SQL Mini Project

Exercise 1:

1.1 Employees from London or Paris

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

1.2 Products stored in bottles

```
SELECT *
FROM Products
WHERE QuantityPerUnit LIKE '%bottle%';
```

1.3 Joining previous with the Suppliers table to show the company name and country

```
-- Removed duplicates.
SELECT DISTINCT sup.CompanyName, sup.Country
FROM Products prod
    INNER JOIN Suppliers sup ON prod.SupplierID = sup.SupplierID
WHERE prod.QuantityPerUnit LIKE '%bottle%';
```

1.4 Number of products in each category

```
-- Showing the highest first.
SELECT cat.CategoryID, cat.CategoryName, COUNT(*) AS "Number of Products"
FROM Products prod
    INNER JOIN Categories cat ON prod.CategoryID = cat.CategoryID
GROUP BY cat.CategoryID, cat.CategoryName
ORDER BY "Number of Products" DESC;
```

1.5 Employees from the UK

```
SELECT CONCAT(TitleOfCourtesy, ' ', FirstName, ' ', LastName) AS "Employee Name",
    City
FROM Employees
WHERE Country = 'UK';
```

1.6 Sales Regions with a Sales Total of more than 1,000,000

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

1.7 Orders with a Freight amount greater than 100 and has either the UK or USA as the ship country

```
SELECT COUNT(*) AS "No. of Orders"
FROM Orders
WHERE Freight > 100.00
    AND ShipCountry IN ('UK', 'USA');
```

1.8 Orders with the highest applied discount

```
-- Some orders have multiple products.
SELECT TOP 1 OrderID,
    FORMAT(SUM(UnitPrice * (1 - Discount) * Quantity), 'C') AS "Money Discounted"
FROM [Order Details]
GROUP BY OrderID
ORDER BY SUM(UnitPrice * (1 - Discount) * Quantity) DESC;
```

Exercise 2:

2.1 Creating the Spartans table

```
CREATE TABLE spartan_table (
    spartan_id INT IDENTITY(1,1) PRIMARY KEY,
    title VARCHAR(5),
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    university VARCHAR(20),
    course VARCHAR(20),
    mark_achieved CHAR(3),
    tech_stream VARCHAR(20)
);
```

2.2 Inserting records for various Spartans

```
INSERT INTO spartan_table
(title, first_name, last_name, university, course, mark_achieved, tech_stream)
VALUES
('Mr', 'William', 'King', 'Swansea', 'Computer Science', '1st', 'DevOps'),
('Mr', 'Benjamin', 'Ranson', 'Essex', 'Computer Science', '1st', 'DevOps'),
('Mr', 'Andrew', 'Asare', 'London Metropolitan', 'Computer Science', '2:1', 'DevOps'),
('Mrs', 'Dunni', 'Adebusuyi', 'Goldsmiths', 'Computer Science', '2:2', 'DevOps'),
('Mr', 'Arun', 'Panesar', 'De Montfort', 'Software Engineering', '1st', 'DevOps'),
('Mr', 'Ayaz', 'Yar', 'Exeter', 'PPE', '2:1', 'DevOps'),
('Mr', 'Jordan', 'Clarke', 'Salford', 'Physics', '2:1', 'DevOps'),
('Mr', 'Jose', 'Torres', 'Madrid', 'Computer Science', '2:1', 'DevOps');
```

Exercise 3:

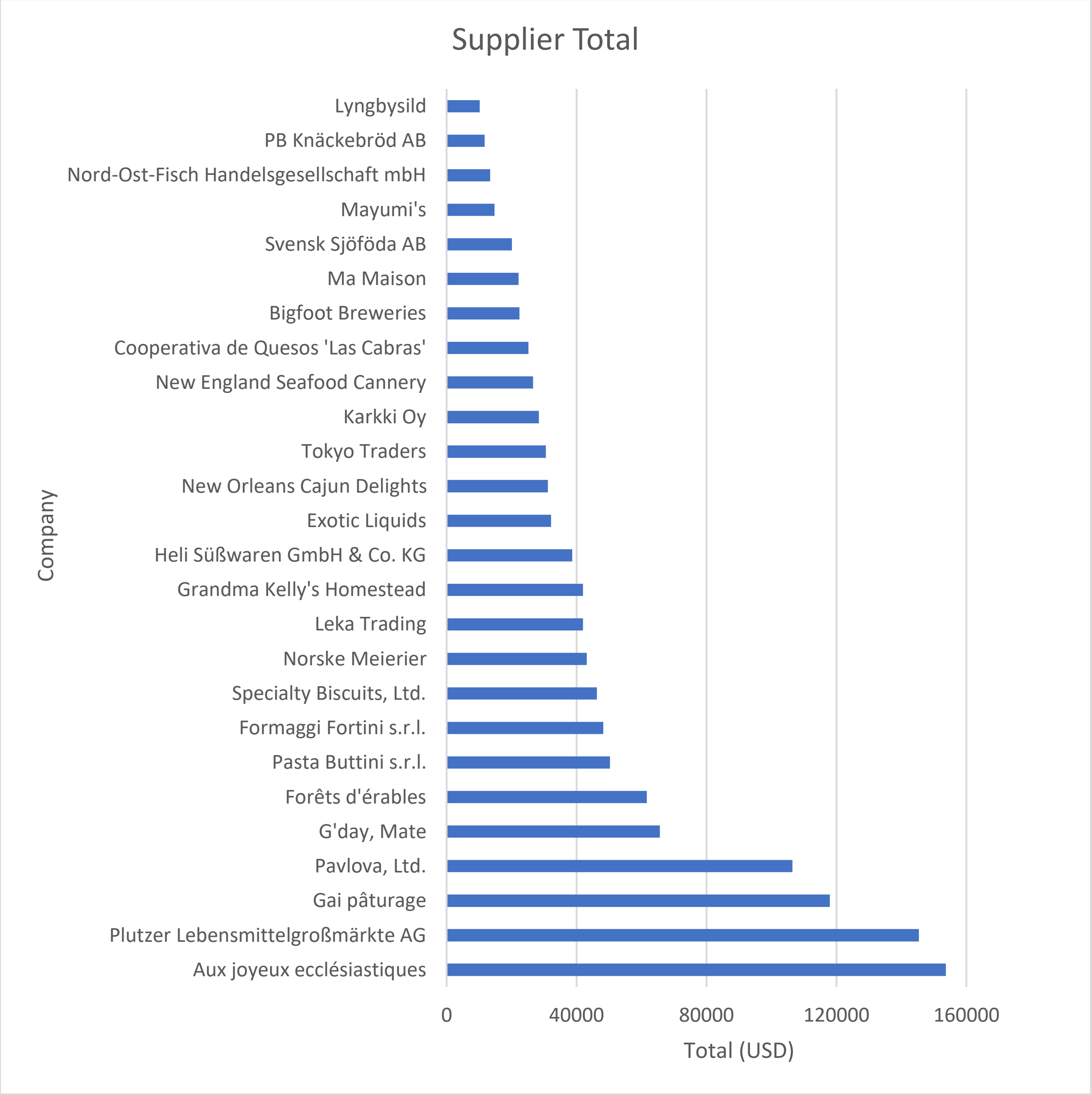
3.1 All employees and who they report to

```
SELECT emp.FirstName + ' ' + emp.LastName AS "Employee Name",
    man.FirstName + ' ' + man.LastName AS "Reports To"
FROM Employees emp
    LEFT JOIN Employees man ON emp.ReportsTo = man.EmployeeID;
```

