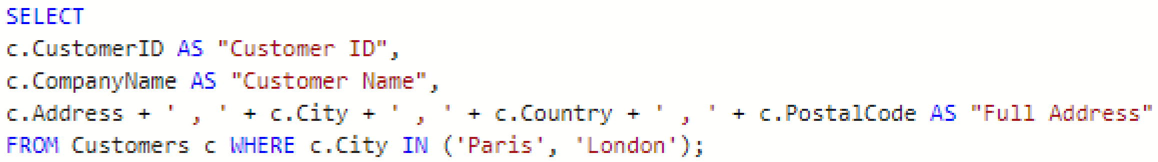
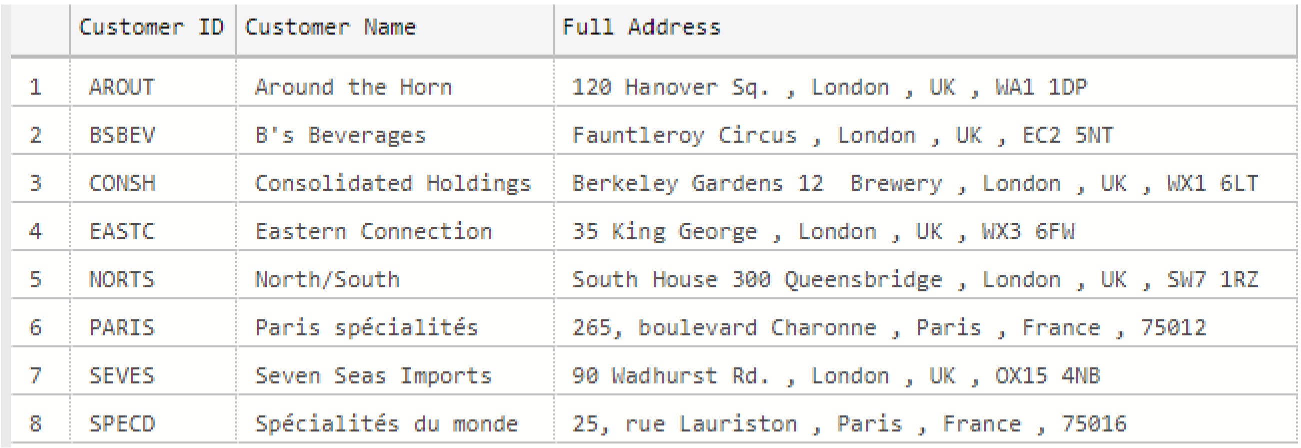
SQL Practical Exercise

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.

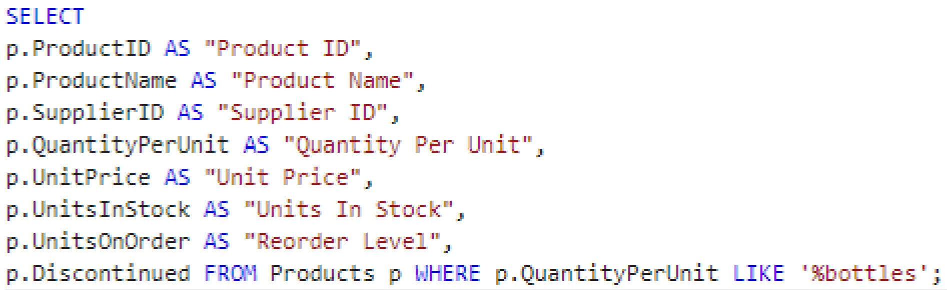


```
SELECT
c.CustomerID AS "Customer ID",
c.CompanyName AS "Customer Name",
c.Address + ' , ' + c.City + ' , ' + c.Country + ' , ' + c.PostalCode AS "Full Address"
FROM Customers c WHERE c.City IN ('Paris', 'London');
```

