SQL Practical Exercise

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)

1.1 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 FROM Customers c
WHERE c.City IN ('Paris', 'London') -- Select Customers and only return customers with Paris/London
```

	CustomerID	CompanyName	Address	City	Region	PostalCode
1	AROUT	Around the Horn	120 Hanover Sq.	London	*NULL*	WA1 1DP
2	BSBEV	B's Beverages	Fauntleroy Circus	London	*NULL*	EC2 5NT
3	CONSH	Consolidated Holdings	Berkeley Gardens 12 Brew…	London	*NULL*	WX1 6LT
4	EASTC	Eastern Connection	35 King George	London	*NULL*	WX3 6FW
5	NORTS	North/South	South House 300 Queensbri…	London	*NULL*	SW7 1RZ
6	PARIS	Paris spécialités	265, boulevard Charonne	Paris	*NULL*	75012
7	SEVES	Seven Seas Imports	90 Wadhurst Rd.	London	*NULL*	OX15 4NB
8	SPECD	Spécialités du monde	25, rue Lauriston	Paris	*NULL*	75016

1.2 List all products stored in bottles.

```
SELECT QuantityPerUnit FROM Products p
WHERE p.QuantityPerUnit LIKE '%Bottle%' -- Selects all products with bottle containers mentioned
```

	QuantityPerUnit
1	24 - 12 oz bottles
2	12 - 550 ml bottles
3	24 - 250 ml bottles
4	24 - 12 oz bottles
5	24 - 12 oz bottles
6	12 - 75 cl bottles
7	750 cc per bottle
8	24 - 500 ml bottles
9	32 - 8 oz bottles
10	24 - 12 oz bottles
11	24 - 355 ml bottles
12	24 - 0.5 l bottles

SQL Practical Exercise

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

```
SELECT p.QuantityPerUnit, s.CompanyName, s.Country FROM Products p
INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
-- Join to the supplier table to Select the data on the first line
WHERE p.QuantityPerUnit LIKE '%Bottle%'
```

	QuantityPerUnit	CompanyName	Country
1	24 - 12 oz bottles	Exotic Liquids	UK
2	12 - 550 ml bottles	Exotic Liquids	UK
3	24 - 250 ml bottles	Mayumi's	Japan
4	24 - 12 oz bottles	Bigfoot Breweries	USA
5	24 - 12 oz bottles	Bigfoot Breweries	USA
6	12 - 75 cl bottles	Aux joyeux ecclésiastiques	France
7	750 cc per bottle	Aux joyeux ecclésiastiques	France
8	24 - 500 ml bottles	Forêts d'érables	Canada
9	32 - 8 oz bottles	New Orleans Cajun Delights	USA
1…	24 - 12 oz bottles	Bigfoot Breweries	USA
1…	24 - 355 ml bottles	Pavlova, Ltd.	Australia
1…	24 - 0.5 l bottles	Plutzer Lebensmittelgroßm…	Germany

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 Products" FROM Products p
INNER JOIN Categories c ON c.CategoryID = p.CategoryID -- Join Products to Categories
GROUP BY CategoryName
-- Grouping here to join all category names together and allow the “count” function to operate for the
second column.
ORDER BY "Number of Products" DESC -- Order from most to least frequent product
```

	CategoryName	Number of Products
1	Confections	13
2	Beverages	12
3	Condiments	12
4	Seafood	12
5	Dairy Products	10
6	Grains/Cereals	7
7	Meat/Poultry	6
8	Produce	5

SQL Practical Exercise

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 TitleOfCourtesy + ' ' + FirstName + ' ' + LastName + ', ' + City AS "Name And City"
FROM Employees
WHERE Country = 'UK' -- Only select those records with UK as the country
```

	Name And City
1	Mr. Steven Buchanan, Lond…
2	Mr. Michael Suyama, London
3	Mr. Robert King, London
4	Ms. Anne Dodsworth, London

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.totalamt),2) AS "Total Sales Figure" FROM Region r
-- Round the sum of sales linked via the series of Inner Joins
INNER JOIN Territories t ON t.RegionID = r.RegionID
INNER JOIN EmployeeTerritories et ON t.TerritoryID = et.TerritoryID
INNER JOIN Employees e ON e.EmployeeID = et.EmployeeID
INNER JOIN Orders o ON o.EmployeeID = e.EmployeeID
INNER JOIN (SELECT OrderID, SUM((UnitPrice * Quantity)*(1-Discount)) AS totalamt FROM [Order
Details] GROUP BY OrderID) od ON od.OrderID = o.OrderID
-- Use a Subquery to ascertain the total sales cost per order (minus the discount) from the
Order Details table. These are summed in total in the initial Select query.
GROUP BY RegionDescription -- Group by Region as per the question
HAVING SUM(od.totalamt) > 1000000 -- Single out only those regions with sales above 1 million
ORDER BY "Total Sales Figure" DESC -- Order by Sales figure
```

