

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 CustomerID, CompanyName, Address, City, PostalCode, Country
FROM Customers
WHERE City = 'London' OR City = 'Paris';
```

Result:

Results		Messages				
	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 Brew...	London	WX1 6LT	UK
4	EASTC	Eastern Connection	35 King George	London	WX3 6FW	UK
5	NORTS	North/South	South House 300 Queensbri...	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.

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

Result:

	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 Pepp...
10	Laughing Lumberjack Lager
11	Outback Lager
12	Rhönbräu Klosterbier

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

```
SELECT p.ProductID, p.ProductName, s.CompanyName, s.Country
FROM Products p
JOIN Suppliers s ON p.SupplierID = s.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottle%';
```

Result:

	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 verte	Aux joyeux ecclésiastiques	France
8	Sirop d'érable	Forêts d'érables	Canada
9	Louisiana Fiery Hot Peppe...	New Orleans Cajun Delights	USA
10	Laughing Lumberjack Lager	Bigfoot Breweries	USA
11	Outback Lager	Pavlova, Ltd.	Australia
12	Rhönbräu Klosterbier	Plutzer Lebensmittelgroßm...	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.

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

Result:

Results Messages

	Category	Number of products in 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 + ' ' + e.FirstName + ' ' + e.LastName AS "Name and Title ", e.City AS "City of Residence"
FROM Employees e
WHERE E.Country = 'UK';
```

Result:

Results Messages

	Name and Title	City of Residence
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.

```
SELECT r.RegionDescription, ROUND(SUM(od.Quantity * od.UnitPrice * (1 - od.Discount)),2) total_sales
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.Quantity * od.UnitPrice),2)>1000000

Result:

Results		Messages
	RegionDescription	total_sales
1	Northern	1048605.58
2	Eastern	2730198.01
3	Western	1615248

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

SELECT COUNT(OrderID) AS "Orders", o.ShipCountry AS "Ship Country"

FROM Orders o

WHERE ShipCountry IN ('UK', 'USA') AND o.Freight > 100

GROUP BY o.ShipCountry

Result:

	Orders	Ship Country
1	9	UK
2	40	USA

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(orr.UnitPrice*orr.Discount*orr.Quantity) FROM [Order Details] orr)

ORDER BY Biggest_amount_of_discount DESC

Result:

	Biggest_amount_of_discount	OrderID
1	2108	10353
2	2108	10372

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.

DROP TABLE Spartan_table

CREATE TABLE Spartan_table (

PersonID **INT IDENTITY**(1,1) **PRIMARY KEY**,

Title **CHAR**(5),

First_Name **VARCHAR**(200),

Last_Name **VARCHAR**(200),

Uni_Attended **VARCHAR**(200),

Course_taken **VARCHAR**(200),

Mark_Achieved **DECIMAL**(2,1),

);

Result:

PersonID	Title	First_Name	Last_Name	Uni_Attended	Course_taken	Mark_Achieved
----------	-------	------------	-----------	--------------	--------------	---------------

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

INSERT INTO Spartan_table (

Title, First_Name, Last_Name, Uni_Attended, Course_taken, Mark_Achieved

)

VALUES(

'Mr.' , 'Kourosh' , 'Philip' , 'University College London' , 'German with Management Studies BA' , '2.1'

),

(

'Miss.' , 'Naveen' , 'Bo' , 'Cardiff University' , 'Data Science BSc' , '1.1'

),

(

'Dr.' , 'Nereida' , 'Tatiana' , 'Oxford University' , 'Applied Medical Sciences BSc' , '2.2'

);

Result:

	PersonID	Title	First_Name	Last_Name	Uni_Attended	Course_taken	Mark_Achieved
1	1	Mr.	Kourosh	Philip	University College London	German with Management St...	2.1
2	2	Miss.	Naveen	Bo	Cardiff University	Data Science BSc	1.1
3	3	Dr.	Nereida	Tatiana	Oxford University	Applied Medical Sciences ...	2.2

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.

