SQL Mini Project

Jamie Hammond

Exercise 1 – Northwind Queries

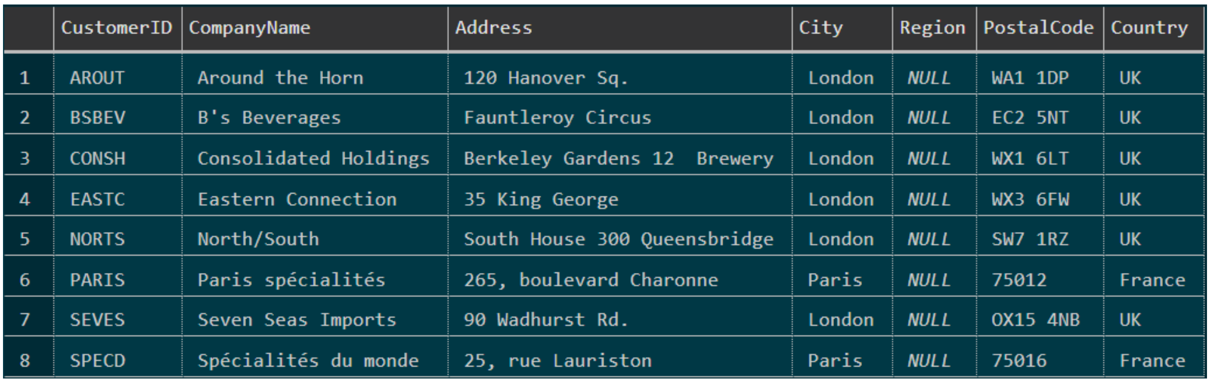
1.1

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

Code:

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

Results:



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

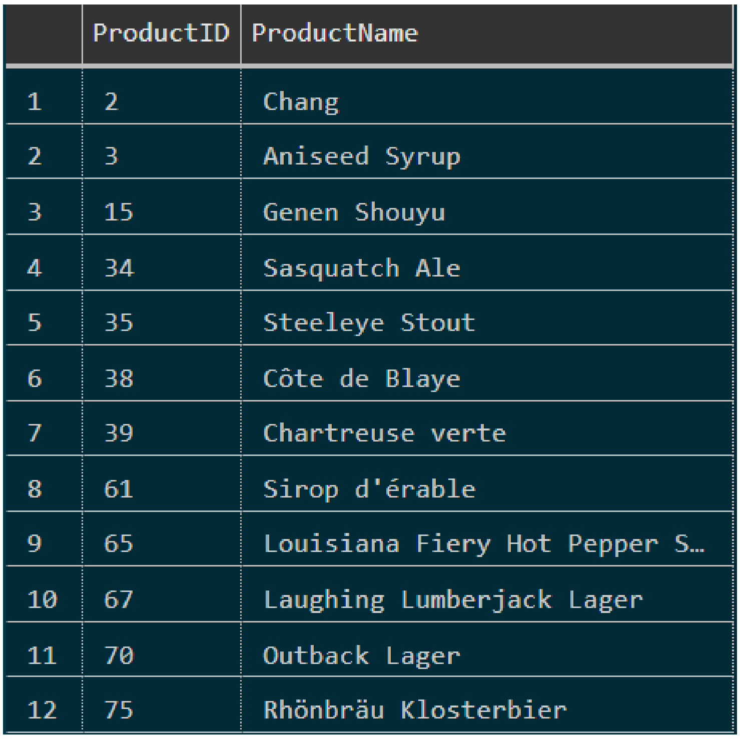
1.2

List all products stored in bottles.

Code:

```sql
SELECT ProductID, ProductName
FROM Products
WHERE CHARINDEX('bottle', QuantityPerUnit) > 0;
```

Results:

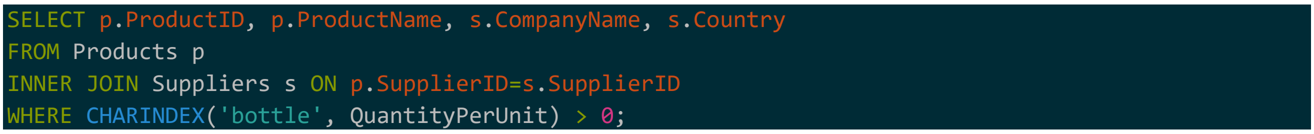


	ProductID	ProductName
1	2	Chang
2	3	Aniseed Syrup
3	15	Genen Shouyu
4	34	Sasquatch Ale
5	35	Steeleye Stout
6	38	Côte de Blaye
7	39	Chartreuse verte
8	61	Sirop d'érable
9	65	Louisiana Fiery Hot Pepper S…
10	67	Laughing Lumberjack Lager
11	70	Outback Lager
12	75	Rhönbräu Klosterbier

1.3

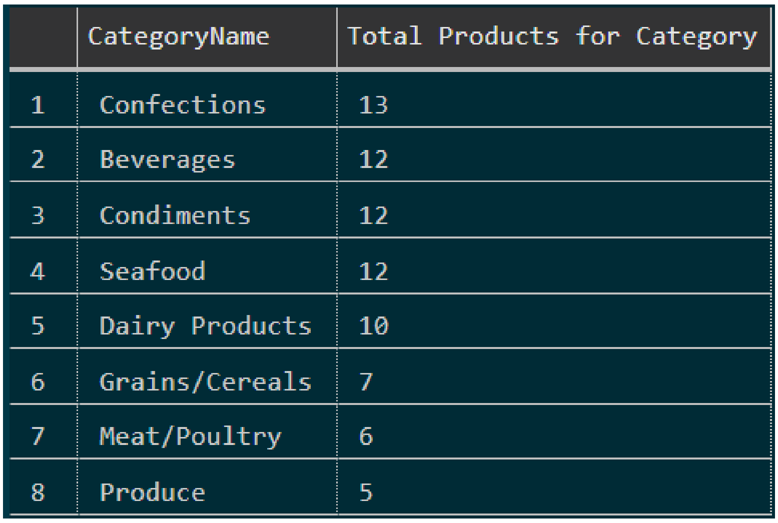
Repeat question above, but add in the Supplier Name and Country.

