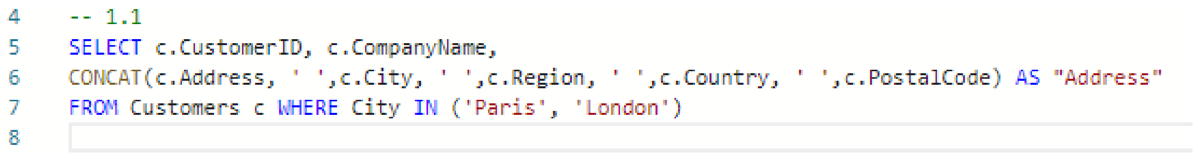
SQL Mini Project

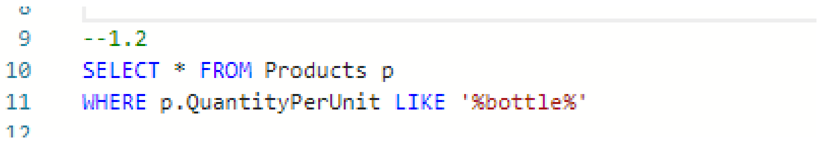
Exercise 1 – Northwind Queries

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



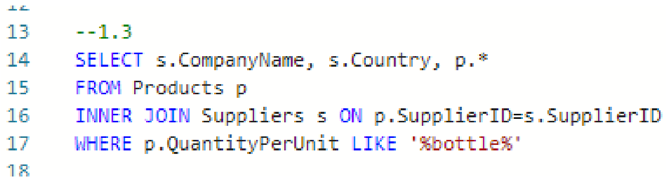
```sql
-- 1.1
SELECT c.CustomerID, c.CompanyName,
CONCAT(c.Address, ' ',c.City, ' ',c.Region, ' ',c.Country, ' ',c.PostalCode) AS "Address"
FROM Customers c WHERE City IN ('Paris', 'London')

```

1.2. List all products stored in bottles.



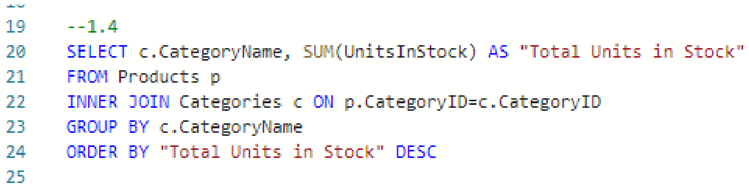
```sql
--1.2
SELECT * FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottle%'
```

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



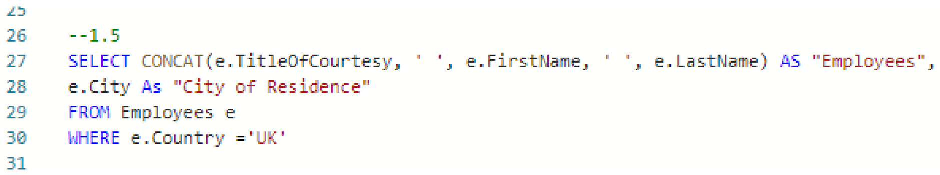
```sql
--1.3
SELECT s.CompanyName, s.Country, p.*
FROM Products p
INNER JOIN Suppliers s ON p.SupplierID=s.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottle%'
```

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.



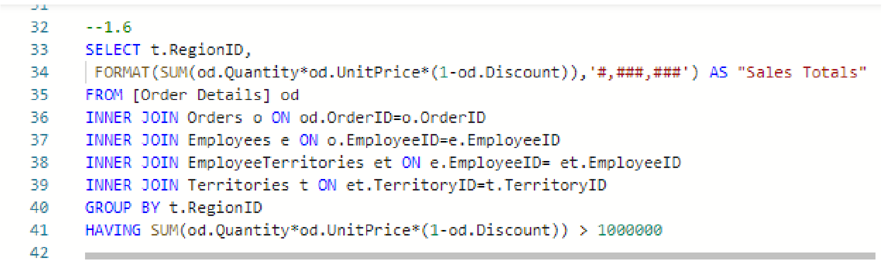
```sql
--1.4
SELECT c.CategoryName, SUM(UnitsInStock) AS "Total Units in Stock"
FROM Products p
INNER JOIN Categories c ON p.CategoryID=c.CategoryID
GROUP BY c.CategoryName
ORDER BY "Total Units in Stock" DESC
```

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.



