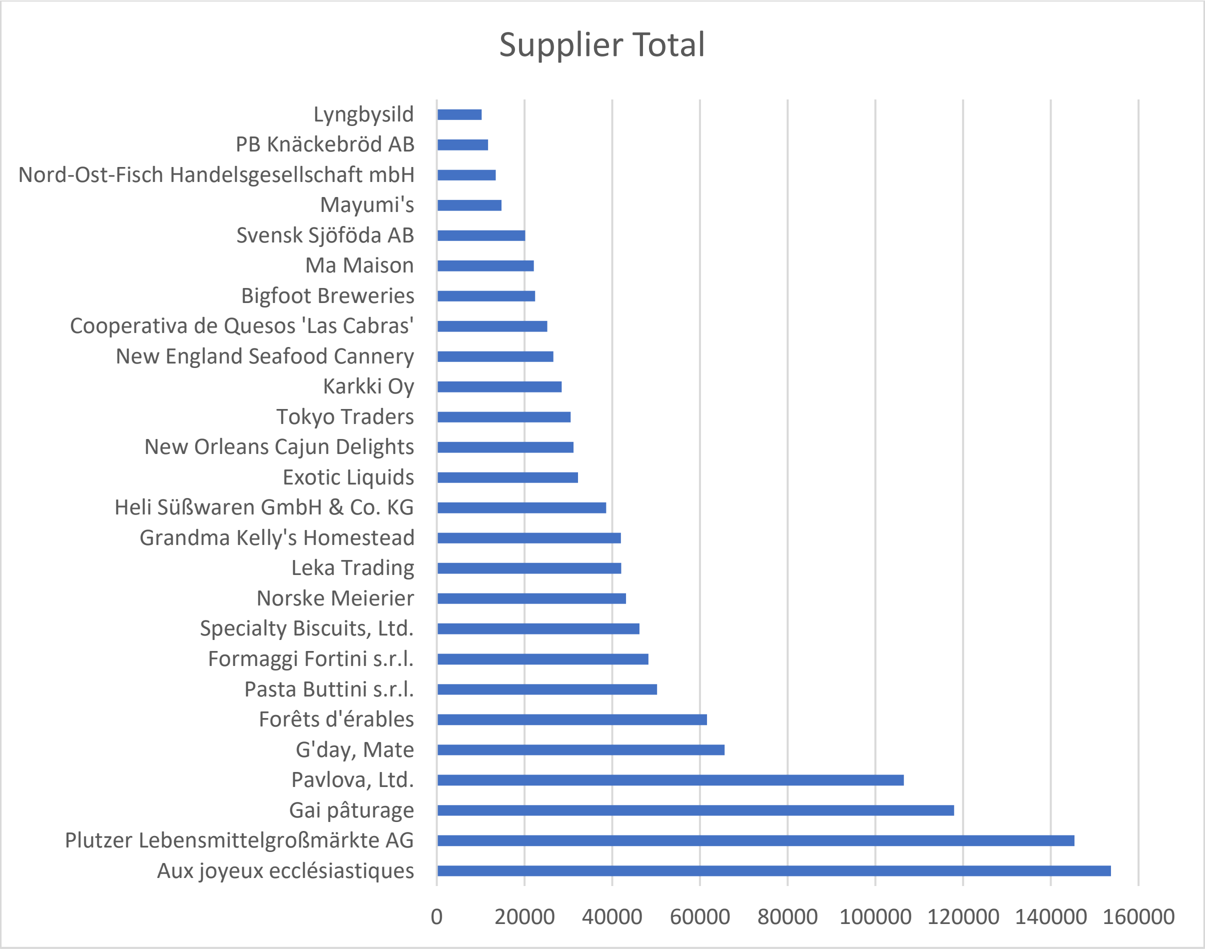
```sql
-- Exercise 1
-- Northwind Queries

USE Northwind


-- 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 AS "Customer ID", c.CompanyName AS "Company Name", c.Address,
    c.City, c.Region, c.Country, c.PostalCode AS "Post Code"
FROM Customers c
WHERE c.city IN ('Paris', 'London')


-- 1.2
-- List all products stored in bottles.

