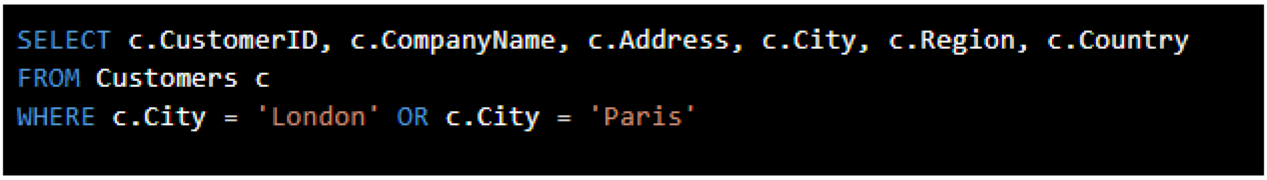
SQL Practical Exercise

Exercise 1 – Northwind Queries (40 marks: 5 for each question)

1.1 Write a query that lists all Customers in either Paris or London. Include Customer ID, Company Name and all address fields.

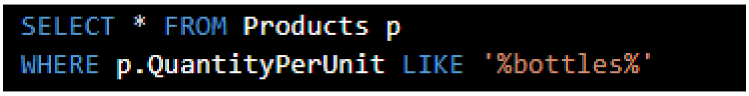
ANSWER



```
SELECT c.CustomerID, c.CompanyName, c.Address, c.City, c.Region, c.Country
FROM Customers c
WHERE c.City = 'London' OR c.City = 'Paris'
```

1.2 List all products stored in bottles.

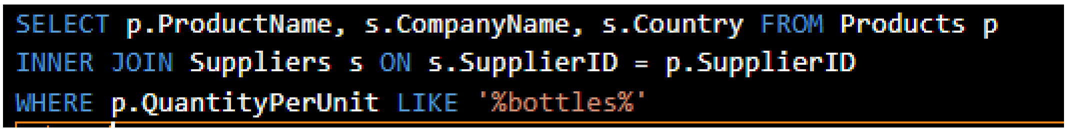
ANSWER



```
SELECT * FROM Products p
WHERE p.QuantityPerUnit LIKE '%bottles%'
```

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

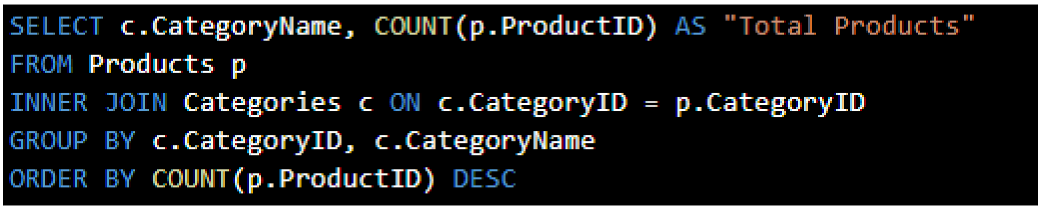
ANSWER



```
SELECT p.ProductName, s.CompanyName, s.Country FROM Products p
INNER JOIN Suppliers s ON s.SupplierID = p.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottles%'
```

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.

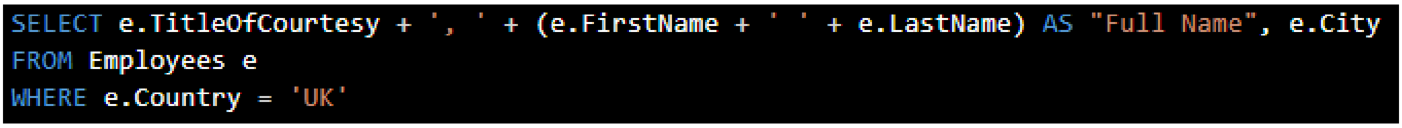
ANSWER



```
SELECT c.CategoryName, COUNT(p.ProductID) AS "Total Products"
FROM Products p
INNER JOIN Categories c ON c.CategoryID = p.CategoryID
GROUP BY c.CategoryID, c.CategoryName
ORDER BY COUNT(p.ProductID) DESC
```

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.

