**EXERCISE 1**

(Tables shown are samples of the full table)

**1.1**

SELECT CustomerID, CompanyName, Address + ', ' + City AS 'Address', PostalCode , Region, Country

FROM Customers;



**1.2**

SELECT ProductName AS 'Product Name'

FROM Products

WHERE QuantityPerUnit LIKE '%bottles';

****

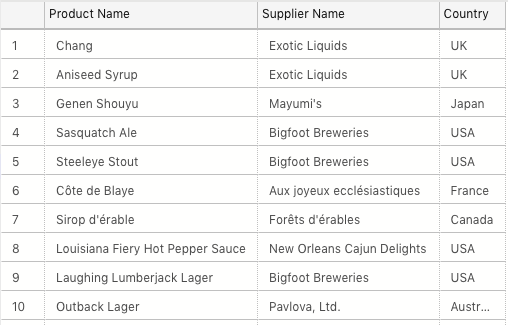
**1.3**

SELECT p.ProductName AS 'Product Name', s.CompanyName AS 'Supplier Name', s.Country

FROM Products p

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

WHERE QuantityPerUnit LIKE '%bottles';

****

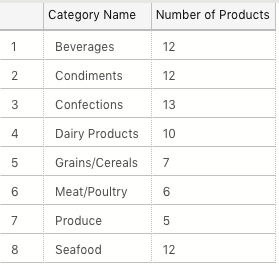
**1.4**

SELECT c.CategoryName AS 'Category Name', COUNT(\*) AS 'Number of Products'

FROM Categories c

INNER JOIN Products p ON c.CategoryID = p.CategoryID

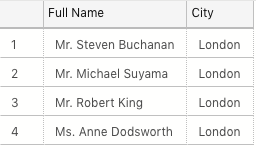
GROUP BY c.CategoryName;

****

**1.5**

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

FROM Employees WHERE Country = 'UK' ;

****

**1.6**

SELECT r.RegionID AS 'Region ID', r.RegionDescription AS 'Region Description', ROUND(SUM(od.Quantity \* od.UnitPrice),0) AS 'Total Sales'

FROM Territories

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

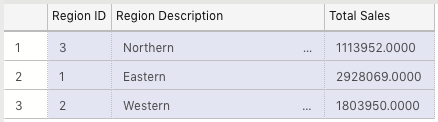
INNER JOIN Region r ON Territories.RegionID = r.RegionID

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

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

GROUP BY r.RegionID, r.RegionDescription

HAVING SUM(od.UnitPrice \* od.Quantity) > 1000000;



**1.7**

SELECT COUNT(\*) AS 'Number of Orders from the UK or USA with a Freight Amount Greater than 100' FROM Orders

WHERE Freight > 100 AND (ShipCountry = 'UK' OR ShipCountry = 'USA') ;



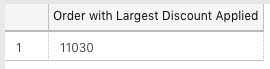
**1.8**

SELECT TOP 1 OrderID AS 'Order with Largest Discount Applied'

FROM [Order Details]

GROUP BY OrderID

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

****

**Exercise 2**

**2.1**

DROP TABLE SpartansTable;

CREATE TABLE SpartansTable

(

Title VARCHAR(10),

FirstName VARCHAR(20),

LastName VARCHAR(20),

University VARCHAR(50),

UniversityCourse VARCHAR(50),

DegreeType VARCHAR(8) DEFAULT 'UNKNOWN',

Grade VARCHAR(7) DEFAULT 'UNKNOWN'

);

**2.2**

(1 example)

INSERT INTO SpartansTable

(

Title, FirstName, LastName, University, UniversityCourse, DegreeType

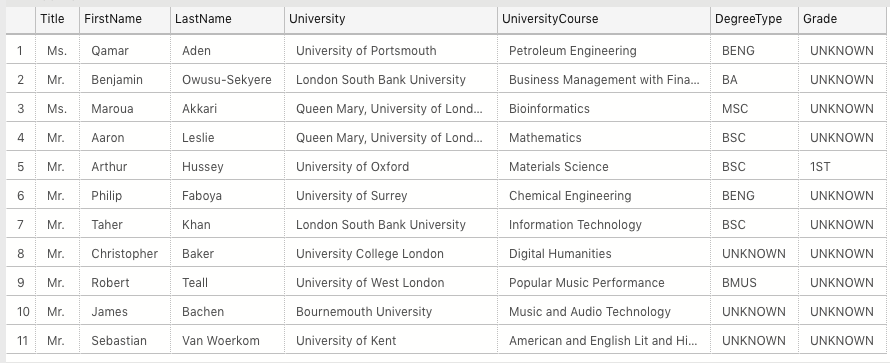
)

