**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, City, PostalCode, Country

FROM Customers

WHERE City = 'London' OR City = 'Paris';

Result:

Table

Description automatically generated

1.2 List all products stored in bottles.

SELECT \*

FROM Products p

WHERE p.QuantityPerUnit LIKE '%bottles%';

Result:

Table

Description automatically generated

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

SELECT p.ProductID, p.ProductName, s.CompanyName, s.Country

FROM Products p

JOIN Suppliers s ON p.SupplierID = s.SupplierID

WHERE p.QuantityPerUnit LIKE '%bottle%';

Result:

Table

Description automatically generated

1.4 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",

COUNT(\*) AS "Number of products in category"

FROM Products p

JOIN Categories c ON p.CategoryID = c.CategoryID

GROUP BY c.CategoryName

ORDER BY COUNT(\*) DESC;

Result:

Table

Description automatically generated

1.5 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 "Name and Title ", e.City AS "City of Residence"

FROM Employees e

WHERE E.Country = 'UK';

Result:

Table

Description automatically generated

1.6 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.Quantity \* od.UnitPrice \* (1 - od.Discount)),2) total\_sales

FROM [Order Details] od

INNER JOIN Orders o ON o.OrderID = od.OrderID

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

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

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

GROUP BY r.RegionDescription HAVING ROUND(SUM(od.Quantity \* od.UnitPrice),2)>1000000

Result:

Table

Description automatically generated

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

SELECT COUNT(OrderID) AS "Orders", o.ShipCountry AS "Ship Country"

FROM Orders o

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

GROUP BY o.ShipCountry

Result:

Table

Description automatically generated

1.8 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.UnitPrice\*od.Quantity\*od.Discount) AS "Biggest\_amount\_of\_discount", od.OrderID

FROM [Order Details] od

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

ORDER BY Biggest\_amount\_of\_discount DESC

Result:

Graphical user interface, text, application

Description automatically generated with medium confidence

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.

DROP TABLE Spartan\_table

CREATE TABLE Spartan\_table (

PersonID INT IDENTITY(1,1) PRIMARY KEY,

Title CHAR(5),

First\_Name VARCHAR(200),

Last\_Name VARCHAR(200),

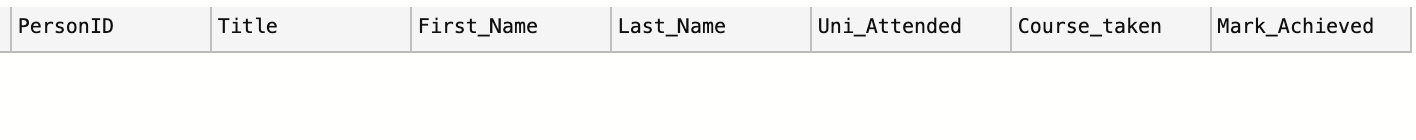
Uni\_Attended VARCHAR(200),

Course\_taken VARCHAR(200),

Mark\_Achieved DECIMAL(2,1),

);

Result:



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

INSERT INTO Spartan\_table (

Title, First\_Name, Last\_Name, Uni\_Attended, Course\_taken, Mark\_Achieved

)

VALUES(

'Mr.' , 'Kourosh' , 'Philip' , 'University College London' , 'German with Management Studies BA' , '2.1'

),

(

'Miss.' , 'Naveen' , 'Bo' , 'Cardiff University' , 'Data Science BSc' , '1.1'

),

(

'Dr.' , 'Nereida' , 'Tatiana' , 'Oxford University' , 'Applied Medical Sciences BSc' , '2.2'

);

Result:

Text

Description automatically generated

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.

SELECT CONCAT(e.TitleOfCourtesy, e.FirstName, ' ' ,e.LastName) AS "Employee Name", CONCAT(id.TitleOfCourtesy, id.FirstName, ' ' ,id.LastName) AS 'Reports To'

FROM Employees e

LEFT JOIN Employees id

ON id.EmployeeID=e.ReportsTo

Result:

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

SELECT s.CompanyName,ROUND(SUM(od.Quantity \* (od.UnitPrice - od.UnitPrice \* od.Discount)),0) AS "Total Sales"

FROM Suppliers s

JOIN Products p ON s.SupplierID = p.SupplierID

JOIN [Order Details] od ON p.ProductID = od.ProductID

GROUP BY s.CompanyName

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

Result:

Table

Description automatically generated

A picture containing 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.

SELECT TOP 10 c.CompanyName AS "Company Name" , ROUND(SUM(od.Quantity\*(od.UnitPrice-od.UnitPrice\*od.Discount)),2) AS "Total Value Of Orders Shipped"

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

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

WHERE YEAR(o.OrderDate) = (SELECT MAX(YEAR(oo.OrderDate)) FROM Orders oo)

GROUP BY c.CompanyName

ORDER BY "Total value of orders shipped" DESC

Result:

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.

SELECT AVG(DATEDIFF(DAY, o.OrderDate , o.ShippedDate)) AS "Avg Ship Time", FORMAT(o.OrderDate , 'MMM-yyyy') AS "Month"

FROM Orders o

GROUP BY FORMAT(o.OrderDate,'MMM-yyyy') , DATEPART(YEAR, o.OrderDate) , DATEPART(MONTH, o.OrderDate)

ORDER BY DATEPART(YEAR, o.OrderDate) , DATEPART(MONTH, o.OrderDate)

Result:

Table

Description automatically generated

Chart, line chart

Description automatically generated