SELECT p.ProductName AS "Product Name"
FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottle%'


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

SELECT p.ProductName AS "Product Name", s.CompanyName AS "Supplier Name", s.Country
FROM Products p
INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
WHERE p.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 AS "Category Name",
    COUNT(c.CategoryName) AS "Products in Category"
FROM Products p
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
GROUP BY p.CategoryID, c.CategoryName
ORDER BY "Products in category" 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 e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Employee Name",
    e.City AS "City of Residence"
FROM Employees e
WHERE e.Country = 'UK'


-- 1.6
-- List Sales Totals for all Sales Regions (via the Territories table using 4 joins)
```

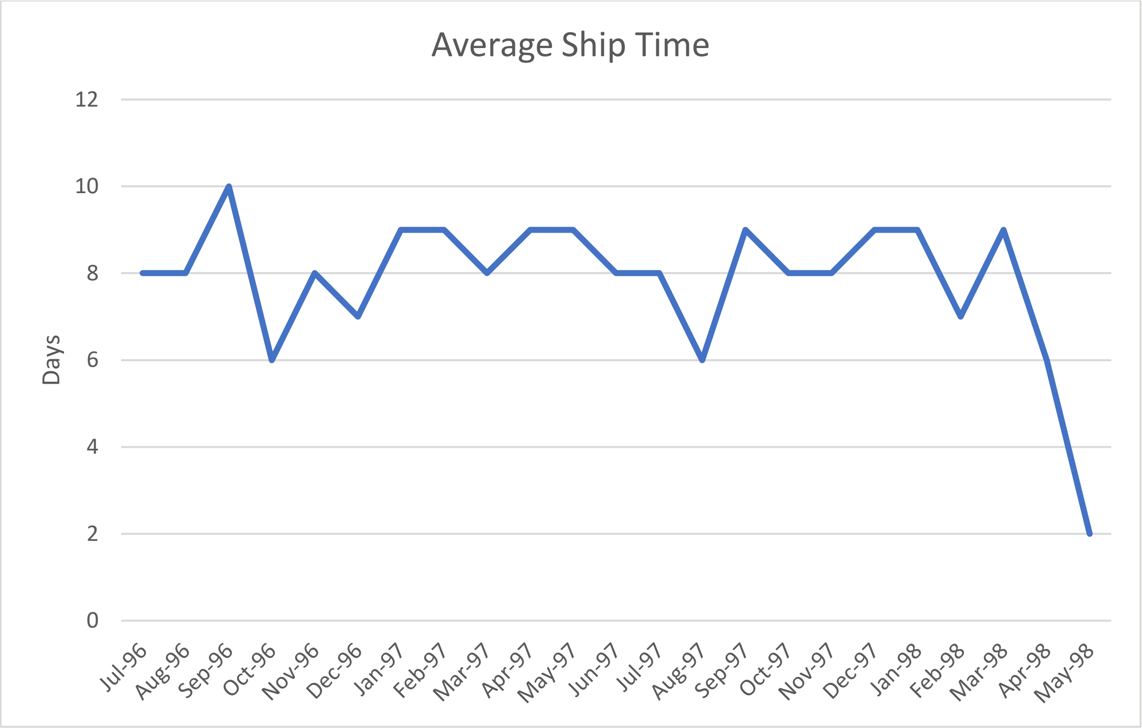
```
-
- with a Sales Total greater than 1,000,000. Use rounding or FORMAT to present the numbers.


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


-- 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 "Number of Orders"
FROM Orders o
WHERE o.Freight > 100.00 AND o.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.
SELECT TOP 1 od.OrderID AS "Order ID"
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY SUM(od.UnitPrice * od.Quantity * od.Discount) DESC



-- Exercise 2
-- Create Spartans Table

CREATE DATABASE isobel_db
USE isobel_db


-- 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_table
(
    spartan_id INT IDENTITY(1,1) PRIMARY KEY,
    title VARCHAR(5),
    first_name VARCHAR(10),
    last_name VARCHAR(15),
    university VARCHAR(20),
    course VARCHAR(20),
```

```sql
    mark VARCHAR(5)
)


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

INSERT INTO spartans_table
VALUES
('Ms.', 'Isobel', 'Fitt-Conway', 'St Andrews', 'Biology', '2:1'),
('Mr.', 'John', 'Smith', 'Edinburgh', 'Physics', '2:2'),
('Ms.', 'Katie','Brown', 'UCL', 'Computer Science', '1st'),
('Mr.', 'Simon', 'Scott', 'Manchester', 'French', '2:1')



-- Exercise 3
-- Northwind Data Analysis linked to Excel

USE Northwind


-- 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 e.FirstName + ' ' + e.LastName AS "Employee Name",
    e2.FirstName + ' ' + e2.LastName AS "Reports To"
FROM Employees e
LEFT JOIN Employees e2 ON e.ReportsTo = e2.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.

SELECT s.CompanyName AS "Company Name",
    ROUND(SUM(od.UnitPrice * od.Quantity * (1 - od.Discount)), 2) AS "Total Sales"
FROM Suppliers s
INNER JOIN Products p ON s.SupplierID = p.SupplierID
INNER 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
```



```
-- 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 AS "Company Name",
    ROUND(SUM(od.UnitPrice * od.Quantity * (1-
od.Discount)), 2) AS "Total Value of Orders (YTD)"
FROM Orders o
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE YEAR(o.OrderDate) = (
    SELECT MAX(YEAR(o.OrderDate))
    FROM Orders o
) AND o.ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY SUM(od.UnitPrice * od.Quantity * (1-od.Discount)) DESC


-- 3.4
-- Plot the Average Ship Time by month for all data in the Orders Table using a line chart.

SELECT FORMAT(o.OrderDate, 'MMM-yy') AS "Month",
    AVG(DATEDIFF(dd, o.OrderDate, o.ShippedDate)) AS "Average Ship Time"
FROM Orders o
WHERE o.ShippedDate IS NOT NULL
GROUP BY FORMAT(o.OrderDate, 'MMM-yy'), FORMAT(o.OrderDate, 'yy-MM')
```

```
ORDER BY FORMAT(o.OrderDate, 'yy-MM')
```


Average Ship Time
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