Exercise 1 – Northwind Queries

```
-- 1.1
-- List all customers in Paris or London
SELECT
    c.CustomerID AS "Customer ID",
    c.CompanyName AS "Company Name",
    CASE WHEN c.Region IS NOT NULL THEN c.Address + ', ' + c.City + ', ' + c.Region + ', ' + c.PostalCode + ', ' + c.Country -- If Region entry exists, include it in Address
    ELSE c.Address + ', ' + c.City + ', ' + c.PostalCode + ', ' + c.Country -- Otherwise exclude Region
    END AS "Address"
FROM Customers c
WHERE c.City IN ('Paris', 'London') -- Check if customer is in Paris or London
```

```
-- 1.2
-- List all products stored in bottles
SELECT
    p.ProductID AS "Product ID",
    p.ProductName AS "Product Name",
    p.QuantityPerUnit AS "Quantity Per Unit"
FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottle%' -- Only if Quantity description contains 'bottle'
```

```
-- 1.3
-- Repeat 1.2 but add Supplier Name and Country
SELECT
    p.ProductID AS "Product ID",
    p.ProductName AS "Product Name",
    p.QuantityPerUnit AS "Quantity Per Unit",
    s.CompanyName AS "Supplier Name",
    s.Country AS "Country"
FROM Products p
    LEFT JOIN Suppliers s -- Joining Suppliers table to Products table
        ON p.SupplierID = s.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottle%' -- Only if Quantity description contains 'bottle'
```

```
-- 1.4
-- How many products are in each category. Includes Category Name and lists highest number first
SELECT
    c.CategoryName AS "Category Name",
    COUNT(p.CategoryID) AS "Number of Products" -- Count the number of products
FROM Products p
    INNER JOIN Categories c -- Join Products and Categories tables
        ON p.CategoryID = c.CategoryID
GROUP BY c.CategoryName -- Group counting by category
ORDER BY "Number of Products" DESC -- Sort by descending
```

```
-- 1.5
-- List all UK employees (Title Of Courtesy + Full Name, City of Residence)
SELECT
    e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Full Name and Title", -- Combine name and title
    e.City AS "City of Residence"
FROM Employees e
WHERE e.Country = 'UK' -- Select only UK employees
```

```sql
-- 1.6
-- 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",
    ROUND(SUM(od.UnitPrice * od.Quantity * (1 - od.Discount)), 2) AS "Sales Total" -- Rounds to two decimal places
FROM Territories t -- Join multiple tables
    INNER JOIN EmployeeTerritories et
        ON t.TerritoryID = et.TerritoryID
    INNER JOIN Region r
        ON r.RegionID = t.RegionID
    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 * od.Quantity * (1- od.Discount)) > 1000000 -- Display only regions with Sales Total over 1000000
```

```sql
-- 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 with Freight > 100 and Ship Country UK or USA" -- Count rows
FROM Orders o
WHERE o.Freight > 100.00 AND (o.ShipCountry IN ('USA', 'UK')) -- If entries meet stated conditions
```

```sql
-- 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 -- Only shows highest discount entry
    od.OrderID AS "Order ID",
    SUM((od.UnitPrice * od.Quantity * od.Discount)) AS "Total value of discount" -- Find total discount value for each order id
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY "Total value of discount" DESC -- Sort by descending
```

Exercise 2 – Create Spartans Table

```sql
-- 2.1
-- Create and use the database
CREATE DATABASE davidEx2_db;
USE davidEx2_db;

-- Create the table
CREATE TABLE spartans (
    title VARCHAR(10),
    firstName VARCHAR(30) NOT NULL,
    lastName VARCHAR(30) NOT NULL,
    university VARCHAR(30),
    course VARCHAR(30),
    mark VARCHAR(30)
)

DROP TABLE spartans;
SP_HELP spartans;

-- 2.2
-- Insert various entries into the table
INSERT INTO spartans
VALUES (
    'Mr. ', 'David', 'Trieu', 'UCL', 'Physics', '1st'
), ('Dr. ', 'Sbeven', 'Squarepants', 'Harvard', 'Music', '2:2')

-- Show table of spartans
SELECT * FROM spartans
```

