## 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.

In this exercise, the following function demonstrates that by selecting customerID, companyname, address, region, postalcode from the customers’ tables where city Is equal to Paris or London will return all the required fields needed.

USE Northwind;

SELECT customerID, companyname, address, region, postalcode

FROM customers WHERE City= 'paris' OR City ='london';

* 1. List all products stored in bottles.

On this occasion we will use the products table to identify which products are stored in bottles. The Modulo (%) selects the column with the product value ‘bottles’.

USE Northwind;

SELECT ProductID, ProductName, QuantityPerUnit FROM products WHERE QuantityPerUnit LIKE '%bottles%';

|  |  |  |
| --- | --- | --- |
| **Product ID** | **Product Name** | **Quantity per Unit** |
| 2 | Chang | 24 - 12 oz bottles |
| 3 | Aniseed Syrup | 12 - 550 ml bottles |
| 15 | Genen Shouyu | 24 - 250 ml bottles |
| 34 | Sasquatch Ale | 24 - 12 oz bottles |
| 35 | Steeleye Stout | 24 - 12 oz bottles |
| 38 | Côte de Blaye | 12 - 75 cl bottles |
| 61 | Sirop d'érable | 24 - 500 ml bottles |
| 65 | Louisiana Fiery Hot Pepper Sauce | 32 - 8 oz bottles |
| 67 | Laughing Lumberjack Lager | 24 - 12 oz bottles |
| 70 | Outback Lager | 24 - 355 ml bottles |
| 75 | Rhönbräu Klosterbier | 24 - 0.5 l bottles |

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

Utilizing the query adopted in question 1.2, two Joints had to be applied in this occasion in order to add the Columns Supplier Name and Country which where found in the Suppliers table.

USE Northwind;

SELECT p.SupplierID, p.ProductID, p.ProductName, p.QuantityPerUnit, s.CompanyName, s.Country

FROM products p

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

WHERE QuantityPerUnit LIKE '%bottles%';

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product ID | Supplier ID | Product Name | Quantity Per Unit | Company Name | Country |
| 1 | 2 | Chang | 24 - 12 oz bottles | Exotic Liquids | UK |
| 1 | 3 | Aniseed Syrup | 12 - 550 ml bottles | Exotic Liquids | UK |
| 6 | 15 | Genen Shouyu | 24 - 250 ml bottles | Mayumi's | Japan |
| 16 | 34 | Sasquatch Ale | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| 16 | 35 | Steeleye Stout | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| 18 | 38 | Côte de Blaye | 12 - 75 cl bottles | Aux joyeux ecclésiastiques | France |
| 29 | 61 | Sirop d'érable | 24 - 500 ml bottles | Forêts d'érables | Canada |
| 2 | 65 | Louisiana Fiery Hot Pepper Sauce | 32 - 8 oz bottles | New Orleans Cajun Delights | USA |
| 16 | 67 | Laughing Lumberjack Lager | 24 - 12 oz bottles | Bigfoot Breweries | USA |
| 7 | 70 | Outback Lager | 24 - 355 ml bottles | Pavlova, Ltd. | Australia |
| 12 | 75 | Rhönbräu Klosterbier | 24 - 0.5 l bottles | Plutzer Lebensmittelgroßmärkte AG | Germany |

* 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.

This query returns the number of products in each category from highest to lowest due to the ORDER BY clause.

USE Northwind;

SELECT p.ProductID, c.CategoryName, p.UnitsInStock AS 'Total Products'

FROM Products p

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

ORDER BY UnitsInStock DESC;

|  |  |  |
| --- | --- | --- |
| ProductID | Category | Total Products |
| 6 | Meat/Poultry | 120 |
| 4 | Dairy Products | 53 |
| 1 | Beverages | 39 |
| 2 | Condiments | 17 |
| 7 | Produce | 15 |
| 3 | Confections | 13 |
| 8 | Seafood | 6 |
| 5 | Grains/Cereals | 0 |

* 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.

To rearrange the following requirements, the following method could be utilized in order to concatenate the following columns.

USE Northwind;

SELECT (Employees.TitleOfCourtesy) + ' ' + (Employees.FirstName) + ' ' + (Employees.LastName)

AS Employee\_name, City AS City

FROM Employees;

|  |  |
| --- | --- |
| **Employee** | **City** |
| Ms. Nancy Davolio | Seattle |
| Dr. Andrew Fuller | Tacoma |
| Ms. Janet Leverling | Kirkland |
| Mrs. Margaret Peacock | Redmond |
| Mr. Steven Buchanan | London |
| Mr. Michael Suyama | London |
| Mr. Robert King | London |
| Ms. Laura Callahan | Seattle |
| Ms. Anne Dodsworth | London |

* 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.

