2021

# SQL Mini Project

ENGINEERING 92 JAVA SDET

JOE HILTON

# Question 1)

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

```
-- 1.1
SELECT c.CustomerID, c.CompanyName,
c.Address + ', ' + c.City + ', ' + c.PostalCode + ', ' + c.Country AS "Address"
FROM Customers c
WHERE c.City = 'Paris' OR c.City = 'London'
```

*Figure 1 - 1.1 Code*

| | CustomerID | CompanyName | Address |
|---|---|---|---|
| 1 | AROUT | Around the Horn | 120 Hanover Sq., London, WA1… |
| 2 | BSBEV | B's Beverages | Fauntleroy Circus, London, E… |
| 3 | CONSH | Consolidated Holdings | Berkeley Gardens 12  Brewery… |
| 4 | EASTC | Eastern Connection | 35 King George, London, WX3 … |
| 5 | NORTS | North/South | South House 300 Queensbridge… |
| 6 | PARIS | Paris spécialités | 265, boulevard Charonne, Par… |
| 7 | SEVES | Seven Seas Imports | 90 Wadhurst Rd., London, OX1… |
| 8 | SPECD | Spécialités du monde | 25, rue Lauriston, Paris, 75… |

*Figure 2 - 1.1 Results*

## 1.2 – List all products stored in bottles

```
-- 1.2
SELECT p.ProductID, p.ProductName, p.QuantityPerUnit
FROM Products p
WHERE QuantityPerUnit LIKE '%bottle%'
```

*Figure 3 - 1.2 Code*

| | ProductID | ProductName | QuantityPerUnit |
|---|---|---|---|
| 1 | 2 | Chang | 24 - 12 oz bottles |
| 2 | 3 | Aniseed Syrup | 12 - 550 ml bottles |
| 3 | 15 | Genen Shouyu | 24 - 250 ml bottles |
| 4 | 34 | Sasquatch Ale | 24 - 12 oz bottles |
| 5 | 35 | Steeleye Stout | 24 - 12 oz bottles |
| 6 | 38 | Côte de Blaye | 12 - 75 cl bottles |
| 7 | 39 | Chartreuse verte | 750 cc per bottle |
| 8 | 61 | Sirop d'érable | 24 - 500 ml bottles |
| 9 | 65 | Louisiana Fiery Hot Pepper S… | 32 - 8 oz bottles |
| 10 | 67 | Laughing Lumberjack Lager | 24 - 12 oz bottles |
| 11 | 70 | Outback Lager | 24 - 355 ml bottles |
| 12 | 75 | Rhönbräu Klosterbier | 24 - 0.5 l bottles |

*Figure 4 - 1.2 Results*

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

```
-- 1.3
SELECT p.ProductID, p.ProductName, p.QuantityPerUnit, s.CompanyName, s.Country
FROM Products p
JOIN Suppliers s ON p.SupplierID = s.SupplierID
WHERE QuantityPerUnit LIKE '%bottle%'
```

*Figure 5 - 1.3 Code*

| | ProductID | ProductName | QuantityPerUnit | CompanyName | Country |
|---|---|---|---|---|---|
| 1 | 2 | Chang | 24 - 12 oz bottles | Exotic Liquids | UK |
| 2 | 3 | Aniseed Syrup | 12 - 550 ml bottles | Exotic Liquids | UK |
| 3 | 15 | Genen Shouyu | 24 - 250 ml bottles | Mayumi's | Japan |
| 4 | 34 | Sasquatch Ale | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| 5 | 35 | Steeleye Stout | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| 6 | 38 | Côte de Blaye | 12 - 75 cl bottles | Aux joyeux ecclésiastiques | France |
| 7 | 39 | Chartreuse verte | 750 cc per bottle | Aux joyeux ecclésiastiques | France |
| 8 | 61 | Sirop d'érable | 24 - 500 ml bottles | Forêts d'érables | Canada |
| 9 | 65 | Louisiana Fiery Hot Pepper S… | 32 - 8 oz bottles | New Orleans Cajun Delights | USA |
| 10 | 67 | Laughing Lumberjack Lager | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| 11 | 70 | Outback Lager | 24 - 355 ml bottles | Pavlova, Ltd. | Australia |
| 12 | 75 | Rhönbräu Klosterbier | 24 - 0.5 l bottles | Plutzer Lebensmittelgroßmärk… | Germany |

*Figure 6 - 1.3 Results*

## 1.4 – Write an SQL Statement that shows how many products there are in each category. Include Category Name in results set and list the highest number first.

