## 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.

USE Northwind;

SELECT CustomerID AS 'Customer ID', --Saves customer id and company name in user friendly format

CompanyName AS 'Company Name',

Address + ', ' + City + ', ' + ISNULL(Region + ', ', '') + Country + ', ' + PostalCode AS 'Full Address'

FROM Customers --Concatenates address from all address fields. If region is null it is not output, if it isn't it outputs the region.

WHERE City IN('Paris', 'London'); --Only displays customers from Paris or London

* 1. List all products stored in bottles.

USE Northwind;--Displays product name in user friendly format

SELECT ProductName AS 'Product Name'

FROM Products

WHERE QuantityPerUnit LIKE '%bottle%';

--Objects stored in bottles have bottle somewhere in their quantity

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

USE Northwind;

SELECT ProductName AS 'Product Name',

CompanyName AS 'Supplier Name',

Country --Already user friendly

FROM Products--As before except suppliers table joined for extra fields

JOIN Suppliers ON Suppliers.SupplierID = Products.SupplierID

WHERE QuantityPerUnit LIKE '%bottle%';

* 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.

USE Northwind;

SELECT CategoryName, --Take name of each category

Count(p.CategoryID) AS 'Amount of Products' --Count the amount of products for each category

FROM Categories c --Save categories table under variable c

JOIN Products p ON c.CategoryID = p.CategoryID --Join products table

GROUP BY CategoryName --Group products by category

ORDER BY [Amount of Products] DESC; --Display category with highest amount of products first.

* 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.

USE Northwind; --Concatenate relevant fields with spaces between

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

City --City is already a user friendly field

FROM Employees

WHERE Country = 'UK' --Only UK Employees

* 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.

USE Northwind;

SELECT RegionDescription AS 'Region', --Take the region description as the region

ROUND(SUM((UnitPrice -Discount) \* Quantity), 2) AS 'Total Sales' --Add up the sales for each region calculated by unit price - discount \* quantity. Round this to 2 dp.

FROM [Order Details] od

JOIN Orders o ON o.OrderID = od.OrderID --Join all relevant tables to get from order details to regions

JOIN EmployeeTerritories et ON et.EmployeeID = o.EmployeeID

JOIN Territories t ON et.TerritoryID = t.TerritoryID

JOIN Region r ON r.RegionID = t.RegionID

GROUP BY r.RegionDescription --Group by region

HAVING SUM((UnitPrice -Discount) \* Quantity) > 1000000; --Having instead of where as selecting groups.

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

USE Northwind; --Count all orders and give relevant title

SELECT Count(\*) AS 'Orders from UK/USA with Freight above 100'

FROM Orders

WHERE Freight > 100 AND ShipCountry IN('UK','USA')

--Only where both the freight is above 100 and shipping from UK or USA

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

USE Northwind; --Take the number 1 order number and give relevant name

SELECT TOP 1 OrderID AS 'Order number with highest discount'

FROM [Order Details]

ORDER BY (Quantity \* Discount) DESC --More products bought = higher discount

--Order by discount so first order has 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.

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

USE Northwind; --In the Northwind database

CREATE TABLE Spartans ( --Create a spartans table

SpartanID int IDENTITY PRIMARY KEY NOT NULL, --Primary key is auto increment int

Title VARCHAR(8),--Title up to 8 characters can be null

FName VARCHAR(16) NOT NULL, -- First name and last name cannot be null and must be 16 characters or less

LName VARCHAR(16) NOT NULL,

Uni VARCHAR(32),--32 characters for uni to accommodate most uni names, can be null if none intended

Course VARCHAR(64),--64 characters to allow for complex course names

Mark CHAR(3),--Mark taken as classification, must be 3 characters

CHECK (Mark IN('1st','2:1','2:2','3rd','DNP'))--Mark must either be the 4 classifications from 1st to 3rd or DNP for did not pass.

);

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

USE Northwind;

INSERT INTO Spartans

VALUES('Mr', 'Matthew', 'Mackreth', 'Queen Mary', 'Computer Science', 'DNP'),

('Miss', 'Payal', 'Nayee', 'Oxford', 'Physics', '1st'),

(NULL, 'Micheal', 'Awosemo', NULL, NULL, NULL),

('Mr', 'Adam', 'Koyuncu', NULL, NULL, NULL),

(NULL, 'Ally', 'Preston', 'Southampton', 'Maths', '2:1')

--Adding values collectively of spartans to demonstrate possible arrangements of data

## 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)

USE Northwind;--Using northwind

SELECT e.TitleOfCourtesy + ' ' + e.FirstName + ' ' + e.LastName AS 'Full Employee Name', --Concatenate the full employee name

r.TitleOfCourtesy + ' ' + r.FirstName + ' ' + r.LastName AS 'Full Supervisor Name'--Concatenate the full refers to, or supervisor, name

FROM Employees e --Take employee table as e

JOIN Employees r ON r.EmployeeID = e.ReportsTo; --Join to itself taking second instance as r.

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)

USE Northwind;

SELECT CompanyName AS 'Company Name', --Take each supplier name

ROUND(SUM(Quantity \* (od.UnitPrice -Discount)),2) AS 'Supplier Sales Total' --Work out total for each company and round to 2dp for currency

FROM [Order Details] od

JOIN Products p ON od.ProductID = p.ProductID --Join relevant tables

JOIN Suppliers s ON s.SupplierID = p.SupplierID

GROUP BY CompanyName --Group by name of supplier

HAVING SUM(Quantity \* (od.UnitPrice -Discount)) > 10000 --Only take where total sales is over 10,000



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)

USE Northwind;

SELECT TOP 10 CompanyName AS 'Customer Name', --Take the top 10 Customers

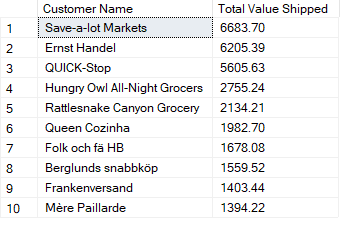
SUM(Freight) AS 'Total Value Shipped' --Show the total value shipped

FROM Orders o --From orders table

JOIN Customers c ON c.CustomerID = o.CustomerID --Joining customers

GROUP BY CompanyName--Group by customer name

ORDER BY [Total Value Shipped] DESC;--Display the highest value first and descend.



3.4 Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below. (10 Marks)

USE Northwind;

SELECT MONTH(ShippedDate) AS 'Shipped Month', --Take the month it was shipped in

AVG(DATEDIFF(d, OrderDate, ShippedDate)) AS 'Ship Time' --Calculate how long shipping took and average across every instance of each month

FROM Orders

WHERE ShippedDate IS NOT NULL AND DATEDIFF(d, OrderDate, ShippedDate) IS NOT NULL --Ignore NULL values

GROUP BY MONTH(ShippedDate)--Group by month

ORDER BY [Shipped Month];--Display in month order

## 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