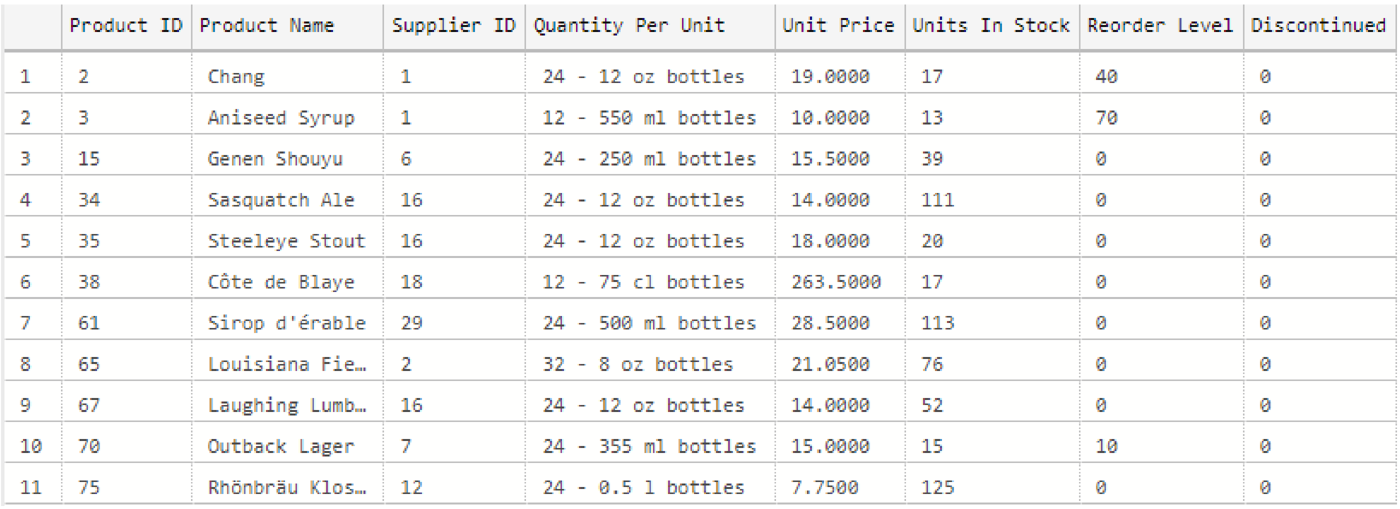


	Customer ID	Customer Name	Full Address
1	AROUT	Around the Horn	120 Hanover Sq. , London , UK , WA1 1DP
2	BSBEV	B's Beverages	Fauntleroy Circus , London , UK , EC2 5NT
3	CONSH	Consolidated Holdings	Berkeley Gardens 12 Brewery , London , UK , WX1 6LT
4	EASTC	Eastern Connection	35 King George , London ,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

1.2 List all products stored in bottles.



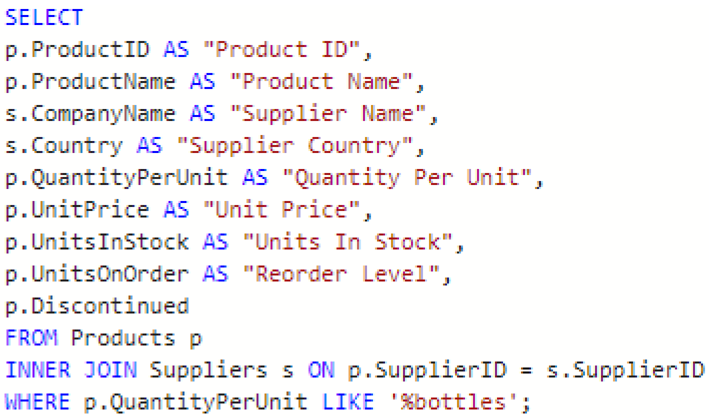
```
SELECT
p.ProductID AS "Product ID",
p.ProductName AS "Product Name",
p.SupplierID AS "Supplier ID",
p.QuantityPerUnit AS "Quantity Per Unit",
p.UnitPrice AS "Unit Price",
p.UnitsInStock AS "Units In Stock",
p.UnitsOnOrder AS "Reorder Level",
p.Discontinued FROM Products p WHERE p.QuantityPerUnit LIKE '%bottles';
```



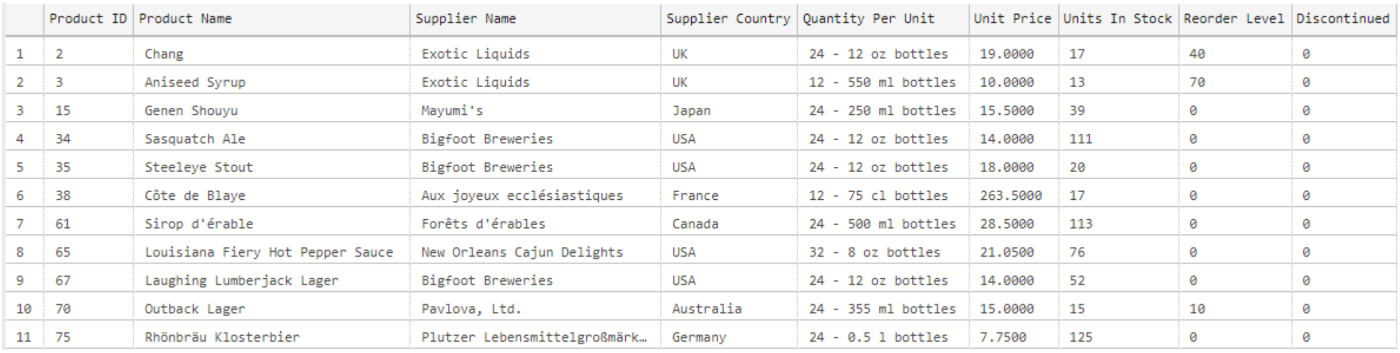
	Product ID	Product Name	Supplier ID	Quantity Per Unit	Unit Price	Units In Stock	Reorder Level	Discontinued
1	2	Chang	1	24 - 12 oz bottles	19.0000	17	40	0
2	3	Aniseed Syrup	1	12 - 550 ml bottles	10.0000	13	70	0
3	15	Genen Shouyu	6	24 - 250 ml bottles	15.5000	39	0	0
4	34	Sasquatch Ale	16	24 - 12 oz bottles	14.0000	111	0	0
5	35	Steeleye Stout	16	24 - 12 oz bottles	18.0000	20	0	0
6	38	Côte de Blaye	18	12 - 75 cl bottles	263.5000	17	0	0
7	61	Sirop d'érable	29	24 - 500 ml bottles	28.5000	113	0	0
8	65	Louisiana Fie…	2	32 - 8 oz bottles	21.0500	76	0	0
9	67	Laughing Lumb…	16	24 - 12 oz bottles	14.0000	52	0	0
10	70	Outback Lager	7	24 - 355 ml bottles	15.0000	15	10	0
11	75	Rhönbräu Klos…	12	24 - 0.5 l bottles	7.7500	125	0	0

