SQL Project

Exercise 1

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

```
-- Exercise 1.1
SELECT c.CustomerID,
    c.CompanyName AS "Company Name",
    c.Address,
    c.City,
    c.Region,
    c.PostalCode AS "Post Code",
    c.Country
FROM Customers c
WHERE City IN ('Paris','London');
```

1.2 List all products stored in bottles.

```
-- Exercise 1.2
SELECT p.ProductName
FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottle%';
```

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

```
-- Exercise 1.3
SELECT p.ProductName AS "Name of Product",
    s.CompanyName AS "Supplier Name",
    s.Country
FROM Products p
INNER JOIN Suppliers s ON s.SupplierID = p.SupplierID
WHERE 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.

```
-- Exercise 1.4
SELECT c.CategoryName AS "Category",
    COUNT(*) AS "Number Of Products in Each Category"
FROM Products p
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY 'Number Of Products in Each Category' 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.

```
-- Exercise 1.5
SELECT CONCAT(TitleOfCourtesy,' ',FirstName,' ',LastName) AS "Name",
    City AS "City Of Residence"
FROM Employees
WHERE 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.

```
-- Exercise 1.6
SELECT r.RegionDescription AS "Region",
    ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) AS "Total Sales"
FROM [Order Details] od
INNER JOIN Orders o
ON od.OrderID=o.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.Quantity*(1-od.Discount)),2)>1000000
ORDER BY "Total Sales" DESC;
```

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

```
-- Exercise 1.7
SELECT
    COUNT(*) AS 'Orders to the UK or USA that are over 100 freight'
FROM Orders
WHERE Freight > 100.00 AND ShipCountry IN ('USA','UK');
```

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.

```
-- Exercise 1.8
SELECT TOP 1 OrderID,
    ROUND(SUM(UnitPrice*Quantity*Discount),2) AS 'Discount Amount'
FROM [Order Details]
GROUP BY OrderID
ORDER BY [Discount Amount] DESC;
```

Exercise 2

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
-- Exercise 2.1
CREATE TABLE Spartans_Table
(
    Person_ID INT IDENTITY(1,1) PRIMARY KEY,
    Title VARCHAR(5) DEFAULT('Mr.'),
    First_Name VARCHAR(30),
    Last_Name VARCHAR(30),
    University VARCHAR(60),
    Course_Taken VARCHAR (100),
    Mark_Achieved VARCHAR(5)
);
```

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

```sql
-- Exercise 2.2
INSERT INTO Spartans_Table
(
    First_Name,
    Last_Name,
    University,
    Course_Taken,
    Mark_Achieved
)
VALUES
(
    'Tom', 'Canfield','Loughborough University','Automotive Engineering','2:1'
),
(
    'Sotiris', 'Loizou', 'Birmingham University', 'Computer Science', '2:1'
),
(
    'Alexander', 'Legon', 'University of Kent', 'Computer Systems Engineering', '2:1'
),
(
    'Alex', 'Barber-Lynch', 'University of Hertfordshire', 'Physics', '2:1'
),
(
    'Karim', 'Wohler', 'University of Greenwich', 'Mechanical Engineering', '2:1'
);
```

Exercise 3

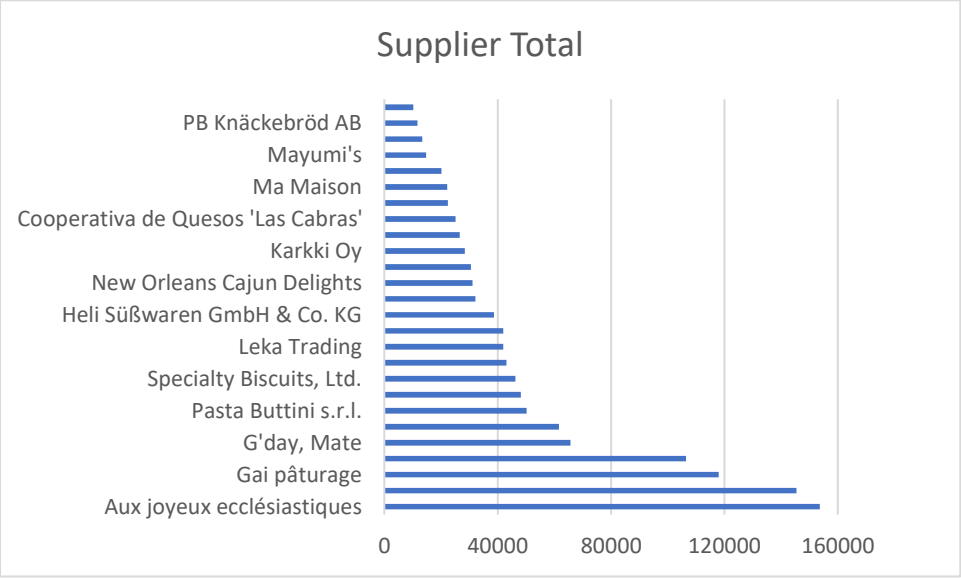
3.1 List all Employees from the Employees table and who they report to. No Excel required. Please mention the Employee Names and the ReportTo names.

```
-- Exercise 3.1
SELECT
    CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "Employee Name",
    (SELECT
     CONCAT(ed.TitleOfCourtesy, ' ', ed.FirstName, ' ', ed.LastName)
     FROM Employees AS "ed"
     WHERE ed.EmployeeID=e.ReportsTo) AS "Reports To"
FROM Employees e;
```

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

```
-- Exercise 3.2

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

```sql
SELECT TOP 10
    c.CompanyName AS "Customer Name",
    ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) AS "Year To Date"
FROM [Order Details] od
INNER JOIN Orders o
ON o.OrderID=od.OrderID
INNER JOIN Customers c
ON c.CustomerID=o.CustomerID
WHERE YEAR(OrderDate)=(SELECT MAX(YEAR(OrderDate)) FROM Orders) AND ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY "Year To Date" DESC;
```

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

```sql
--Exercise 3.4

SELECT
    FORMAT(OrderDate, 'MMM-yy') AS "Date",
    MONTH(OrderDate) AS "Month",
    YEAR(OrderDate) AS "Year",
    AVG(CAST((DATEDIFF(d,OrderDate,ShippedDate)) AS decimal(4,2))) AS "Average Shipping Time"
FROM Orders
GROUP BY FORMAT(OrderDate, 'MMM-yy'), MONTH(OrderDate),YEAR(OrderDate)
ORDER BY 3,2;
```