# 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.

SELECT c.CustomerID, c.CompanyName, c.Address, c.City, c.Region, c.PostalCode, c.Country

FROM Customers c

WHERE c.City IN ('Paris', 'London')

-- 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.ContactName, 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, COUNT(p.ProductID) AS "Number of Products"

FROM Products p

INNER JOIN Categories c ON c.CategoryID = p.CategoryID

GROUP BY c.CategoryName

ORDER BY "Number of Products" 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 ROUND(SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)), 0) AS "Sales", r.RegionDescription AS "Territory Description"

FROM Orders o

INNER JOIN [Order Details] od 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 t.TerritoryID = et.TerritoryID

INNER JOIN Region r ON t.RegionID = r.RegionID

GROUP BY 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(o.OrderID) AS "Number of Orders"

FROM Orders o

WHERE (o.Freight > 100.00)

AND (o.ShipCountry = 'USA' OR o.ShipCountry = '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.UnitPrice\*od.Quantity - od.UnitPrice\*od.Quantity\*(1-od.Discount)) AS "Discount Amount"

FROM [Order Details] od

WHERE od.Discount > 0

ORDER BY "Discount Amount" DESC

# Exercise 2 - Create Spartans Table

-- 2.1 Write the correct SQL statement to create the following table

DROP TABLE IF EXISTS Spartans;

CREATE TABLE Spartans (

spartan\_id INT IDENTITY PRIMARY KEY,

title VARCHAR(5),

first\_name VARCHAR(20),

last\_name VARCHAR(20),

university VARCHAR(50),

course VARCHAR(50),

mark VARCHAR(30)

)

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

INSERT INTO Spartans VALUES

('Mr', 'Svilen', 'Petrov', 'London Metropolitan University', 'BSc Computing', 'First'),

('Mr', 'Reece', 'Louch', 'University Of Warwick', 'Computer Science', '2:2'),

('Mr', 'Saleh', 'Sandhu', 'University Of Westminister', 'Computer Science', '2:1'),

('Mr', 'Ben', 'Swift', 'Nottingham Trent University', 'Computer Science', '2:1'),

('Mr', 'Toyin', 'Ajani', 'University Of Bath', 'Chemical engineering', 'First'),

('Mr', 'Chris', 'Cunningham', 'Loughborough', 'Computer Science', '2:1'),

('Ms', 'Janja', 'Kovacevic', 'University of Massachusetts Amherst', 'Computer Science and Computational Mathematics', '3.9'),

('Mr', 'Abdullah', 'Muhammad', 'University of Southampton', 'Physics', 'First'),

('Mr', 'Shahid', 'Enayat', 'Brunel University', 'Electronic and Electrical Engineering', '2:2'),

('Mr', 'Dami', 'Oshidele', 'King''s College London', 'Electronic Engineering with Management', '2:1'),

('Mr', 'Emmanuel', 'Buraimo', 'King''s College London', 'Computer Science', '2:1');

# Exercise 3 - Northwind Data Analysis linked to Excel

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

SELECT (emp.FirstName + ' ' + emp.LastName) AS "Employee", (mngr.FirstName + ' ' + mngr.LastName) AS "Manager"

FROM Employees emp

LEFT JOIN Employees mngr ON emp.ReportsTo = mngr.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 "Supplier", SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)) AS "Total Net Sales"

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.UnitPrice\*od.Quantity\*(1-od.Discount)) > 10000

ORDER BY "Total Net 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.

SELECT TOP 10 c.CompanyName, ROUND(SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)),2) AS "Total Value of Shipped Orders"

FROM Orders o

INNER JOIN Customers c ON o.CustomerID = c.CustomerID

INNER JOIN [Order Details] od ON od.OrderID = o.OrderID

WHERE (o.OrderDate >=

(SELECT MAX(YEAR(o2.OrderDate))

FROM Orders o2))

AND o.ShippedDate IS NOT NULL

GROUP BY c.CompanyName

ORDER BY "Total Value of Shipped Orders" DESC

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

SELECT FORMAT(o.OrderDate, 'MMMM') AS "Month", YEAR(o.OrderDate) AS "Year", AVG(DATEDIFF(d, o.OrderDate, o.ShippedDate)) AS "Ship Time (Days)"

FROM Orders o

GROUP BY FORMAT(o.OrderDate, 'MMMM'), YEAR(o.OrderDate)

ORDER BY "Year", DATEPART(MM, FORMAT(o.OrderDate, 'MMMM')+'01 1900')

-- 3.4 Just like the graph (ALTERNATE SOLUTION)

SELECT CONVERT(varchar(15), FORMAT(o.OrderDate, 'MMMM-yy')) AS "Month/Year", AVG(DATEDIFF(d, o.OrderDate, o.ShippedDate)) AS "Ship Time (Days)"

FROM Orders o

GROUP BY CONVERT(varchar(15), FORMAT(o.OrderDate, 'MMMM-yy'))

ORDER BY MIN(o.OrderDate)

