## 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**

**FROM CUSTOMERS c**

**WHERE City IN('Paris ','London')**

* 1. List all products stored in bottles.

**SELECT p.ProductName,p.QuantityPerUnit FROM products p**

**WHERE p.QuantityPerUnit LIKE '%bottles%'**

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

**SELECT s.CompanyName,s.Country,p.QuantityPerUnit,p.ProductName FROM products p**

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

**WHERE p.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.

**SELECT c.CategoryName,COUNT(ProductID) AS "Product\_count" FROM Products p**

**INNER JOIN Categories c On c.CategoryID=p.CategoryID**

**GROUP BY c.CategoryName**

**ORDER BY Product\_count 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 e.TitleOfCourtesy+' '+ e.FirstName+' '+e.LastName AS "Employees in uk",e.Country,e.City**

**FROM Employees e**

**WHERE e.Country='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.

**SELECT r.RegionDescription,ROUND (SUM(od.UnitPrice\*od.Quantity) ,0)AS "SALES Total" FROM Territories t**

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

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

**INNER JOIN Employees e ON e.EmployeeID=et.EmployeeID**

**INNER JOIN Orders o ON o.EmployeeID=e.EmployeeID**

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

**GROUP BY r.RegionDescription**

**HAVING ROUND (SUM(od.UnitPrice\*od.Quantity) ,0) > 1000000**

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

**SELECT COUNT(o.OrderID) as "Num\_Orders"**

**FROM Orders o**

**WHERE o.Freight>100 and o.ShipCountry IN ('USA' , 'UK')**

* 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 od.OrderID, SUM(od.Quantity\*od.UnitPrice\*od.Discount)AS "Dis\_price"**

**FROM[Order Details] od**

**GROUP BY od.OrderID**

**ORDER BY "Dis\_price" 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.

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

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

CREATE DATABASE POORNIMA\_DBs;

CREATE TABLE Spartans\_Table(

Spartans\_id INT IDENTITY(1,1) PRIMARY KEY,

First\_Name VARCHAR(100),

Last\_Name VARCHAR(100),

Separate\_Title VARCHAR(30),

University VARCHAR(100),

Course VARCHAR(100),

Mark\_achieved VARCHAR(50))

INSERT INTO Spartans\_Table(First\_Name,Last\_Name,Separate\_Title,University,course,mark\_achieved) VALUES

('Amith','SH','MR','Prineton','Java SDET','A+'),

('Ravi','DH','Miss','Yale','Engineering','A+'),

('Jhon','LM','MR','Yale','Engineering','A+'),

('Sasha','RH','Miss','Princeton','Engineering','A+'),

('Mike','PH','MR','Yale','Java SDET','A+');

UPDATE Spartans\_Table SET University = 'Princeton' WHERE Spartans\_id=1;

ALTER TABLE Spartans\_Table ALTER Column Last\_Name Varchar(50);

ALTER TABLE Spartans\_Table ADD Email varchar(255);

ALTER TABLE Spartans\_Table ADD PhoneNumber varchar(255)NOT NULL;

UPDATE Spartans\_Table SET Email = 'amith@gmail.com' WHERE Spartans\_id=1;

UPDATE Spartans\_Table SET Email = 'ravi@gmail.com' WHERE Spartans\_id=2;

UPDATE Spartans\_Table SET Email = 'jhon@gmail.com' WHERE Spartans\_id=3;

UPDATE Spartans\_Table SET Email = 'sasha@gmail.com' WHERE Spartans\_id=4;

UPDATE Spartans\_Table SET Email = 'mike@gmail.com' WHERE Spartans\_id=5;

SELECT \*FROM Spartans\_Table;

--DELETE Spartans\_Table Where Spartans\_id=6;

--DROP TABLE Spartans\_Table;

--[SP\_HELP] shows created table table

SP\_HELP Spartans\_Table;

## 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 ee.FirstName AS "EmployeeName",e.FirstName as "ReportsTo" from Employees e**

**LEFT JOIN Employees ee on e.EmployeeID = ee.ReportsTo**

**select ee.FirstName AS "EmployeeName",e.FirstName as "ReportsTo" from Employees e**

**INNER JOIN Employees ee on e.EmployeeID = ee.ReportsTo**

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,SUM((od.UnitPrice\*od.Quantity)\*(1-od.Discount)) AS "Sales"**

**FROM [Order Details] od**

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

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

**GROUP BY s.CompanyName**

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

**ORDER BY "Sales" DESC**



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 c.CompanyName,YEAR(o.OrderDate) AS "OrderYear" ,COUNT(o.orderID)AS "No\_orders"**

**FROM Customers c INNER JOIN Orders o ON c.CustomerID=o.CustomerID**

**WHERE o.ShippedDate IS NOT NULL AND YEAR(o.OrderDate)=(SELECT MAX(YEAR(OrderDate)) FROM Orders)**

**GROUP BY c.CompanyName,YEAR(o.OrderDate)**

**ORDER BY "No\_orders" 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 CONCAT( MONTH(o.OrderDate),'-',YEAR(o.OrderDate)) AS "MonthYear",AVG(DATEDIFF(DAY, o.OrderDate,**

**o.ShippedDate) )**

**AS "Average Ship Time" FROM Orders o**

**WHERE o.ShippedDate IS NOT NULL**

**GROUP BY YEAR(o.OrderDate),MONTH(o.OrderDate)**

**ORDER BY YEAR(o.OrderDate) DESC, MONTH(o.OrderDate)**



## 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