

SQL Project Work

Name: MacAnderson Uche



August 1, 2021

## 

## TABLE OF CONTENT

[Introduction TABLE OF CONTENT 1](#_Toc78374526)

[Exercise 1 – Northwind Queries (40 marks: 5 for each question) 2](#_Toc78374527)

[Exercise 2 – Create Spartans Table (20 marks – 10 each) 3](#_Toc78374528)

[Exercise 3 – Northwind Data Analysis linked to Excel (30 marks) 4](#_Toc78374529)

[Standards (10 marks) 5](#_Toc78374530)

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 CustomerID, ContactName, Address

FROM Customers

WHERE City IN ('Paris', 'London')

* 1. List all products stored in bottles.

SELECT QuantityPerUnit

FROM Products

WHERE QuantityPerUnit LIKE '%bottle%'

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

SELECT QuantityPerUnit, ContactName AS "Supplier Name", Country

FROM Products p INNER JOIN Suppliers s ON p.SupplierID = s.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.

SELECT COUNT(\*) AS "Products per Category", c.CategoryName

FROM Categories c INNER JOIN Products p

    ON c.CategoryID = p.CategoryID

GROUP BY p.CategoryID, c.CategoryName

ORDER BY COUNT(p.CategoryID) 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 TitleOfCourtesy+ ' ' + FirstName + ' ' + LastName

AS "Full Details", City

FROM Employees

WHERE 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 FORMAT(SUM((UnitPrice\*Quantity\*(1-Discount))),'C', 'en-gb') AS 'Total Sales', r.RegionID

FROM Region r INNER JOIN Territories t

    ON t.RegionID= r.RegionID

    INNER JOIN EmployeeTerritories et

    ON et.TerritoryID = t.TerritoryID

    INNER JOIN Employees e

    ON et.EmployeeID = e.EmployeeID

    INNER JOIN Orders o

    ON o.EmployeeID = e.EmployeeID

    INNER JOIN [Order Details] od

    ON od.OrderID = o.OrderID

GROUP BY r.RegionID

HAVING (SUM((UnitPrice\*Quantity\*(1-Discount)))) > 1000000

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.

SELECT COUNT (\*) AS "Number of Orders Greather Than 100 and Either UK or US"

FROM Orders

WHERE Freight > 100.00 AND ShipCountry IN ('UK', 'USA')

* 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

    od.OrderID  AS "Order Number(s)", SUM(od.UnitPrice\*od.Quantity\*od.Discount) AS "Discounted Value"

FROM [Order Details] od

GROUP BY od.OrderID

ORDER BY SUM(od.UnitPrice\*od.Quantity\*od.Discount) DESC

## Exercise 2 – Create Spartans Table (20 marks – 10 each)

2.1 Write the correct SQL statement to create the following table.

CREATE DATABASE MacSQLq2\_db

USE MacSQLq2\_db

CREATE TABLE spartans\_table

(

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

    title VARCHAR(30),

    first\_name VARCHAR(30),

    last\_name VARCHAR(30),

    university\_attended VARCHAR(50),

    course\_taken VARCHAR(30),

    Mark VARCHAR(30)

)

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, first\_name, last\_name, university\_attended, course\_taken, Mark

    )

VALUES

    (

        'Mr', 'Joe', 'Hilton', 'Uni of Bolton', 'Animal Science', '69%'

),

    ('Mr', 'Nirel', 'Warde', 'Oxford Brookes', 'Politics', '39%'),

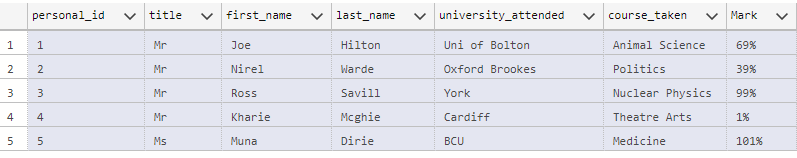
    ('Mr', 'Ross', 'Savill', 'York', 'Nuclear Physics', '99%'),

    ('Mr', 'Kharie', 'Mcghie', 'Cardiff', 'Theatre Arts', '1%'),

    ('Ms', 'Muna', 'Dirie', 'BCU', 'Medicine', '101%')

SELECT \*

FROM 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 d1.FirstName +' '+ d1.LastName AS 'Employee Name', d2.FirstName +' '+ d2.LastName AS 'Report TO Names'

FROM Employees d1

    LEFT JOIN Employees d2

    ON d1.ReportsTo = d2.EmployeeID

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 'Total Sale'

FROM [Order Details] od INNER JOIN Products p

    ON od.ProductID = p.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 (SUM((od.UnitPrice\*od.Quantity)\*(1 - od.Discount))) 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, ROUND(SUM(od.Quantity\*(od.UnitPrice-od.Discount)),2) AS "Total"

FROM Orders o

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

    INNER JOIN Customers c ON o.CustomerID = c.CustomerID

WHERE YEAR(o.OrderDate)=(SELECT MAX(YEAR(o.OrderDate))

FROM Orders o)

GROUP BY c.CompanyName

ORDER BY "Total" 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 FORMAT("Date",'MM/yyyy') AS "Month/Year",

    AVG("ShipTime") AS "AverageShipTimeByMonth"

FROM (SELECT o.OrderDate AS "Date", MONTH(o.OrderDate) AS "Month", YEAR(o.OrderDate) AS "Year",

        DATEDIFF(DAY, o.OrderDate, o.ShippedDate) AS "ShipTime"

    FROM Orders o) AS "ShipTimes"

GROUP BY FORMAT("Date",'MM/yyyy'), "Month", "Year"

ORDER BY "Year" ASC, "Month" ASC

## 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