## Introduction

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

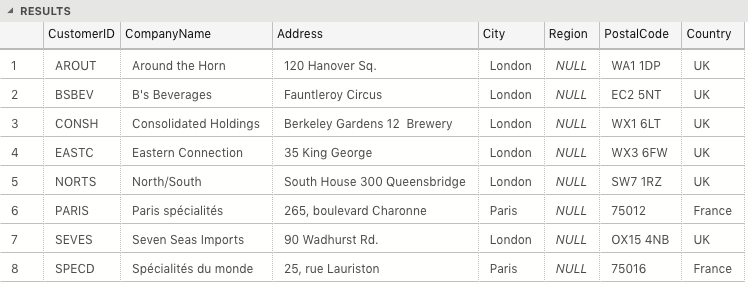
|  |  |
| --- | --- |
| CREATE TABLE | Concatenation |
| SQL Data Types | Formatting dates and numbers |
| INSERT INTO | Column aliases |
| SELECT | Simple JOIN statements |
| WHERE clause | Complex JOIN statements |
| LIKE and wildcards | Subquery |

## 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;

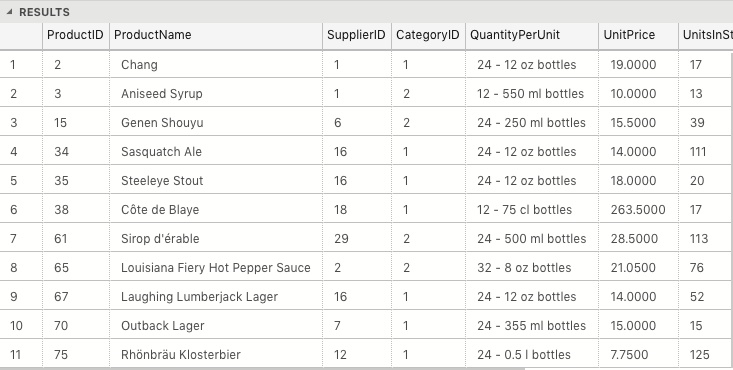
WHERE City = 'Paris' OR City = 'London';



* 1. List all products stored in bottles.

SELECT \* FROM Products

WHERE QuantityPerUnit LIKE '%bottles%';



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

SELECT p.ProductID, p.ProductName, s.CompanyName AS 'SupplierName', s.Country FROM Products p  
INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID  
WHERE QuantityPerUnit LIKE '%bottle%';



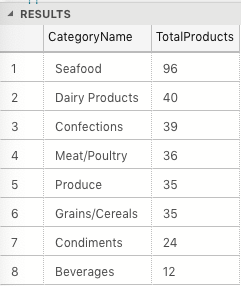
* 1. Write an SQL Statement that shows how many products there are in each category. Include Category Name in result set and list the highest number first.

SELECT c.CategoryName, SUM(c.CategoryID) AS 'TotalProducts' FROM Categories c

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

GROUP BY c.CategoryName

ORDER BY 'TotalProducts' DESC;



* 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 'FullName', City FROM Employees;



* 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 r.RegionID, r.RegionDescription, ROUND(SUM(od.Quantity \* od.UnitPrice), 2) AS 'TotalSales' FROM Region r

INNER JOIN Territories t on r.RegionID = t.RegionID

INNER JOIN EmployeeTerritories et ON t.TerritoryID = et.TerritoryID

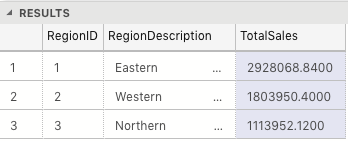
INNER JOIN Orders o ON et.EmployeeID = o.EmployeeID

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

GROUP BY r.RegionID, r.RegionDescription

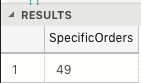
HAVING SUM(od.Quantity \* od.UnitPrice) > 1000000

ORDER BY r.RegionID;



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

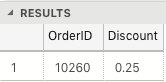
SELECT COUNT(\*) AS 'SpecificOrders' FROM Orders  
 WHERE Freight > 100.00 AND (ShipCountry = 'USA' OR ShipCountry = 'UK');



* 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 \* FROM [Order Details]

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:

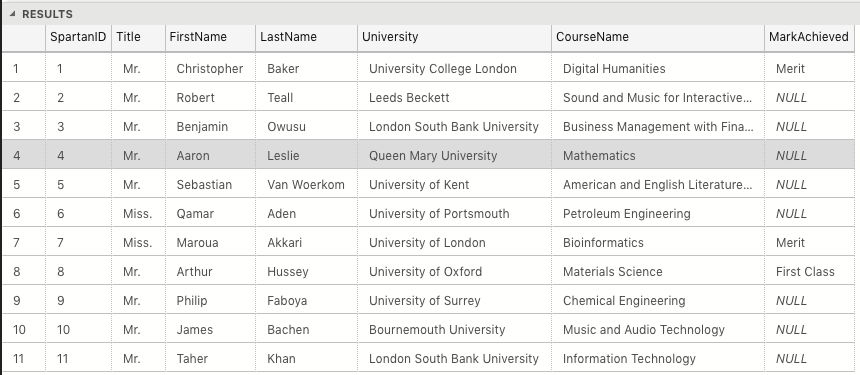
Spartans Table – include details about all the Spartans on this course. Separate Title, First Name and Last Name into separate columns, and include University attended, course taken and mark achieved. Add any other columns you feel would be appropriate.