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

SQL Practical Exercise

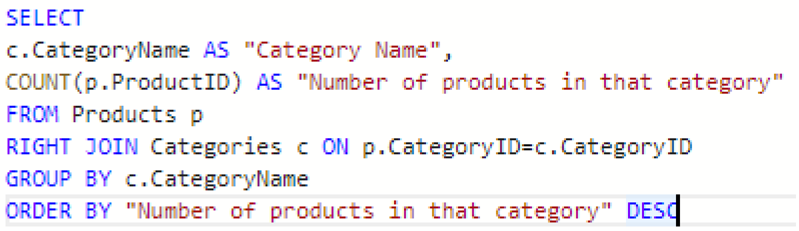


```
SELECT
p.ProductID AS "Product ID",
p.ProductName AS "Product Name",
s.CompanyName AS "Supplier Name",
s.Country AS "Supplier Country",
p.QuantityPerUnit AS "Quantity Per Unit",
p.UnitPrice AS "Unit Price",
p.UnitsInStock AS "Units In Stock",
p.UnitsOnOrder AS "Reorder Level",
p.Discontinued
FROM Products p
INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottles';
```

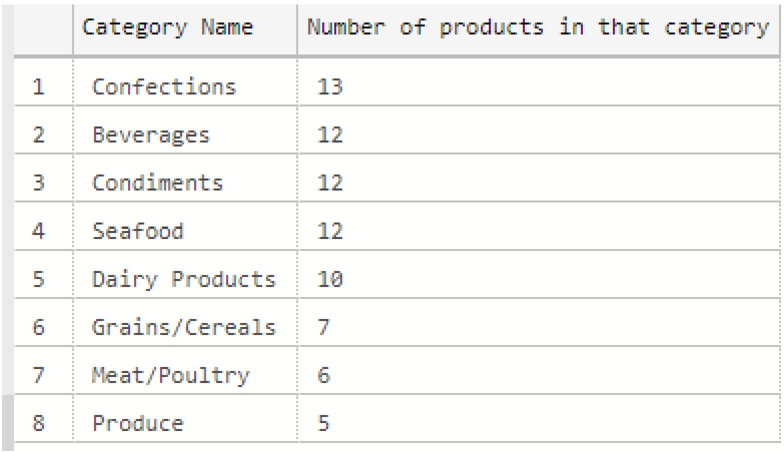


	Product ID	Product Name	Supplier Name	Supplier Country	Quantity Per Unit	Unit Price	Units In Stock	Reorder Level	Discontinued
1	2	Chang	Exotic Liquids	UK	24 - 12 oz bottles	19.0000	17	40	0
2	3	Aniseed Syrup	Exotic Liquids	UK	12 - 550 ml bottles	10.0000	13	70	0
3	15	Genen Shouyu	Mayumi's	Japan	24 - 250 ml bottles	15.5000	39	0	0
4	34	Sasquatch Ale	Bigfoot Breweries	USA	24 - 12 oz bottles	14.0000	111	0	0
5	35	Steeleye Stout	Bigfoot Breweries	USA	24 - 12 oz bottles	18.0000	20	0	0
6	38	Côte de Blaye	Aux joyeux ecclésiastiques	France	12 - 75 cl bottles	263.5000	17	0	0
7	61	Sirop d'érable	Forêts d'érables	Canada	24 - 500 ml bottles	28.5000	113	0	0
8	65	Louisiana Fiery Hot Pepper Sauce	New Orleans Cajun Delights	USA	32 - 8 oz bottles	21.0500	76	0	0
9	67	Laughing Lumberjack Lager	Bigfoot Breweries	USA	24 - 12 oz bottles	14.0000	52	0	0
10	70	Outback Lager	Pavlova, Ltd.	Australia	24 - 355 ml bottles	15.0000	15	10	0
11	75	Rhönbräu Klosterbier	Plutzer Lebensmittelgroßmärk…	Germany	24 - 0.5 l bottles	7.7500	125	0	0

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.

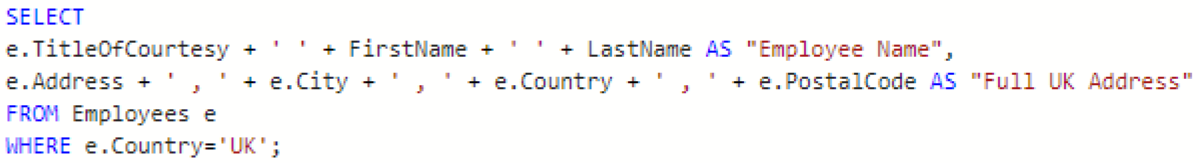


```
SELECT
c.CategoryName AS "Category Name",
COUNT(p.ProductID) AS "Number of products in that category"
FROM Products p
RIGHT JOIN Categories c ON p.CategoryID=c.CategoryID
GROUP BY c.CategoryName
ORDER BY "Number of products in that category" DESC
```



