Homework Answers

Q1.1)

USE Northwind;

SELECT CustomerID AS 'Customer ID',

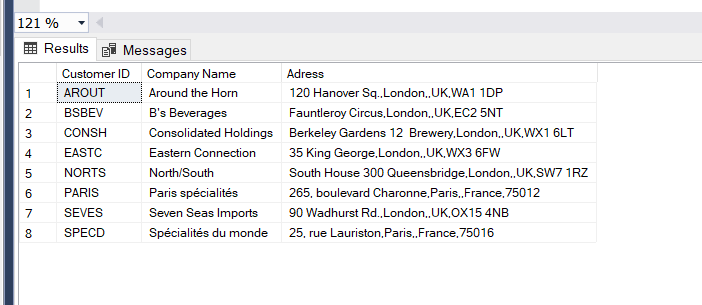
CompanyName AS 'Company Name',

address + ',' + City +',' + ISNULL (Region + ',' ,',') + country + ',' + postalcode AS 'Adress'

FROM

Customers

WHERE city IN ('paris','london');



Q1.2)

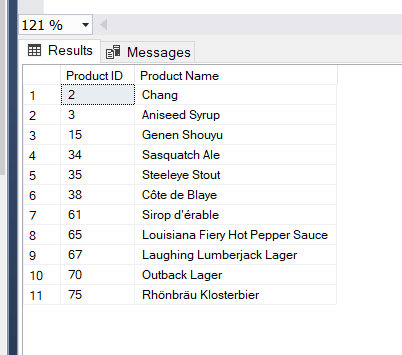
SELECT ProductID AS 'Product ID',

ProductName AS 'Product Name'

FROM

Products

WHERE QuantityPerUnit LIKE '%bottles%';



Q1.3)

-- Selecting columns and aliasing it to its appropriate names

SELECT ProductID AS 'Product ID', ProductName AS 'Product Name', Suppliers.CompanyName

AS 'Suppliers Name', Suppliers.Country AS 'Suppliers Country'

FROM Products

--joining suppliers table

join Suppliers ON Products.SupplierID = suppliers.SupplierID

--where clause, using LIKE to find all the products stored in bottle

WHERE QuantityPerUnit LIKE '%bottles%';



Q1.4)

-- Selecting columns and aliasing it to its appropriate names

--using count function, to count the amount of products in a table

SELECT COUNT(Products.ProductName) AS 'Number of Products', categories.CategoryName AS 'Category Name'

FROM Products

--joining categories table

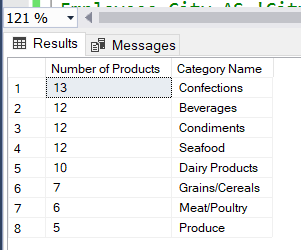
-- group ny clause to group the items by the column

--order by clause to order the list from highest to lowest

JOIN Categories ON categories.CategoryID = Products.CategoryID

Group BY categories.CategoryName

ORDER BY 'Number of Products' DESC;



Q 1.5)

-- concatenating title, first name and last name into one column adding city into another

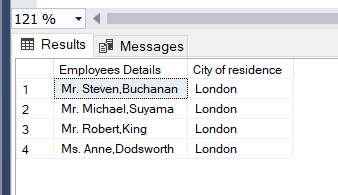
SELECT Employees.TitleOfCourtesy + ' ' + Employees.FirstName + ',' + Employees.LastName AS 'Employees Details',

Employees.City AS 'City of residence'

FROM Employees

WHERE Employees.Country = 'UK';

--where clause to find employees whos country is uk



Q 1.6)

--Selecting the number of total sales (using sum function (quantity\*price) then using the round function tot round to 0)

SELECT regionDescription AS 'Region', ROUND(SUM(unitprice\*quantity),0) AS 'Total Number of Sales'

FROM [Order Details]

-- joininh orders table,employeeTerroritories,territories and region table using joins

JOIN orders ON orders.OrderID = [Order Details].OrderID

JOIN EmployeeTerritories ON orders.EmployeeID = EmployeeTerritories.EmployeeID

JOIN Territories ON EmployeeTerritories.TerritoryID = Territories.TerritoryID

JOIN Region ON Region.RegionID = Territories.RegionID

--group by clause, grouping the rigons

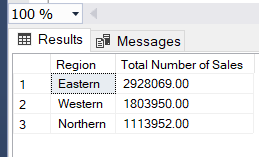
GROUP BY RegionDescription

--having caluse where it shows the total number of sales which ic greater than 1000000.00

HAVING ROUND(SUM(unitprice\*quantity),0) > 1000000.00

--order by the highest number

ORDER BY 'Total Number of Sales' DESC;



Q 1.7)

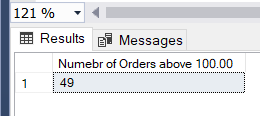
--count functioing counting the numbers of ID in a column

--aliasing it to its correct name

SELECT COUNT(OrderID) AS 'Numebr of Orders above 100.00' FROM Orders

--where clause- show me data where freight is above 100.00 and it is in UK and USA

Where Orders.Freight > 100.00 AND Orders.ShipCountry IN ('UK','USA');



Q1.8)

--select TOP first record

--select amount of discount applied to an item (unitprice\*discount)

SELECT TOP 1 OrderID AS 'Number of Order',

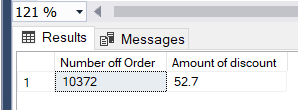
Discount \* UnitPrice AS 'Amount of discount'

FROM

[Order Details]

