SQL Practical Exercise

### 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],PostalCode,City,Country

FROM Customers

Where City = 'London' OR City = 'Paris';

* 1. List all products stored in bottles.

SELECT ProductName "Product Name" FROM Products

WHERE QuantityPerUnit LIKE '%bottles%';

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

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

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

WHERE QuantityPerUnit LIKE '%bottles%';

*/\* In this I have found products stored in bottle, I have used an INNER JOIN*

*for Suppliers and Tables to obtain the Country and Company Name\*/*

* 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 AS "Category Name", SUM(c.CategoryID) AS "Total"

FROM Categories c

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

GROUP BY c.CategoryID, c.CategoryName

ORDER BY "Total" DESC;

*/\*The orders are grouped by the CategoryID From the Category table. The products*

*have also been ordered from Highest to Lowest using the Total value.\*/*

* 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 'Full Name' , City

FROM Employees;

*--Using the Concatenate (+) for the name values*

* 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(UnitPrice \* Quantity \* (1-Discount)),-1) 'Sales'

FROM Territories t

INNER JOIN Region r ON r.RegionID = t.RegionID

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

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

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

GROUP BY r.RegionID,r.RegionDescription

HAVING 'Sales' > '1000000'

ORDER BY 'Sales' DESC

*/\* This query uses 4 Inner Joins between 4 Data Sets in order to find the needed Data Set.*

*Beginning from Territories have used Region for the RegionID, EmployeeTerritories for*

*the Territory ID, Orders for the EmployeeID and Order Details to obtain the Order ID.*

*Using these I have grouped the Regions together to group the sales, and searched for the regions*

*which have 'Sales' Greater than a 1000000, which is all of them.*

*Finally, I have ordered them by Sales \*/*

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

SELECT Count(OrderID) AS "Countries with Freight over 100.00"

FROM Orders

WHERE (ShipCountry ='UK' OR ShipCountry = 'USA') AND (Freight > 100.00);

*/\*I have counted the Orders where the Freight number is greater than 100.00*

*This was done using WHERE =, Although it would be more efficient for the program to use WHERE IN\*/*

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

SELECT (UnitPrice\*Quantity\*Discount) AS "Total Ammount", OrderID FROM

[Order Details]

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

*/\*Using a subquery to SELECT the MAX discount FROM Order Details, I have listed the orders with the highest discount.\*/*

### 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.

*--Exercise 2*

CREATE DATABASE spartans\_db

USE spartans\_db

*--DROP TABLE Engineering54*

CREATE TABLE Engineering54 (

SpartaID INT IDENTITY(1,1),

title VARCHAR(20),

firstName VARCHAR(225) NOT NULL,

lastName VARCHAR(225) NOT NULL,

universityName VARCHAR(225),

courseTaken VARCHAR(225),

universityGrade VARCHAR(4),

City VARCHAR(15) DEFAULT 'London' NOT NULL,

PRIMARY KEY (SpartaID)

);

*/\* To create the table, a database was created first then I used the CREATE TABLE with the various columns*

*I have used VARCHAR(225) in firstName, lastName, universityName and courseTaken as the values in the columns*

*do vary in length\*/*

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

INSERT INTO Engineering54(title,firstName, lastName,

universityName, universityGrade,City,courseTaken)

VALUES ('Mr','Victor', 'Sibanda', 'University of Lincoln', '1st','London','Electrical Engineering'),

('Miss','Jade', 'Meyer', 'Imperial University', '2:1','London','English'),

('Dr','Ella', 'Jacobs', 'Kings College London', '2:1','London','Japanese Culture Studies'),

('Mr','Ayo', 'Adebayo', 'University of Birmingham', '1st','Birmingham','Mathematics'),

('Mrs','Aisha', 'Wood', 'University of Cambridge', '3rd','London','Computer Science'),

('Mr','John', 'Smith', 'University of East Anglia', '2:2','London','Social Studies');

*--Inserts Values to the table*

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

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

SELECT EmployeeID,TitleOfCourtesy+' ' +firstName + ' ' + lastName AS "Name" ,

ReportsTo AS "Employee Reports To"

FROM Employees;

*--Tells us the EmployeeID of whom the Employee Reports to.*

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.SupplierID, s.CompanyName AS "Company Name",

FORMAT(SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)),'N', 'en-uk') AS "Total Sales"

FROM Suppliers s

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

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

GROUP BY s.SupplierID, s.CompanyName

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

ORDER BY s.SupplierID ;

*/\*Suppliers with sales greater than 1000, I have used two INNER Joins for Suppliers on Products and Order Details*

*As I need the Order Details to get the sum of SALES. FORMAT is used here to give the Sales as a Number with 2 Decimal Places*

*in english\*/*

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 o.CustomerID,c.CompanyName AS "Company Name",

ROUND(SUM(od.UnitPrice\*od.Quantity\*(1-od.Discount)),2) AS 'Total Value' ,

Year(ShippedDate) AS "Year"

FROM ((Orders o

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

INNER JOIN Customers c ON c.CustomerID=o.CustomerID)

GROUP BY o.CustomerID, c.CompanyName, Year(ShippedDate)

HAVING Year(ShippedDate) = (SELECT Year(MAX(ShippedDate)) FROM Orders)

ORDER BY 'Total Value' DESC ;

*/\* Used Top 10 to get top 10 companies from the latest year in the Data which is the MAX Year*

*Value in the data. This was done using the ship date. \*/*

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(ShippedDate) AS "Month",

AVG(DATEDIFF(DD,OrderDate,ShippedDate)) AS "Average Shipping Time(Days)"

FROM Orders

GROUP BY MONTH(ShippedDate)

HAVING AVG(DATEDIFF(DD,OrderDate,ShippedDate)) IS NOT NULL

ORDER BY "Month";

*--USE of DATEDIFF and AVG to calculate the Average shipping time.*