SQL Project

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.

```
1 SELECT c.CustomerID, c.CompanyName, c.Address, c.City, c.Region, c.Country, c.PostalCode
2 FROM Customers c
3 WHERE c.City IN('Paris','London');
```

1.2 List all products stored in bottles.

```
SELECT p.ProductName AS "Product Name", p.QuantityPerUnit
FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottle%'
```

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

```
SELECT p.ProductName AS "Product Name", p.QuantityPerUnit, s.CompanyName AS "Supplier Name", s.Country
FROM Products p
JOIN Suppliers s ON s.SupplierID=p.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottle%'
```

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, COUNT(*) AS "No. of Products in Category"
FROM Categories c
JOIN Products p ON c.CategoryID=p.CategoryID
GROUP BY 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.

```
SELECT CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "Full name", e.City AS "City of Residence"
FROM Employees e
WHERE e.Country = 'UK'
```

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.RegionID,r.Regiondescription,
FORMAT(SUM(od.UnitPrice*(1-od.Discount)*od.Quantity),'###,###') AS "Sales Total for Region"
FROM [Order Details] od
    JOIN Orders o ON od.OrderID=o.OrderID
    JOIN EmployeeTerritories et ON o.EmployeeID=et.EmployeeID
    JOIN Territories t ON et.TerritoryID=t.TerritoryID
    JOIN Region r ON t.regionID=r.regionID
GROUP BY r.Regiondescription, r.RegionID
HAVING SUM(od.UnitPrice*(1-od.Discount)*od.Quantity) > 1000000
ORDER BY "Sales Total for Region" DESC
```

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 "No. of orders"
FROM Orders o
WHERE o.Freight > 100 AND o.ShipCountry IN ('UK', 'US');
```

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 o.OrderID,
(o.Discount*o.UnitPrice*o.Quantity) AS "Discount Amount"
FROM [Order Details] o
ORDER BY "Discount Amount" DESC
```

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.

```
DROP TABLE IF EXISTS engineering72
CREATE TABLE engineering72 (
    student_id INT IDENTITY PRIMARY KEY NOT NULL,
    title CHAR(2),
    first_name VARCHAR(15),
    last_name VARCHAR(15),
    university VARCHAR(50),
    course_taken VARCHAR(100),
    mark VARCHAR(10)
)
```

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

```
INSERT INTO engineering72 VALUES
('Mr','Toyin', 'Ajani', 'Univeristy of Bath','Chemical Engineering','first'),
('Mr','Reece', 'Louch', 'Univeristy of Warwick','Computer Science','2:1'),
('Mr','Saleh', 'Sandhu', 'Univeristy of Westminster','Computer Science','2:1'),
('Mr','Svillen', 'Petrov', 'London Metropolitan University','Computing','first'),
('Mr','Ben', 'Swift', 'Nottingham Trent University','Computer Science','2:1'),
('Mr','Christopher', 'Cunningham', 'Loughborough University','Computer Science','2:1'),
('Ms','Janja', 'Kovacevic', 'Univerity of Massachusetts Amherst','Computer Science and Computational Mathematics','3.9'),
('Mr','Dami', 'Oshidele', 'Kings College London','Electronic Engineering with Management','2:1'),
('Mr','Abdullah', 'Muhammad', 'Univeristy of Southampton','Physics','first'),
('Mr','Shahid', 'Enayat', 'Brunel University','Electronic and Electrical Engineering','2:2'),
('Mr','Emmanuel', 'Buraimo', 'Kings College London','Computer Science','2:1')
```

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

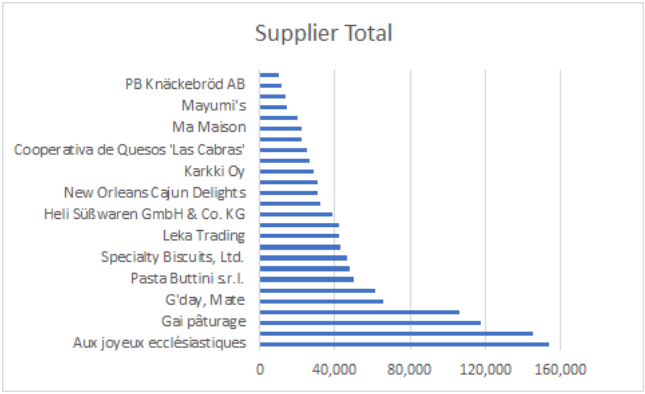
3.1 List all Employees from the Employees table and who they report to. No Excel required. (5 Marks)

```
SELECT e.EmployeeID, CONCAT(e.FirstName, ' ', e.LastName) AS "Employee Name", e1.FirstName + ' ' + e1.LastName AS "Reports to"
FROM Employees e
LEFT JOIN Employees e1 ON e.ReportsTo=e1.EmployeeID
```

```
-- Using Concat
SELECT e.EmployeeID, CONCAT(e.FirstName, ' ', e.LastName) AS "Employee Name", CONCAT(e1.FirstName ,' ' ,e1.LastName) AS "Report
FROM Employees e
LEFT JOIN Employees e1 ON e.ReportsTo=e1.EmployeeID
```

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)

```
SELECT  s.CompanyName, FORMAT(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)), '#,###,###') AS "Total Sales"
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
```

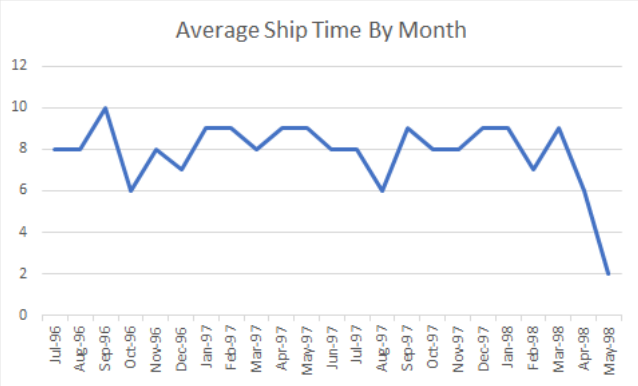


3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped. No Excel required. (10 Marks)

```
SELECT TOP 10 c.CompanyName, SUM(od.Quantity*od.UnitPrice*(1-od.Discount)) AS "YTD Sales"
FROM Orders o
JOIN Customers c ON o.CustomerID=c.CustomerID
JOIN [Order Details] od ON o.OrderID=od.OrderID
WHERE  YEAR(o.OrderDate) =  (SELECT TOP 1 YEAR(o1.OrderDate)
    FROM Orders o1
    ORDER BY o1.OrderDate DESC) AND o.ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY SUM(od.Quantity*od.UnitPrice*(1-od.Discount)) DESC
```

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 MONTH(o.OrderDate) AS "Month", YEAR(o.OrderDate) AS "Year", AVG(DATEDIFF(d,o.OrderDate,o.ShippedDate)) AS "Ship time"
FROM Orders o
WHERE o.ShippedDate IS NOT NULL
GROUP BY MONTH(o.OrderDate), YEAR(o.OrderDate)
ORDER BY "Year", "Month"
```


Average Ship Time By Month
12
10
8
6
4
2
0
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