VALUES

(

'Mr.', 'James', 'Bachen', 'Bournemouth University', 'Music and Audio Technology', 'BMUS'

);



**Exercise 3**

**3.1**

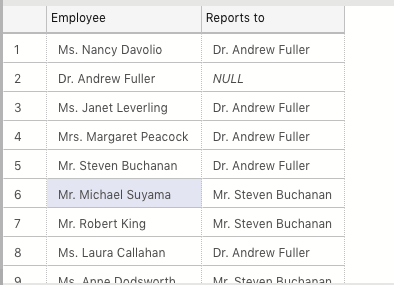
SELECT e1.TitleOfCourtesy + ' ' + e1.FirstName + ' ' + e1.LastName AS 'Employee' ,

e2.TitleOfCourtesy + ' ' + e2.FirstName + ' ' + e2.LastName AS 'Reports to'

FROM Employees e1

LEFT OUTER JOIN Employees e2

ON e1.ReportsTo = e2.EmployeeID



**3.2**

SELECT s.CompanyName, SUM(od.UnitPrice \* od.Quantity) AS 'Total Sales'

FROM [Suppliers] s

INNER JOIN Products p

ON s.SupplierID = p.SupplierID

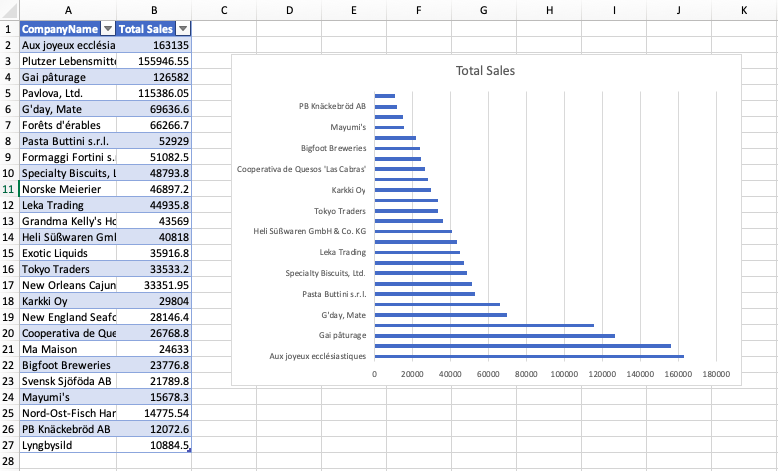
INNER JOIN [Order Details] od

ON p.ProductID = od.ProductID

GROUP BY s.CompanyName

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

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

****

**3.3**

SELECT TOP 10 o.CustomerID, SUM(od.UnitPrice \* od.Quantity) AS 'Total Value of Orders Shipped'

FROM [Orders] o

INNER JOIN [Order Details] od

ON o.OrderID = od.OrderID

GROUP BY CustomerID

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

****

**3.4**

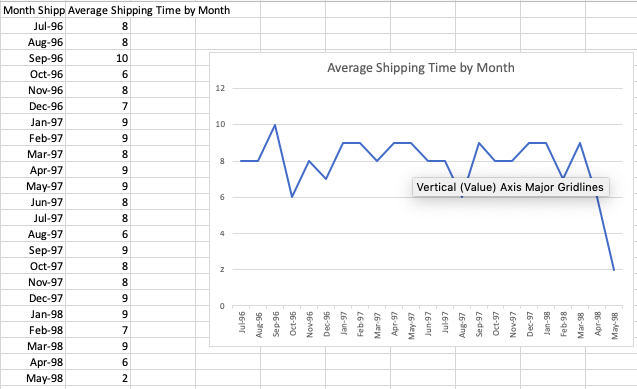
SELECT MONTH(OrderDate) AS 'Month Shipped', YEAR(OrderDate) AS 'Year Shipped', AVG(DATEDIFF(day, OrderDate, ShippedDate)) AS 'Average Days Taken to Ship'

FROM Orders

WHERE DATEDIFF(day, OrderDate, ShippedDate) IS NOT NULL

GROUP BY MONTH(OrderDate), YEAR(OrderDate)

ORDER BY 'Year Shipped', 'Month Shipped' ASC

****