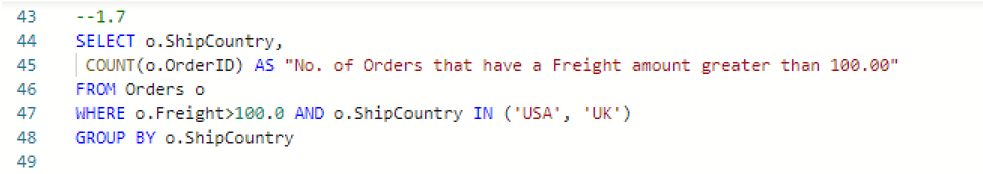
```sql
--1.5
SELECT CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "Employees",
e.City As "City of Residence"
FROM Employees e
WHERE e.Country ='UK'
```

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.



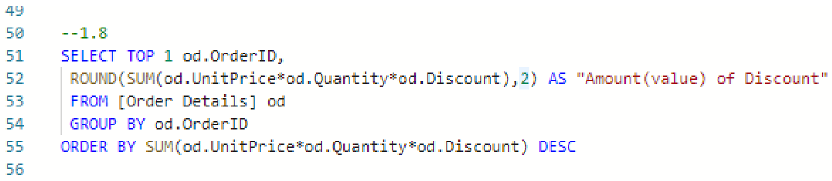
```sql
--1.6
SELECT t.RegionID,
 FORMAT(SUM(od.Quantity*od.UnitPrice*(1-od.Discount)),'#,###,###') AS "Sales Totals"
FROM [Order Details] od
INNER JOIN Orders o ON od.OrderID=o.OrderID
INNER JOIN Employees e ON o.EmployeeID=e.EmployeeID
INNER JOIN EmployeeTerritories et ON e.EmployeeID= et.EmployeeID
INNER JOIN Territories t ON et.TerritoryID=t.TerritoryID
GROUP BY t.RegionID
HAVING SUM(od.Quantity*od.UnitPrice*(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.



