Trainee Name: John Byrne **Subject:** SQL **Trainer:** Astha Shaw

Mini-Project

Set: 26/06/2020 **Due:** 29/06/2020

Following Aspects of SQL are included in exercises

Exercise 1 – Northwind Queries (40 marks: 5 for each question)

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

```
01. SELECT c.CustomerID, c.CompanyName, CONCAT(c.Address, ', ', c.City, ', ', c.PostalCode, ', ' , c.Country) AS "Address"
02. FROM Customers c
03. WHERE c.City = 'Paris' OR c.City = 'London'
```

Answer

```
01. SELECT CustomerID AS “Customer ID”, CompanyName AS “Customer Name”, Address + ‘, ‘ + City + ‘, ‘ + PostalCode + ‘, ‘ + Country AS “Address”
02. FROM Customers
03.     WHERE City IN ('Paris', 'London');
```

1.2	List all products stored in bottles.

```
01. SELECT p.ProductName
02. FROM Products p
03. WHERE p.QuantityPerUnit LIKE '%bottle%'
```

Answer

```
01. SELECT ProductName, QuantityPerUnit
02. FROM Products
03.     WHERE QuantityPerUnit LIKE '%Bottle%';
```

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

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

Answer

```
01. SELECT ProductName, QuantityPerUnit, CompanyName, Country
02.     FROM Products p
03.             INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
04.     WHERE QuantityPerUnit LIKE '%Bottle%';
```

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, COUNT(p.ProductID) AS "Products in Category"
FROM Products p
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY [Products in Category] DESC
```

Answer

```
SELECT  c.CategoryName "Category Name", COUNT(*) as "No of Products"
FROM Products p
        INNER JOIN Categories c ON p.CategoryID=c.CategoryID
GROUP BY c.CategoryName
ORDER BY COUNT(*) 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 CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName, ' From ', e.City) AS "UK Employees"
FROM Employees e
WHERE e.Country = 'UK'
```

Answer

```
SELECT TitleOfCourtesy + ' ' + FirstName + ' ' + LastName As Employee, City
  FROM Employees
  WHERE 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 ROUND(SUM((od.UnitPrice * od.Quantity) - (od.UnitPrice * od.Discount * od.Quantity)), 2) AS "Total Sales",
r.RegionDescription
FROM [Order Details] od
INNER JOIN Orders o ON o.OrderID = od.OrderID
INNER JOIN EmployeeTerritories et ON et.EmployeeID = o.EmployeeID
INNER JOIN Territories t ON t.TerritoryID = et.TerritoryID
INNER JOIN Region r ON r.RegionID = t.RegionID
GROUP BY r.RegionDescription
HAVING ROUND(SUM((od.UnitPrice * od.Quantity) - (od.UnitPrice * od.Discount * od.Quantity)), 2)  > 1000000
```

Answer

```
SELECT r.RegionID, r.RegionDescription AS Region,
FORMAT(SUM((UnitPrice * Quantity) * (1-Discount)),'C')
AS "Sales Total by Region"
    FROM Orders AS o
        INNER JOIN [Order Details] AS od ON od.OrderID = o.OrderID
        INNER JOIN EmployeeTerritories AS et ON o.EmployeeID = et.EmployeeID
        INNER JOIN Territories AS t ON et.TerritoryID = t.TerritoryID
        INNER JOIN Region AS r ON t.RegionID = r.RegionID
    GROUP BY r.RegionDescription, r.RegionID
    HAVING SUM((UnitPrice * Quantity) * (1-Discount)) > 1000000
    ORDER BY "Sales Total by Region" DESC;
```

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.orderID) AS "Frieght Greater than 100.00"
FROM Orders o
WHERE o.Freight > 100.000 AND o.ShipCountry IN ('UK', 'USA')
```

Answer

```sql
SELECT COUNT(*) AS 'No of Orders >100 from US or UK'
    FROM Orders
  WHERE Freight>100 AND ShipCountry IN ('USA','UK');
```

