SQL Practical Exercise

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

```
SELECT c.CustomerID, c.CompanyName, c.Address, c.City, c.Region, c.PostalCode, c.Country
FROM Customers c
WHERE c.City='Paris' OR c.City='London'
```

1.2 List all products stored in bottles.

```
SELECT p.ProductName, p.QuantityPerUnit
From Products p
WHERE p.QuantityPerUnit LIKE '%bottles%'
```

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

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

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 "Categories", COUNT(p.ProductID) AS "Number of Products"
FROM [Categories] c
INNER JOIN Products p ON c.CategoryID=p.CategoryID
GROUP BY c.CategoryName
ORDER BY COUNT(p.ProductID) 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 e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Full Name", e.Country,
e.City
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 SUM(od.UnitPrice * od.Quantity) AS "Total Sales", r.RegionDescription
FROM [Order Details] od
INNER JOIN Orders o ON od.OrderID = o.OrderID
INNER JOIN EmployeeTerritories e ON o.EmployeeID = e.EmployeeID
INNER JOIN Territories t ON e.TerritoryID = t.TerritoryID
INNER JOIN Region r on r.RegionID = t.RegionID
GROUP BY r.RegionDescription
HAVING ROUND(SUM(od.UnitPrice* od.Quantity),2) >1000000
```

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

```sql
SELECT COUNT(o.Freight), o.ShipCountry
FROM Orders o
WHERE o.Freight > 100.00 AND o.ShipCountry IN ('UK','USA')
GROUP BY o.ShipCountry
```

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

```sql
SELECT TOP 1 od.OrderID, ROUND(SUM((od.UnitPrice * od.Quantity) * od.Discount),2) AS "Total Discount"
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY "Total Discount" DESC
```

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

```sql
CREATE TABLE spartan_table
(
    spartan_id INT IDENTITY(1,1) PRIMARY KEY,
    spartan_fname VARCHAR(25),
    spartan_lname VARCHAR(25),
    university VARCHAR(25),
    course VARCHAR(25),
    mark DECIMAL(2,1)
)
```

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

```sql
INSERT INTO spartan_table
(spartan_fname, spartan_lname, university, course, mark )
VALUES
('Julia', 'Kowalska', 'Imperial College', 'software dev', 2.1),
('John', 'Smith', 'University of Leeds', 'networking', 2.2),
('Ann', 'Walker', 'University of Manchester', 'computing', 1.1),
('Lisa', 'Lee', 'Southampton', 'computing', 2.1),
('Kyle', 'Green', 'University of Leeds', 'networking', 2.2),
('Mary', 'Rainbow', 'University of Manchester', 'computing', 1.1),
('Nate', 'Moon', 'Southampton', 'software dev', 2.1),
('Lilly', 'Grey', 'University of Leeds', 'networking', 2.2),
('Natalie', 'Book', 'University of Manchester', 'computing', 1.1),
('Dan', 'Baker', 'University of London', 'software dev', 2.1)
```

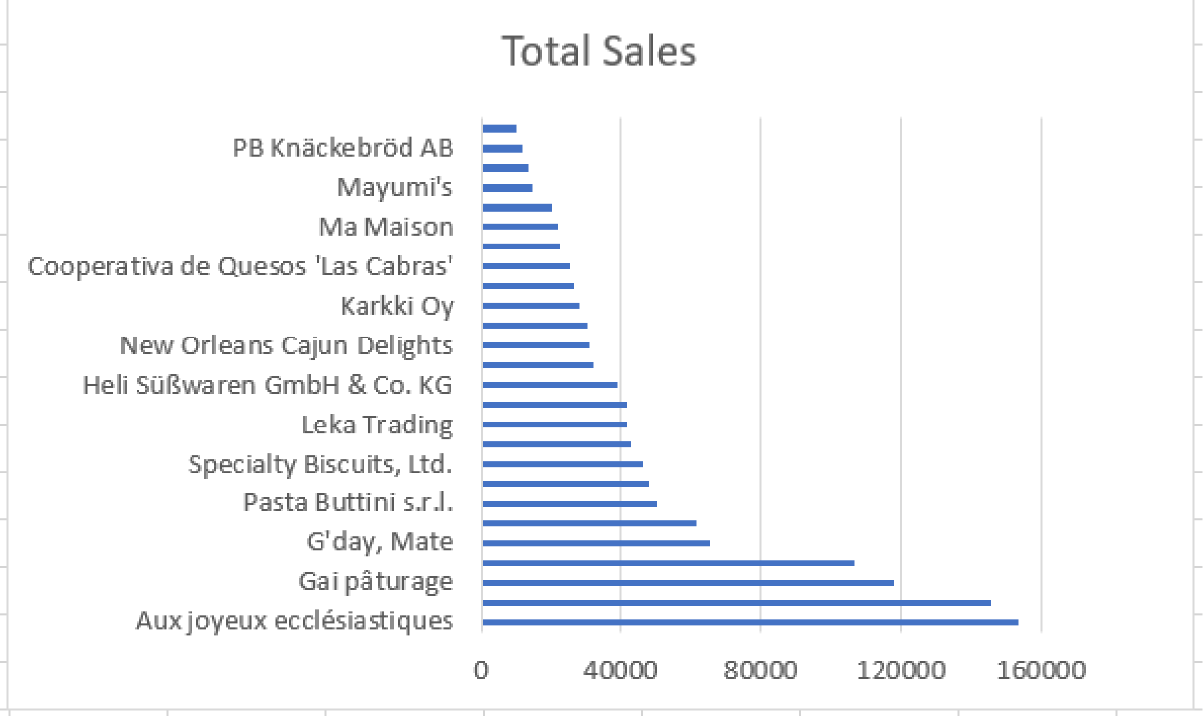
Exercise 3 – Northwind Data Analysis linked to Excel

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

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

3.2 List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table.