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

CREATE TABLE Spartans  
(  
SpartanID int IDENTITY(1,1) PRIMARY KEY,  
Title varchar(5),  
FirstName varchar(100),  
LastName varchar(100),  
University varchar(100),  
CourseName varchar(100),  
MarkAchieved varchar(40)  
);

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

INSERT INTO Spartans (Title, FirstName, LastName, University, CourseName, MarkAchieved)  
VALUES ('Mr.', 'Christopher', 'Baker', 'University College London', 'Digital Humanities', 'Merit'),  
('Mr.', 'Robert', 'Teall', 'Leeds Beckett', 'Sound and Music for Interactive Games', NULL),  
('Mr.', 'Benjamin', 'Owusu', 'London South Bank University', 'Business Management with Finance', NULL),  
('Mr.', 'Aaron', 'Leslie', 'Queen Mary University', 'Mathematics', NULL),  
('Mr.', 'Sebastian', 'Van Woerkom', 'University of Kent', 'American and English Literature & History of Art', NULL),  
('Miss.', 'Qamar', 'Aden', 'University of Portsmouth', 'Petroleum Engineering', NULL),  
('Miss.', 'Maroua', 'Akkari', 'University of London', 'Bioinformatics', 'Merit'),  
('Mr.', 'Arthur', 'Hussey', 'University of Oxford', 'Materials Science', 'First Class'),  
('Mr.', 'Philip', 'Faboya', 'University of Surrey', 'Chemical Engineering', NULL),  
('Mr.', 'James', 'Bachen', 'Bournemouth University', 'Music and Audio Technology', NULL),  
('Mr.', 'Taher', 'Khan', 'London South Bank University', 'Information Technology', NULL);

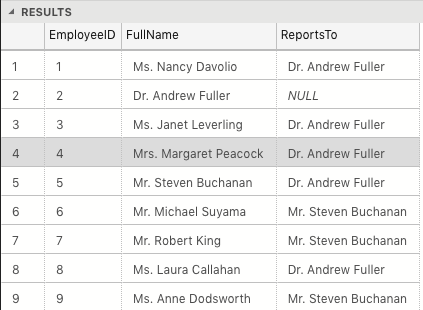


## Exercise 3 – Northwind Data Analysis linked to Excel (30 marks)

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.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS 'FullName', ee.TitleOfCourtesy + ' ' + ee.FirstName + ' ' + ee.LastName AS 'ReportsTo' FROM Employees e

LEFT OUTER JOIN Employees ee ON e.reportsto = ee.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.UnitPrice \* od.Quantity \* (1 - od.Discount)) AS 'TotalSales' FROM Suppliers s

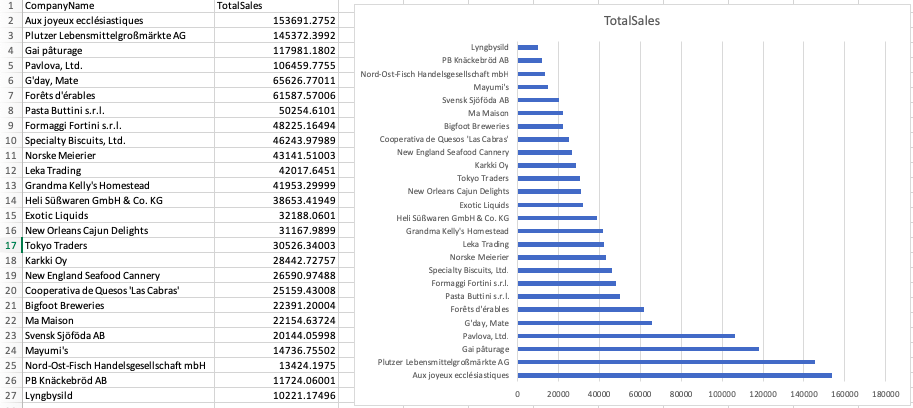
INNER JOIN Products p ON s.SupplierID = p.SupplierID

INNER JOIN [Order Details] od ON p.ProductID = od.ProductID

GROUP BY s.CompanyName

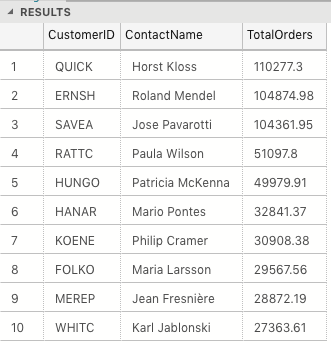
HAVING SUM(od.UnitPrice \* od.Quantity \* (1 - od.Discount)) > 10000

ORDER BY 'TotalSales' 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.CustomerID, c.ContactName, ROUND(SUM(od.UnitPrice \* od.Quantity \* (1 - od.Discount)), 2) AS 'TotalOrders' FROM Customers c  
INNER JOIN Orders o ON c.CustomerID = o.CustomerID  
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID  
WHERE OrderDate > 1998-01-00   
GROUP BY c.CustomerID, c.ContactName  
ORDER BY 'TotalOrders' 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)

## 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