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, od.UnitPrice, od.Quantity, od.Discount, (od.UnitPrice * od.Discount * od.Quantity) AS "Total Discount"
FROM [Order Details] od
ORDER BY [Total Discount] DESC
```

Answer

```sql
SELECT OrderID AS 'Order ID',
       FORMAT((UnitPrice * Quantity) * Discount,'C') AS 'Discount Amount'
    FROM [Order Details]
    ORDER BY [Discount Amount] DESC;
```

Exercise 2 - Create Spartans Table (20 marks – 10 each)

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 DATABASE Mini_projectDB_John

CREATE TABLE [Spartans] (
    [SpartansID] INTEGER NOT NULL IDENTITY(1, 1),
    [Title] VARCHAR(255) NULL,
    [FirstName] VARCHAR(255) NULL,
    [Surname] VARCHAR(255) NULL,
    [University] VARCHAR(255) NULL,
    [Course] VARCHAR(255) NULL,
    [Mark] VARCHAR(255) NULL,
    PRIMARY KEY ([SpartansID])
);
```

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

```sql
INSERT INTO Spartans([Title],[FirstName],[Surname],[University],[Course],[Mark])
VALUES('Mrs.','Georgina','Bartlett','Newcastle University','Archaeology','2:1'),
('Mr.','Humza','Malak','University of Kent','Computing with Games Development','2:2'),
('Mr.','Ibrahim','Bocus','University of Leicester','Computer Science','2:1'),
('Mr.','Bari','Allali','Lancaster University','Business Economics','2:2'),
('Mr.','Nola','Alston','University of Warwick','International Business & Management','3:3'),
('Dr.','Aspen','Reed','University of Leicester','Computing with Games Development','3:3'),
('Ms.','Ezekiel','Espinoza','University of Greenwich','Product Design','2:2'),
('Mr.','Aretha','Berry','Newcastle University','Aerospace Engineering','1:1'),
('Dr.','Ivan','Harrell','Edinburgh','Computing with Games Development','2:1'),
('Mrs.','Sydnee','Evans','Aston University','International Business & Management','2:2');
```

Exercise 3 – Northwind Data Analysis linked to Excel (30 marks)

3.1	List all Employees from the Employees table and who they report to. No Excel required. (5 Marks)

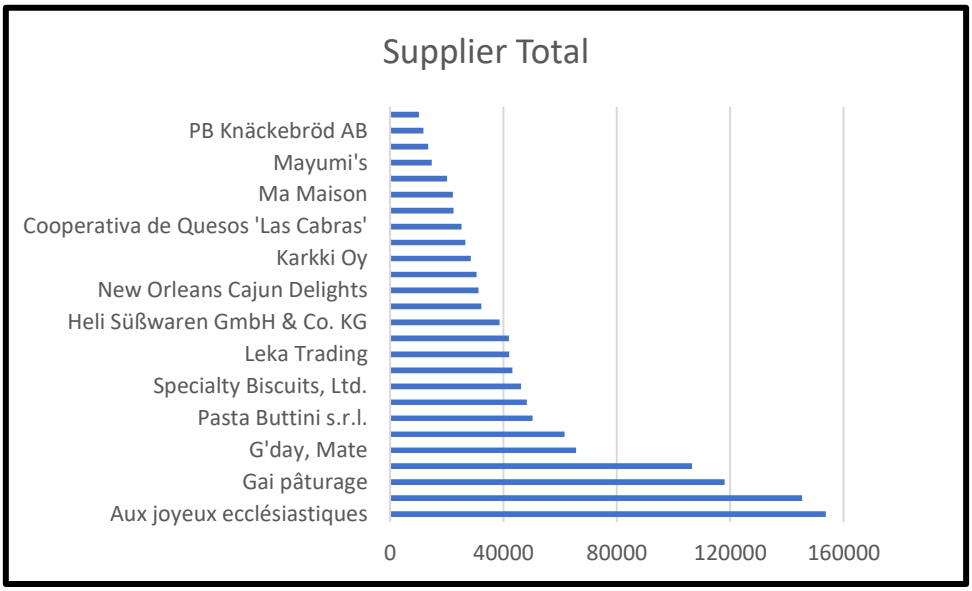
```sql
SELECT CONCAT(e1.FirstName,' ', e1.LastName, ' Reports to') AS "Employee", CONCAT(e2.FirstName,' ', e2.LastName) AS "Superior"
FROM Employees e1, Employees e2
WHERE e1.ReportsTo = e2.employeeID
```

