# SQL Practical Exercise

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

--to find customers in either paris or london we could have used IN Clause or OR condition

-- and displayed all the address details with the customerID using relevent columns

SELECT CustomerID, CompanyName, Address, City, Region, PostalCode, Country FROM Customers WHERE City = 'Paris' OR City = 'London';

OR

SELECT CustomerID, CompanyName, Address, City, Region, PostalCode, Country FROM Customers WHERE City IN ('Paris', 'London');

* 1. List all products stored in bottles.

--to list the product names, the word bottles was only on Column QuantityPerUnit

--to search by bottles we had to use the wild card (%) to specifically find bottles within each cell

--% sign was used in either side of bottles because the bottles might me in between other words

SELECT \* FROM Products

SELECT ProductName FROM Products WHERE QuantityPerUnit LIKE '%bottles%';

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

--in order to get the supplier name and country we had to INNER JOIN another table (suppliers table)

--INNER JOIN was used becuase supplier id was common between both tables

SELECT p.ProductName, s.CompanyName, s.Country FROM Products p

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

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.

--since CategoryID was common between both tables we had to INNER JOIN Products and ----Category table to get all the detials

--to COUNT the number of products in each category we had to group by category

--and count how many products were in each group (i.e category)

-- to list the highest number we used ORDER BY DESC

SELECT c.CategoryName, COUNT(p.CategoryID) AS 'Number of products' FROM Products p

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

GROUP BY c.CategoryName

ORDER BY COUNT(p.CategoryID) 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 'UK Emplyees', City FROM Employees;

-- to concatenate different columsn we use + symbol and give space between the column names we simply use ' ' with space in between.

* 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.

--for this question we had to use 4 joins

--since order details table was needed to calculate total sum,

--we had to join it to other tables to get to territories table

--we grouped them by regionID and found the total sum by sum(od.UnitPrice\*od.Quantity)

--since the total sum was already greater than 1000,000 there was no need to write an statement for it.

SELECT SUM(od.UnitPrice\*od.Quantity) AS 'Sales Totals for Regoins' FROM Territories t

JOIN EmployeeTerritories et ON t.TerritoryID = et.TerritoryID

JOIN Employees e ON et.EmployeeID = e.EmployeeID

JOIN Orders o ON e.EmployeeID = o.EmployeeID

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

GROUP BY t.RegionID

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

--for this question all the required details were on Orders table

--to count the number of orders we will be using COUNT function and counting all the orders (\*)

--to find the ship country, we can use IN () which is another way of using OR condition e.g. ShipCountry = UK OR USA

--we the AND condition to find Freight greater than 100

select COUNT(\*) AS 'Number of Freight Orders' FROM Orders WHERE ShipCountry IN ('USA', 'UK') AND Freight>100.00;

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

--on the ORDER DETIALS table the order Number was OrderID

--we can use DESC order to find the largest number and ORDER BY Clause

--max discount is Discount\*UnitPrice

## SELECT TOP 1 OrderID, Discount\*UnitPrice FROM [Order Details] ORDER BY Discount\*UnitPrice 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.

\*/

--to create a database we do CREATE DATABASE <name of database>

--CREATE DATABASE Spartans;

USE Spartans --to specify which database to use

/\*

to create a table within a database we do

CREATE TABLE <name of the table>

(

<name of col> <type>

.

.

);

we can specify the data type depending on the type of data e.g. for name we use VARCHAR(SIZE)

we can also make the ID auto increment by using IDENTITY(<start>,<increment by>)

we can set the ID as primary key using PRIMARY KEY keyword.

CREATE TABLE DevOps\_Spartans

(

Student\_ID int IDENTITY(1,1) PRIMARY KEY,

Title VARCHAR(5),

St\_Name VARCHAR(20) NOT NULL,

St\_Surname VARCHAR(20) NOT NULL,

Uni\_Attented VARCHAR(15),

Uni\_Course VARCHAR(15),

Marks\_achieved DEC(4,2), --4 digits, 2 digits after decimal point

Hobbies VARCHAR(20)

);

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

to insert data into a table:

INSERT INTO <table name> (<col1 name>,<col2 name>,...) VALUES (<col1 value>,<col2 value>...),

(<col1 value2>,<col2 value2>...);

INSERT INTO DevOps\_Spartans (Title, St\_Name, St\_Surname, Uni\_Attented, Uni\_Course, Marks\_achieved, Hobbies)

VALUES ('Mr','Mujeebullah', 'Noori', 'Queen Mary', 'Electronics', 75.54, 'Go out'),

('Mr', 'Adam', ' Koyuncu', NULL, 'Business', 90.00, 'to teach'),

('Mr', 'Alley', 'Preston', 'Brunel', 'MAthematics', 90.00, 'Chill');

--I only managed to find details of 3 people including myself, since it was weekend I wasn’t able to get in touch with everyone to get their details. But the concept is the same if I was to add everyone else’s details.

--to print out the content of the table

SELECT \* FROM DevOps\_Spartans;

\*/

**Exercise 3 – Northwind Data Analysis linked to Excel**

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

--to do this question, one must first find each employee that reports to which ---------employee

--that was done by 'select ReportsTo from Employees'

--then used that result as subquery for the outer query, where we are finding their ----full name

--the link between the two query was done using the employeeID

SELECT EmployeeID, FirstName, LastName, ReportsTo FROM Employees

WHERE EmployeeID IN (select ReportsTo from Employees)

Another way:

--is by using joins

SELECT e.EmployeeID, e.LastName, e.FirstName, em.ReportsTo from Employees e

JOIN Employees em ON e.ReportsTo = em.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:

--initially i did:

-- joined the order details table and suppliers table

--group by companyname

SELECT \* FROM (

SELECT s.CompanyName, round(sum(od.UnitPrice\*Quantity-Discount),2) AS 'Total Sales' from Products p

JOIN Suppliers s ON p.SupplierID = s.SupplierID

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

group by s.CompanyName) Inner\_query\_name

WHERE [Total Sales] > 10000

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.

--this is to get the top 10 customers from order table by latest date

SELECT top 10 \* FROM Orders order by OrderDate desc

--this is to get the orderid and the total value of orders using group by clause

SELECT OrderID ,SUM(UnitPrice\*Quantity) FROM [Order Details]

GROUP BY OrderID

-- i tried to join the order details table with orders table to get the customerID but i was facing errors

SELECT top 10 SUM(od.UnitPrice\*od.Quantity) FROM [Order Details] od

INNER JOIN Orders o ON od.OrderID = o.OrderID

GROUP BY od.OrderID

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

--since i was not possible to find the avg of date we had to use cast function

--i found the avg ship time by year, however i was not able to find the ship date by month

SELECT CAST(AVG(CAST(ShippedDate AS FLOAT)) AS DATETIME) AS 'Average ship date' FROM Orders

GROUP BY CustomerID