```sql
--1.7
SELECT o.ShipCountry,
 COUNT(o.OrderID) AS "No. of Orders that have a Freight amount greater than 100.00"
FROM Orders o
WHERE o.Freight>100.0 AND o.ShipCountry IN ('USA', 'UK')
GROUP BY o.ShipCountry
```

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.



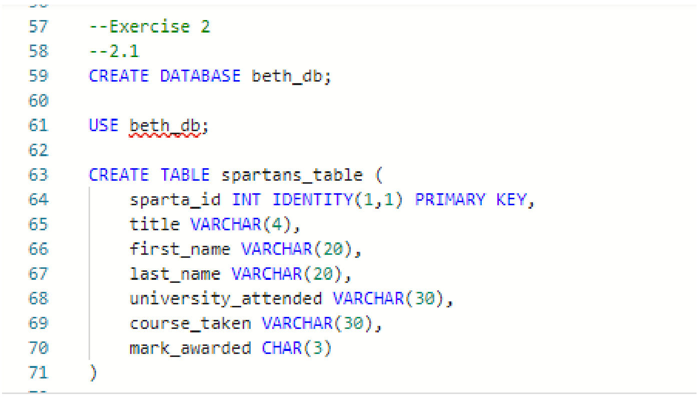
```sql
--1.8
SELECT TOP 1 od.OrderID,
 ROUND(SUM(od.UnitPrice*od.Quantity*od.Discount),2) AS "Amount(value) of Discount"
 FROM [Order Details] od
 GROUP BY od.OrderID
ORDER BY SUM(od.UnitPrice*od.Quantity*od.Discount) DESC
```

Exercise 2 – Create Spartans Table

2.1. Write the correct SQL statement to create the following table:

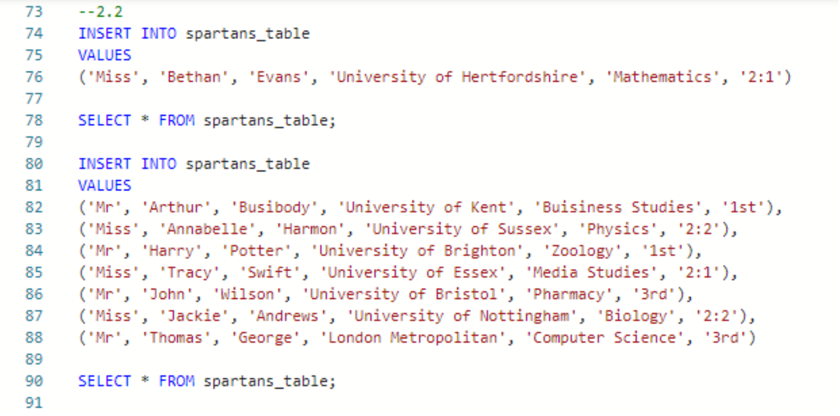
Spartans Table – include details about all the Spartans on this course. Separate Title, First Name and Last Name into separate columns, and include University attended, course taken, and mark achieved. Add any other columns you feel would be appropriate.

IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.



```sql
--Exercise 2
--2.1
CREATE DATABASE beth_db;

USE beth_db;

CREATE TABLE spartans_table (
    sparta_id INT IDENTITY(1,1) PRIMARY KEY,
    title VARCHAR(4),
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    university_attended VARCHAR(30),
    course_taken VARCHAR(30),
    mark_awarded CHAR(3)
)
```

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

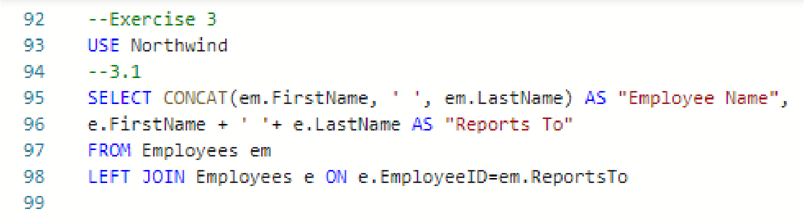


```
--2.2
INSERT INTO spartans_table
VALUES
('Miss', 'Bethan', 'Evans', 'University of Hertfordshire', 'Mathematics', '2:1')

SELECT * FROM spartans_table;

INSERT INTO spartans_table
VALUES
('Mr', 'Arthur', 'Busibody', 'University of Kent', 'Buisiness Studies', '1st'),
('Miss', 'Annabelle', 'Harmon', 'University of Sussex', 'Physics', '2:2'),
('Mr', 'Harry', 'Potter', 'University of Brighton', 'Zoology', '1st'),
('Miss', 'Tracy', 'Swift', 'University of Essex', 'Media Studies', '2:1'),
('Mr', 'John', 'Wilson', 'University of Bristol', 'Pharmacy', '3rd'),
('Miss', 'Jackie', 'Andrews', 'University of Nottingham', 'Biology', '2:2'),
('Mr', 'Thomas', 'George', 'London Metropolitan', 'Computer Science', '3rd')