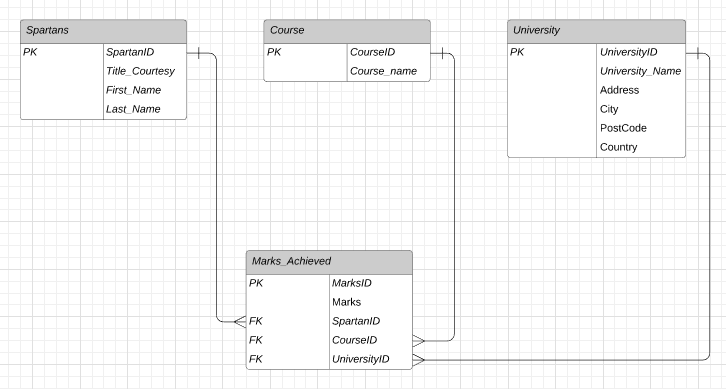
--order the list from hieghest to lowest

ORDER BY Discount \* UnitPrice DESC;



**Exercise 2:**

Q 2.1)



CREATE DATABASE Sparttans;

-- creating Spartn table

CREATE TABLE Spartans (

SpartanID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

courtesy\_title VARCHAR (10),

first\_name VARCHAR (30),

last\_name VARCHAR (30),

);

--Creatring course table

Create table course (

courseID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

course\_name VARCHAR (30)

);

--creating university table

CREATE TABLE university (

uniID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

uni\_name VARCHAR (50),

uni\_address VARCHAR (100),

city VARCHAR (20),

PostCode VARCHAR (10),

Country VARCHAR (20)

);

--creating marks\_achieved table

CREATE TABLE marks\_achieved (

marksID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

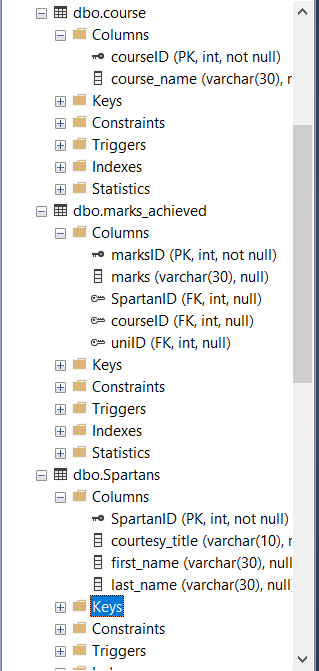
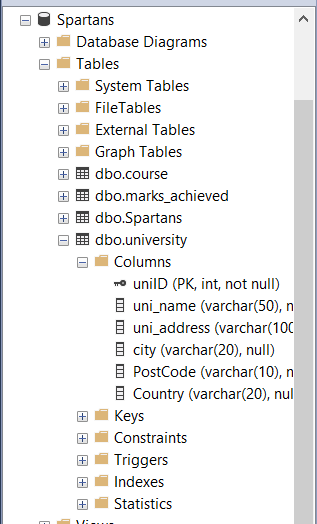
marks VARCHAR (30),

SpartanID INT FOREIGN KEY REFERENCES spartans ( SpartanID),

courseID INT FOREIGN KEY REFERENCES Course ( courseID),

uniID INT FOREIGN KEY REFERENCES university ( uniID),

);

Q 2.2)

--inserting multiple values into the different Tables

INSERT INTO Spartans

(courtesy\_title, first\_name, last\_name)

VALUES

('Mr','Omid','Ayoobe'),

('Miss','Kim','Jones'),

('Mr','james','anderson');

INSERT INTO course

(course\_name)

VALUES

('Computer Science'),

('Economics'),

('Law');

INSERT INTO University

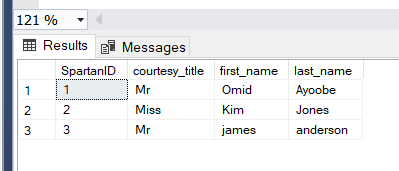
(uni\_name, uni\_address,city,PostCode,Country)

VALUES

('Goldsmiths University of london','8 Lewisham Way, New Cross','London','SE14 6NW','UK'),

('Brunel university','Kingston Ln, Uxbridge','London','UB83PH','UK'),

('Queen Mary','Mile End Rd, Bethnal Green','London','E1 4NS','UK');



**Exercise 3:**

Q 3.1)

----selecting first name of employees from employees table

SELECT

e.firstname + ' ' + e.lastname AS 'employee',

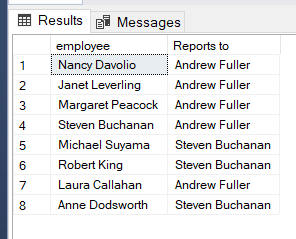
r.firstname + ' ' + r.lastname AS 'Reports to'

----since employees table is linked back, join can be used

FROM

Employees e -- linking the employeeID to the reportToID

JOIN Employees r ON r.EmployeeID = e.ReportsTo



Q 3.2)

--SELECTING sum of sales (unitprice\*quanitiy) and company name

SELECT sum([Order Details].UnitPrice \* [Order Details].Quantity) AS 'Sales', CompanyName AS 'Company Name'

FROM

[Order Details]

--joining suppliers table throu products table then suppliers tbale using primary keys

JOIN Products ON Products.ProductID = [Order Details].ProductID

JOIN Suppliers ON Suppliers.SupplierID = Products.SupplierID

-- grouping by company name - having clause = give me sales is greater than 10000

GROUP BY CompanyName

HAVING sum(quantity\*[Order Details].UnitPrice) > 10000

ORDER BY 'sales' DESC; -- ordering it by highest sales