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

*Figure 7 - 1.4 Code*

| | CategoryName | 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 |

*Figure 8 - 1.4 Results*

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.

```
-- 1.5
SELECT e.TitleOfCourtesy + ' ' +  e.FirstName + ' ' + e.LastName AS "UK Employee Name",
e.City
FROM Employees e
WHERE e.Country = 'UK'
```

*Figure 9 - 1.5 Code*

| | UK Employee Name | City |
|---|---|---|
| 1 | Mr. Steven Buchanan | London |
| 2 | Mr. Michael Suyama | London |
| 3 | Mr. Robert King | London |
| 4 | Ms. Anne Dodsworth | London |

*Figure 10 - 1.5 Results*

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.

```
-- 1.6
SELECT r.RegionDescription,
ROUND(SUM((od.UnitPrice*od.Quantity)*(1-od.Discount)), 2) AS "Total Sales"
FROM Region r
JOIN Territories t ON t.RegionID = r.RegionID
JOIN EmployeeTerritories et ON et.TerritoryID = t.TerritoryID
JOIN Employees e ON e.EmployeeID = et.EmployeeID
JOIN Orders o ON o.EmployeeID = e.EmployeeID
JOIN [Order Details] od ON od.OrderID = o.OrderID
GROUP BY r.RegionDescription
HAVING SUM((od.UnitPrice*od.Quantity)*1-od.Discount) > 1000000
```

*Figure 11 - 1.6 Code*

| | RegionDescription | Total Sales |
|---|---|---|
| 1 | Northern … | 1048605.58 |
| 2 | Eastern … | 2730198.01 |
| 3 | Western … | 1615248 |

*Figure 12 - 1.6 Results*

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

```
-- 1.7
SELECT COUNT(*) AS "Orders with Freight > 100 in USA or UK"
FROM Orders
WHERE (Freight > 100 AND (ShipCountry = 'USA' OR ShipCountry = 'UK'))
```

*Figure 13 - 1.7 Code*

| | Orders with Freight > 100 in USA or UK |
|---|---|
| 1 | 49 |

*Figure 14 - 1.7 Results*

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.

```
-- 1.8
SELECT TOP 1 od.OrderID,
SUM((od.UnitPrice*od.Quantity)*od.Discount) AS "Discount Value"
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY"Discount Value" DESC
```

*Figure 15 - 1.8 Code*

| | OrderID | Discount Value |
|---|---|---|
| 1 | 11030 | 3706.8499755859375 |

*Figure 16 - 1.8 Result*

# Question 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.

```
-- Q2
CREATE DATABASE joe_db
USE joe_db

-- 2.1
CREATE TABLE spartans(
title VARCHAR(5),
firstName VARCHAR(15),
lastName VARCHAR(15),
university VARCHAR(30),
courseName VARCHAR (30),
mark INT)
```

*Figure 17 - 2.1 Code*

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

```
-- 2.2
INSERT INTO spartans
VALUES ("Mr.", "Joe", "Hilton", "Brunel University", "BSc Computer Science", 70)

INSERT INTO spartans (title, firstName, lastName, university, courseName, mark)
VALUES ("Mr.", "Mac", "Uche", "Essex University", "BSc Engineering", 75)
```

*Figure 18 - 2.2 Code*

| | title | firstName | lastName | university | courseName | mark |
|---|---|---|---|---|---|---|
| 1 | Mr. | Joe | Hilton | Brunel University | BSc Computer Science | 70 |
| 2 | Mr. | Mac | Uche | Essex University | BSc Engineering | 75 |

*Figure 19 - Sample Spartan Data for Q2*

# Question 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.

```
-- 3.1
SELECT e.FirstName + ' ' + e.LastName AS "Employee Name",
(SELECT s.FirstName + ' ' + s.LastName
FROM Employees s
WHERE s.EmployeeID = e.ReportsTo) AS "Reports to"
FROM Employees e
```

*Figure 20 - 3.1 Code*

| | Employee Name | Reports to |
|---|---|---|
| 1 | Nancy Davolio | Andrew Fuller |
| 2 | Andrew Fuller | *NULL* |
| 3 | Janet Leverling | Andrew Fuller |
| 4 | Margaret Peacock | Andrew Fuller |
| 5 | Steven Buchanan | Andrew Fuller |
| 6 | Michael Suyama | Steven Buchanan |
| 7 | Robert King | Steven Buchanan |
| 8 | Laura Callahan | Andrew Fuller |
| 9 | Anne Dodsworth | Steven Buchanan |

*Figure 21 - 3.1 Results*

## 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.

```
-- 3.2
SELECT s.CompanyName,
SUM((od.UnitPrice * od.Quantity)*(1-od.Discount)) AS "Total Sales"
FROM Suppliers s
JOIN Products p ON p.SupplierID = s.SupplierID
JOIN [Order Details] od ON od.ProductID = p.ProductID
GROUP BY s.CompanyName
HAVING SUM((od.UnitPrice * od.Quantity)*(1-od.Discount)) > 10000
ORDER BY "Total Sales" DESC
```