Code:



```
SELECT p.ProductID, p.ProductName, s.CompanyName, s.Country
FROM Products p
INNER JOIN Suppliers s ON p.SupplierID=s.SupplierID
WHERE CHARINDEX('bottle', QuantityPerUnit) > 0;
```

Results:

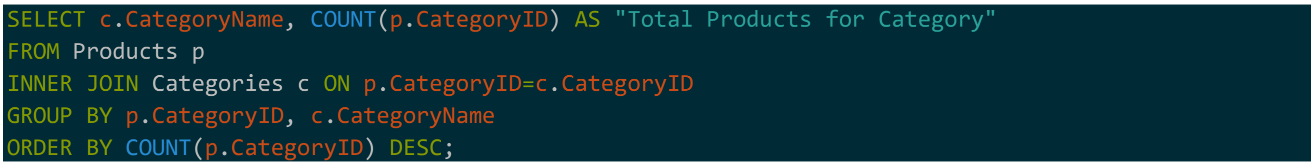


	CategoryName	Total Products for Category
1	Confections	13
2	Beverages	12
3	Condiments	12
4	Seafood	12
5	Dairy Products	10
6	Grains/Cereals	7
7	Meat/Poultry	6
8	Produce	5

1.4

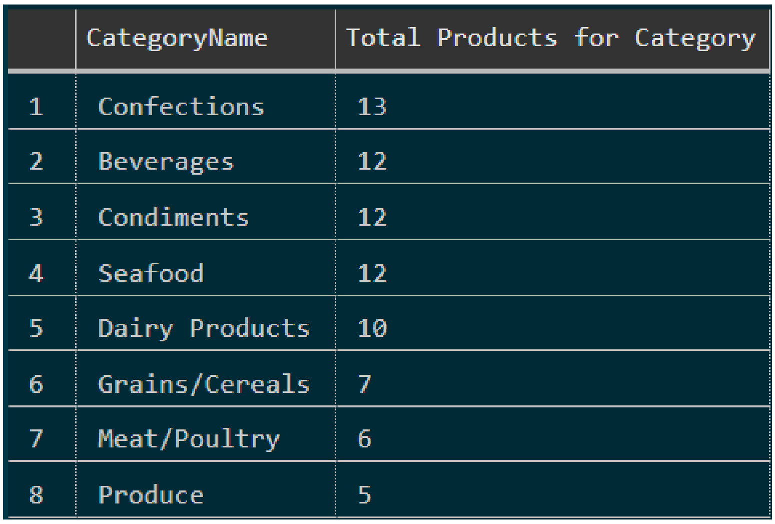
Write an SQL Statement that shows how many products there are in each category. Include Category Name in result set and list the highest number first.

Code:



```
SELECT c.CategoryName, COUNT(p.CategoryID) AS "Total Products for Category"
FROM Products p
INNER JOIN Categories c ON p.CategoryID=c.CategoryID
GROUP BY p.CategoryID, c.CategoryName
ORDER BY COUNT(p.CategoryID) DESC;
```

Results:

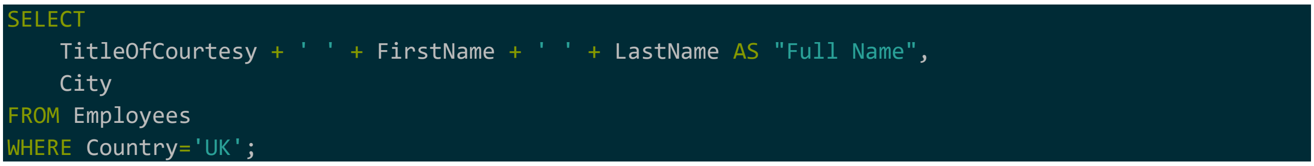


	CategoryName	Total Products for Category
1	Confections	13
2	Beverages	12
3	Condiments	12
4	Seafood	12
5	Dairy Products	10
6	Grains/Cereals	7
7	Meat/Poultry	6
8	Produce	5

1.5

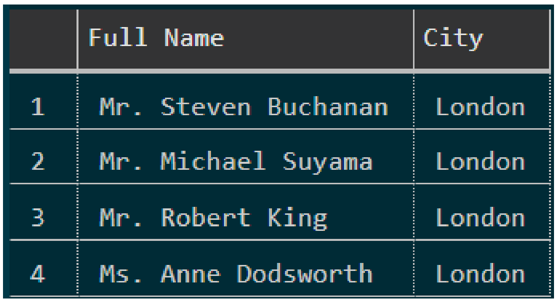
List all UK employees using concatenation to join their title of courtesy, first name and last name together. Also include their city of residence.

Code:



```
SELECT
    TitleOfCourtesy + ' ' + FirstName + ' ' + LastName AS "Full Name",
    City
FROM Employees
WHERE Country='UK';
```

Results:

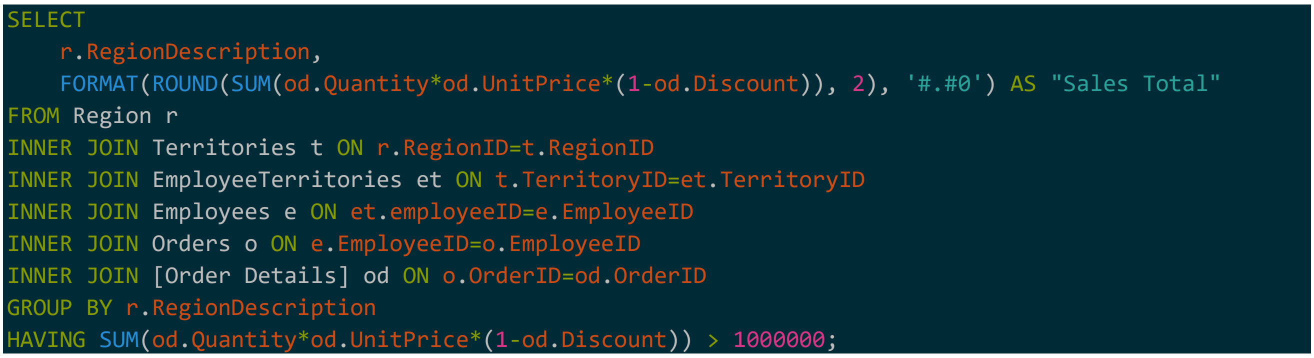


	Full Name	City
1	Mr. Steven Buchanan	London
2	Mr. Michael Suyama	London
3	Mr. Robert King	London
4	Ms. Anne Dodsworth	London

1.6

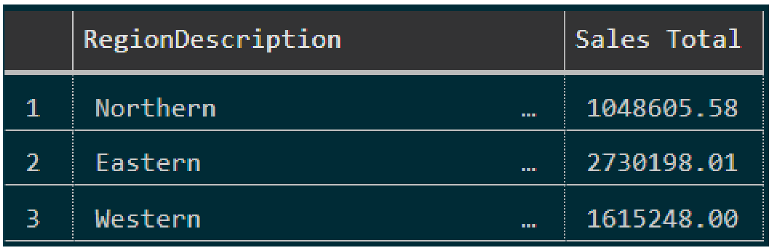
List Sales Totals for all Sales Regions (via the Territories table using 4 joins) with a Sales Total greater than 1,000,000. Use rounding or FORMAT to present the numbers.