ANSWER

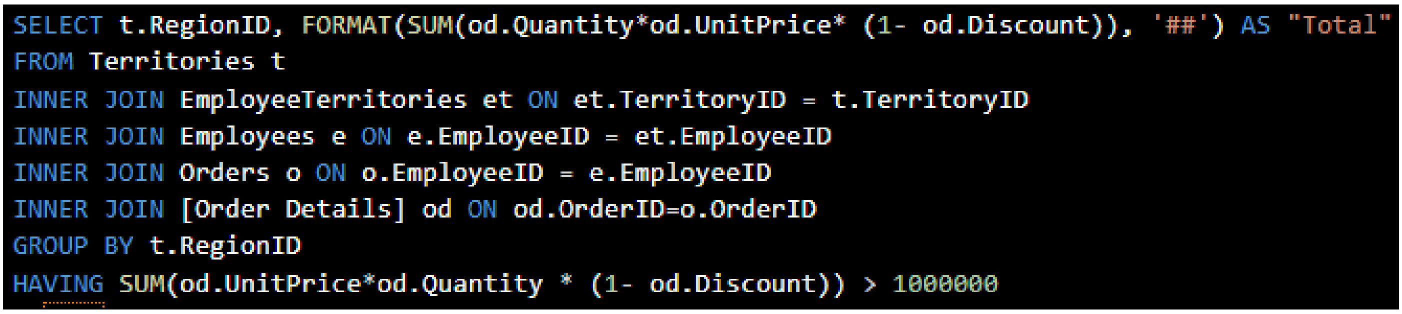


```
SELECT e.TitleOfCourtesy + ', ' + (e.FirstName + ' ' + e.LastName) AS "Full Name", e.City
FROM Employees e
WHERE e.Country = 'UK'
```

SQL Practical Exercise

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.

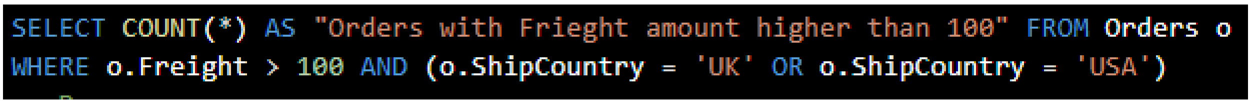
ANSWER



```
SELECT t.RegionID, FORMAT(SUM(od.Quantity*od.UnitPrice* (1- od.Discount)), '##') AS "Total"
FROM Territories t
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 t.RegionID
HAVING SUM(od.UnitPrice*od.Quantity * (1- od.Discount)) > 1000000
```

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

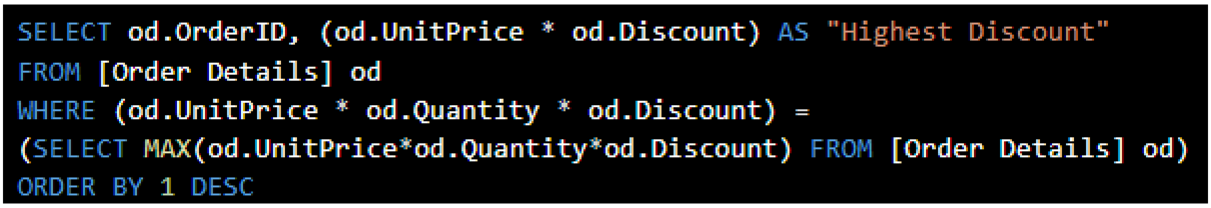
ANSWER



```
SELECT COUNT(*) AS "Orders with Frieght amount higher than 100" FROM Orders o
WHERE o.Freight > 100 AND (o.ShipCountry = 'UK' OR o.ShipCountry = 'USA')
```

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.

ANSWER



```
SELECT od.OrderID, (od.UnitPrice * od.Discount) AS "Highest Discount"
FROM [Order Details] od
WHERE (od.UnitPrice * od.Quantity * od.Discount) =
(SELECT MAX(od.UnitPrice*od.Quantity*od.Discount) FROM [Order Details] od)
ORDER BY 1 DESC
```

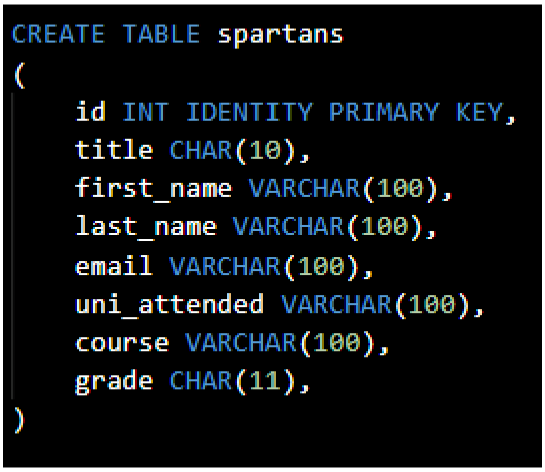
SQL Practical Exercise

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.