	RegionDescription	Total Sales Figure
1	Eastern …	2730198.01
2	Western …	1615248
3	Northern …	1048605.58

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 "Count of Freight = 100+ to UK/USA" FROM Orders
WHERE Freight > 100 AND ShipCountry IN ('USA', 'UK')
-- Single out Freight count above 100 and specifically in the USA and UK
```

	Count of Freight = 100+ to UK/USA
1	49

SQL Practical Exercise

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 OrderID, (UnitPrice*Quantity) AS "Pre-Discount Price", -- See Initial Pre-Discount Price
       CONCAT(Discount*100, '%') AS "Discount", -- View Discount Percentage
       (UnitPrice*Quantity)*Discount AS "Total Discount Value", -- Total Sale Discount
       (UnitPrice*Quantity)*(1-Discount) AS "Post-Discount Price" -- View amount post-discount
FROM [Order Details]
ORDER BY "Total Discount Value" DESC -- Order by Discount Value and Select Top 1 Only Via Top Function
```

	OrderID	Pre-Discount Price	Discount	Total Discount Value	Post-Discount Price
1	10353	10540.0000	20%	2108	8432

SQL Practical Exercise

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.

```
CREATE TABLE spartans_table(
    spartan_id INT IDENTITY(1,1) PRIMARY KEY,
    spartan_title VARCHAR(10),
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    university VARCHAR(30),
    course VARCHAR(30),
    mark VARCHAR(20),
    date_graduated DATE,
)
```

spartan_id	spartan_title	first_name	last_name	university	course	mark	date_graduated

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

```
INSERT INTO spartans_table
VALUES
('Mr', 'Ross', 'Savill', 'Brunel University', 'History', '2:1 With Honours', '05-05-2008'),
('Dr', 'Muna', 'Dirie', 'Hampstead University', 'Politics', '1:1 With Distinction', '06-30-2018'),
('Professor', 'Mac', 'Uche', 'Kings College', 'English Literature', '2:1 With Honours', '04-29-2019'),
('Dr', 'Connor', 'Platts', 'Middlesex University', 'Computer Science', '2:2 With Merit', '06-25-2019'),
('Mr', 'Joe', 'Hilton', 'Southampton University', 'Physics', '2:1 With Honours', '04-15-2016'),
('Dr', 'Poornima', 'Harsha', 'Oxford University', 'English Language', '2:1 With Honours', '05-20-2015'),
('Mr', 'Cameron', 'Matthias-Yearwood', 'Cambridge University', 'Chemistry', '1:1 With Distinction', '01-07-2018'),
('Professor', 'Khari', 'McGhie', 'Edinburgh University', 'Geography', '2:1 With Honours', '06-22-2018'),
('Professor', 'Nirel', 'Warde', 'Glasgow University', 'Maths', '2:2 With Merit', '05-24-2017'),
('Mr', 'Daanyaal', 'Chaudry', 'Newcastle University', 'Statistics', '2:1 With Honours', '04-23-2019'),
('Mr', 'Muhammad', 'Butt', 'Liverpool University', 'Graphic Design', '1:1 With Distinction', '02-05-2019'),
('Mr', 'Yasin', 'Rahman', 'Manchester University', 'Art', '2:1 With Honours', '01-06-2014'),
('Dr', 'Joshua', 'Maccarthy', 'Durham University', 'Artificial Intelligence', '2:1 With Honours', '09-05-2019'),
('Professor', 'Sonam', 'Gurung', 'Cardiff University', 'French', '2:1 With Honours', '07-05-2018')
```

	spartan_id	spartan_title	first_name	last_name	university	course	mark	date_graduated
1	5	Mr	Ross	Savill	Brunel University	History	2:1 With Honours	2008-05-05
2	6	Dr	Muna	Dirie	Hampstead University	Politics	1:1 With Distinction	2018-06-30
3	7	Professor	Mac	Uche	Kings College	English Literature	2:1 With Honours	2019-04-29
4	8	Dr	Connor	Platts	Middlesex University	Computer Science	2:2 With Merit	2019-06-25
5	9	Mr	Joe	Hilton	Southampton University	Physics	2:1 With Honours	2016-04-15
6	10	Dr	Poornima	Harsha	Oxford University	English Language	2:1 With Honours	2015-05-20
7	11	Mr	Cameron	Matthias-Yearwood	Cambridge University	Chemistry	1:1 With Distinction	2018-01-07
8	12	Professor	Khari	McGhie	Edinburgh University	Geography	2:1 With Honours	2018-06-22
9	13	Professor	Nirel	Warde	Glasgow University	Maths	2:2 With Merit	2017-05-24
10	14	Mr	Daanyaal	Chaudry	Newcastle University	Statistics	2:1 With Honours	2019-04-23
11	15	Mr	Muhammad	Butt	Liverpool University	Graphic Design	1:1 With Distinction	2019-02-05
12	16	Mr	Yasin	Rahman	Manchester University	Art	2:1 With Honours	2014-01-06
13	17	Dr	Joshua	Maccarthy	Durham University	Artificial Intelligence	2:1 With Honours	2019-09-05
14	18	Professor	Sonam	Gurung	Cardiff University	French	2:1 With Honours	2018-07-05