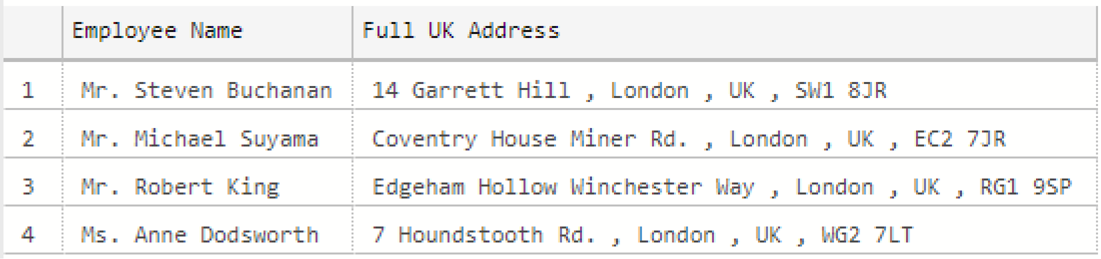
	Category Name	Number of products in that category
1	Confections	13
2	Beverages	12
3	Condiments	12
4	Seafood	12
5	Dairy Products	10
6	Grains/Cereals	7
7	Meat/Poultry	6
8	Produce	5

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.



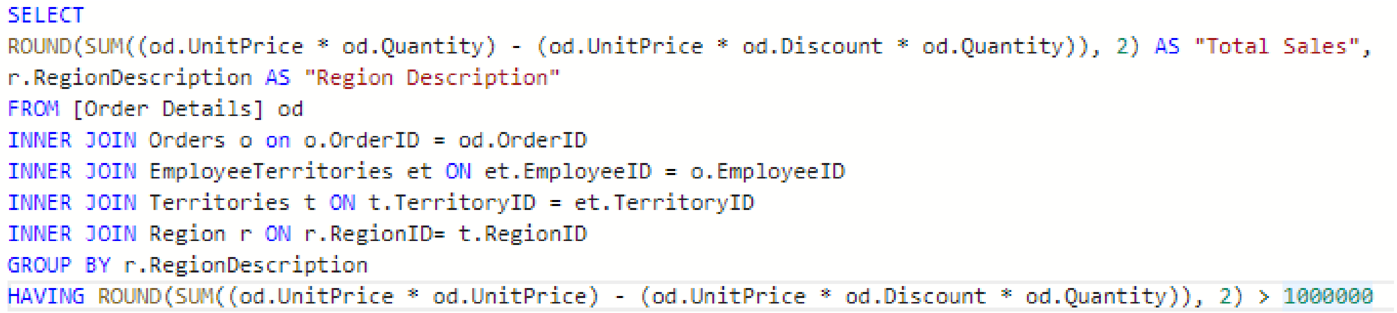
```
SELECT
e.TitleOfCourtesy + ' ' + FirstName + ' ' + LastName AS "Employee Name",
e.Address + ' , ' + e.City + ' , ' + e.Country + ' , ' + e.PostalCode AS "Full UK Address"
FROM Employees e
WHERE e.Country='UK';
```

SQL Practical Exercise

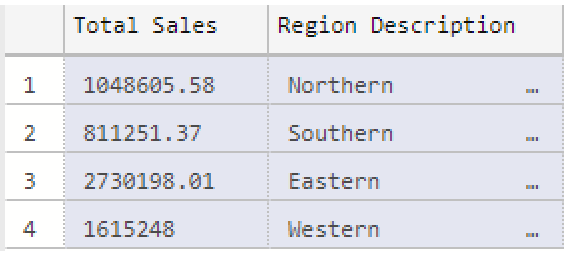


	Employee Name	Full UK Address
1	Mr. Steven Buchanan	14 Garrett Hill , London , UK , SW1 8JR
2	Mr. Michael Suyama	Coventry House Miner Rd. , London , UK , EC2 7JR
3	Mr. Robert King	Edgeham Hollow Winchester Way , London , UK , RG1 9SP
4	Ms. Anne Dodsworth	7 Houndstooth Rd. , London , UK , WG2 7LT

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.

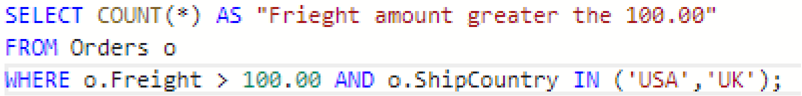


```
SELECT
ROUND(SUM((od.UnitPrice * od.Quantity) - (od.UnitPrice * od.Discount * od.Quantity)), 2) AS "Total Sales",
r.RegionDescription AS "Region Description"
FROM [Order Details] od
INNER JOIN Orders o on o.OrderID = od.OrderID
INNER JOIN EmployeeTerritories et ON et.EmployeeID = o.EmployeeID
INNER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

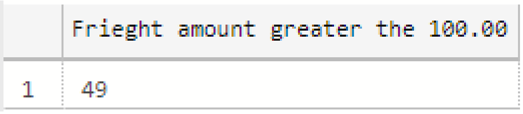


	Total Sales	Region Description
1	1048605.58	Northern …
2	811251.37	Southern …
3	2730198.01	Eastern …
4	1615248	Western …

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

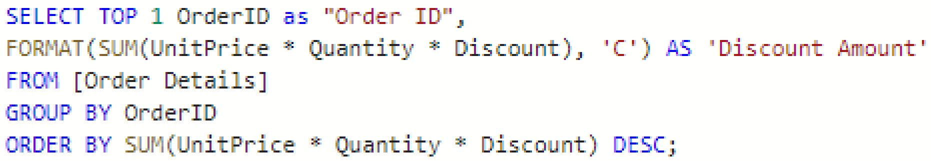


```
SELECT COUNT(*) AS "Frieght amount greater the 100.00"
FROM Orders o
WHERE o.Freight > 100.00 AND o.ShipCountry IN ('USA','UK');
```

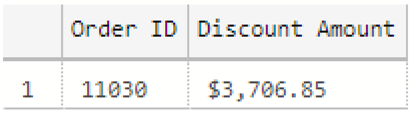


	Frieght amount greater the 100.00
1	49

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.



