Week 2 Project Write-up

Class	SQL
Created	@Jun 30, 2020 11:26 AM
Type	Assignment

Introduction

The following is the project I completed as a part of my SQL training. This involved forming queries that manipulated the Northwind database to answer the questions given.

Queries

Write a query that lists all Customers in either Paris or London.

Include Customer ID, Company Name and all address fields.

```
/* Address field to go: Address, City, Country then postal code*/
SELECT
     c.CustomerID
    ,c.CompanyName
    ,CONCAT(c.Address,', ', c.City,', ', c.Country,', ', c.PostalCode) AS "Full Adress"
FROM
    Customers c
WHERE
    c.City IN ('Paris', 'London')
```

List all products stored in bottles.

```
SELECT
     p.ProductName AS "Product Name"
    ,p.QuantityPerUnit
FROM
    Products p
WHERE
    p.QuantityPerUnit LIKE '%bottle%'
/* To search for the word bottle within the quantity per unit column */
ORDER BY
    p.ProductName ASC
```

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 AS "Country of Origin"
FROM
    Products p
INNER JOIN Suppliers s ON
    p.SupplierID=s.SupplierID
/*The Suppliers table contains the information we are looking for */
```

```
WHERE
    p.QuantityPerUnit LIKE '%bottle%'
```

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 "Employee Name"
    ,e.City AS "City Of Residence"
FROM
    Employees e
WHERE
    e.Country LIKE 'UK' /*The question asks for UK employees*/
```

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
     t.RegionID
    ,r.RegionDescription
    ,ROUND(SUM((od.Quantity * od.UnitPrice)*(1-od.Discount)),2) AS "Total Sales"
FROM
    Employees e
INNER JOIN Orders o ON
    e.EmployeeID = o.EmployeeID
INNER JOIN [Order Details] od ON
    o.OrderID = od.OrderID
INNER JOIN EmployeeTerritories et ON
    e.EmployeeID=et.EmployeeID
INNER JOIN Territories t ON
    et.TerritoryID=t.TerritoryID
INNER JOIN Region r ON
    r.RegionID=t.RegionID
GROUP BY
    t.RegionID,r.RegionDescription
HAVING
    SUM((od.Quantity * od.UnitPrice)*(1-od.Discount)) > 1000000
/*This ensures that the total sales is being summed for each region*/
ORDER BY
    REGIONID ASC
```

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

```
SELECT
    COUNT(*) AS "Number of Orders with Freight Amount Greater Than 100"
/*The number of rows in the table will be the number of orders we are looking for*/
FROM
    Orders o
WHERE
    Freight > 100 AND o.ShipCountry IN ('USA', 'UK')
```

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.

```
CREATE DATABASE ibs_db
USE ibs_db
CREATE TABLE spartans_table
(

    title VARCHAR(50),
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    university_attended VARCHAR(50),
    graduation_date VARCHAR(50),
    course_taken VARCHAR(50),
    mark_achieved INT
)
```

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

```
INSERT INTO spartans_table (title, first_name , last_name , university_attended , graduation_date , course_taken , mark_achieved)
    VALUES

