## 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.

1. SELECT CustomerID, CompanyName, Address, City, Region, PostalCode, Country
2. FROM Customers
3. WHERE (City IN ('Paris', 'London'));
   1. List all products stored in bottles.
4. SELECT ProductID AS 'Product ID',
5. SupplierID AS 'Supplier ID',
6. CategoryID AS 'Category ID',
7. ProductName AS 'Product Name',
8. QuantityPerUnit AS 'Quantity Per Unit',
9. UnitPrice AS 'Unit Price',
10. UnitsInStock AS 'Units In Stock',
11. UnitsOnOrder AS 'Units On Order',
12. ReorderLevel AS 'Reorder Level',
13. Discontinued
14. FROM Products
15. WHERE QuantityPerUnit LIKE '%bottles%';
    1. Repeat question above, but add in the Supplier Name and Country.
16. SELECT
17. ProductID AS 'Product ID',
18. p.SupplierID AS 'Supplier ID',
19. CategoryID AS 'Category ID',
20. CompanyName AS 'Company Name',
21. ProductName AS 'Product Name',
22. QuantityPerUnit AS 'Quantity Per Unit',
23. UnitPrice AS 'Unit Price',
24. UnitsInStock AS 'Units In Stock',
25. UnitsOnOrder AS 'Units On Order',
26. ReorderLevel AS 'Reorder Level',
27. Discontinued,
28. Country
29. FROM (Products AS p INNER JOIN
30. (SELECT SupplierID, CompanyName, Country FROM Suppliers) AS s
31. ON p.SupplierID = s.SupplierID)
32. 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.
33. SELECT p.CategoryID AS 'Category ID',
34. c.CategoryName AS 'Category Name',
35. COUNT(ProductName) AS 'Total'
36. FROM (Products AS p INNER JOIN
37. (SELECT CategoryID, CategoryName FROM Categories) AS c
38. ON c.CategoryID = p.CategoryID)
39. GROUP BY p.CategoryID, c.CategoryName
40. ORDER BY COUNT(ProductName) 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.
41. SELECT TitleOfCourtesy + FirstName + ' ' + LastName AS 'Name',
42. City AS 'City of Residence'
43. FROM Employees
44. WHERE Country LIKE '%UK%';
    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.
45. SELECT RegionDescription AS 'Region',
46. FORMAT(SUM((UnitPrice\*Quantity\*(1-Discount))), 'N','en-uk') AS 'Total Sales'
47. FROM Region r
48. INNER JOIN
49. (SELECT TerritoryID, RegionID FROM Territories) t ON r.RegionID = t.RegionID
50. INNER JOIN
51. (SELECT \* FROM EmployeeTerritories) e ON e.TerritoryID=t.TerritoryID
52. INNER JOIN
53. (SELECT EmployeeID, OrderID FROM Orders) o ON o.EmployeeID=e.EmployeeID
54. INNER JOIN
55. [Order Details] od ON o.OrderID=od.OrderID
56. GROUP BY RegionDescription HAVING SUM((od.UnitPrice\*od.Quantity\*(1-od.Discount))) >= 1000000
57. ORDER BY [Total Sales] DESC;
    1. Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.
58. SELECT COUNT(Freight) AS 'Total Orders (> 100 AND UK OR USA)' FROM Orders
59. WHERE (Freight > 100.00) AND (ShipCountry IN ('USA','UK'));
    1. Write an SQL Statement to identify the Order Number of the Order with the highest amount of discount applied to that order.
60. SELECT OrderID AS 'Order ID',
61. (UnitPrice\*Quantity\*Discount) AS 'Discount Amount'
62. FROM [Order Details]
63. WHERE (UnitPrice\*Quantity\*Discount) =
64. (SELECT MAX(UnitPrice\*Quantity\*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:

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.

IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.

CREATE TABLE [Spartans Table](

studentID INT IDENTITY(1,1),

title varchar(10) NOT NULL,

firstName varchar(30) NOT NULL,

lastName varchar(30) NOT NULL,

universityAttended varchar(40) DEFAULT NULL,

courseAttended varchar(40) DEFAULT NULL,

markAchieved INT DEFAULT NULL,

PRIMARY KEY (studentID)

);

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]

(title, firstName, lastName, universityAttended, courseAttended, markAchieved)

VALUES ('Mr.', 'Ayman', 'Yousfi', 'West London', 'Computer Science', 69),

('Mr.', 'Camile', 'Malungu', 'Brunel', 'Computer Science', 57),

('Miss', 'Sara', 'Abdrabu', 'Westminster', 'Computer Networks with Security', 67),

('Mr.', 'Adam', 'Mohsen', 'Sussex', 'Computer Science', 70),

('Mr.', 'Abdullah', 'Ayyaz', 'Westminster', 'Business Economics', 69),

('Mr.', 'Ash', 'Isbitt', 'Brunel', 'Visual Effect and Motion Graphics', 68),

('Mr.', 'Elliot', 'Harris', 'CCC', 'History', 58),

('Mr.', 'James', 'Hovell', 'Portsmouth', 'Mathematics', 67),

('Mr.', 'Mahan', 'Yousfi', 'Portsmouth', 'Mathematics', 82),

('Mr.', 'Maksaud', 'Ahmed', '', '', ''),

('Mr.', 'Mohammad', 'Uddin', 'Greenwich', 'Computer Science', 46),

('Mr.', 'Victor', 'Sibanda', 'Lincoln', 'Electrical Engineering', 74),

('Mr.', 'Zack', 'Davenport', 'UEA', 'Film and TV', 55);

## 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)

SELECT EmployeeID AS 'Employee ID',

TitleOfCourtesy + ' ' + FirstName + ' ' + LastName AS 'Name',

ReportsTo AS 'Reports To' FROM Employees;

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 AS 'Supplier ID', s.CompanyName AS 'Company Name',

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

FROM (([Order Details] od

INNER JOIN Products p ON od.ProductID=p.ProductID)

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

GROUP BY s.SupplierID, s.CompanyName

HAVING SUM((od.UnitPrice\*od.Quantity\*(1-od.Discount)))>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. (10 Marks)

SELECT TOP 10 o.CustomerID AS 'Customer ID', 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;

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 'AVG Ship Time (Days)'

FROM Orders

GROUP BY MONTH(ShippedDate)

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

ORDER BY [Month];

## 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