## Introduction

This exercise requires you to know the following aspects of SQL:

|  |  |
| --- | --- |
| 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 |

## Exercise 1 – Northwind Queries (40 marks: 5 for each question)

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

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

FROM Customers

WHERE City = 'Paris' OR City = 'London';



* 1. List all products stored in bottles.

SELECT QuantityPerUnit FROM Products

WHERE QuantityPerUnit LIKE '%bottles'; --Find entries that end in ‘bottles’



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

SELECT ProductName As 'Product Name', QuantityPerUnit As 'Quantity Per Unit', Suppliers.CompanyName As 'Company Name', Suppliers.Country

FROM Products

INNER JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID –Connect Suppliers table via SupplierID to get Suppliers Name

WHERE QuantityPerUnit LIKE '%bottles';



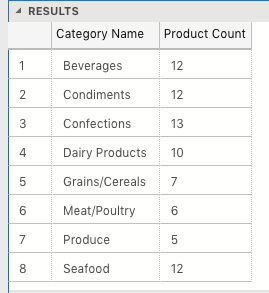
* 1. Write an SQL Statement that shows how many products there are in each category. Include Category Name in result set and list the highest number first.

SELECT Categories.CategoryName As 'Category Name', COUNT(Products.CategoryID) AS 'Product Count' –Count number of entries in Category ID

FROM Products

INNER JOIN Categories ON Products.CategoryID = Categories.CategoryID

GROUP BY Categories.CategoryName; --Group by the Catergory Name

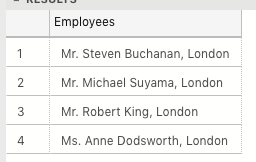


* 1. List all UK employees using concatenation to join their title of courtesy, first name and last name together. Also include their city of residence.

SELECT TitleofCourtesy + ' ' + FirstName + ' ' + LastName + ', ' AS Employees, City

FROM Employees

WHERE Country = 'UK';



* 1. 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 Region.RegionID As 'Region ID', Region.RegionDescription As 'Region Description', ROUND(SUM([Order Details].UnitPrice \* [Order Details].Quantity),0) AS 'Total Sales' --Multiply Unit Price by Quantity to get Total Sales

FROM Territories

INNER JOIN EmployeeTerritories ON Territories.TerritoryID = EmployeeTerritories.TerritoryID

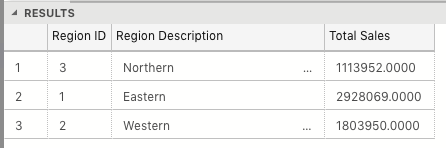
INNER JOIN Region ON Territories.RegionID = Region.RegionID

INNER JOIN Orders ON EmployeeTerritories.EmployeeID = Orders.EmployeeID

INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID

GROUP BY Region.RegionID, Region.RegionDescription

HAVING SUM([Order Details].UnitPrice \* [Order Details].Quantity) > 1000000;



* 1. Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.

SELECT ShipCountry As 'Ship Country', COUNT(ShipCountry) AS 'Count'

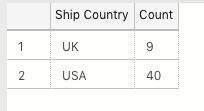
FROM Orders

WHERE (Freight > 100.00)

AND (ShipCountry = 'UK'

OR ShipCountry='USA')

GROUP BY ShipCountry;



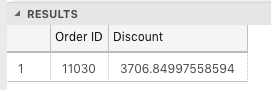
* 1. Write an SQL Statement to identify the Order Number of the Order with the highest amount of discount applied to that order.

SELECT TOP 1 OrderID AS 'Order ID', SUM(Discount \* Quantity \* UnitPrice) AS 'Discount'

FROM [Order Details]

GROUP BY OrderID

ORDER BY 'Discount' DESC;



## Exercise 2 – Create Spartans Table (20 marks – 10 each)

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.

CREATE TABLE SpartansTable (

PersonID INT NOT NULL IDENTITY PRIMARY KEY,

FirstName VARCHAR(25),

LastName VARCHAR(25),

University VARCHAR(50),

CourseTaken VARCHAR(50),

MarkAchieved VARCHAR(5),

DateFinished DATE,

);



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

INSERT INTO SpartansTable

(FirstName, LastName, University, CourseTaken, MarkAchieved, DateFinished)

VALUES

('Seb', 'Van Woerkom', 'University of Kent', 'English', '2.1', '2018-06-30'),

('Qamar', 'Aden', 'University of Portsmouth', 'Petroleum Engineering', '2.1', '2018-06-30'),

('Ben', 'Owusu', 'NULL', 'Business with Finance', '2.1', '2018-06-30'),

('Maroua', 'Akkari', 'Queen Mary University', 'Bioinformatics', '2.1', '2018-06-30'),