    ('Mr', 'Ibrahim', 'Bocus', 'University of Leicester','31-12-2018','Mechanical Engineering', (RAND()*((100-90)+90))),
    ('Mr', 'Andrew', 'Osborne', 'University of Andrew','31-12-2020','Biomedical Science', (RAND()*((100-1)+1))),
    ('Mr', 'Daniel', 'Teegan', 'University of Brighton','31-12-2020','Product Design', (RAND()*((100-1)+1))),
    ('Mr', 'Abdelbari', 'Allali', 'University of Lancaster','31-12-2020','Business and Economics', (RAND()*((100-1)+1))),
    ('Mr', 'Mehdi', 'Shamaa', 'University of Nottingham','31-12-2020','Philosophy and Economics', (RAND()*((100-1)+1))),
    ('Mr', 'Anais', 'Tang', 'University Edinburgh','31-12-2020','Modern Languages', (RAND()*((100-1)+1))),
    ('Mr', 'Saheed', 'Lamina', 'University of Warwick','31-12-2020','Politics and International Studies', (RAND()*((100-1)+1))),
    ('Mr', 'Man-Wai', 'Tse', 'University of Hertfordshire','31-12-2020','Aerospace Engineering', (RAND()*((100-1)+1))),
    ('Mr', 'Sohaib', 'Sohail', 'Brunel University London','31-12-2020','Modern Languages', (RAND()*((100-1)+1))),
    ('Miss', 'Ugne', 'Okmanaite', 'Aston University','31-12-2020','Internal Business and Management', (RAND()*((100-1)+1))),
    ('Mr', 'John', 'Byrne', 'University of Greenwich','31-12-2020','Computing and Games Development', (RAND()*((100-1)+1))),
    ('Mr', 'Daniel', 'Teegan', 'University of Brighton','31-12-2020','Product Design', (RAND()*((100-1)+1))),
    ('Mr', 'Max', 'Palmer', 'University of Birmingham','31-12-2020','Ancient History Languages', (RAND()*((100-1)+1)))
)
/*A random number generator was used for marks achieved to not offend anyone.
However, if you look at the top row (mine),you will see that not all random number
generators are made equally*/
```

List all Employees from the Employees table and who they report to.

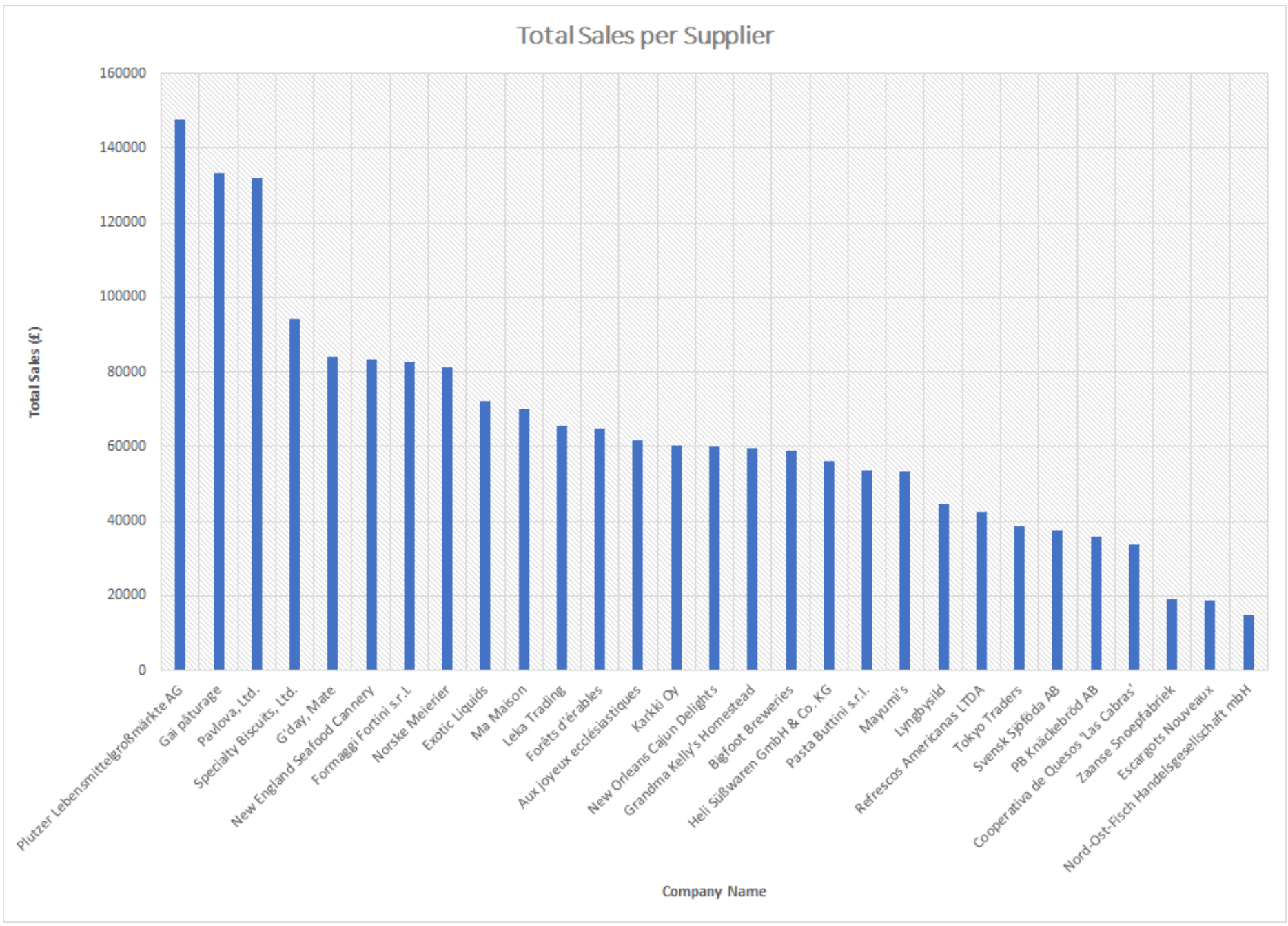
```
/* This is an example of a self join as the data in the table refers to itself.
Andrew is not shown in the output table as he has no manager */