*Figure 22 - 3.2 Code*

| | CompanyName | Total Sales |
|---|---|---|
| 1 | Aux joyeux ecclésiastiques | 153691.2751789093 |
| 2 | Plutzer Lebensmittelgroßmärk… | 145372.39916038513 |
| 3 | Gai pâturage | 117981.18016052246 |
| 4 | Pavlova, Ltd. | 106459.77550125122 |
| 5 | G'day, Mate | 65626.77010917664 |
| 6 | Forêts d'érables | 61587.57006072998 |
| 7 | Pasta Buttini s.r.l. | 50254.61009979248 |
| 8 | Formaggi Fortini s.r.l. | 48225.16494369507 |
| 9 | Specialty Biscuits, Ltd. | 46243.97989368439 |

*Figure 23 - Results Sample for 3.2*

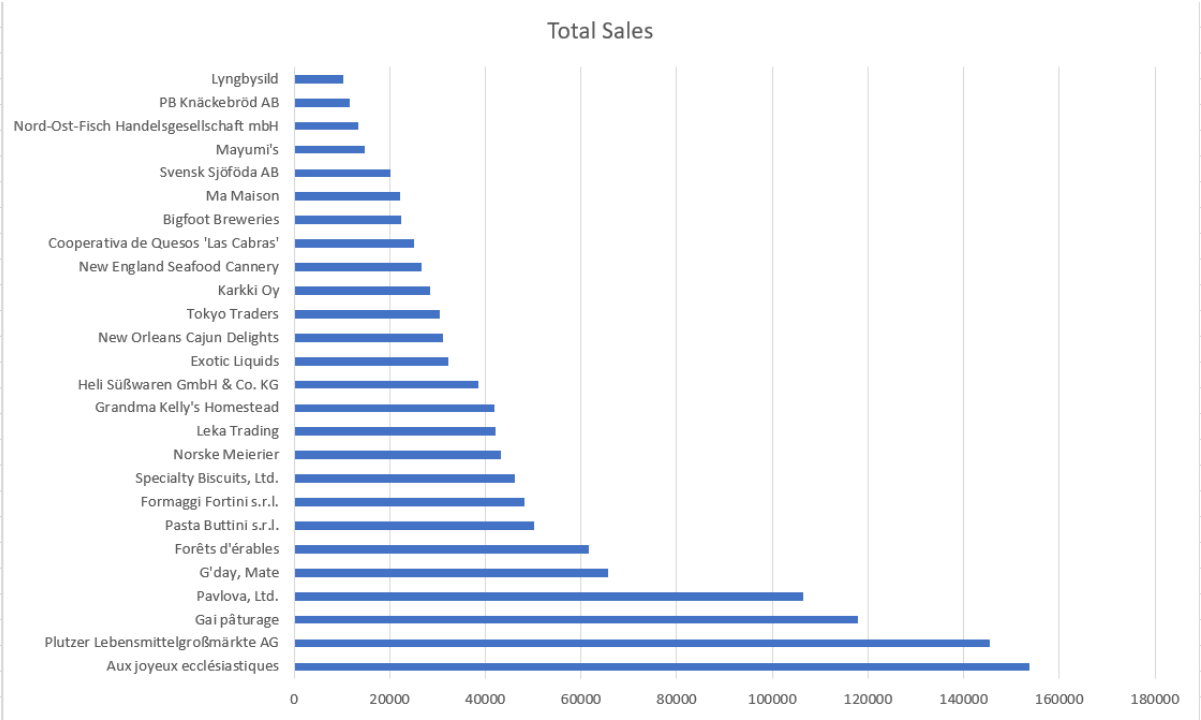


*Figure 24 - Suppliers with over $10,000 in total sales*

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

```
-- 3.3
-- Lastest year is 1998
SELECT TOP 10 c.CompanyName,
SUM((od.UnitPrice * od.Quantity)*(1-od.Discount)) AS "Total Sales"
FROM [Order Details] od
JOIN Orders o ON o.OrderID = od.OrderID
JOIN Customers c ON c.CustomerID = o.CustomerID
WHERE o.OrderDate >= '1998-01-01'
GROUP BY c.CompanyName
ORDER BY "Total Sales" DESC
```