Answer

```sql
SELECT e.FirstName + ' ' + e.LastName AS "Employee Name",
       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 and present as a bar chart as below: (5 Marks)

```sql
SELECT s.CompanyName, ROUND(SUM((od.UnitPrice * od.Quantity) - (od.UnitPrice * od.Discount * od.Quantity)), 2) AS "Total Sales"
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)) > 10000
```



Answer

```sql
SELECT s.CompanyName,SUM(od.UnitPrice*od.Quantity*(1-od.Discount)) As "Supplier Total"
    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)) DESC;
```

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. (10 Marks)

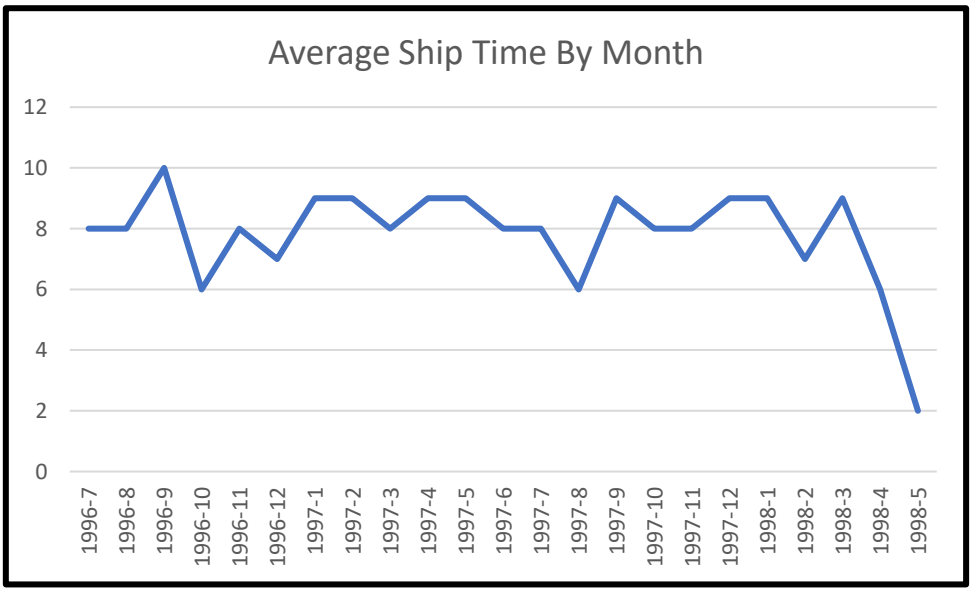
```
in DESC to reveal SELECT TOP 10
    c.CompanyName,
    ROUND(SUM((od.UnitPrice * od.Quantity) - (od.UnitPrice * od.Discount * od.Quantity)), 2) AS "Total value of orders shipped"
FROM Customers c
INNER JOIN Orders o ON c.CustomerID = o.CustomerID
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
WHERE YEAR(o.OrderDate) = (SELECT MAX(YEAR(orderDate)) FROM Orders) AND o.ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY [Total value of orders shipped] DESC
```

Answer

```
SELECT TOP 10 c.CustomerID AS "Customer ID", c.CompanyName As "Company",
FORMAT(SUM(UnitPrice * Quantity * (1-Discount)),'C')
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;
```

3.4	Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below. (10 Marks)

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



Answer

```
SELECT MONTH(OrderDate) Month, YEAR(OrderDate) Year,
AVG(CAST(DATEDIFF(d, OrderDate, ShippedDate) As DECIMAL(10,2))) As ShipTime
    FROM orders
    WHERE ShippedDate IS NOT NULL
    GROUP BY YEAR(OrderDate),MONTH(OrderDate)
    ORDER BY Year ASC, Month ASC
```