```
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 "Companies In London or Paris"
, c.Address
, c.City
, c.Country
, c.PostalCode AS "Post Code"
FROM Customers AS c
WHERE c.City IN ('London' , 'Paris')
```

1.2 List all products stored in bottles.

```
SELECT
p.ProductName AS "Products with Bottles"
FROM Products AS p
WHERE p.QuantityPerUnit LIKE '%bottles%'
```

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

```
SELECT p.ProductName AS "Products with Bottles"
, s.CompanyName AS "Supplier Name"
, s.Country
FROM Products AS p
INNER JOIN Suppliers AS 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 Cat egory Name in result set and list the highest number first.

```
SELECT p.CategoryID AS "Category ID"
, c.CategoryName AS "Category Name"
, COUNT(*)  AS "Number of Products in Category"
 FROM Products p
INNER JOIN Categories c
 ON c.CategoryID = p.CategoryID
GROUP BY p.CategoryID, c.CategoryName
ORDER BY "Number of Products in Category" DESC
```

1.5 List all UK employees using concatenation to join their title of courtesy, first name and last na me together. Also include their city of residence.

```
SELECT
e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Employee Full Name",
e.city AS "City of Residence"
FROM Employees AS 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 gr eater than 1,000,000. Use rounding or FORMAT to present the numbers.

```
SELECT r.regionDescription
, ROUND(SUM(od.UnitPrice * od.Quantity * (1- od.Discount)), 2) AS "Sales Total"
FROM [Order Details] od
INNER JOIN Orders o
ON od.OrderID = o.OrderID
INNER JOIN Employees e
ON e.EmployeeID = o.EmployeeID
INNER JOIN EmployeeTerritories et
ON et.EmployeeID = e.EmployeeID
INNER JOIN Territories t
ON et.TerritoryID = t.TerritoryID
INNER JOIN Region r
ON r.RegionID = t.RegionID
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 Sh ip Country.

```
SELECT
COUNT(*) AS "Number of UK or USA Orders with more than 100 Freight"
FROM Orders o
WHERE o.Freight > 100 AND o.ShipCountry IN ('USA', 'UK')
```

1.8 Write an SQL Statement to identify the Order Number of the Order with the highest amount(v alue) of discount applied to that order.

```
SELECT
o.OrderID AS "Order ID of Order(s) with Highest Discount"
, od.UnitPrice * od.Quantity * od.Discount AS "Discount"
FROM Orders o
INNER JOIN [Order Details] od
ON od.OrderID = o.OrderID
WHERE od.UnitPrice * od.Quantity * od.Discount = ( SELECT MAX(od.UnitPrice * od.Quantity *
od.Discount) FROM[Order Details] od )
ORDER BY od.UnitPrice * od.Quantity * od.Discount 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 DATABASE timin_db
USE timin_db
DROP TABLE spartans_table
CREATE TABLE spartans_table(
    title VARCHAR(20),
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    uni_attended VARCHAR(30),
    courses_taken VARCHAR(20),
    mark_achieved INT
    )
```

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

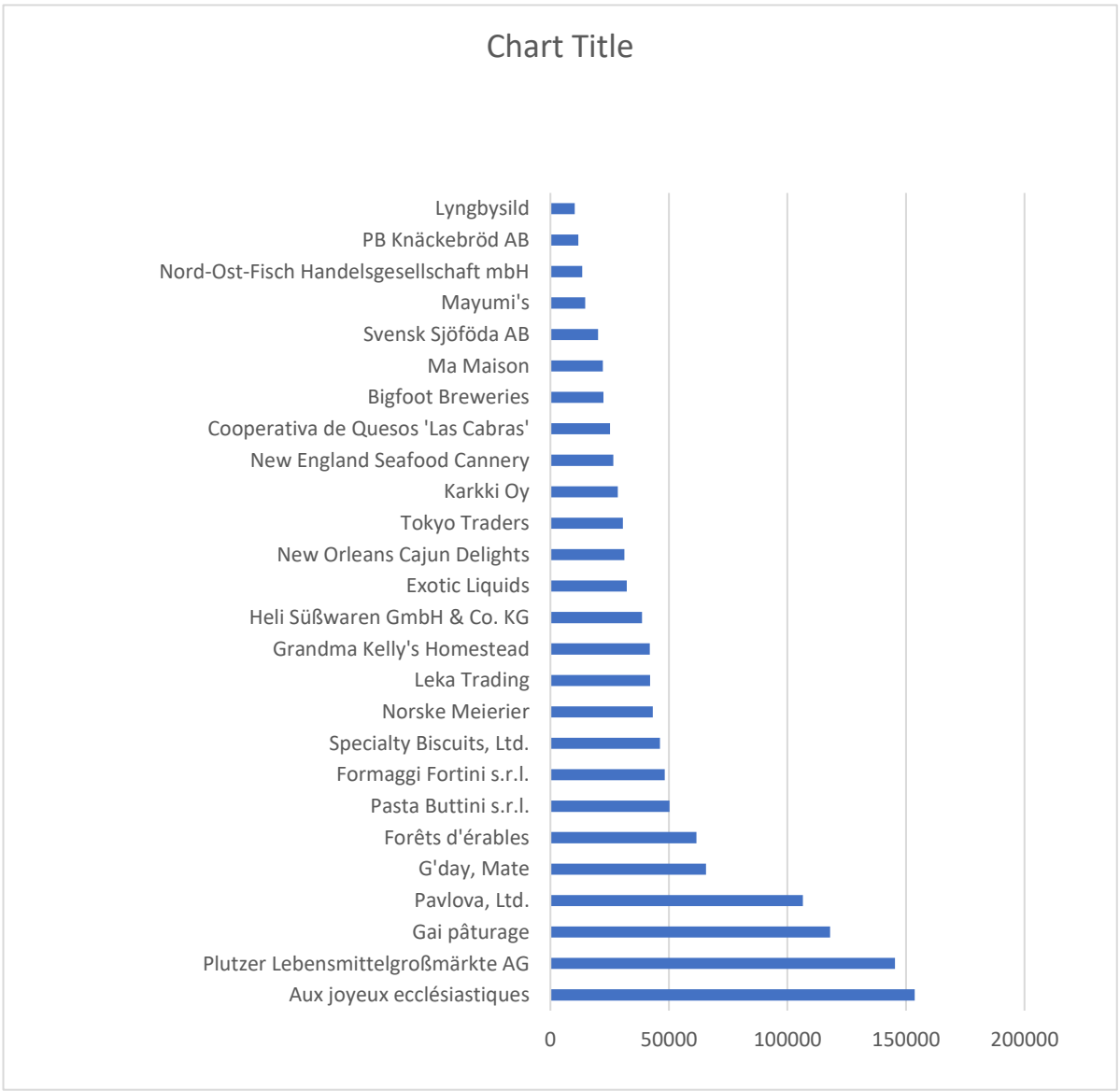
```
INSERT INTO spartans_table
VALUES
    ('Mr', 'Alex', 'Ng', 'UOA', 'PE, Music, Dance', 0),
    ('Mr', 'Ahmed', 'Rahman', 'UOB', 'PE, Dance', 10),
    ('Mr', 'Andrei', 'Pavel', 'UOC', 'Dance', 20),
    ('Mr', 'Asakar', 'Hussain', 'UOD', 'PE, Music', 30),
    ('Mr', 'Ben', 'Middlehurst', 'UOE', 'PE, Dance', 40),
    ('Mr', 'Benjamin', 'Balls', 'UOF', 'French', 50),
    ('Mr', 'Daniel', 'Alldritt', 'UOG', 'PE, PE, PE', 60),
    ('Mr', 'Gregory', 'Spratt', 'UOH', 'Dance', 70),
    ('Mr', 'Ismail', 'Kadir', 'UOI', 'Music', 80),
    ('Mr', 'James', 'Fletcher', 'UOJ', 'Archery', 90),
    ('Mr', 'Jamie', 'Hammond', 'UOK', 'PE, Dance', 100),
    ('Mr', 'Josh', 'Weeden', 'UOL', 'Fighting', 90),
    ('Mr', 'Nathan', 'Johnston', 'UOM', 'Dance', 80),
    ('Mr', 'Rashawn', 'Henry', 'UON', 'PE, Dance, Music', 70),
    ('Mr', 'Sidhant', 'Khosla', 'UOO', 'PE, Dance', 60),
    ('Mr', 'Timin', 'Rickaby', 'KCL', 'Maths', 50),
    ('Mr', 'Yusuf', 'Uddin', 'UOP', 'Quizzing, Dance', 40)