This exercise was a little tricky to join the regions with the total sales generated in each region. I only managed to conduct the total sales generated by each product individually. The issue was that I was not able to draw out a clear map in order to use joins appropriately.

USE Northwind;

SELECT t.RegionID, t.TerritoryDescription,SUM(((UnitPrice-Discount)\*Quantity)) AS "Gross Total"

FROM [Order Details]

RIGHT JOIN Territories t ON RegionID=UnitPrice

GROUP BY t.RegionID, t.TerritoryDescription,((UnitPrice-Discount)\*Quantity)

HAVING SUM(((UnitPrice-Discount)\*Quantity)) <1000000;

OR

SELECT ps.ProductSales, ps.ProductName, p.ProductName FROM

[Product Sales for 1997] ps JOIN Products p ON

ps.ProductName=p.ProductName

WHERE ProductSales >= 10000

ORDER BY ProductSales DESC;

|  |  |
| --- | --- |
| Total Sales | Product Name |
| 46563.09 | Côte de Blaye |
| 33616.55 | Raclette Courdavault |
| 33109.51 | Thüringer Rostbratwurst |
| 32604 | Gnocchi di nonna Alice |
| 23550.02 | Manjimup Dried Apples |
| 20762.82 | Tarte au sucre |
| 20652.28 | Camembert Pierrot |
| 16580.85 | Alice Mutton |
| 15950 | Carnarvon Tigers |
| 14041.8 | Gudbrandsdalsost |
| 12854.28 | Rössle Sauerkraut |
| 12784.13 | Perth Pasties |
| 11838.6 | Mozzarella di Giovanni |
| 11225.66 | Gumbär Gummibärchen |
| 11069.9 | Ipoh Coffee |
| 10974 | Schoggi Schokolade |

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

In this exercise, two tables (two syntax) functions were adopted in order to find the quantity of order with Freight more than 100 in the UK and USA.

Syntax 1:

USE Northwind;

SELECT OrderID, ShipCountry, Freight FROM Orders

WHERE ShipCountry = 'UK'

GROUP BY ShipCountry, Freight, OrderID

HAVING Freight > 100.00 ;

AND

Syntax 2:

USE Northwind;

SELECT OrderID, ShipCountry, Freight FROM Orders

WHERE ShipCountry = 'USA'

GROUP BY ShipCountry, Freight, OrderID

HAVING Freight > 100.00 ;

|  |  |  |  |
| --- | --- | --- | --- |
| **Count** | **OrderID** | **Location** | **Freight** |
| 1 | 10359 | UK | 288.43 |
| 2 | 10547 | UK | 178.43 |
| 3 | 10768 | UK | 146.32 |
| 4 | 10800 | UK | 137.44 |
| 5 | 10829 | UK | 154.72 |
| 6 | 10869 | UK | 143.28 |
| 7 | 10987 | UK | 185.48 |
| 8 | 11023 | UK | 123.83 |
| 9 | 11056 | UK | 278.96 |
| 10 | 10294 | USA | 147.26 |
| 11 | 10305 | USA | 257.62 |
| 12 | 10316 | USA | 150.15 |
| 13 | 10324 | USA | 214.27 |
| 14 | 10329 | USA | 191.67 |
| 15 | 10346 | USA | 142.08 |
| 16 | 10369 | USA | 195.68 |
| 17 | 10393 | USA | 126.56 |
| 18 | 10452 | USA | 140.26 |
| 19 | 10479 | USA | 708.95 |
| 20 | 10510 | USA | 367.63 |
| 21 | 10555 | USA | 252.49 |
| 22 | 10607 | USA | 200.24 |
| 23 | 10612 | USA | 544.08 |
| 24 | 10616 | USA | 116.53 |
| 25 | 10627 | USA | 107.46 |
| 26 | 10657 | USA | 352.69 |
| 27 | 10660 | USA | 111.29 |
| 28 | 10678 | USA | 388.98 |
| 29 | 10693 | USA | 139.34 |
| 30 | 10696 | USA | 102.55 |
| 31 | 10706 | USA | 135.63 |
| 32 | 10713 | USA | 167.05 |
| 33 | 10748 | USA | 232.55 |
| 34 | 10805 | USA | 237.34 |
| 35 | 10816 | USA | 719.78 |
| 36 | 10847 | USA | 487.57 |
| 37 | 10852 | USA | 174.05 |
| 38 | 10855 | USA | 170.97 |
| 39 | 10889 | USA | 280.61 |
| 40 | 10894 | USA | 116.13 |
| 41 | 10904 | USA | 162.95 |
| 42 | 10941 | USA | 400.81 |
| 43 | 10965 | USA | 144.38 |
| 44 | 10983 | USA | 657.54 |
| 45 | 10984 | USA | 211.22 |
| 46 | 11002 | USA | 141.16 |
| 47 | 11030 | USA | 830.75 |
| 48 | 11031 | USA | 227.22 |
| 49 | 11032 | USA | 606.19 |