```
SELECT TOP 1 OrderID as "Order ID",
FORMAT(SUM(UnitPrice * Quantity * Discount), 'C') AS 'Discount Amount'
FROM [Order Details]
GROUP BY OrderID
ORDER BY SUM(UnitPrice * Quantity * Discount) DESC;
```



	Order ID	Discount Amount
1	11030	$3,706.85

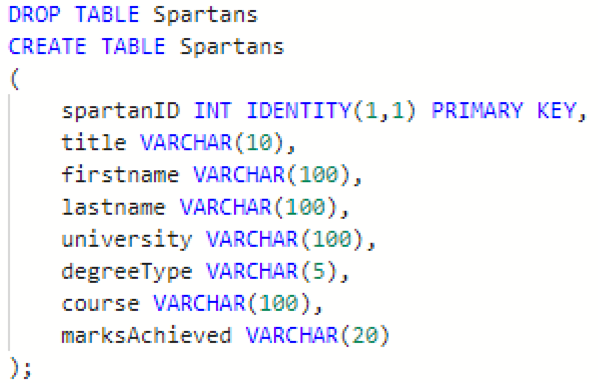
Exercise 2 – Create Spartans Table (20 marks – 10 each)

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.

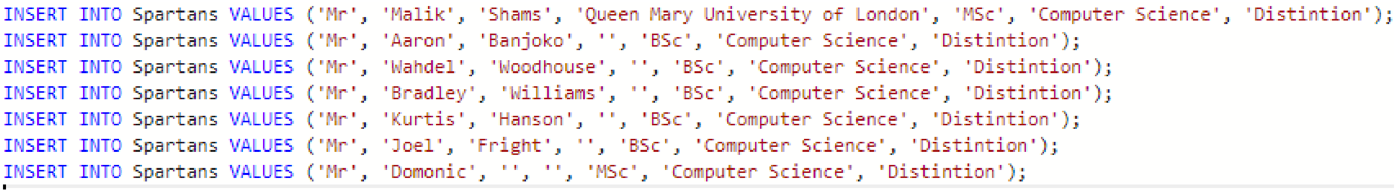
SQL Practical Exercise

IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.



```
DROP TABLE Spartans
CREATE TABLE Spartans
(
    spartanID INT IDENTITY(1,1) PRIMARY KEY,
    title VARCHAR(10),
    firstname VARCHAR(100),
    lastname VARCHAR(100),
    university VARCHAR(100),
    degreeType VARCHAR(5),
    course VARCHAR(100),
    marksAchieved VARCHAR(20)
);
```

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

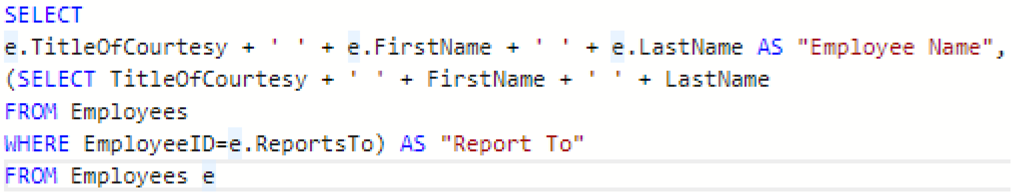