Code:



```
SELECT
    r.RegionDescription,
    FORMAT(ROUND(SUM(od.Quantity*od.UnitPrice*(1-od.Discount)), 2), '#.#0') AS "Sales Total"
FROM Region r
INNER JOIN Territories t ON r.RegionID=t.RegionID
INNER JOIN EmployeeTerritories et ON t.TerritoryID=et.TerritoryID
INNER JOIN Employees e ON et.employeeID=e.EmployeeID
INNER JOIN Orders o ON e.EmployeeID=o.EmployeeID
INNER JOIN [Order Details] od ON o.OrderID=od.OrderID
GROUP BY r.RegionDescription
HAVING SUM(od.Quantity*od.UnitPrice*(1-od.Discount)) > 1000000;
```

Results:

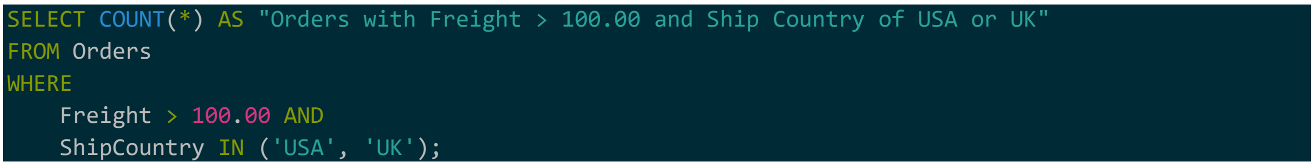


	RegionDescription	Sales Total
1	Northern …	1048605.58
2	Eastern …	2730198.01
3	Western …	1615248.00

1.7

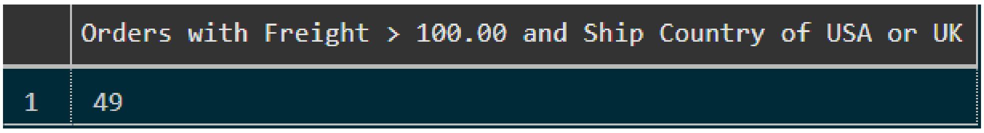
Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.

Code:



```
SELECT COUNT(*) AS "Orders with Freight > 100.00 and Ship Country of USA or UK"
FROM Orders
WHERE
    Freight > 100.00 AND
    ShipCountry IN ('USA', 'UK');
```

Results:

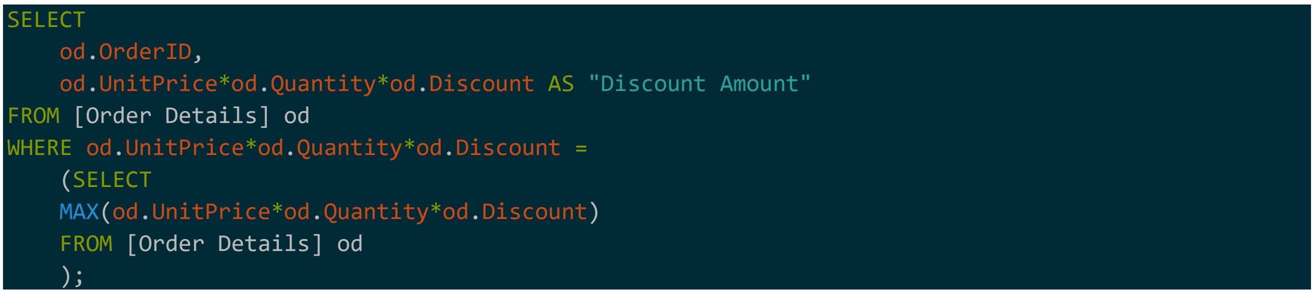


	Orders with Freight > 100.00 and Ship Country of USA or UK
1	49

1.8

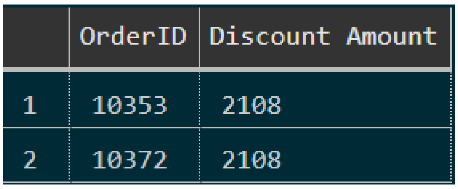
Write an SQL Statement to identify the Order Number of the Order with the highest amount(value) of discount applied to that order.

Code:



```
SELECT
    od.OrderID,
    od.UnitPrice*od.Quantity*od.Discount AS "Discount Amount"
FROM [Order Details] od
WHERE od.UnitPrice*od.Quantity*od.Discount =
    (SELECT
    MAX(od.UnitPrice*od.Quantity*od.Discount)
    FROM [Order Details] od
    );
```

Results:



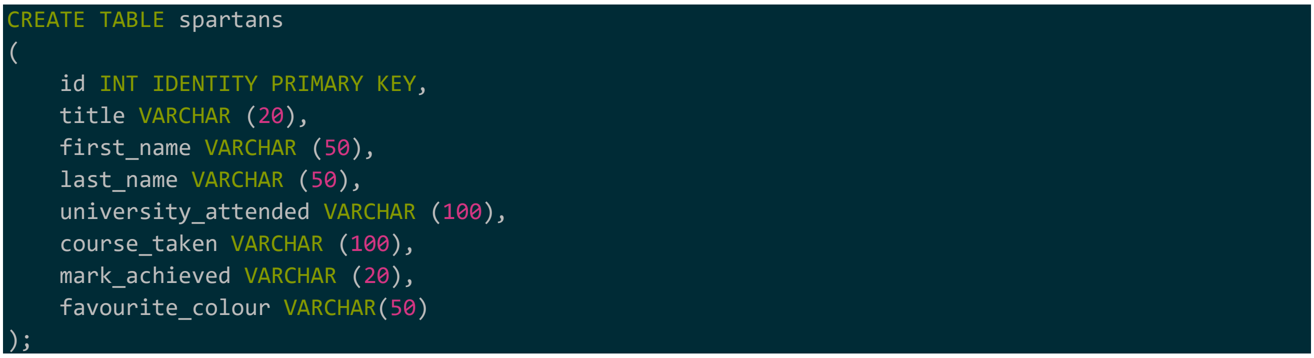
	OrderID	Discount Amount
1	10353	2108
2	10372	2108

Exercise 2 – Create Spartans Table

2.1

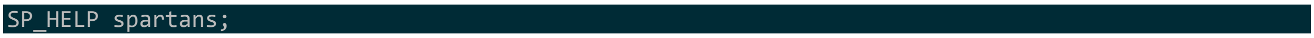
Write the correct SQL statement to create the following table: Spartans Table – include details about all the Spartans on this course. Separate Title, First Name and Last Name into separate columns, and include University attended, course taken and mark achieved. Add any other columns you feel would be appropriate.