**Total Orders is 49.**

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

The following syntax adopts a subquery in the WHERE clause that enables the ability to find the order Number with the highest applied discount which is order 10372 with a discount of 52.7.

USE Northwind;SELECT OrderID, UnitPrice\*Discount AS 'Discount'

FROM [Order Details]

WHERE UnitPrice\*Discount = (SELECT MAX(UnitPrice\*discount) FROM [Order Details]);

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

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

To create a database is very simple, you simply follow the syntax bellow.

CREATE DATABASE Spartans\_Table;

Once the Spartans Table database is created, we can now add the table. This is an important step towards creating a new table because you need to identify whether the columns will contain an interger, VARCHAR characters, if so how long? CHAR, DATE, DATETIME etc.

USE Spartans\_Table;

CREATE TABLE Spartans(

StudentID INT,

First\_Name VARCHAR(18),

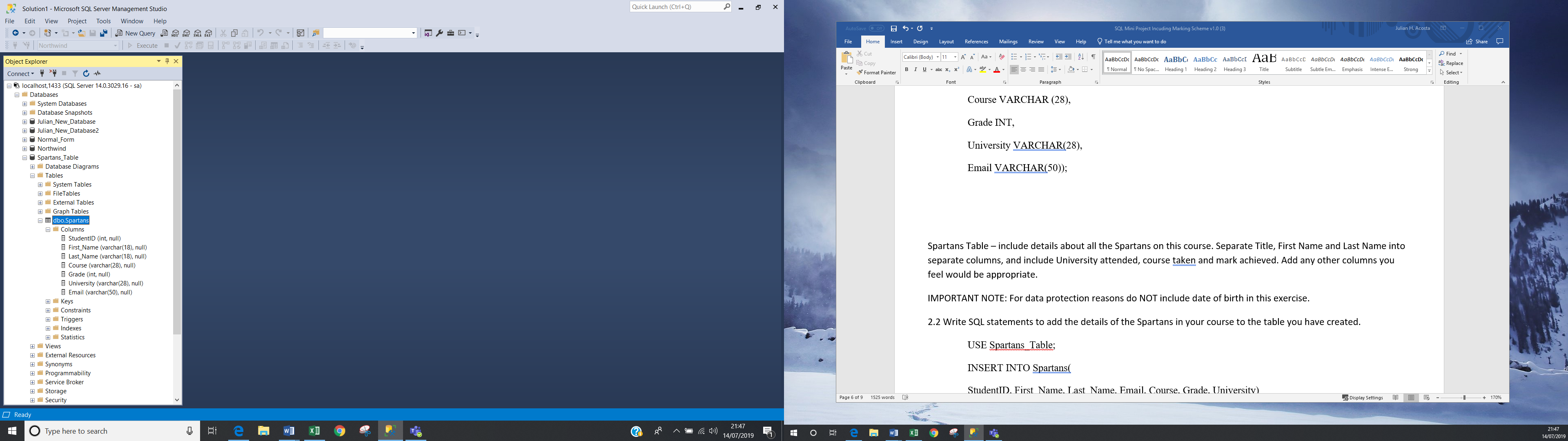
Last\_Name VARCHAR (18),

Course CHAR (28),

Grade INT,

University VARCHAR(28),

Email VARCHAR(50));



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

The following syntax is adopted to manually insert data into the table created on question 2.1. Since SQL is only conducted row by row, to insert data, always mention the column names and the values with the same structure.