ANSWER

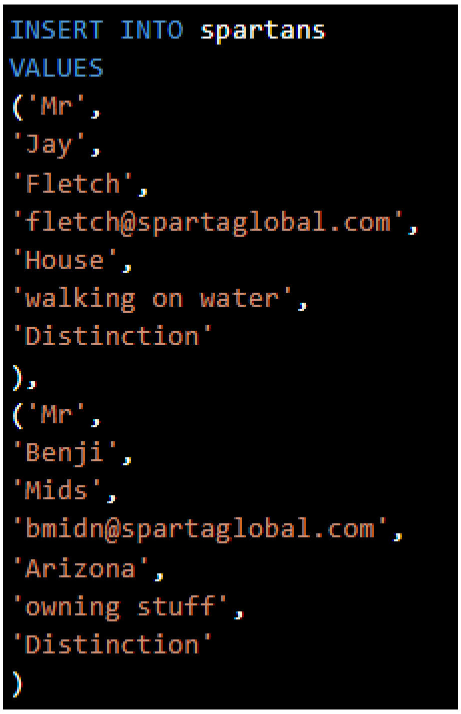


```
CREATE TABLE spartans
(
    id INT IDENTITY PRIMARY KEY,
    title CHAR(10),
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    email VARCHAR(100),
    uni_attended VARCHAR(100),
    course VARCHAR(100),
    grade CHAR(11),
)
```

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.

ANSWER



```
INSERT INTO spartans
VALUES
('Mr',
'Jay',
'Fletch',
'fletch@spartaglobal.com',
'House',
'walking on water',
'Distinction'
),
('Mr',
'Benji',
'Mids',
'bmidn@spartaglobal.com',
'Arizona',
'owning stuff',
'Distinction'
)
```

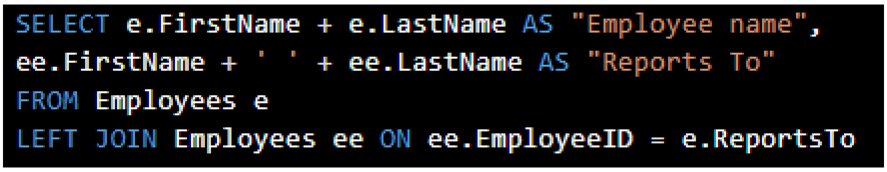
SQL Practical Exercise

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. (5 Marks)