SQL Practical Exercise

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. Please mention the Employee Names and the ReportTo names. (5 Marks)

```
SELECT e.FirstName + ' ' + e.LastName AS "Employee Name", m.FirstName + ' ' + m.LastName AS
"Reports To" FROM Employees e
LEFT JOIN Employees m ON m.EmployeeID = e.ReportsTo
      -- Use Left Join to ensure all employees are pulled, Inner Join to the SAME TABLE and
      from the "Reports To" attribute to the same table's Primary Key to identify who each
      employee reports to. Andrew Fuller reports to no-one and has a null field.
```

	Employee Name	Reports To
1	Nancy Davolio	Andrew Fuller
2	Andrew Fuller	*NULL*
3	Janet Leverling	Andrew Fuller
4	Margaret Peacock	Andrew Fuller
5	Steven Buchanan	Andrew Fuller
6	Michael Suyama	Steven Buchanan
7	Robert King	Steven Buchanan
8	Laura Callahan	Andrew Fuller
9	Anne Dodsworth	Steven Buchanan

SQL Practical Exercise

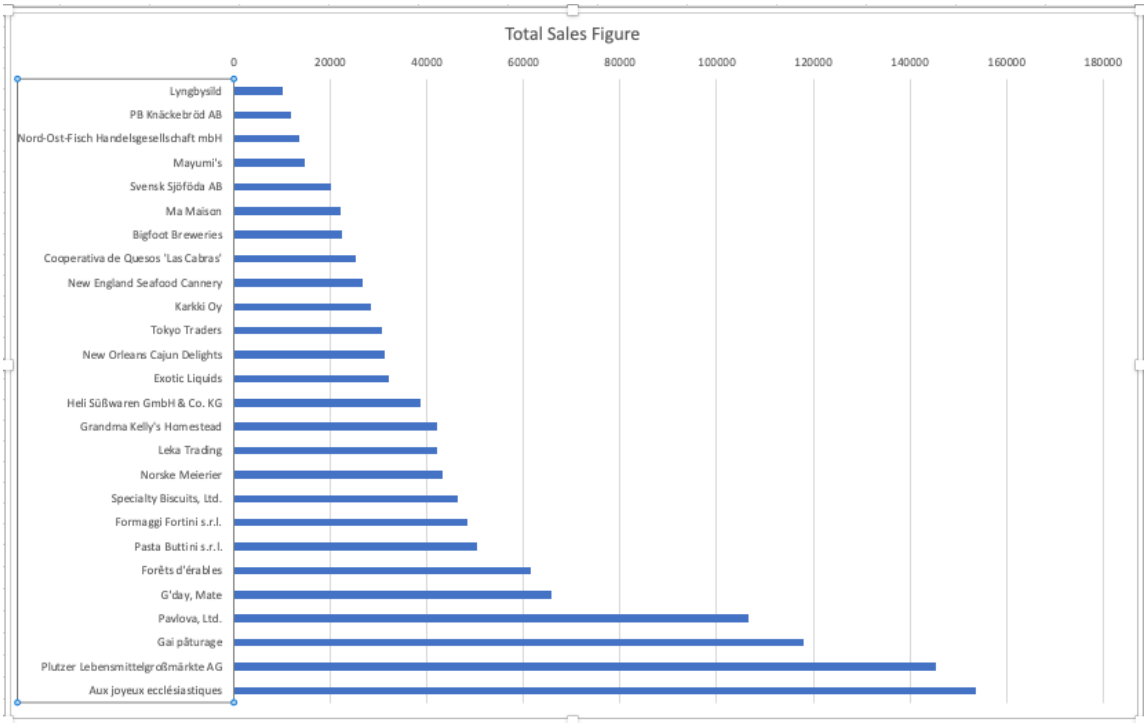
3.2 List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table and present as a bar chart as below: (5 Marks)

```
SELECT s.CompanyName AS "Company Name", ROUND(SUM("Sales Figures"),2) AS "Total Sales Figure"
FROM Suppliers s
INNER JOIN Products p ON s.SupplierID = p.SupplierID
INNER JOIN (SELECT ProductID, (UnitPrice*Quantity)*(1-Discount) AS "Sales Figures" FROM [Order
Details]) od ON p.ProductID = od.ProductID
GROUP BY CompanyName
HAVING SUM("Sales Figures") > 10000
ORDER BY "Total Sales Figure" ASC
-- Join from the Suppliers table to the Order Details table to pull across the sum of sales
(minus discount) and place that side by side with the supplier's company name. Use Having
function to weed out total sales below 10,000 and order by the remaining sales figures.
```

