SQL PRACTICAL EXERCISE

PATRICK WALSH

pwalsh@spartaglobal.com · www.linkedin.com/in/patrick-walsh-jsdet

This exercise requires you to know the following aspects of SQL:

CREATE TABLE	Concatenation
SQL Data Types	Formatting dates and numbers
INSERT INTO	Column aliases
SELECT	Simple JOIN statements
WHERE clause	Complex JOIN statements
LIKE and wildcards	Subquery

EXERCISE 1 - NORTHWIND QUERIES

EXERCISE 1.1

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

```
SELECT CustomerID, CompanyName, Address, City, Region, PostalCode, Country
FROM Customers
WHERE City IN ('Paris','London')
```

EXERCISE 1.2

List all products stored in bottles.

```
SELECT *
FROM Products
WHERE QuantityPerUnit LIKE '%bottles%'
```

EXERCISE 1.3

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

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

EXERCISE 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 COUNT(*) AS "NumberOfProductsInEachCategory", c.CategoryName
FROM Products p
INNER JOIN Categories c
ON p.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY COUNT(*) DESC
```

EXERCISE 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 CONCAT(e.TitleOfCourtesy, e.FirstName,' ',e.LastName,', ',e.City)
FROM Employees e
WHERE Country = 'UK'
```

EXERCISE 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 R.RegionDescription, ROUND(SUM(UnitPrice * Quantity * (1-
Discount)),2) AS "SALES TOTALS"
FROM [Order Details] od
INNER JOIN Orders o ON od.OrderID = o.OrderID
INNER JOIN EmployeeTerritories et ON o.EmployeeID = et.EmployeeID
INNER JOIN Territories t ON et.TerritoryID = t.TerritoryID
INNER JOIN Region r ON t.RegionID = r.RegionID
GROUP BY R.RegionDescription
HAVING SUM(UnitPrice * Quantity * (1-Discount)) > 1000000
```

EXERCISE 1.7

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

```
SELECT COUNT(*) AS "Freight more than 100 with USA/UK as Ship Country"
FROM ORDERS
WHERE Freight > 100 AND ShipCountry IN ('USA','UK')
```

EXERCISE 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.

```
SELECT TOP 1 OrderID,  ROUND(SUM(UnitPrice * Quantity * (Discount)),2) AS "AMOUNT OF DISCOUNT"
FROM [Order Details]
GROUP BY OrderID
ORDER BY SUM(UnitPrice * Quantity * (Discount)) DESC
```

EXERCISE 2 – CREATE SPARTANS TABLE

EXERCISE 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.
IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.

```
CREATE TABLE spartans(
    spartan_id int PRIMARY KEY IDENTITY(1,1),
    title VARCHAR(20),
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    university VARCHAR(255),
    course VARCHAR(255),
    grade VARCHAR(20)
)
```

EXERCISE 2.2

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

```
INSERT INTO spartans(title, first_name, last_name, university, course, grade)
VALUES('title', 'Patrick','Walsh', 'London Metropolitan', 'Computer Science', '1st')
```

EXERCISE 3 – NORTHWIND DATA ANALYSIS LINKED TO EXCEL

EXERCISE 3.1

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

```
SELECT CONCAT(E.FirstName,' ',E.LastName,' REPORTS TO '),M.FirstName,M.LastName
FROM Employees e
LEFT JOIN Employees m
ON E.ReportsTo = M.EmployeeID
```

EXERCISE 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 (Fig 1)

```
SELECT S.CompanyName, ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2)
FROM [Order Details] od
INNER JOIN Products p
ON od.ProductID = p.ProductID
INNER JOIN Suppliers s
ON P.SupplierID = S.SupplierID
GROUP BY S.CompanyName
HAVING SUM(od.UnitPrice*od.Quantity*(1-od.Discount)) > 10000
ORDER BY SUM(od.UnitPrice*od.Quantity*(1-od.Discount))
```

EXERCISE 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.

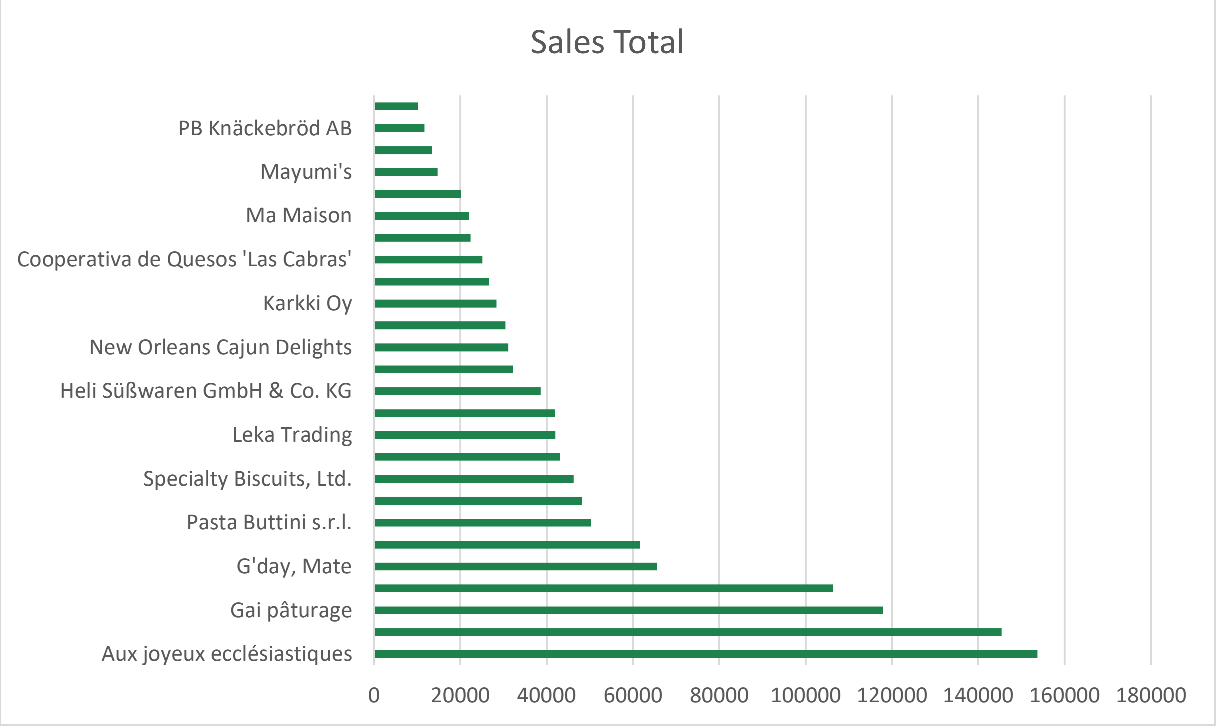
```
SELECT TOP 10 c.CompanyName, ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2)
FROM [Order Details] od
INNER JOIN Orders o
ON od.OrderID = o.OrderID
INNER JOIN Customers c
ON o.CustomerID = c.CustomerID