SELECT e.EmployeeID
    ,CONCAT(e.FirstName, ' ', e.LastName) AS "Employee Name"
    ,CONCAT(e2.FirstName, ' ', e2.LastName) AS "Manager Name"
FROM
        Employees e, Employees e2
    /*This selfjoin is necessary because ReportsTo references the Employee ID of that person*/
WHERE
        e2.EmployeeID = e.ReportsTo
    /*This enforces the condition we are looking for*/
```

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
    ,FORMAT((SUM((od.UnitPrice*od.Quantity)*(1-od.Discount))),'N','em-uk') AS "Total Sales (£)"
/* This formatting puts it into correct UK format with commas*/
        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)*(1-od.Discount)) > 10000
        ORDER BY
            SUM((od.UnitPrice*od.Quantity)*(1-od.Discount)) DESC
```

Output table of above question:



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.CustomerID AS "Customer ID"
    ,c.CompanyName As "Company"
```

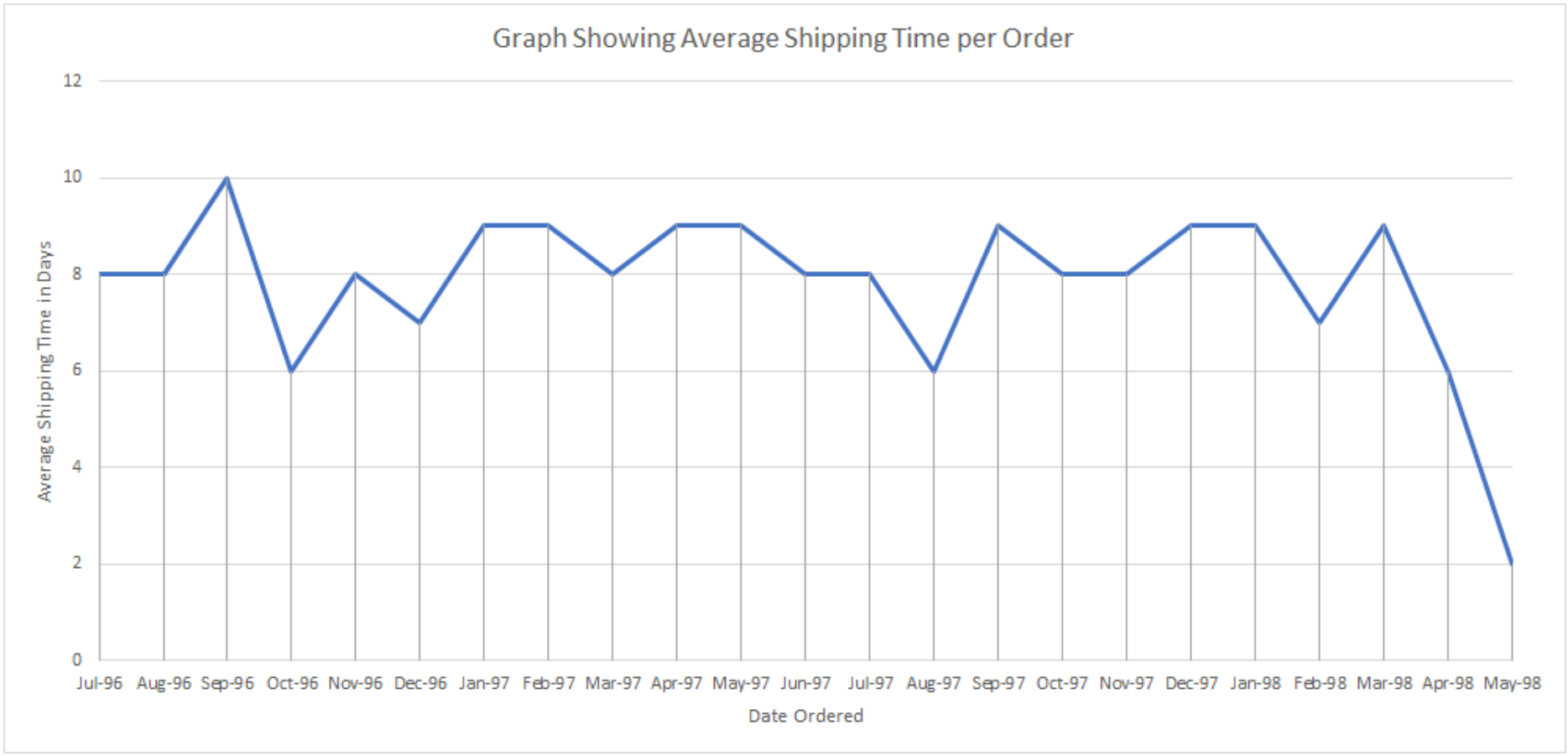
```
    ,FORMAT(SUM(UnitPrice * Quantity * (1-Discount)),'C','em-uk') AS "YTD Sales"
FROM
    Customers c
INNER JOIN Orders o ON
    o.CustomerID=c.CustomerID
INNER JOIN [Order Details] od ON
    od.OrderID=o.OrderID
WHERE
    YEAR(OrderDate)= (SELECT MAX(YEAR(OrderDate)) From Orders) AND o.ShippedDate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

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

```
