

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

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 2 - 1.2 Code

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 3 - 1.3 Code

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 4 - 1.4 Code

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 5 - 1.5 Code

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 6 - 1.6 Code

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(*)
FROM Orders
WHERE (Freight > 100 AND (ShipCountry = 'USA' OR ShipCountry = 'UK'))
```

Figure 7 - 1.7 Code

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 8 - 1.8 Code

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 9 - 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 10 - 2.2 Code

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 11 - 3.1 Code

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 12 - 3.2 Code

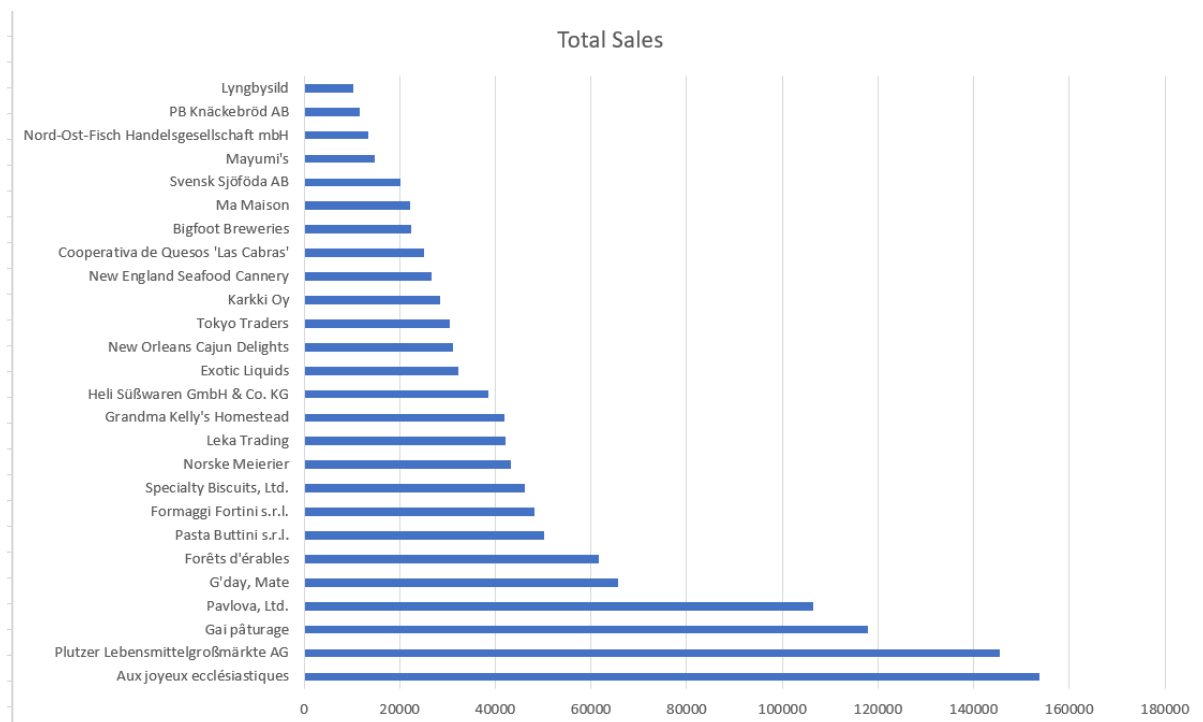


Figure 13 - Suppliers with over \$10,000 in total sales

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 14 - 3.3 Code

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 15 - 3.4 Code

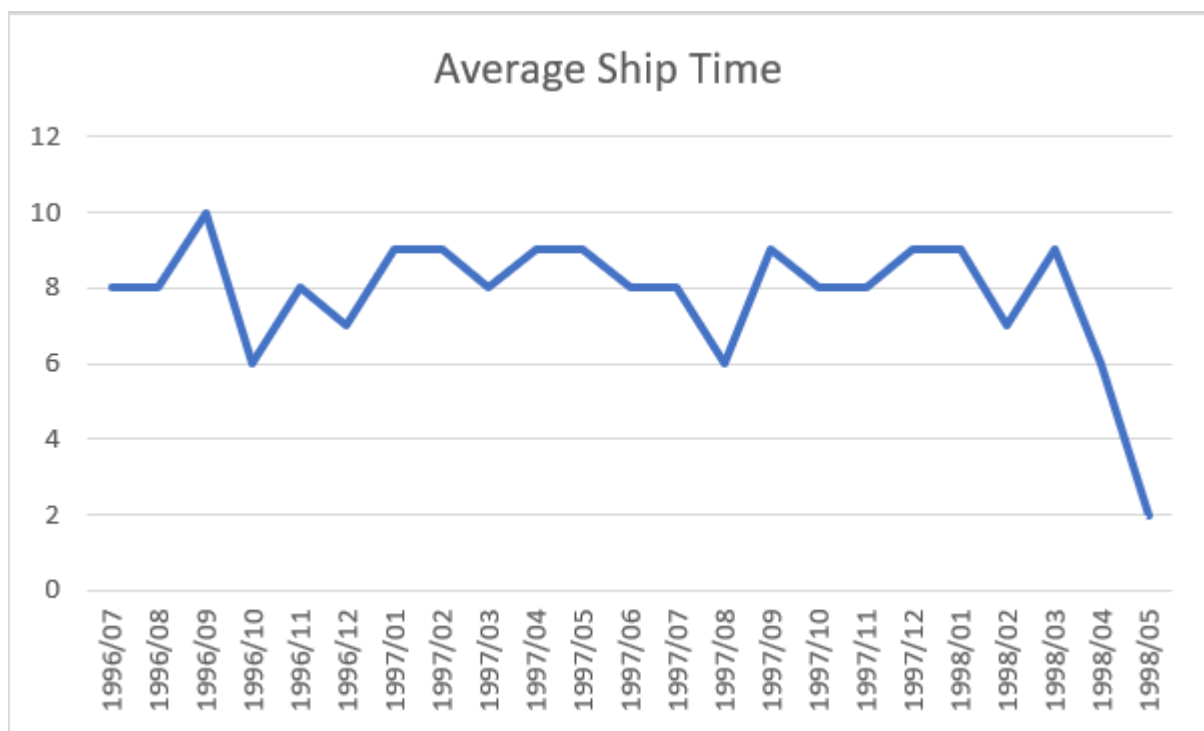


Figure 16 - Average Ship Time per Month