

SQL Northwind Mini Project

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

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

1.2 List all products stored in bottles.

```
SELECT p.ProductName FROM Products p
WHERE CHARINDEX('Bottle', p.QuantityPerUnit) > 0
```

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

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

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 AS "Category", COUNT(p.ProductID) AS "Num. of Products" FROM Products p
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
GROUP BY c.CategoryName
ORDER BY "Num. of Products" 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) AS "Employees", e.City
FROM Employees e
WHERE e.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 r.RegionDescription AS "Region", SUM((od.UnitPrice * (1-od.Discount)) * od.Quantity) AS
"SalesTotal"
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 SUM((od.UnitPrice * (1-od.Discount)) * od.Quantity) > 1000000
```

1.7 Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as the ship country

```
SELECT COUNT(o.Freight) AS "Count" FROM Orders o
WHERE o.Freight > 100.00 AND (o.ShipCountry = 'UK' OR o.ShipCountry = 'USA')
```

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 od.OrderID AS "Order Num.", SUM((od.UnitPrice * od.Discount) * od.Quantity) AS
"Discounted"
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY [Discounted] DESC
```

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.

```
CREATE TABLE Spartans
(
    SpartanID INTEGER IDENTITY(1,1) PRIMARY KEY,
    Title VARCHAR(50),
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    UniversityAttended VARCHAR(50),
    CourseTaken VARCHAR(50),
    GradeAchieved VARCHAR(50)
)
```

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, FirstName, LastName, UniversityAttended, CourseTaken, GradeAchieved)
VALUES
    ('Mr', 'Aaron', 'Banjoko', 'University for the Creative Arts', 'Music & Management', '1:1'),
    ('Mr', 'Dominic', 'Cogan-Tucker', 'University of Birmingham', 'Engineering', '2:1'),
    ('Mr', 'Joel', 'Fright', 'University of Hull', 'Physics', '2:1'),
    ('Mr', 'Kurtis', 'Hanson', 'University of Lincoln', 'Computer Security', '1:1'),
    ('Mr', 'Malik', 'Shams', 'Leicester University', 'Mathematics', '1:1')
```

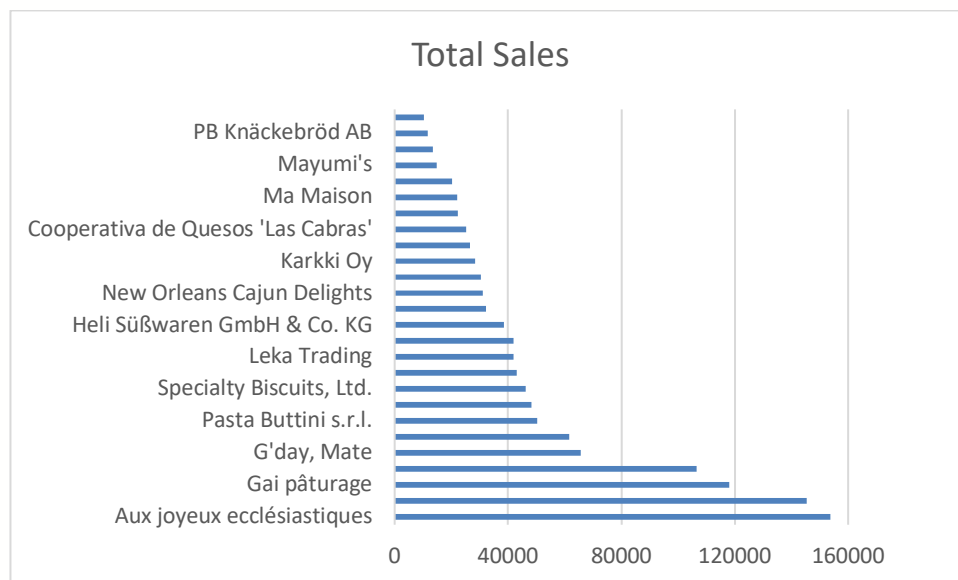
Write SQL statements to extract the data required for the following charts (create these in Excel):

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

```
SELECT CONCAT(e.FirstName, ' ', e.LastName) AS "Employees", r.FirstName + ' ' + r.LastName AS  
"Reports To"  
FROM Employees e  
LEFT JOIN Employees r ON e.ReportsTo = r.EmployeeID
```

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)

```
SELECT s.SupplierID AS "Supplier ID", s.CompanyName AS "Company Name", FORMAT(SUM((od.UnitPrice *  
(1-od.Discount)) * od.Quantity), 'C') AS "Total Sales"  
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.SupplierID, s.CompanyName  
HAVING SUM((od.UnitPrice * (1-od.Discount)) * od.Quantity) > 10000  
ORDER BY SUM((od.UnitPrice * (1-od.Discount)) * od.Quantity) 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)

```
SELECT TOP 10 c.CompanyName AS "Company", o.OrderDate AS "Ordered On", FORMAT(SUM((od.UnitPrice *  
od.Discount) * od.Quantity), 'C') AS "Total Value"  
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 YEAR(MAX(o.OrderDate)) FROM Orders o)  
    AND o.ShippedDate IS NOT NULL  
GROUP BY c.CompanyName, o.OrderDate  
ORDER BY [Total Value] 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(MONTH(o.OrderDate), '/', YEAR(o.OrderDate)) AS "Date", AVG(DATEDIFF(DAY, o.OrderDate, o.ShippedDate)) AS "Average"
FROM Orders o
GROUP BY MONTH(o.OrderDate), YEAR(o.OrderDate)
```