```
INSERT INTO Spartans VALUES ('Mr', 'Malik', 'Shams', 'Queen Mary University of London', 'MSc', 'Computer Science', 'Distintion');
INSERT INTO Spartans VALUES ('Mr', 'Aaron', 'Banjoko', '', 'BSc', 'Computer Science', 'Distintion');
INSERT INTO Spartans VALUES ('Mr', 'Wahdel', 'Woodhouse', '', 'BSc', 'Computer Science', 'Distintion');
INSERT INTO Spartans VALUES ('Mr', 'Bradley', 'Williams', '', 'BSc', 'Computer Science', 'Distintion');
INSERT INTO Spartans VALUES ('Mr', 'Kurtis', 'Hanson', '', 'BSc', 'Computer Science', 'Distintion');
INSERT INTO Spartans VALUES ('Mr', 'Joel', 'Fright', '', 'BSc', 'Computer Science', 'Distintion');
INSERT INTO Spartans VALUES ('Mr', 'Domonic', '', '', 'MSc', 'Computer Science', 'Distintion');
```

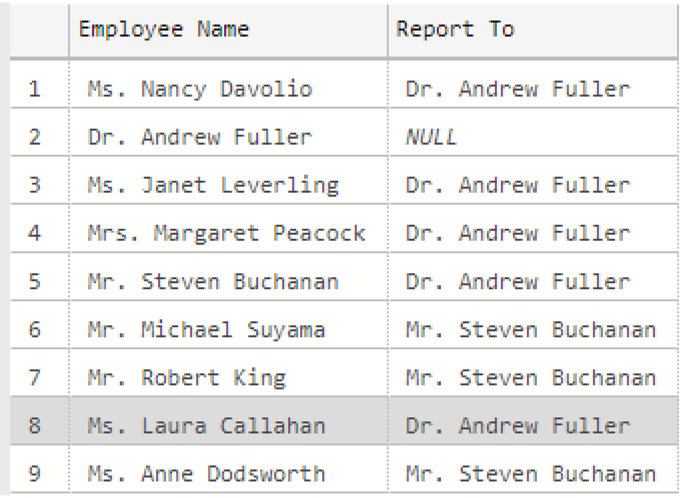
Exercise 3 – Northwind Data Analysis linked to Excel (30 marks)

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. (5 Marks)



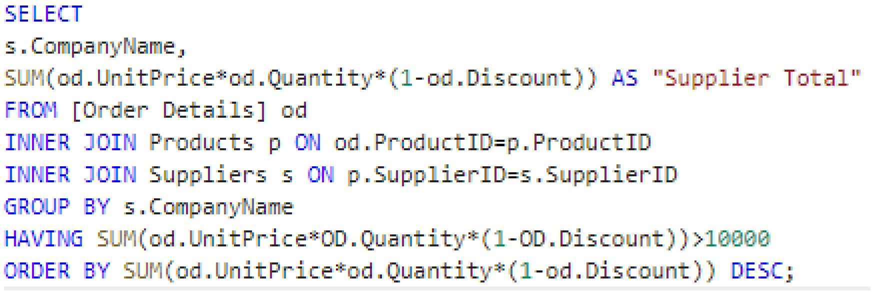
```
SELECT
e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Employee Name",
(SELECT TitleOfCourtesy + ' ' + FirstName + ' ' + LastName
FROM Employees
WHERE EmployeeID=e.ReportsTo) AS "Report To"
FROM Employees e
```



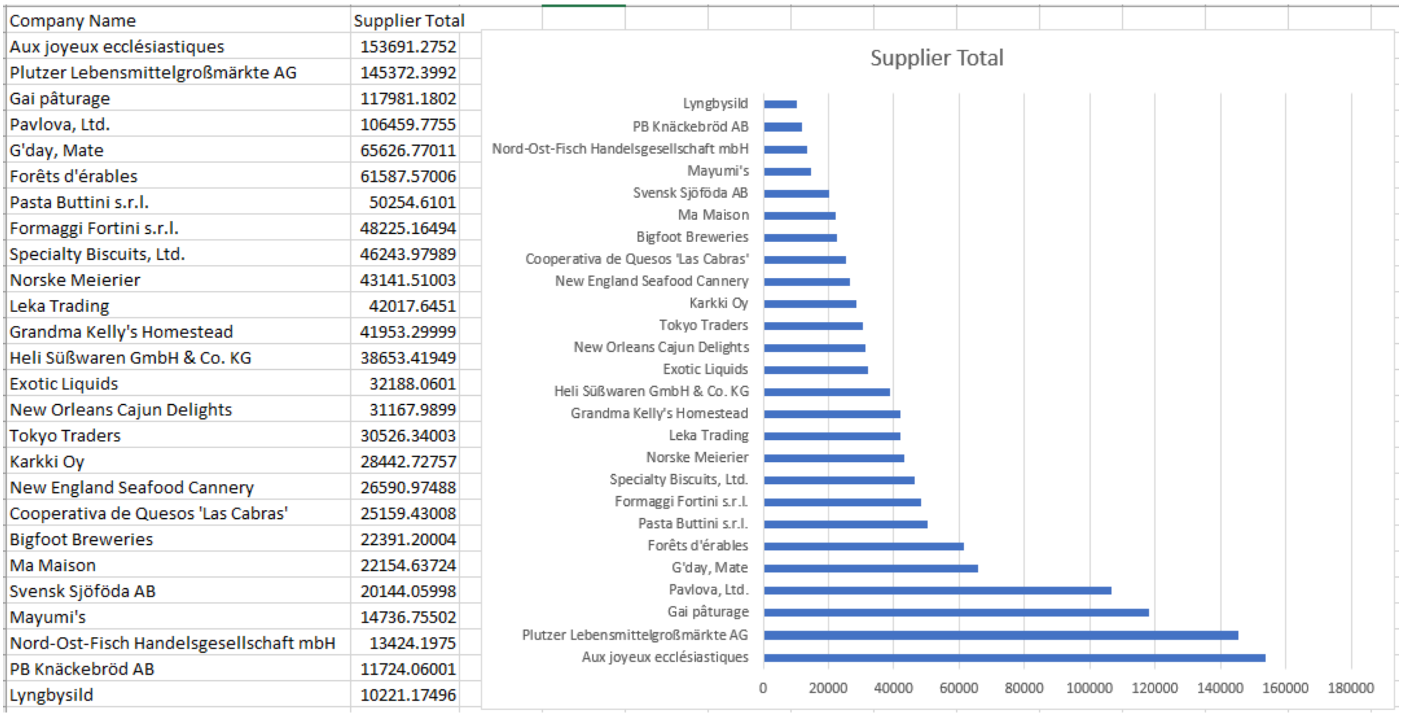
	Employee Name	Report To
1	Ms. Nancy Davolio	Dr. Andrew Fuller
2	Dr. Andrew Fuller	*NULL*
3	Ms. Janet Leverling	Dr. Andrew Fuller
4	Mrs. Margaret Peacock	Dr. Andrew Fuller
5	Mr. Steven Buchanan	Dr. Andrew Fuller
6	Mr. Michael Suyama	Mr. Steven Buchanan
7	Mr. Robert King	Mr. Steven Buchanan
8	Ms. Laura 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)

SQL Practical Exercise



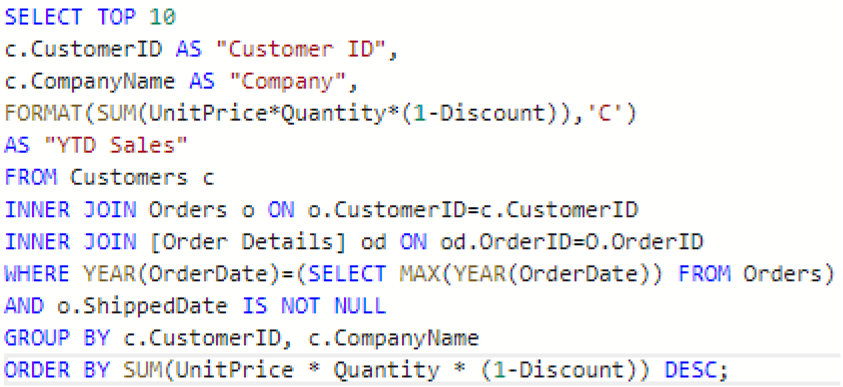
```
SELECT
s.CompanyName,
SUM(od.UnitPrice*od.Quantity*(1-od.Discount)) AS "Supplier Total"
FROM [Order Details] od
INNER JOIN Products p ON od.ProductID=p.ProductID
INNER JOIN Suppliers s ON p.SupplierID=s.SupplierID
GROUP BY s.CompanyName
HAVING SUM(od.UnitPrice*OD.Quantity*(1-OD.Discount))>10000
ORDER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



