SQL Practical Exercise

Introduction

Exercise to learn the SQL commands

CREATE TABLE	Concatenation
SQL Data Types	Formatting dates and numbers
INSERT INTO	Column aliases
SELECT	Simple JOIN statements
WHERE clause	Complex JOIN statements
LIKE and wildcards	Subquery

Connect to the database

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,
    c.CompanyName,
    c.Address, c.City,
    c.Region,
    c.PostalCode,
    c.Country
FROM Customers c
WHERE c.City IN ('London','Paris');
```

1.2 List all products stored in bottles.

```
SELECT p.ProductName
FROM Products p
WHERE p.QuantityPerUnit LIKE ('%bottle%')
```

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

```
SELECT p.ProductName, s.CompanyName, s.Country
FROM Products p
LEFT 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, COUNT(*) AS "Number of Condiments"
FROM Products p
LEFT JOIN Categories c ON p.CategoryID=c.CategoryID
GROUP BY c.CategoryName
ORDER BY 2 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
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.regionDescription,
    ROUND(SUM(od.UnitPrice * od.Quantity * (1- od.Discount)), 2)
    AS "Total sales"
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 Ship Country.

```
SELECT COUNT(*) AS "Number of Freight Greater than 100"
FROM Orders o
WHERE o.shipcountry IN ('UK', 'USA')
    AND o.Freight > 100.00
```

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,
    od.unitprice * od.quantity * od.Discount AS "Discount Amount"
FROM [Order Details] od
ORDER BY 2 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.

```
CREATE DATABASE greg_db
USE greg_db

DROP TABLE IF EXISTS spartans
CREATE TABLE spartans
(
```

```
    spartan_id INT IDENTITY PRIMARY KEY,
    title CHAR(2),
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    university VARCHAR(60),
    course VARCHAR(60),
    grade INT,
    favourite_number INT,
    can_drive BIT

)
```

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

Here are two examples of inserting data, I have added more in the actual SQL file:

```
INSERT INTO spartans
VALUES (
    'Mr',
    'Gregory',
    'Spratt',
    'Monsters University',
    'Canister Design',
    21,
    2,
    1
)
INSERT INTO spartans
VALUES (
    'Mr',
    'Ahmed',
    'Rahman',
    'Empire State University',
    'Biophysics',
    43,
    3,
    0
)
SELECT * FROM spartans
```

Exercise 3 – Northwind Data Analysis linked to Excel

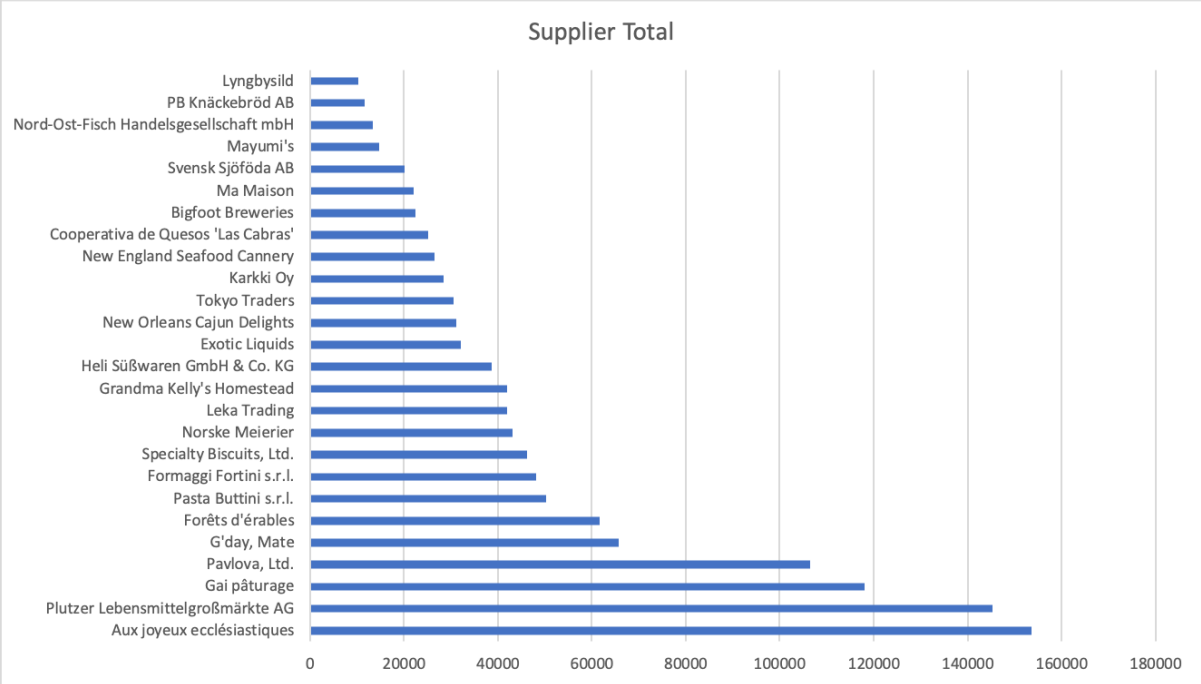
3.1 List all Employees from the Employees table and who they report to. No Excel required.

```
USE NorthWind
```

```
SELECT e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS "Employee",
    em.TitleOfCourtesy + ' ' + em.FirstName + ' ' + em.LastName AS "Employee
Reports to"
FROM Employees e
LEFT JOIN Employees em ON e.ReportsTo=em.EmployeeID
=em.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: (5 Marks)

```
SELECT s.CompanyName,
    SUM(od.Quantity * od.UnitPrice * (1-od.Discount)) AS "Individual Sale"
FROM Suppliers s
INNER JOIN Products p ON s.SupplierID=p.SupplierID
INNER JOIN [Order Details] od ON p.ProductID=od.ProductID
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. (10 Marks)

```
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 below. (10 Marks)

```
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" -- Might need to format here as not sure if getting
the correct answer with rounding
FROM Orders o
GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate), FORMAT(o.OrderDate, 'MMM-yy')
ORDER BY 1, 2
```