```
SELECT CONCAT(e.TitleOfCourtesy, e.FirstName, ' ', e.LastName) AS "Employee Name",  
CONCAT(id.TitleOfCourtesy, id.FirstName, ' ', id.LastName) AS 'Reports To'  
FROM Employees e  
LEFT JOIN Employees id  
ON id.EmployeeID=e.ReportsTo
```

Result:

	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

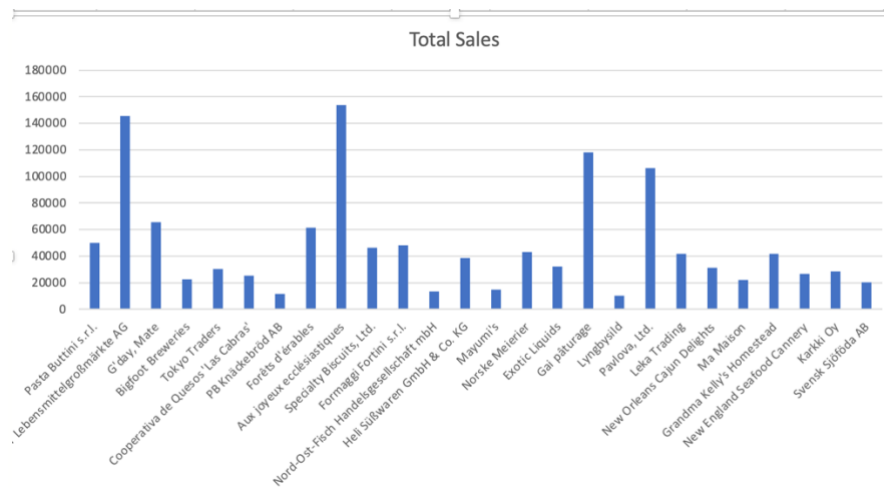
```

SELECT s.CompanyName,ROUND(SUM(od.Quantity * (od.UnitPrice - od.UnitPrice * od.Discount)),0) AS "Total
Sales"
FROM Suppliers s
JOIN Products p ON s.SupplierID = p.SupplierID
JOIN [Order Details] od ON p.ProductID = od.ProductID
GROUP BY s.CompanyName
HAVING SUM(od.Quantity*(od.UnitPrice-od.UnitPrice*od.Discount)) > 10000

```

Result:

	CompanyName	Total Sales
1	Pasta Buttini s.r.l.	50255
2	Plutzer Lebensmittelgroßm...	145372
3	G'day, Mate	65627
4	Bigfoot Breweries	22391
5	Tokyo Traders	30526
6	Cooperativa de Quesos 'La...	25159
7	PB Knäckebröd AB	11724
8	Forêts d'érables	61588
9	Aux joyeux ecclésiastiques	153691
10	Specialty Biscuits, Ltd.	46244
11	Formaggi Fortini s.r.l.	48225
12	Nord-Ost-Fisch Handelsges...	13424
13	Heli Süßwaren GmbH & Co. ...	38653
14	Mayumi's	14737
15	Norske Meierier	43142
16	Exotic Liquids	32188
17	Gai pâturage	117981
18	Lyngbysild	10221
19	Pavlova, Ltd.	106460
20	Leka Trading	42018
21	New Orleans Cajun Delights	31168
22	Ma Maison	22155
23	Grandma Kelly's Homestead	41953
24	New England Seafood Canne...	26591
25	Karkki Oy	28443
26	Svensk Sjöföda AB	20144



3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped.

```

SELECT TOP 10 c.CompanyName AS "Company Name" , ROUND(SUM(od.Quantity*(od.UnitPrice-
od.UnitPrice*od.Discount)),2) AS "Total Value Of Orders Shipped"
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN [Order Details] od ON o.OrderID = od.OrderID
WHERE YEAR(o.OrderDate) = (SELECT MAX(YEAR(oo.OrderDate)) FROM Orders oo)
GROUP BY c.CompanyName
ORDER BY "Total value of orders shipped" DESC

```

Result:

	Company Name	Total Value Of Orders Shipped (£)
1	Ernst Handel	41210.65
2	QUICK-Stop	37217.32
3	Save-a-lot Markets	36310.11
4	Hanari Carnes	23821.2
5	Rattlesnake Canyon Grocery	21238.27
6	Hungry Owl All-Night Groc...	20402.12
7	Königlich Essen	19582.77
8	White Clover Markets	15278.9
9	Folk och få HB	13644.07
1...	Suprêmes délices	11644.6

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

```
SELECT AVG(DATEDIFF(DAY, o.OrderDate , o.ShippedDate)) AS "Avg Ship Time", FORMAT(o.OrderDate ,  
'MMM-yyyy') AS "Month"  
FROM Orders o  
GROUP BY FORMAT(o.OrderDate,'MMM-yyyy') , DATEPART(YEAR, o.OrderDate) , DATEPART(MONTH,  
o.OrderDate)  
ORDER BY DATEPART(YEAR, o.OrderDate) , DATEPART(MONTH, o.OrderDate)
```

Result:

	Avg Ship Time	Month
1	8	Jul-1996
2	8	Aug-1996
3	10	Sep-1996
4	6	Oct-1996
5	8	Nov-1996
6	7	Dec-1996
7	9	Jan-1997
8	9	Feb-1997
9	8	Mar-1997
10	9	Apr-1997
11	9	May-1997
12	8	Jun-1997
13	8	Jul-1997
14	6	Aug-1997
15	9	Sep-1997
16	8	Oct-1997
17	8	Nov-1997
18	9	Dec-1997
19	9	Jan-1998
20	7	Feb-1998
21	9	Mar-1998
22	6	Apr-1998
23	2	May-1998