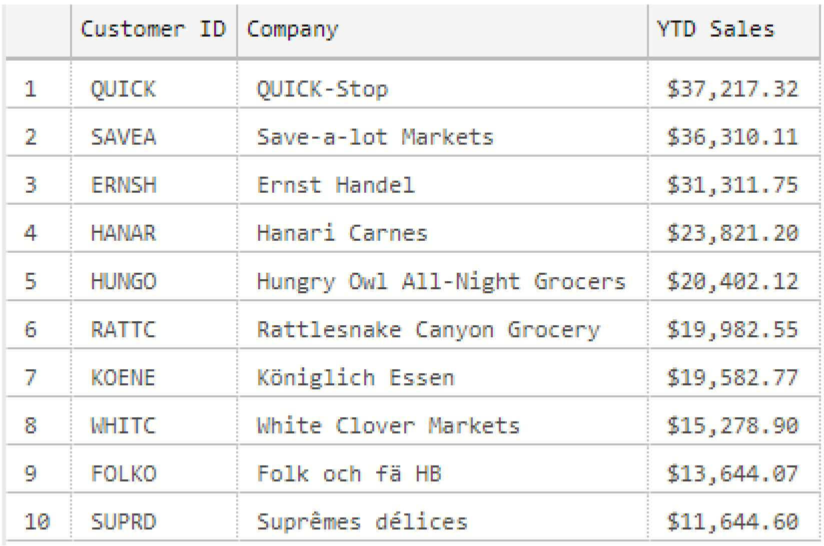
Company Name	Supplier Total
Aux joyeux ecclésiastiques	153691.2752
Plutzer Lebensmittelgroßmärkte AG	145372.3992
Gai pâturage	117981.1802
Pavlova, Ltd.	106459.7755
G'day, Mate	65626.77011
Forêts d'érables	61587.57006
Pasta Buttini s.r.l.	50254.6101
Formaggi Fortini s.r.l.	48225.16494
Specialty Biscuits, Ltd.	46243.97989
Norske Meierier	43141.51003
Leka Trading	42017.6451
Grandma Kelly's Homestead	41953.29999
Heli Süßwaren GmbH & Co. KG	38653.41949
Exotic Liquids	32188.0601
New Orleans Cajun Delights	31167.9899
Tokyo Traders	30526.34003
Karkki Oy	28442.72757
New England Seafood Cannery	26590.97488
Cooperativa de Quesos 'Las Cabras'	25159.43008
Bigfoot Breweries	22391.20004
Ma Maison	22154.63724
Svensk Sjöföda AB	20144.05998
Mayumi's	14736.75502
Nord-Ost-Fisch Handelsgesellschaft mbH	13424.1975
PB Knäckebröd AB	11724.06001
Lyngbysild	10221.17496

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)

SQL Practical Exercise

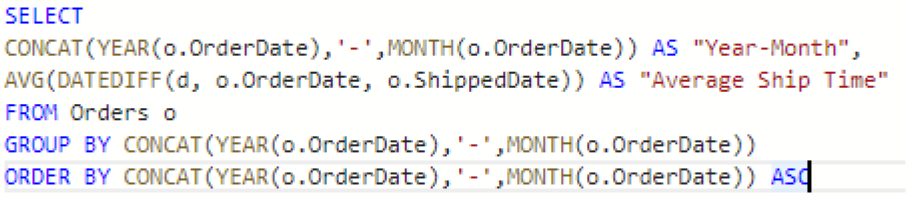


```
SELECT TOP 10
c.CustomerID AS "Customer ID",
c.CompanyName AS "Company",
FORMAT(SUM(UnitPrice*Quantity*(1-Discount)),'C')
AS "YTD Sales"
FROM Customers c
INNER JOIN Orders o ON o.CustomerID=c.CustomerID
INNER JOIN [Order Details] od ON od.OrderID=O.OrderID
WHERE YEAR(OrderDate)=(SELECT MAX(YEAR(OrderDate)) FROM Orders)
AND o.ShippedDate IS NOT NULL
GROUP BY c.CustomerID, c.CompanyName
ORDER BY SUM(UnitPrice * Quantity * (1-Discount)) DESC;
```