SELECT * FROM spartans_table;

```

Exercise 3 – Northwind Data Analysis linked to Excel

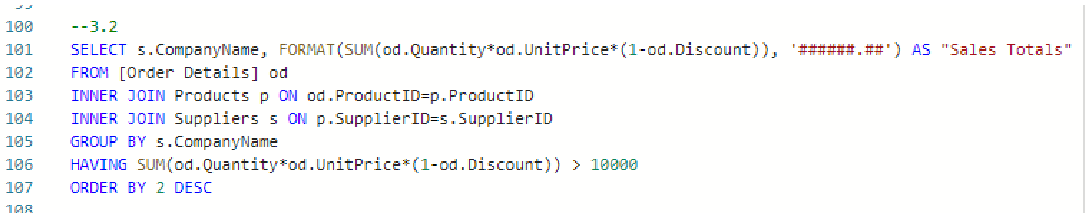
Write SQL statements to extract the data required for the following charts (create these in Excel):

3.1. List all Employees from the Employees table and who they report to. No Excel required. Please mention the Employee Names and the Report To names.

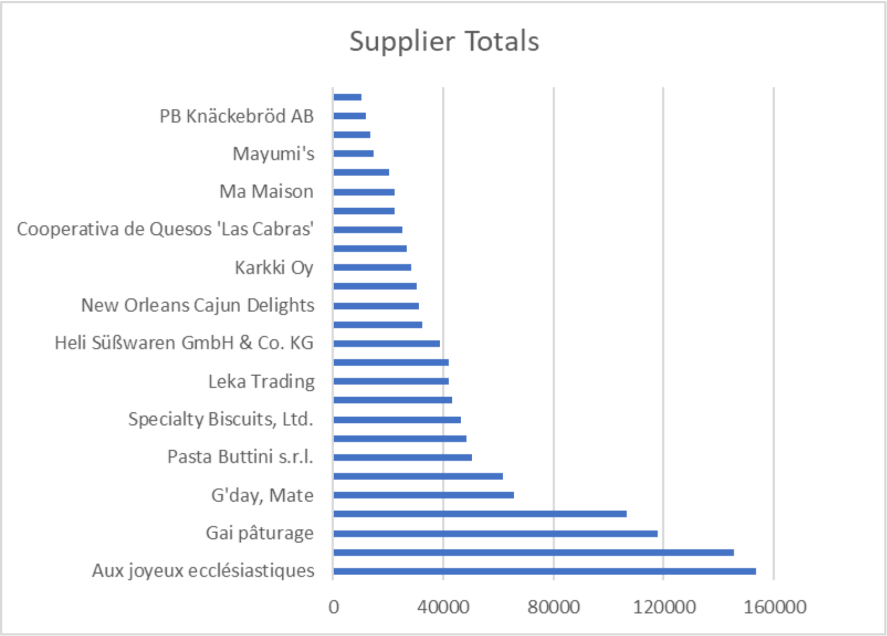


```
--Exercise 3
USE Northwind
--3.1
SELECT CONCAT(em.FirstName, ' ', em.LastName) AS "Employee Name",
e.FirstName + ' '+ e.LastName AS "Reports To"
FROM Employees em
LEFT JOIN Employees e ON e.EmployeeID=em.ReportsTo

```

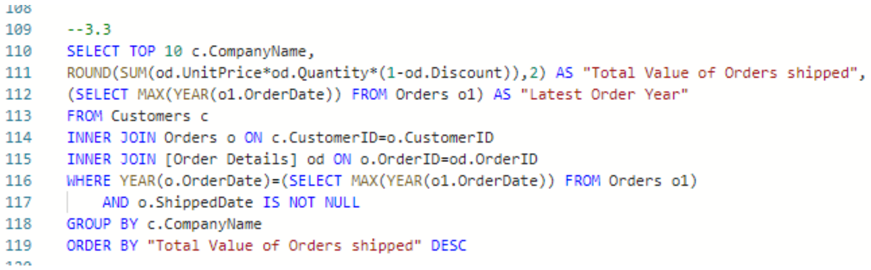
3.2. List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table and present as a bar chart



