## 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 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

**Table

Description automatically generated**

* 1. **List all products stored in bottles.**

SELECT QuantityPerUnit FROM Products p

WHERE p.QuantityPerUnit LIKE '%Bottle%' –- Selects all products with bottle containers mentioned

**Table

Description automatically generated**

* 1. **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%'

**Table

Description automatically generated**

* 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, 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

**Table

Description automatically generated**

* 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 + ', ' + City AS "Name And City"

FROM Employees

WHERE Country = 'UK' –- Only select those records with UK as the country

**Table

Description automatically generated**

* 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.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

**Table

Description automatically generated**

* 1. **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

**Graphical user interface, text, application

Description automatically generated**

* 1. **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



## 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,

)

****

**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')

Graphical user interface, table

Description automatically generated

## 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.

Table

Description automatically generated

**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.

**Table

Description automatically generated**

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

**Chart

Description automatically generated**

**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.

**Table

Description automatically generated**

**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

**Table

Description automatically generated**

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

Chart, line chart

Description automatically generated