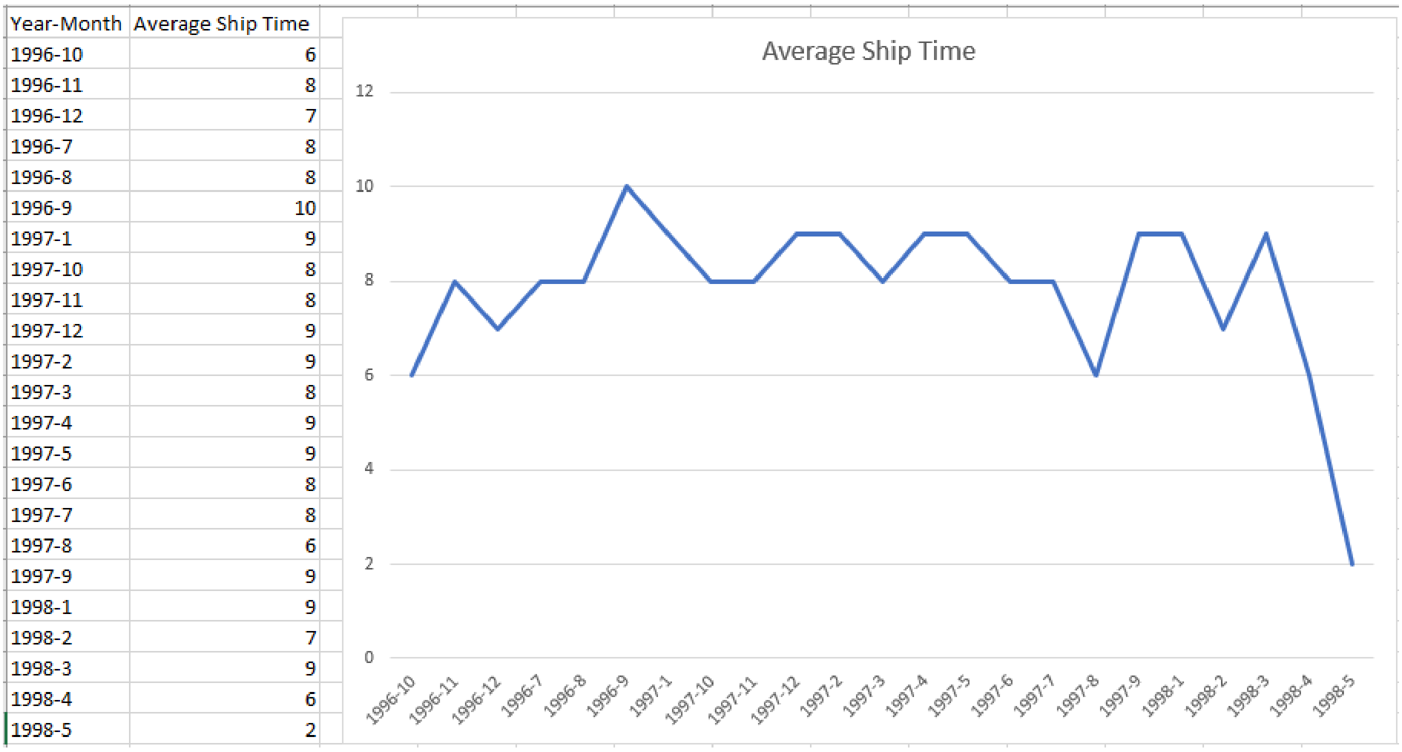
	Customer ID	Company	YTD Sales
1	QUICK	QUICK-Stop	$37,217.32
2	SAVEA	Save-a-lot Markets	$36,310.11
3	ERNSH	Ernst Handel	$31,311.75
4	HANAR	Hanari Carnes	$23,821.20
5	HUNGO	Hungry Owl All-Night Grocers	$20,402.12
6	RATTC	Rattlesnake Canyon Grocery	$19,982.55
7	KOENE	Königlich Essen	$19,582.77
8	WHITC	White Clover Markets	$15,278.90
9	FOLKO	Folk och fä HB	$13,644.07
10	SUPRD	Suprêmes délices	$11,644.60

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



```
SELECT
CONCAT(YEAR(o.OrderDate),'-',MONTH(o.OrderDate)) AS "Year-Month",
AVG(DATEDIFF(d, o.OrderDate, o.ShippedDate)) AS "Average Ship Time"
FROM Orders o
GROUP BY CONCAT(YEAR(o.OrderDate),'-',MONTH(o.OrderDate))
ORDER BY CONCAT(YEAR(o.OrderDate),'-',MONTH(o.OrderDate)) ASC
```

SQL Practical Exercise



Year-Month	Average Ship Time
1996-10	6
1996-11	8
1996-12	7
1996-7	8
1996-8	8
1996-9	10
1997-1	9
1997-10	8
1997-11	8
1997-12	9
1997-2	9
1997-3	8
1997-4	9
1997-5	9
1997-6	8
1997-7	8
1997-8	6
1997-9	9
1998-1	9
1998-2	7
1998-3	9
1998-4	6
1998-5	2