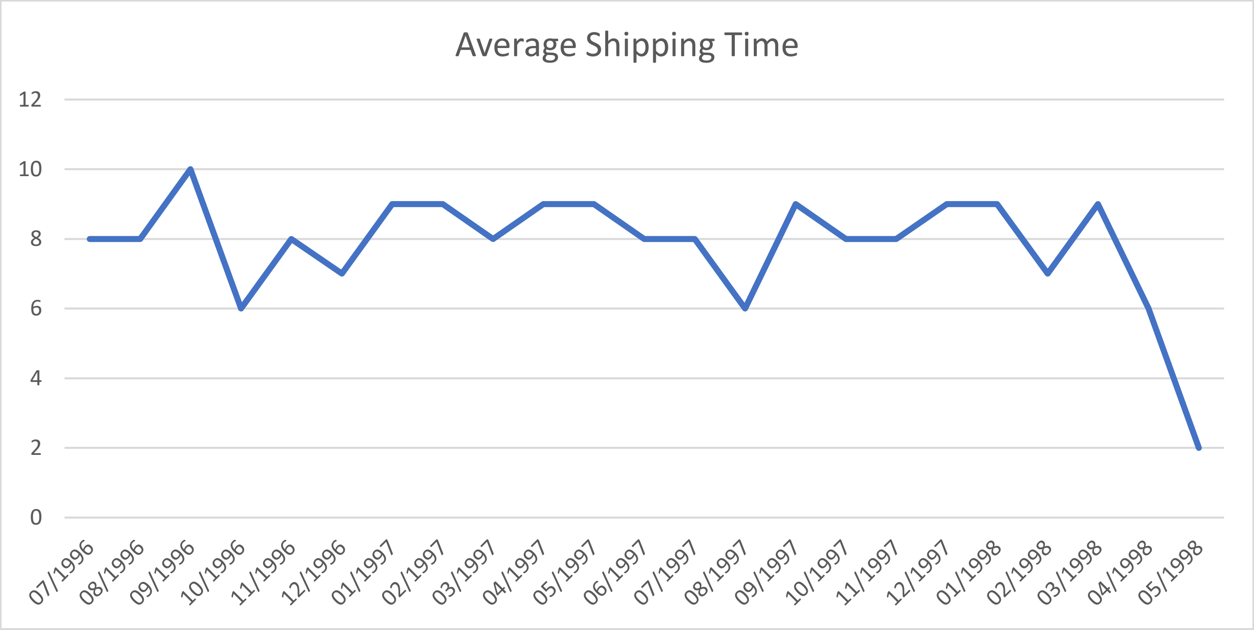
	CompanyName	Total
1	Save-a-lot Markets	42598.4
2	Ernst Handel	42537.5
3	QUICK-Stop	40421.54
4	Hanari Carnes	24224.5
5	Hungry Owl All-Night Grocers	22756.44
6	Rattlesnake Canyon Grocery	21702.19
7	Königlich Essen	20183.5
8	Folk och fä HB	15909.8
9	White Clover Markets	15278.9
10	Bottom-Dollar Markets	12187.65

3.4 Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below. (10 Marks)

Query:

```
-- Selecting the date in the desired format and the average shipping time from the subquery
SELECT FORMAT("Date",'MM/yyyy') AS "Date",
 AVG("Shipping Time") AS "Average Shipping Time"
-- Subquery to find the average shipping time
FROM (SELECT o.OrderDate AS "Date", MONTH(o.OrderDate) AS "Month", YEAR(o.OrderDate) AS "Ye
ar",
DATEDIFF(DAY, o.OrderDate, o.ShippedDate) AS "Shipping Time"
FROM Orders o) AS "Shipping Times"
-- Grouping shipping times by month
GROUP BY FORMAT("Date",'MM/yyyy'), "Month", "Year"
-- Ordering chronologically, first by year and then month
ORDER BY "Year" ASC, "Month" ASC
```

Response:


Average Shipping Time
12
10
8
6
4
2
0
07/1996
08/1996
09/1996
10/1996
11/1996
12/1996
01/1997
02/1997
03/1997
04/1997
05/1997
06/1997
07/1997
08/1997
09/1997
10/1997
11/1997
12/1997
01/1998
02/1998
03/1998
04/1998
05/1998