Code:

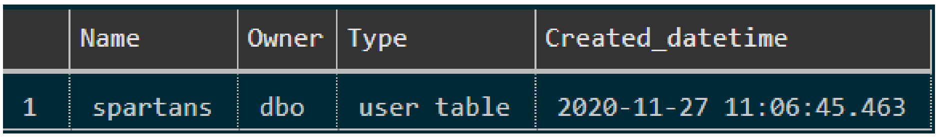


```
CREATE TABLE spartans
(
    id INT IDENTITY PRIMARY KEY,
    title VARCHAR (20),
    first_name VARCHAR (50),
    last_name VARCHAR (50),
    university_attended VARCHAR (100),
    course_taken VARCHAR (100),
    mark_achieved VARCHAR (20),
    favourite_colour VARCHAR(50)
);
```

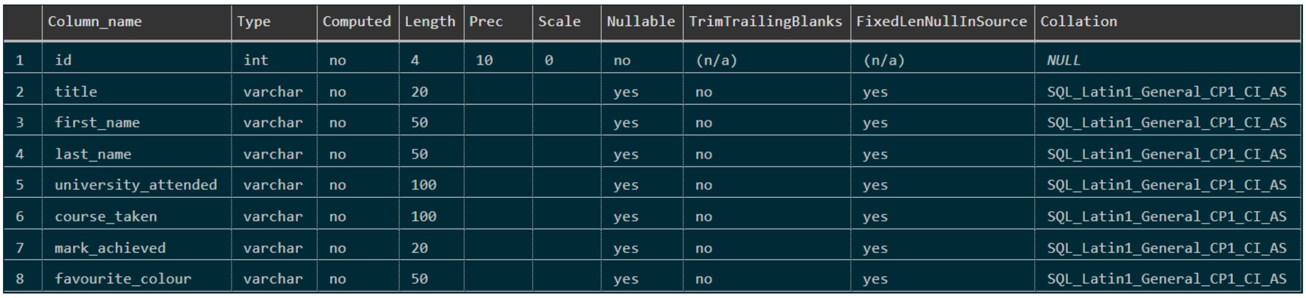
Results:



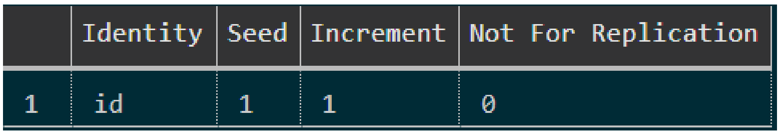
```
SP_HELP spartans;
```



	Name	Owner	Type	Created_datetime
1	spartans	dbo	user table	2020-11-27 11:06:45.463



	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	id	int	no	4	10	0	no	(n/a)	(n/a)	NULL
2	title	varchar	no	20			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
3	first_name	varchar	no	50			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
4	last_name	varchar	no	50			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
5	university_attended	varchar	no	100			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
6	course_taken	varchar	no	100			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
7	mark_achieved	varchar	no	20			yes	no	yes	SQL_Latin1_General_CP1_CI_AS
8	favourite_colour	varchar	no	50			yes	no	yes	SQL_Latin1_General_CP1_CI_AS



	Identity	Seed	Increment	Not For Replication
1	id	1	1	0

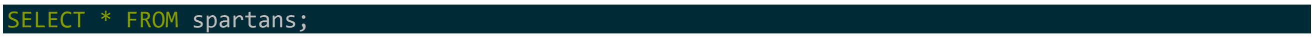
2.2
Write SQL statements to add the details of the Spartans in your course to the table you have created.

Code:

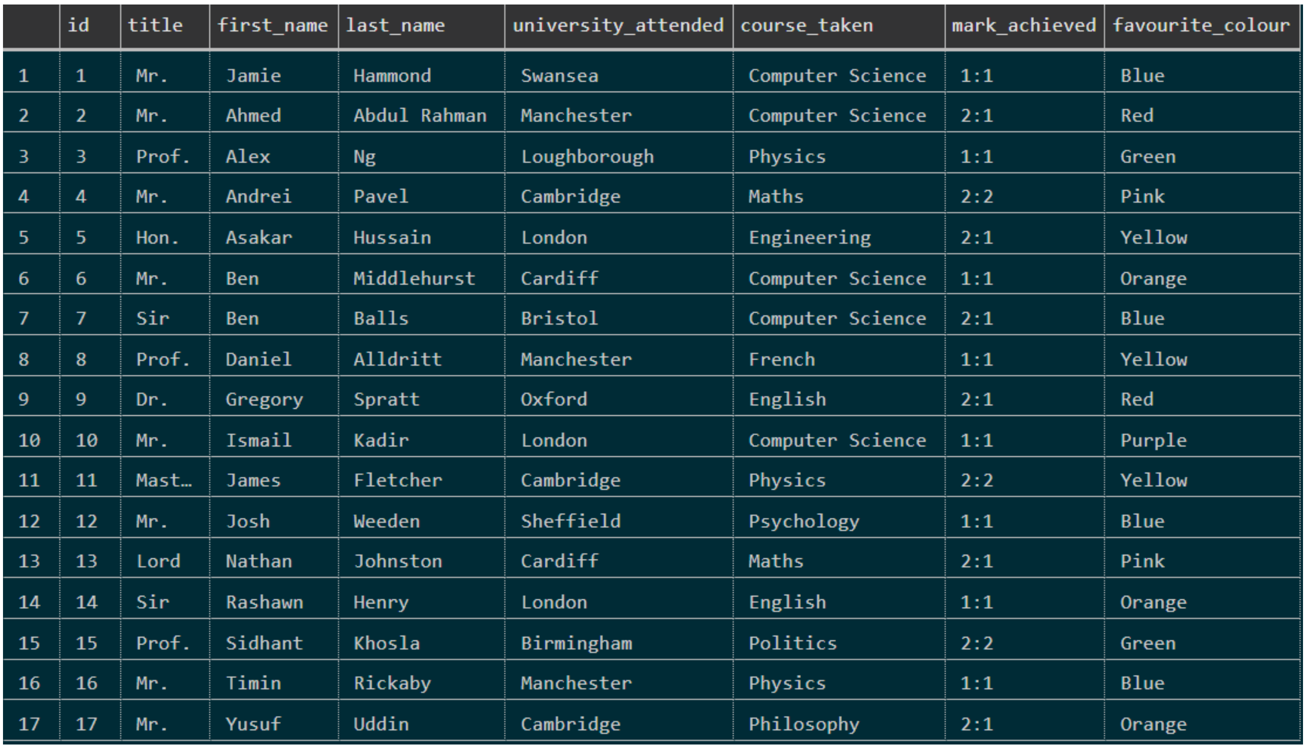
```
INSERT INTO spartans
VALUES
(
    'Mr.',
    'Jamie',
    'Hammond',
    'Swansea',
    'Computer Science',
    '1:1',
    'Blue'
);

INSERT INTO spartans
VALUES
(
    'Mr.',
    'Ahmed',
    'Abdul Rahman',
    'Manchester',
    'Computer Science',
    '2:1',
    'Red'
);

INSERT INTO spartans
VALUES
(
    'Prof.',
    'Alex',
    'Ng',
    'Loughborough',
    'Physics',
    '1:1',
    'Green'
);
...
```