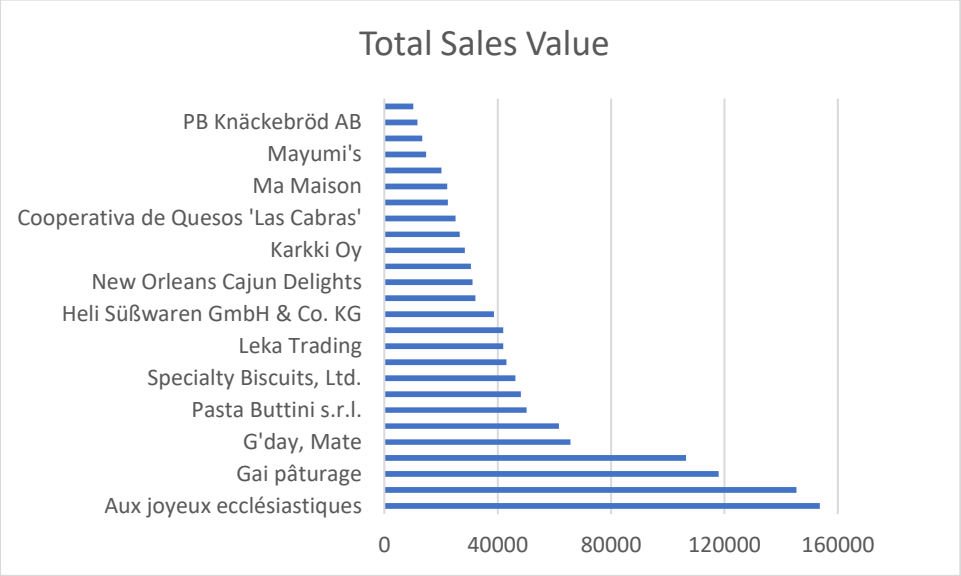
Exercise 3 – Northwind Data Analysis linked to Excel

```sql
-- 3.1
-- List all Employees from the Employees table and who they report to.

SELECT
    e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Employee Name", -- Format Employee Name
    e2.TitleOfCourtesy + ' ' + e2.FirstName + ' ' + e2.LastName AS "Reports To"
FROM Employees e
    LEFT JOIN Employees e2 -- Join table to itself based on who each person reports to
        ON e.ReportsTo = e2.EmployeeID
```

```sql
-- 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 "Supplier Name",
    SUM(od.UnitPrice * od.Quantity * (1- od.Discount)) AS "Total Sales" -- Calculate total sales per supplier
FROM Suppliers s -- Join relevant tables
    INNER JOIN Products p
        ON s.SupplierID = p.SupplierID
    INNER JOIN [Order Details] od
        ON p.ProductID = od.ProductID
GROUP BY s.CompanyName -- Groups by supplier
HAVING SUM(od.UnitPrice * od.Quantity * (1- od.Discount)) > 10000 -- Checks for total sales above 10000
ORDER BY "Total Sales" DESC -- Sort by Total Sales descending
```

3.2 - Bar Chart




```sql
-- 3.3
-- List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped.
SELECT TOP 10 -- Select top 10
    c.CompanyName AS "Company Name",
    SUM(od.UnitPrice * od.Quantity * (1- od.Discount)) AS "Total Sales" -- Calculate total sales for each company
FROM Customers c -- Join relevant tables
    INNER JOIN Orders o
        ON c.CustomerID = o.CustomerID
    INNER JOIN [Order Details] od
        ON o.OrderID = od.OrderID
WHERE o.ShippedDate >= '1998/1/1' -- Only consider orders shipped within the latest year (1998)
GROUP BY c.CompanyName
ORDER BY "Total Sales" DESC
```

```sql
-- 3.4
-- Plot the Average Ship Time by month for all data in the Orders Table using a line chart.
SELECT
    FORMAT(o.OrderDate, 'MM') AS "Order Month",
    FORMAT(o.OrderDate, 'yyyy') AS "Order Year",
    AVG(DATEDIFF(d,o.OrderDate,o.ShippedDate)) AS "Ship Days"
FROM Orders o
GROUP BY FORMAT(o.OrderDate, 'MM'), FORMAT(o.OrderDate, 'yyyy')
```

3.4 Line Chart