```sql
--3.2
SELECT s.CompanyName, FORMAT(SUM(od.Quantity*od.UnitPrice*(1-od.Discount)), '######.##') AS "Sales Totals"
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.Quantity*od.UnitPrice*(1-od.Discount)) > 10000
ORDER BY 2 DESC
```

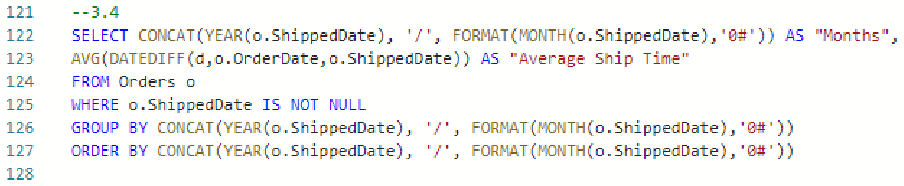


3.3. List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped. No Excel required.

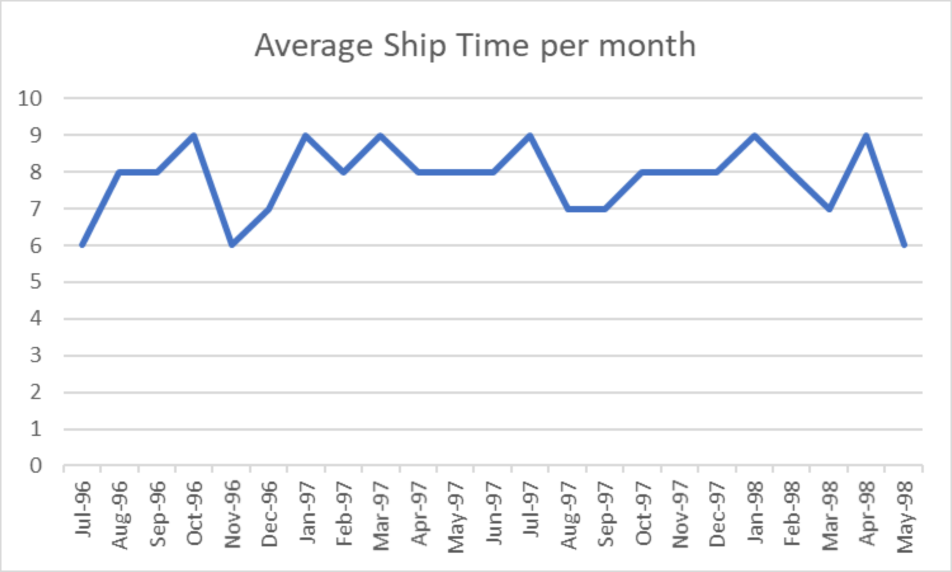


```sql
--3.3
SELECT TOP 10 c.CompanyName,
ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) AS "Total Value of Orders shipped",
(SELECT MAX(YEAR(o1.OrderDate)) FROM Orders o1) AS "Latest Order Year"
FROM Customers c
INNER JOIN Orders o ON c.CustomerID=o.CustomerID
INNER JOIN [Order Details] od ON o.OrderID=od.OrderID
WHERE YEAR(o.OrderDate)=(SELECT MAX(YEAR(o1.OrderDate)) FROM Orders o1)
    AND o.ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY "Total Value of Orders shipped" DESC
```

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



```sql
--3.4
SELECT CONCAT(YEAR(o.ShippedDate), '/', FORMAT(MONTH(o.ShippedDate),'0#')) AS "Months",
AVG(DATEDIFF(d,o.OrderDate,o.ShippedDate)) AS "Average Ship Time"
FROM Orders o
WHERE o.ShippedDate IS NOT NULL
GROUP BY CONCAT(YEAR(o.ShippedDate), '/', FORMAT(MONTH(o.ShippedDate),'0#'))
ORDER BY CONCAT(YEAR(o.ShippedDate), '/', FORMAT(MONTH(o.ShippedDate),'0#'))
```


Average Ship Time per month
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