Not shown are several more entries of the same format, for the sake of saving space.

Results:



```
SELECT * FROM spartans;
```



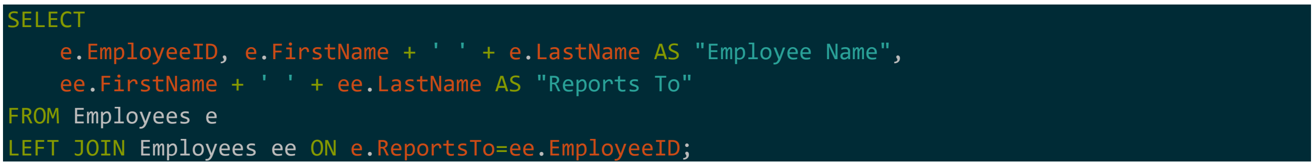
	id	title	first_name	last_name	university_attended	course_taken	mark_achieved	favourite_colour
1	1	Mr.	Jamie	Hammond	Swansea	Computer Science	1:1	Blue
2	2	Mr.	Ahmed	Abdul Rahman	Manchester	Computer Science	2:1	Red
3	3	Prof.	Alex	Ng	Loughborough	Physics	1:1	Green
4	4	Mr.	Andrei	Pavel	Cambridge	Maths	2:2	Pink
5	5	Hon.	Asakar	Hussain	London	Engineering	2:1	Yellow
6	6	Mr.	Ben	Middlehurst	Cardiff	Computer Science	1:1	Orange
7	7	Sir	Ben	Balls	Bristol	Computer Science	2:1	Blue
8	8	Prof.	Daniel	Alldritt	Manchester	French	1:1	Yellow
9	9	Dr.	Gregory	Spratt	Oxford	English	2:1	Red
10	10	Mr.	Ismail	Kadir	London	Computer Science	1:1	Purple
11	11	Mast…	James	Fletcher	Cambridge	Physics	2:2	Yellow
12	12	Mr.	Josh	Weeden	Sheffield	Psychology	1:1	Blue
13	13	Lord	Nathan	Johnston	Cardiff	Maths	2:1	Pink
14	14	Sir	Rashawn	Henry	London	English	1:1	Orange
15	15	Prof.	Sidhant	Khosla	Birmingham	Politics	2:2	Green
16	16	Mr.	Timin	Rickaby	Manchester	Physics	1:1	Blue
17	17	Mr.	Yusuf	Uddin	Cambridge	Philosophy	2:1	Orange

Exercise 3 – Northwind Data Analysis Linked to Excel

3.1

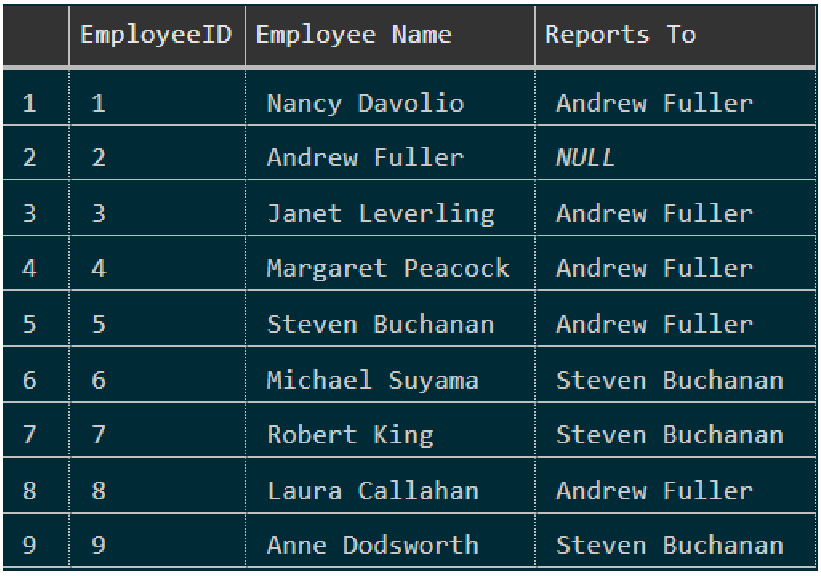
List all Employees from the Employees table and who they report to. No Excel required.

Code:



```
SELECT
    e.EmployeeID, e.FirstName + ' ' + e.LastName AS "Employee Name",
    ee.FirstName + ' ' + ee.LastName AS "Reports To"
FROM Employees e
LEFT JOIN Employees ee ON e.ReportsTo=ee.EmployeeID;
```

Results:

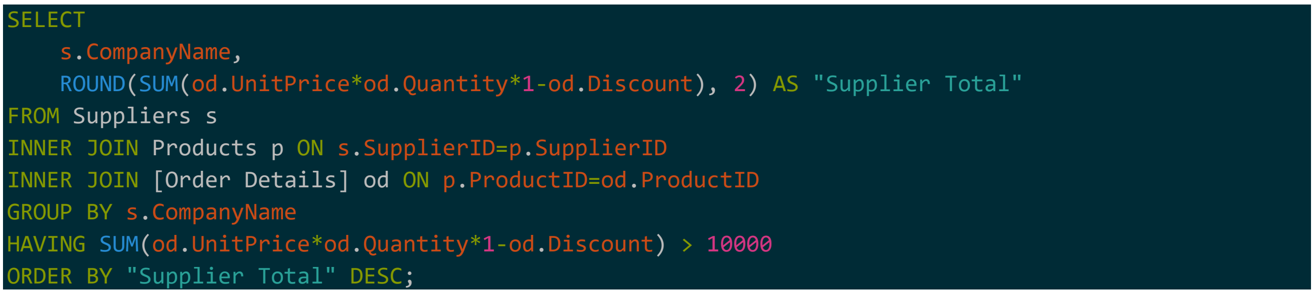


	EmployeeID	Employee Name	Reports To
1	1	Nancy Davolio	Andrew Fuller
2	2	Andrew Fuller	*NULL*
3	3	Janet Leverling	Andrew Fuller
4	4	Margaret Peacock	Andrew Fuller
5	5	Steven Buchanan	Andrew Fuller
6	6	Michael Suyama	Steven Buchanan
7	7	Robert King	Steven Buchanan
8	8	Laura Callahan	Andrew Fuller
9	9	Anne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

3.2

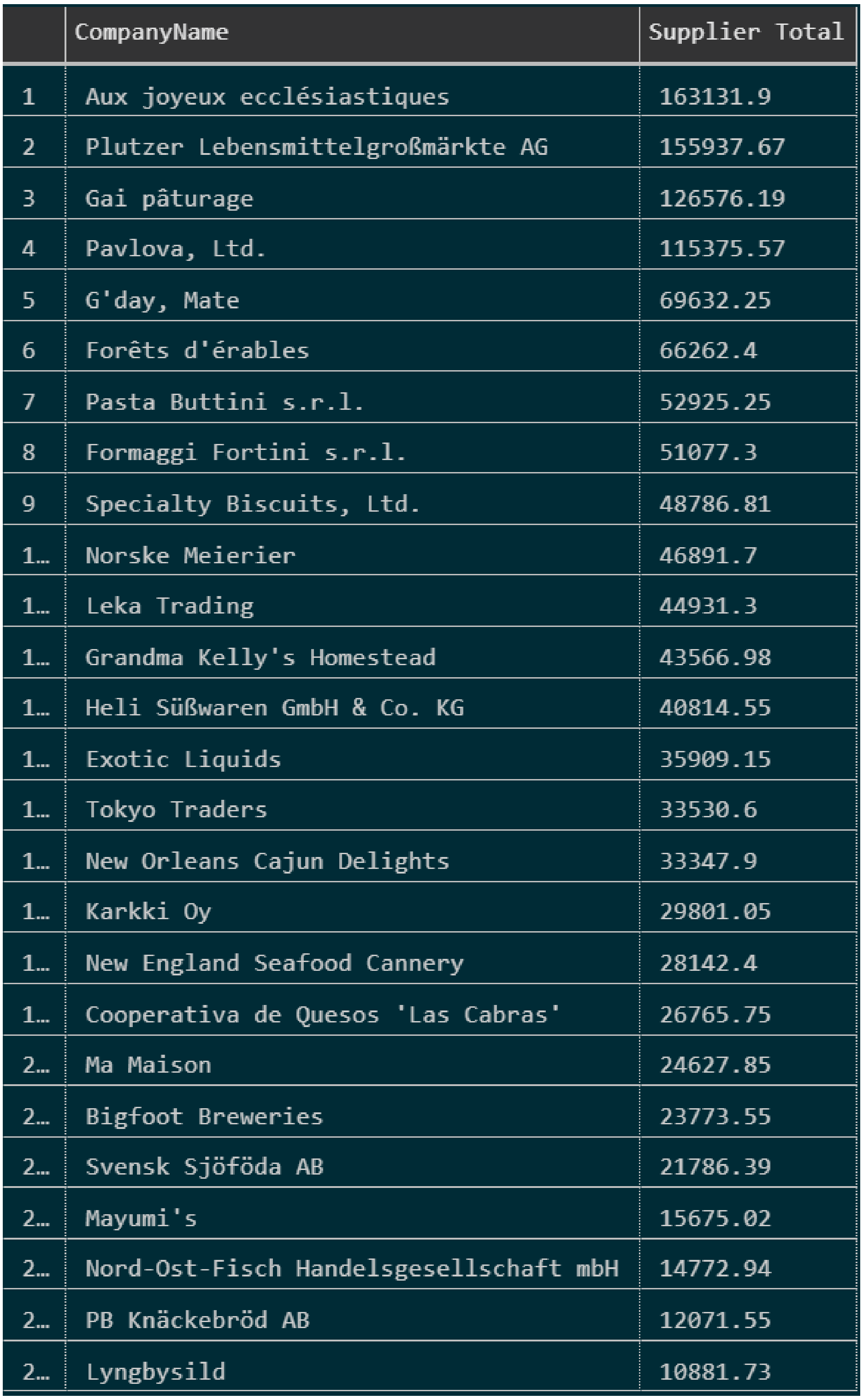
List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table and present as a bar chart as below.

Code:

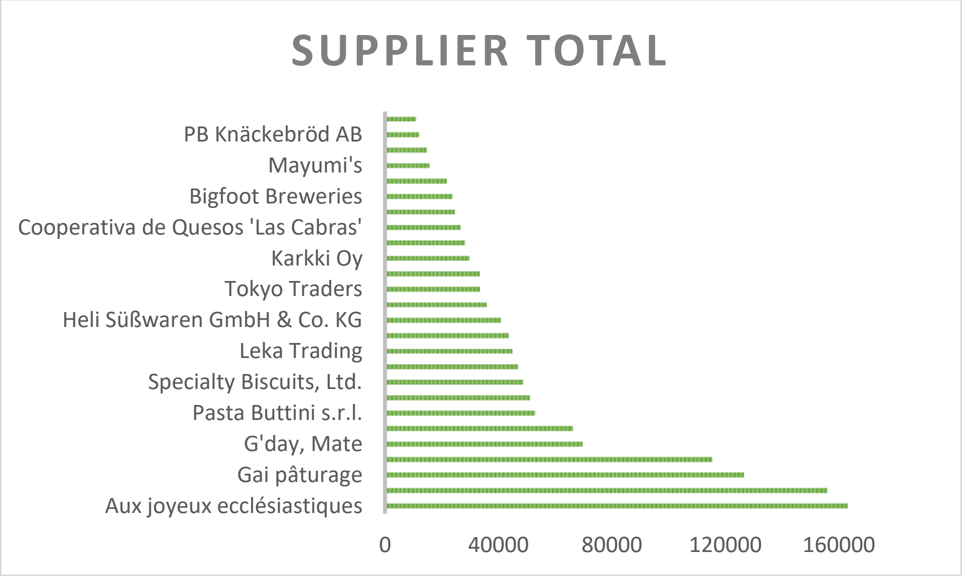


```
SELECT
    s.CompanyName,
    ROUND(SUM(od.UnitPrice*od.Quantity*1-od.Discount), 2) AS "Supplier Total"
FROM Suppliers s
INNER JOIN Products p ON s.SupplierID=p.SupplierID
INNER JOIN [Order Details] od ON p.ProductID=od.ProductID
GROUP BY s.CompanyName
HAVING SUM(od.UnitPrice*od.Quantity*1-od.Discount) > 10000
ORDER BY "Supplier Total" DESC;
```