*Figure 25 - 3.3 Code*

| | CompanyName | Total Sales |
|---|---|---|
| 1 | Ernst Handel | 41210.65002441406 |
| 2 | QUICK-Stop | 37217.315002441406 |
| 3 | Save-a-lot Markets | 36310.10977935791 |
| 4 | Hanari Carnes | 23821.199989318848 |
| 5 | Rattlesnake Canyon Grocery | 21238.270441055298 |
| 6 | Hungry Owl All-Night Grocers | 20402.11993408203 |
| 7 | Königlich Essen | 19582.773986816406 |
| 8 | White Clover Markets | 15278.89998626709 |
| 9 | Folk och fä HB | 13644.067497253418 |
| 10 | Suprêmes délices | 11644.599998474121 |

*Figure 26 - 3.3 Results*

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

```
-- 3.4
SELECT LEFT(CONVERT(VARCHAR(10), o.OrderDate, 111), 7) AS "Year/Month",
AVG(DATEDIFF(day, o.OrderDate, o.ShippedDate)) AS "Average Ship Time"
FROM Orders o
WHERE o.ShippedDate IS NOT NULL
GROUP BY LEFT(CONVERT(VARCHAR(10), o.OrderDate, 111), 7)
ORDER BY LEFT(CONVERT(VARCHAR(10), o.OrderDate, 111), 7)
```

*Figure 27 - 3.4 Code*

| | Year/Month | Average Ship Time |
|---|---|---|
| 1 | 1996/07 | 8 |
| 2 | 1996/08 | 8 |
| 3 | 1996/09 | 10 |
| 4 | 1996/10 | 6 |
| 5 | 1996/11 | 8 |
| 6 | 1996/12 | 7 |
| 7 | 1997/01 | 9 |
| 8 | 1997/02 | 9 |
| 9 | 1997/03 | 8 |
| 10 | 1997/04 | 9 |

*Figure 28 - 3.4 Results Sample*

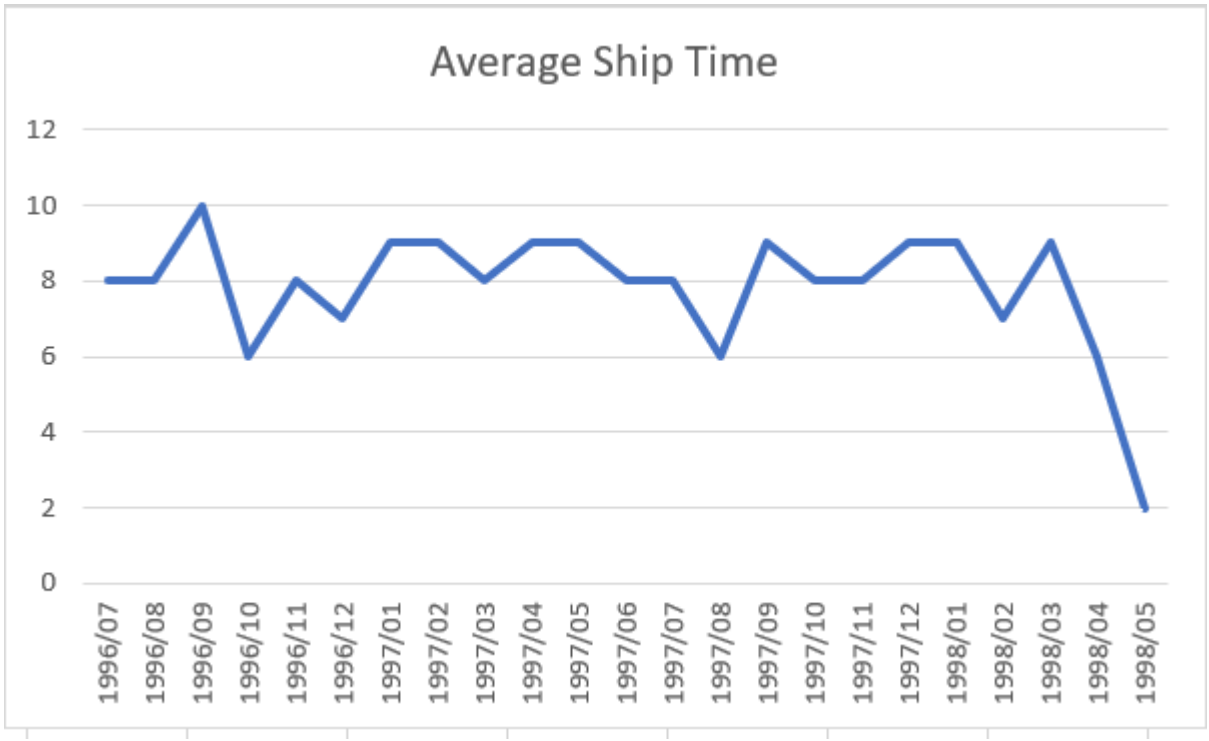


*Figure 29 - Average Ship Time per Month*