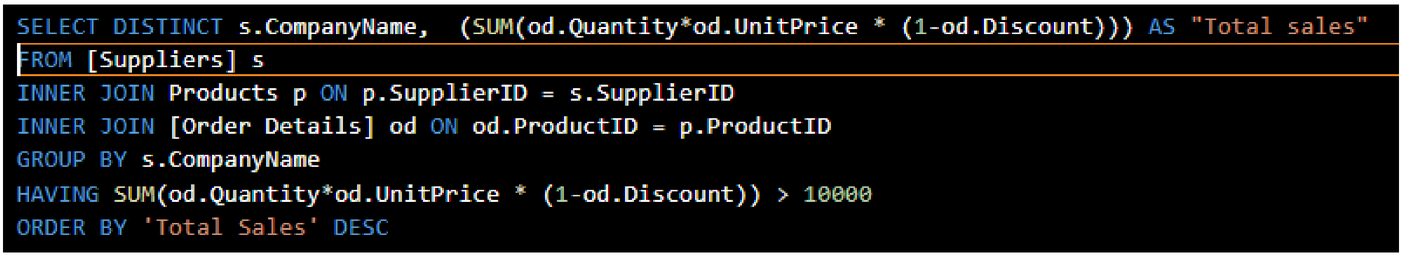
ANSWER



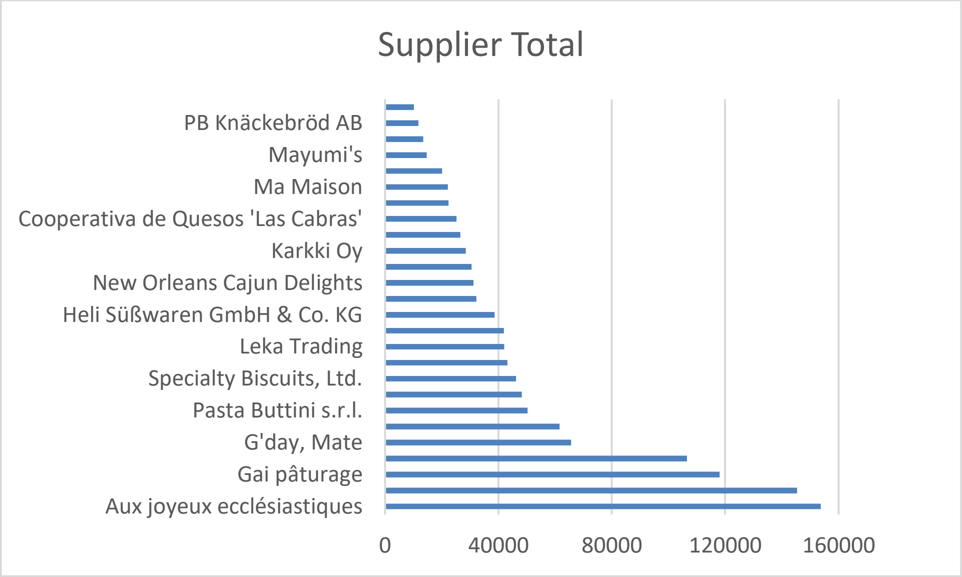
```
SELECT e.FirstName + e.LastName AS "Employee name",
ee.FirstName + ' ' + ee.LastName AS "Reports To"
FROM Employees e
LEFT JOIN Employees ee ON ee.EmployeeID = e.ReportsTo
```

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)

ANSWER



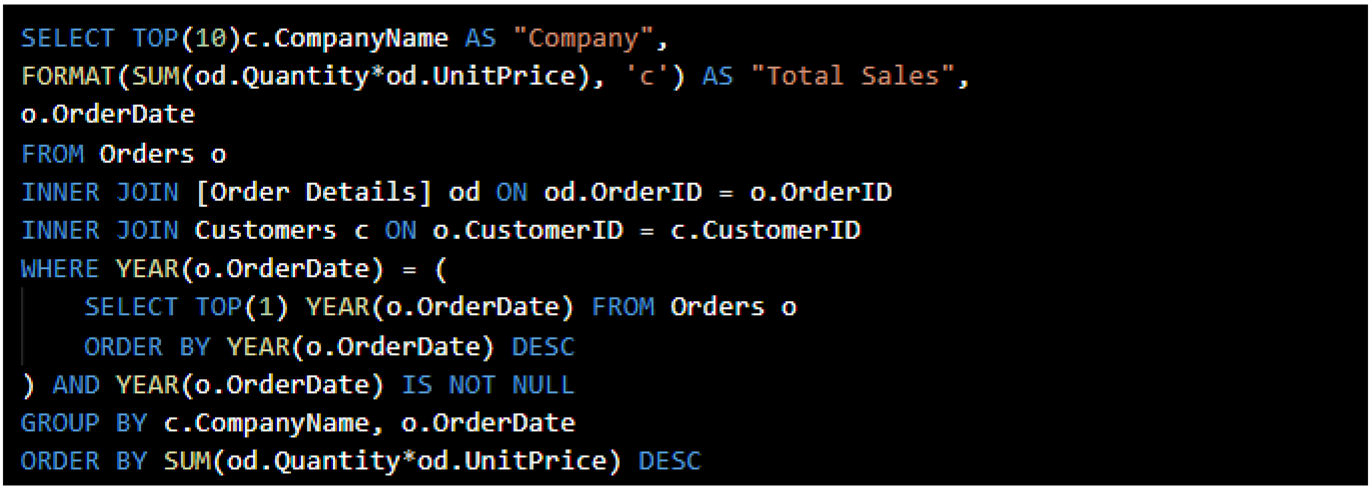
```
SELECT DISTINCT s.CompanyName,  (SUM(od.Quantity*od.UnitPrice * (1-od.Discount))) AS "Total sales"
FROM [Suppliers] s
INNER JOIN Products p ON p.SupplierID = s.SupplierID
INNER JOIN [Order Details] od ON od.ProductID = p.ProductID
GROUP BY s.CompanyName
HAVING SUM(od.Quantity*od.UnitPrice * (1-od.Discount)) > 10000
ORDER BY 'Total Sales' DESC
```



SQL Practical Exercise

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)