WHERE YEAR(O.OrderDate) = (SELECT TOP 1 YEAR(OrderDate)
                           FROM ORDERS
                           ORDER BY YEAR(OrderDate)DESC)

GROUP BY c.CompanyName
ORDER BY ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) DESC
```

EXERCISE 3.4

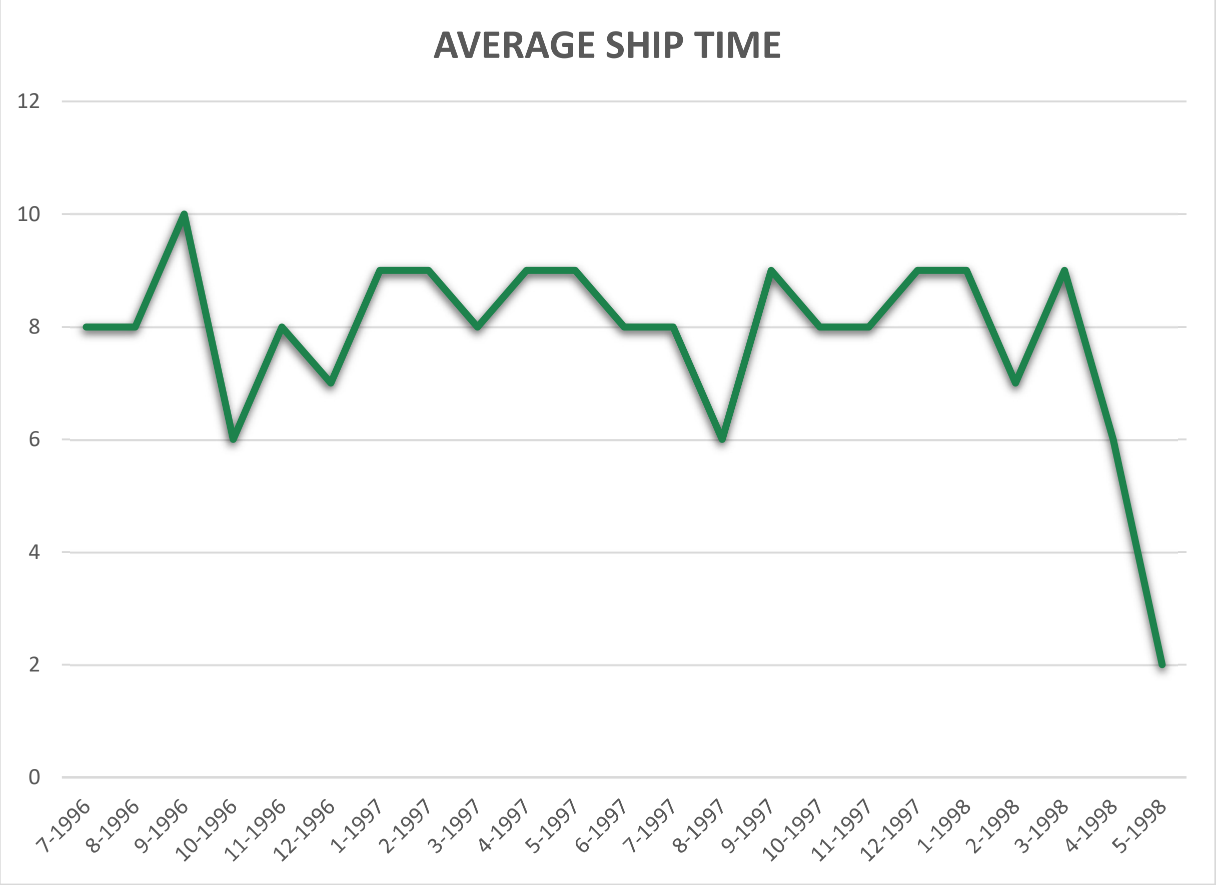
Plot the Average Ship Time by month for all data in the Orders Table using a line chart (Fig 2)

```
SELECT MONTH(OrderDate), YEAR(OrderDate), SUM(DATEDIFF(D,OrderDate,ShippedDate))
FROM ORDERS
WHERE ShippedDate IS NOT NULL
GROUP BY MONTH(OrderDate), YEAR(OrderDate)
```

EXERCISE 3.2 - FIGURE 1



EXERCISE 3.4 – FIRGURE 2


AVERAGE SHIP TIME