USE Spartans\_Table;

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100849, 'Joy','Paterson','ctfijban@sharklasers.com','Physics',2.1,'Brunel');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100800, 'Humairaa','Quintana','dmohamed.almagi@thepieter.com','Chemestry',2.2,'Aston');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100751, 'Madiha','Woods','lsandy.hack32u@glocknershop.com','Maths',1,'Birmingham');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100702, 'Izabel','Rosario','xt-fiks@ms.vcss.eu.org','Psychology',2.1,'Queen Mary');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100653, 'Alasdair','Vaughn','fsanae.soso03@memberheality.ga','Business', 2.1,'Sussex');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100604, 'Guy','Harrington','2hammo-1997m@toppieter.com','Computer Science',1,'Birmingham');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100555, 'Phoebe','Croft','abousandala5@cryptontrade.ga','Economics',1,'Queen Mary');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100506, 'Nikolas','Townsend','jrymadilmi8@daniarturos.club','Physics',2.1,'Oxford');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100457, 'Brandy','Stark','3munozjim@edu.hstu.eu.org','Maths',2.1,'Portsmouth');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100408, 'Zachariah','Beasley','tmsh.msh.522z@mail1.mungmung.o-r.kr','English',1,'Brighton');

INSERT INTO Spartans(

StudentID, First\_Name, Last\_Name, Email, Course, Grade, University)

VALUES(

100359, 'Kiara','Sawyer','zflorentina\_g@tolongsaya.me','Computer Science',1,'Portsmouth');

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student ID** | **First Name** | **Last Name** | **Course** | **Grade** | **University** | **Email** |
| 100898 | Neive | Moody | Economics | 1 | Portsmouth | hsvi1a+53z3psopblvlo@sharklasers.com |
| 100849 | Joy | Paterson | Physics | 2.2 | Brunel | ctfijban@sharklasers.com |
| 100800 | Humairaa | Quintana | Chemestry | 2.1 | Aston | dmohamed.almagi@thepieter.com |
| 100751 | Madiha | Woods | Maths | 1 | Birmingham | lsandy.hack32u@glocknershop.com |
| 100702 | Izabel | Rosario | Psychology | 2.1 | Queen Mary | xt-fiks@ms.vcss.eu.org |
| 100653 | Alasdair | Vaughn | Business | 2,1 | Sussex | fsanae.soso03@memberheality.ga |
| 100604 | Guy | Harrington | Computer Science | 1 | Birmingham | 2hammo-1997m@toppieter.com |
| 100555 | Phoebe | Croft | Economics | 1 | Queen Mary | abousandala5@cryptontrade.ga |
| 100506 | Nikolas | Townsend | Physics | 2.1 | Oxford | jrymadilmi8@daniarturos.club |
| 100457 | Brandy | Stark | Maths | 2,2 | Portsmouth | 3munozjim@edu.hstu.eu.org |
| 100408 | Zachariah | Beasley | English | 1 | Brighton | tmsh.msh.522z@mail1.mungmung.o-r.kr |
| 100359 | Kiara | Sawyer | Computer Science | 1 | Portsmouth | zflorentina\_g@tolongsaya.me |

## 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)

The following syntax is used to provide an overview of the employee structure

USE Northwind;

SELECT EmployeeID, FirstName, LastName, ReportsTo, Title

FROM Employees;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Emp.ID** | **First Name** | **Last Name** | **Report Id** | **Reports to** |
| 1 | Nancy | Davolio | 2 | Sales Representative |
| 2 | Andrew | Fuller | NULL | Vice President, Sales |
| 3 | Janet | Leverling | 2 | Sales Representative |
| 4 | Margaret | Peacock | 2 | Sales Representative |
| 5 | Steven | Buchanan | 2 | Sales Manager |
| 6 | Michael | Suyama | 5 | Sales Representative |
| 7 | Robert | King | 5 | Sales Representative |
| 8 | Laura | Callahan | 2 | Inside Sales Coordinator |
| 9 | Anne | Dodsworth | 5 | Sales Representative |

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)

The foolwing sintax provides an output in order to be able to list all the suppliers with total sales above $10,000.

USE Northwind;

SELECT ps.ProductSales, ps.ProductName, p.SupplierID,

s.SupplierID, CompanyName, p.ProductID

FROM [Product Sales for 1997] ps

JOIN Products p ON ps.ProductName=p.ProductName

JOIN Suppliers s ON p.SupplierID=s.SupplierID

WHERE ProductSales >= 10000

ORDER BY ProductSales 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)

USE Northwind;

SELECT o.OrderDate, o.Freight FROM Orders o

WHERE OrderDate BETWEEN 1998-01-06-00-00-00-00 AND 1998-01-01-00-00-00-000;

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