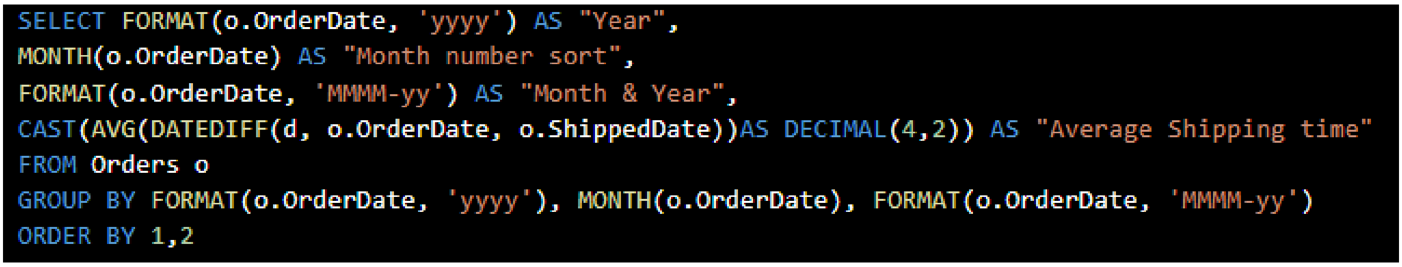
ANSWER



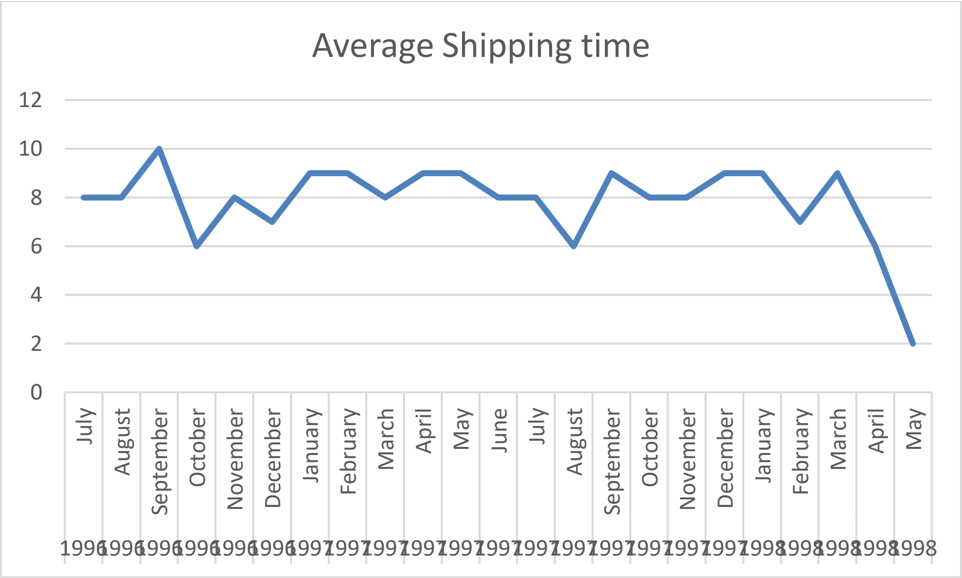
```
SELECT TOP(10)c.CompanyName AS "Company",
FORMAT(SUM(od.Quantity*od.UnitPrice), 'c') AS "Total Sales",
o.OrderDate
FROM Orders o
INNER JOIN [Order Details] od ON od.OrderID = o.OrderID
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE YEAR(o.OrderDate) = (
    SELECT TOP(1) YEAR(o.OrderDate) FROM Orders o
    ORDER BY YEAR(o.OrderDate) DESC
) AND YEAR(o.OrderDate) IS NOT NULL
GROUP BY c.CompanyName, o.OrderDate
ORDER BY SUM(od.Quantity*od.UnitPrice) 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)

ANSWER



```
SELECT FORMAT(o.OrderDate, 'yyyy') AS "Year",
MONTH(o.OrderDate) AS "Month number sort",
FORMAT(o.OrderDate, 'MMMM-yy') AS "Month & Year",
CAST(AVG(DATEDIFF(d, o.OrderDate, o.ShippedDate))AS DECIMAL(4,2)) AS "Average Shipping time"
FROM Orders o
GROUP BY FORMAT(o.OrderDate, 'yyyy'), MONTH(o.OrderDate), FORMAT(o.OrderDate, 'MMMM-yy')
ORDER BY 1,2
```



SQL Practical Exercise

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