SELECT * FROM spartans_table

-- demonstration of IS NOT NULL
-- INSERT INTO spartans_table
-- VALUES ('Mr', 'Timin', 'Rickaby', 'KCL', 'maths', 50)

-- INSERT INTO spartans_table (title, first_name, last_name, uni_attended)
-- VALUES ('Mrs', 'Timina', 'Rickabya', NULL)

-- INSERT INTO spartans_table
-- VALUES ('Sr', 'Timino', 'Rickabyo', 'Mexico University', NULL)

-- SELECT first_name AS "Name" FROM spartans_table WHERE uni_attended IS NOT NULL

USE Northwind
```

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

```
SELECT
e1.TitleOfCourtesy + ' ' + e1.FirstName + ' ' + e1.LastName AS "Full Name"
, e2.FirstName + ' ' + e2.LastName AS "Reports to Name"
FROM Employees e1
LEFT JOIN Employees e2
ON e1.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 as below

```
SELECT
s.CompanyName
, SUM(od.UnitPrice*od.Quantity*(1-od.Discount)) AS "Total Sales"
FROM [Order Details] od
INNER JOIN Products p
ON p.ProductID = od.ProductID
INNER JOIN Suppliers s
ON s.SupplierID = p.SupplierID
GROUP BY s.SupplierID, 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 ord ers shipped. No Excel required

```
SELECT TOP 10 c.CompanyName, ROUND(SUM((1-
od.Discount)*od.Quantity * od.UnitPrice),2) AS "sales"
FROM [Order Details] od
INNER JOIN Orders o
ON o.OrderID = od.OrderID
INNER JOIN Customers c
ON o.CustomerID = c.CustomerID
GROUP BY c.CompanyName, o.ShippedDate
HAVING o.ShippedDate > '1997-12-31'
ORDER BY sales DESC
```

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

```
SELECT YEAR(o.OrderDate) AS "Year"
, MONTH(o.OrderDate) AS "Month"
, FORMAT(o.OrderDate, 'MMM-yy') AS "Year-Month"
, AVG(CAST(DATEDIFF(d, o.OrderDate, o.ShippedDate) AS Decimal(4,2))) AS "Average Number of
Ship Days"
FROM Orders o
GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate), FORMAT(o.OrderDate, 'MMM-yy')
ORDER BY "Year", "Month"
```