Results:



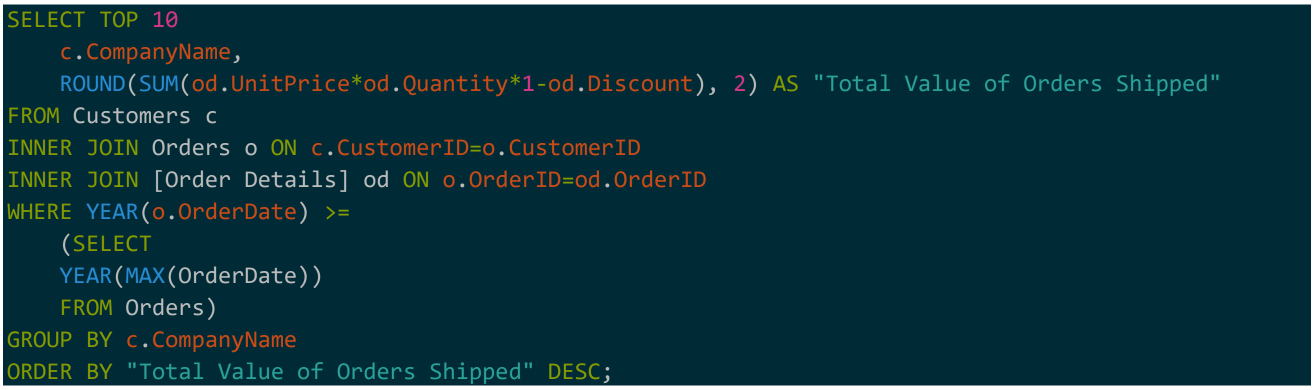
	CompanyName	Supplier Total
1	Aux joyeux ecclésiastiques	163131.9
2	Plutzer Lebensmittelgroßmärkte AG	155937.67
3	Gai pâturage	126576.19
4	Pavlova, Ltd.	115375.57
5	G'day, Mate	69632.25
6	Forêts d'érables	66262.4
7	Pasta Buttini s.r.l.	52925.25
8	Formaggi Fortini s.r.l.	51077.3
9	Specialty Biscuits, Ltd.	48786.81
1...	Norske Meierier	46891.7
1...	Leka Trading	44931.3
1...	Grandma Kelly's Homestead	43566.98
1...	Heli Süßwaren GmbH & Co. KG	40814.55
1...	Exotic Liquids	35909.15
1...	Tokyo Traders	33530.6
1...	New Orleans Cajun Delights	33347.9
1...	Karkki Oy	29801.05
1...	New England Seafood Cannery	28142.4
1...	Cooperativa de Quesos 'Las Cabras'	26765.75
2...	Ma Maison	24627.85
2...	Bigfoot Breweries	23773.55
2...	Svensk Sjöföda AB	21786.39
2...	Mayumi's	15675.02
2...	Nord-Ost-Fisch Handelsgesellschaft mbH	14772.94
2...	PB Knäckebröd AB	12071.55
2...	Lyngbysild	10881.73



3.3

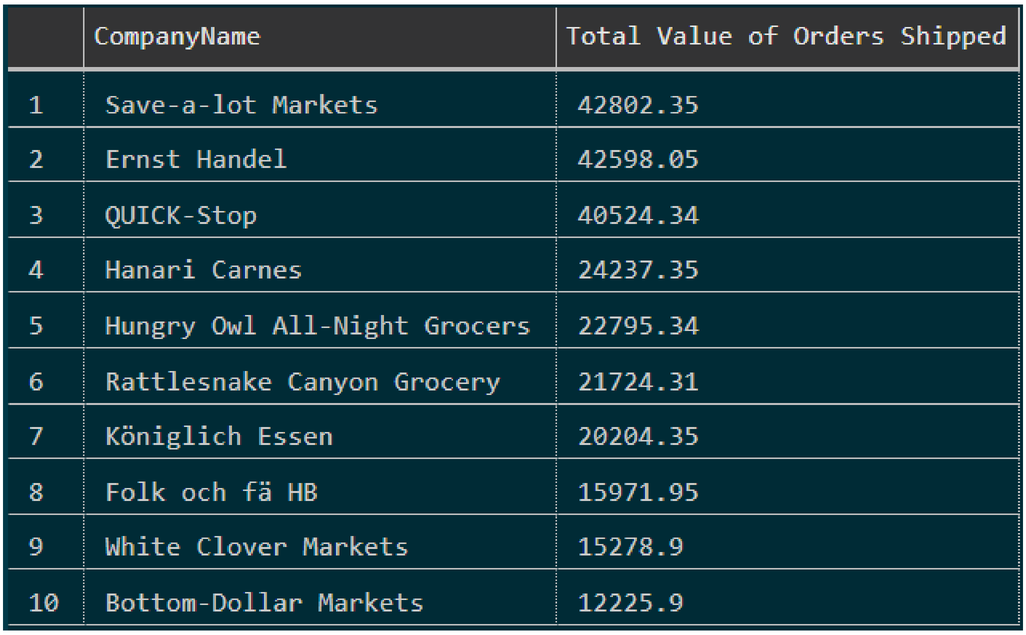
List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped. No Excel required.