3.2 Suppliers with a Total Sales over $10,000

```
-- Discount applied.
SELECT s.CompanyName AS "Company",
    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;
```

3.2 Excel Table:



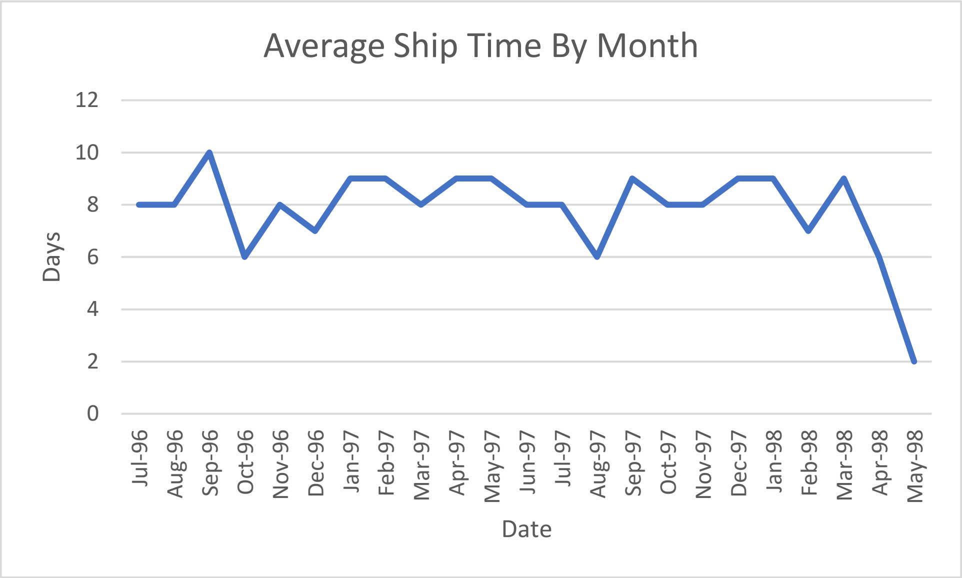
3.3 Top 10 Customers YTD for the latest year, based on the total value of orders shipped

```sql
-- Only using records with a shipping date.
SELECT TOP 10 o.CustomerID,
    FORMAT(SUM(od.UnitPrice * (1 - od.Discount) * od.Quantity), 'C')
    AS "Total Value of Orders in YTD"
FROM Orders o
    INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
WHERE o.ShippedDate IS NOT NULL
    AND YEAR(o.OrderDate) IN (SELECT MAX(YEAR(OrderDate))
                              FROM Orders)
GROUP BY o.CustomerID
ORDER BY SUM(od.UnitPrice * (1 - od.Discount) * od.Quantity) DESC;
```

3.4 Average ship time by month

```sql
-- Only using records with a shipping date.
SELECT CONCAT(MONTH(OrderDate), '-', YEAR(OrderDate)) AS "Date",
    AVG(DATEDIFF(day, OrderDate, ShippedDate)) AS "Average Ship Time (Days)"
FROM Orders
WHERE ShippedDate IS NOT NULL
GROUP BY YEAR(OrderDate), MONTH(OrderDate)
ORDER BY YEAR(OrderDate), MONTH(OrderDate) ASC;
```

3.4 Excel Table (months converted in Excel):


Average Ship Time By Month
Days
Jul-96
Aug-96
Sep-96
Oct-96
Nov-96
Dec-96
Jan-97
Feb-97
Mar-97
Apr-97
May-97
Jun-97
Jul-97
Aug-97
Sep-97
Oct-97
Nov-97
Dec-97
Jan-98
Feb-98
Mar-98
Apr-98
May-98
Date