('Philip', 'Faboya', 'University of Surrey', 'Chemical Engineering', '2.1', '2018-06-30'),

('Arthur', 'Hussey', 'University of Oxford', 'Course Materials Science', '2.1', '2018-06-30'),

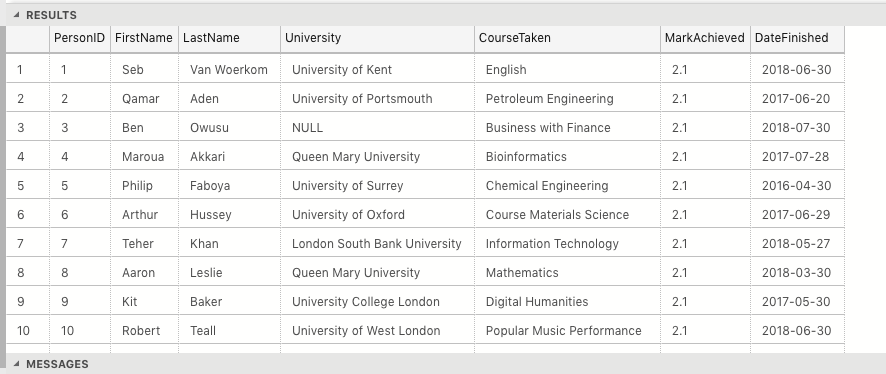
('Teher', 'Khan', 'London South Bank University', 'Information Technology', '2.1', '2018-06-30'),

('Aaron', 'Leslie', 'Queen Mary University', 'Mathematics', '2.1', '2018-06-30'),

('Kit', 'Baker', 'University College London', 'Digital Humanities', '2.1', '2017-05-30'),

('Robert', 'Teall', 'University of West London', 'Popular Music Performance', '2.1', '2018-06-30'),

('James', 'Bachen', 'Bournemouth University', 'Music and Audio Technology', '2.1', '2018-06-30');



## Exercise 3 – Northwind Data Analysis linked to Excel (30 marks)

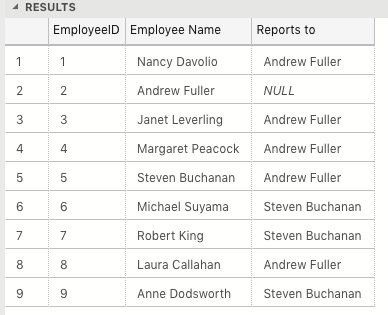
Write SQL statements to extract the data required for the following charts (create these in Excel):

3.1 List all Employees from the Employees table and who they report to. No Excel required. (5 Marks)

SELECT e.EmployeeID, e.FirstName + ' '+ e.LastName AS 'Employee Name', e1.FirstName + ' ' + e1.LastName AS 'Reports to'

FROM Employees e

LEFT JOIN Employees e1 ON e.ReportsTo = e1.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 Suppliers.SupplierID, Suppliers.CompanyName, SUM([Order Details].UnitPrice \* [Order Details].Quantity) AS 'Total Sales'

FROM Suppliers

INNER JOIN Products ON Suppliers.SupplierID = Products.SupplierID

INNER JOIN [Order Details] ON Products.ProductID = [Order Details].ProductID

GROUP BY Suppliers.SupplierID, Suppliers.CompanyName

HAVING SUM([Order Details].UnitPrice \* [Order Details].Quantity) > 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 orders shipped. No Excel required. (10 Marks)

SELECT TOP 10 Orders.CustomerID, Customers.[ContactName], SUM([Order Details].UnitPrice \* [Order Details].Quantity) AS 'Total Sales'

FROM Orders

INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID

INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE Orders.ShippedDate IS NOT NULL AND YEAR(Orders.ShippedDate) = 1998

GROUP BY Orders.CustomerID,Customers.[ContactName]

ORDER BY 'Total 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 MONTH(OrderDate) AS 'Month', YEAR(OrderDate) AS 'Year', AVG(DateDiff(DAY,OrderDate,ShippedDate)) AS 'Days Taken to Ship'

--Works out difference between Order Date and Shipped Date

FROM Orders

WHERE DateDiff(day,OrderDate,ShippedDate) IS NOT NULL --Remove entries that have not shipped

GROUP BY MONTH(OrderDate), YEAR(OrderDate)

ORDER BY 'Year', 'Month' ASC



## Standards (10 marks)

Remember to apply all the following standards:

* Use consistent capitalisation and indentation of SQL Statements
* Use concise and consistent table alias names
* Use column aliases to ensure tidy column headings (spaces and consistent capitalisation)
* Concatenate any closely related columns e.g. First Name and Last Name or Address and City etc
* Put comments throughout