Code:



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

Results:

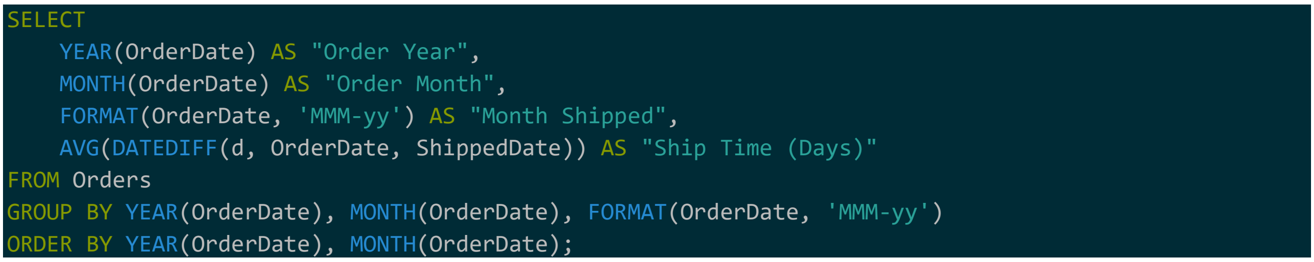


	CompanyName	Total Value of Orders Shipped
1	Save-a-lot Markets	42802.35
2	Ernst Handel	42598.05
3	QUICK-Stop	40524.34
4	Hanari Carnes	24237.35
5	Hungry Owl All-Night Grocers	22795.34
6	Rattlesnake Canyon Grocery	21724.31
7	Königlich Essen	20204.35
8	Folk och fä HB	15971.95
9	White Clover Markets	15278.9
10	Bottom-Dollar Markets	12225.9

3.4

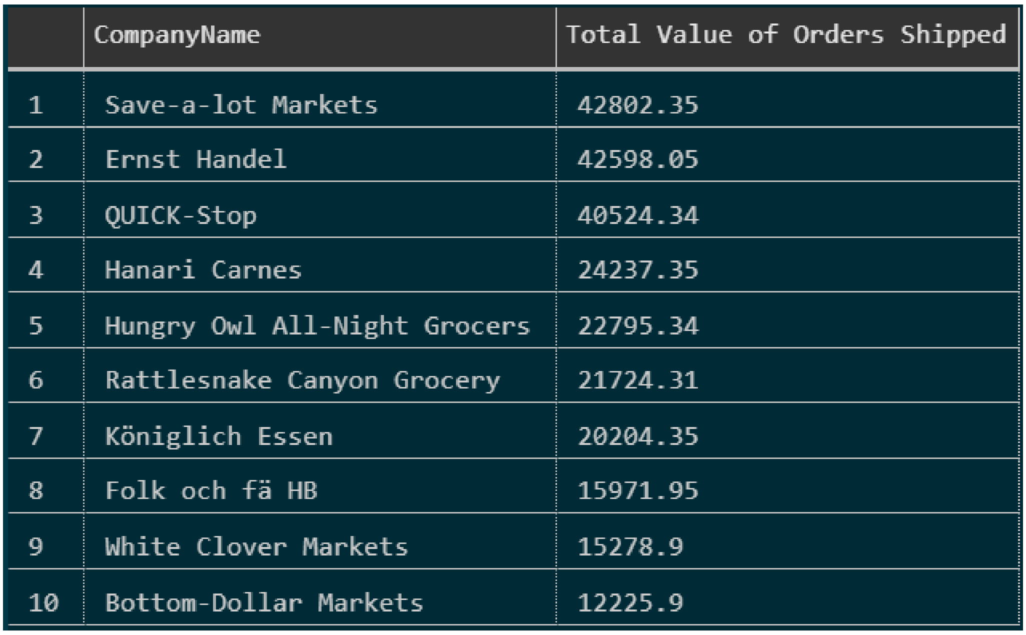
Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below.

Code:

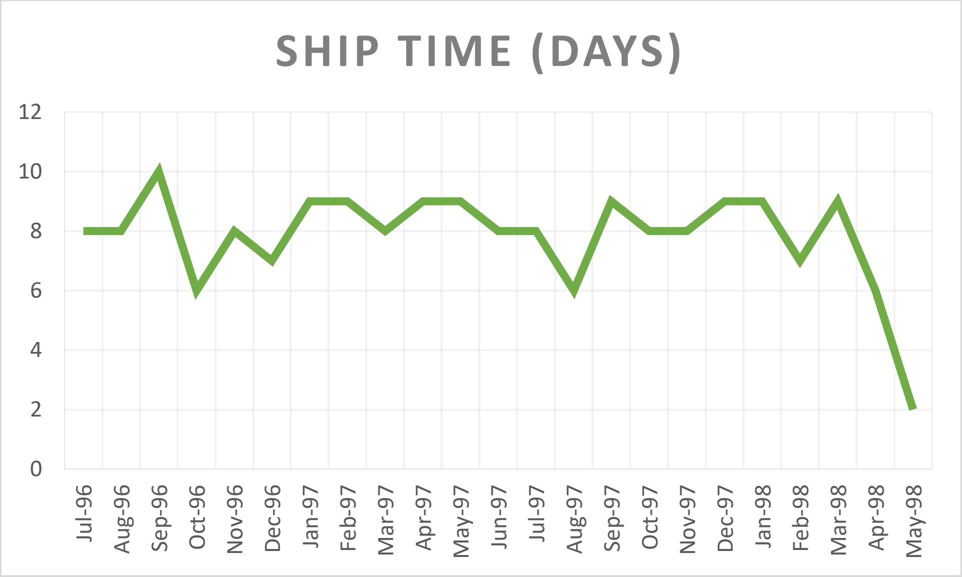


```sql
SELECT
    YEAR(OrderDate) AS "Order Year",
    MONTH(OrderDate) AS "Order Month",
    FORMAT(OrderDate, 'MMM-yy') AS "Month Shipped",
    AVG(DATEDIFF(d, OrderDate, ShippedDate)) AS "Ship Time (Days)"
FROM Orders
GROUP BY YEAR(OrderDate), MONTH(OrderDate), FORMAT(OrderDate, 'MMM-yy')
ORDER BY YEAR(OrderDate), MONTH(OrderDate);
```

Results:



	CompanyName	Total Value of Orders Shipped
1	Save-a-lot Markets	42802.35
2	Ernst Handel	42598.05
3	QUICK-Stop	40524.34
4	Hanari Carnes	24237.35
5	Hungry Owl All-Night Grocers	22795.34
6	Rattlesnake Canyon Grocery	21724.31
7	Königlich Essen	20204.35
8	Folk och fä HB	15971.95
9	White Clover Markets	15278.9
10	Bottom-Dollar Markets	12225.9


SHIP TIME (DAYS)
Jul-96
Aug-96
Sep-96
Oct-96
Nov-96
Dec-96
Jan-97
Feb-97
Mar-97
Apr-97
May-97
Jun-97
Jul-97
Aug-97
Sep-97
Oct-97
Nov-97
Dec-97
Jan-98
Feb-98
Mar-98
Apr-98
May-98