	Company Name	Total Sales Figure
1	Lyngbysild	10221.17
2	PB Knäckebröd AB	11724.06
3	Nord-Ost-Fisch Handelsges…	13424.2
4	Mayumi's	14736.76
5	Svensk Sjöföda AB	20144.06
6	Ma Maison	22154.64
7	Bigfoot Breweries	22391.2
8	Cooperativa de Quesos 'La…	25159.43
9	New England Seafood Canne…	26590.97
10	Karkki Oy	28442.73
11	Tokyo Traders	30526.34
12	New Orleans Cajun Delights	31167.99
13	Exotic Liquids	32188.06
14	Heli Süßwaren GmbH & Co. …	38653.42
15	Grandma Kelly's Homestead	41953.3
16	Leka Trading	42017.65
17	Norske Meierier	43141.51
18	Specialty Biscuits, Ltd.	46243.98
19	Formaggi Fortini s.r.l.	48225.16
20	Pasta Buttini s.r.l.	50254.61
21	Forêts d'érables	61587.57
22	G'day, Mate	65626.77
23	Pavlova, Ltd.	106459.78
24	Gai pâturage	117981.18
25	Plutzer Lebensmittelgroßm…	145372.4
26	Aux joyeux ecclésiastiques	153691.28

SQL Practical Exercise

(Exported Table to Excel and screenshot below – Excel document also on Github)



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 MAX(OrderDate) FROM Orders -- Latest Year = 1998

SELECT TOP 10 c.CompanyName, ROUND(SUM("Sales Figures"),2) AS "Total_Company_Sales_Value" FROM Customers c
INNER JOIN Orders o ON o.CustomerID = c.CustomerID
INNER JOIN (SELECT OrderID, (UnitPrice*Quantity)*(1-Discount) AS "Sales Figures" FROM [Order Details]) od ON
o.OrderID = od.OrderID -- Subquery used to find total sales (minus discount)
WHERE YEAR(o.OrderDate) = '1998' -- Only draw orders from the latest year
GROUP BY CompanyName -- Group by company name and round/sum total sales for company in top Select.
ORDER BY "Total_Company_Sales_Value" DESC -- Order sales by largest first.
```

	CompanyName	Total_Company_Sales_Value
1	Ernst Handel	41210.65
2	QUICK-Stop	37217.32
3	Save-a-lot Markets	36310.11
4	Hanari Carnes	23821.2
5	Rattlesnake Canyon Grocery	21238.27
6	Hungry Owl All-Night Grocers	20402.12
7	Königlich Essen	19582.77
8	White Clover Markets	15278.9
9	Folk och fä HB	13644.07
10	Suprêmes délices	11644.6

SQL Practical Exercise

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 LEFT(CONVERT(VARCHAR(10),OrderDate,102),7) AS "Year_Month", -- Turn date into a more manageable
VARCHAR format.
SUM(DATEDIFF(DAY, OrderDate, ShippedDate)) / COUNT(*) AS "Avg_Days_Before_Shipping"
FROM Orders -- Sum all the days between orders received and finally shipped, divide them by the total number
of orders
WHERE ShippedDate IS NOT NULL -- Avoid errors relating to null values in the shipped column.
GROUP BY LEFT(CONVERT(VARCHAR(10),OrderDate,102),7) -- Group by the date as per the question.
ORDER BY LEFT(CONVERT(VARCHAR(10),OrderDate,102),7) ASC -- Date format is Year first to correctly order
```

	Year_Month	Avg_Days_Before_Shipping
1	1996.07	8
2	1996.08	8
3	1996.09	10
4	1996.10	6
5	1996.11	8
6	1996.12	7
7	1997.01	9
8	1997.02	9
9	1997.03	8
10	1997.04	9
11	1997.05	9
12	1997.06	8
13	1997.07	8
14	1997.08	6
15	1997.09	9
16	1997.10	8
17	1997.11	8
18	1997.12	9
19	1998.01	9
20	1998.02	7
21	1998.03	9
22	1998.04	6
23	1998.05	2

(Exported Table to Excel and screenshot below – Excel document available on Github)