```
SELECT s.CompanyName AS "Company", ROUND(SUM((od.UnitPrice * od.Quantity) - (od.UnitPrice *
 od.Discount * od.Quantity)),2) AS "Total Sales" -(total order-discount order)
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)- (od.UnitPrice * od.Discount * od.Quantity)) > 1000
0
ORDER BY SUM((od.UnitPrice * od.Quantity)- (od.UnitPrice * od.Discount * od.Quantity)) DESC
```

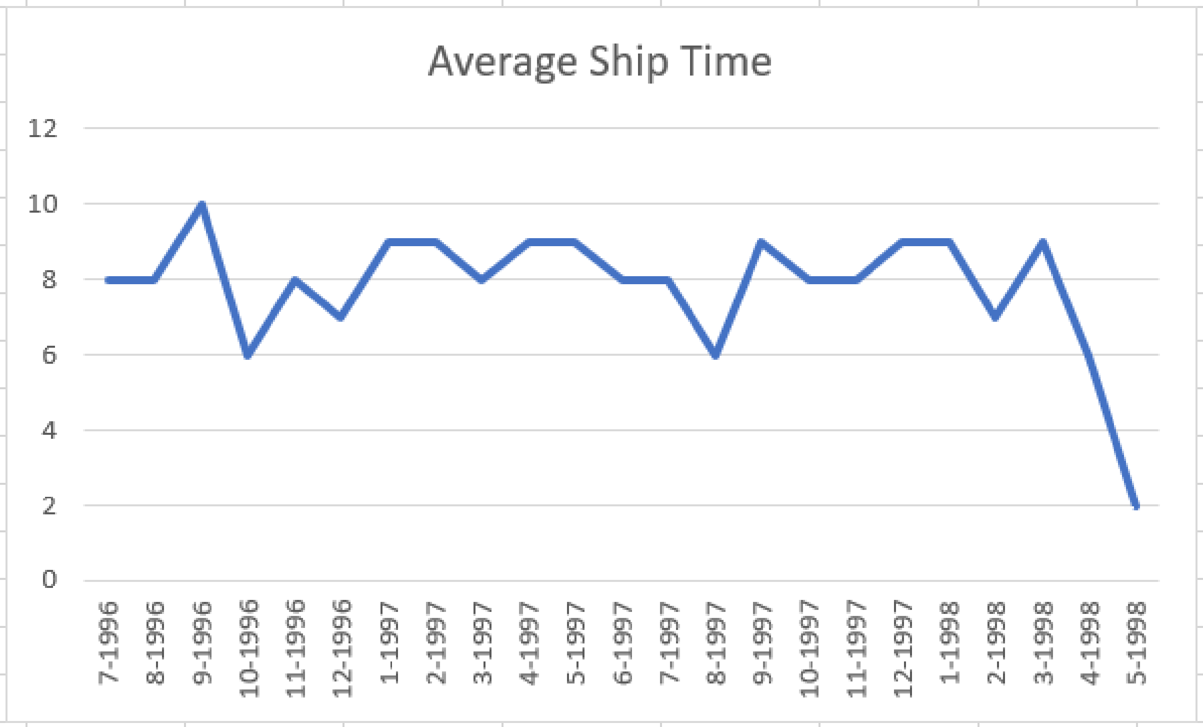


```
3.3 List the Top 10 Customers YTD for the latest year in the Orders file.
Based on total value of orders shipped.

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

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

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


Average Ship Time