SELECT
     CONCAT(sq1.MonthName,' ', sq1.YearOrdered) "Date Ordered"
    ,AVG("ShipTimePerproductindays") "AverageShipTimePerproductindays"
/*This uses the subquery information and averages the amount of shipping time for the column*/
FROM

        (SELECT
             DATEDIFF(d,o.orderdate,o.ShippedDate) "ShipTimePerproductindays"
            ,MONTH(o.OrderDate) "MonthOrdered"
            ,YEAR(o.orderdate) "YearOrdered"
            ,DateName(MONTH,DATEADD(MONTH,MONTH(o.orderdate) , 0 ) - 1 ) "MonthName"
        /* This converts the month number to month name */
        FROM
            Orders o) sq1, Orders o
WHERE
    o.ShippedDate IS NOT NULL
/* This subquery gives the time it took to ship each order, and also the month that the order was made*/
GROUP BY
     sq1.YearOrdered
/*This tells the AVG function to only average the shiptimes for each month*/
    ,sq1.MonthName
    ,sq1.MonthOrdered
ORDER BY
     CONVERT(datetime, CONCAT(sq1.YearOrdered,'/',sq1.MonthOrdered,'/','1'))
/* This puts it in a nice format for excel*/
```

Output table of above question:



Conclusion

This project was my first introduction to coding, and I believe that SQL is a good language to start with as it is syntactically quite simple. This project was very fulfilling to complete, and provided a great opportunity to practice applying the theory I have learned during my classes.