

# SQL mini-project

## Introduction



This document contains the code for an SQL project I created, based on the Northwind database to test my knowledge of SQL commands.

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

```
SELECT c.CustomerID
, c.CompanyName
, CONCAT(c.Address, c.City, c.PostalCode, c.Country) AS "Full Address"
FROM Customers c
WHERE c.City IN ('Paris', 'London');
```

### 1.2 List all products stored in bottles.

```
Select p.ProductName
, p.QuantityPerUnit
FROM Products p
WHERE QuantityPerUnit
LIKE '%bottle%';
```

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

```
Select p.ProductName
, p.QuantityPerUnit
, s.CompanyName
, s.Country
FROM Products p
JOIN Suppliers s ON s.supplierID = p.SupplierID
WHERE QuantityPerUnit
LIKE ('%bottle%');
```

### 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
COUNT(p.UnitsInStock) AS "Amount of stock"
, c.CategoryName
FROM products p
JOIN Categories c ON c.CategoryID = p.CategoryID
```

```
GROUP BY c.CategoryName
ORDER BY COUNT (P.UnitsInStock) 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.**

```
SELECT
CONCAT (e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "UK Employees"
, e.City AS "City of Residence"
FROM Employees e
WHERE e.Country
IN ('UK');
```

**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 ROUND(SUM((od.UnitPrice * od.Quantity) * (1.00-Discout)),2) AS "Total Sales", t.RegionID
FROM Territories t
JOIN EmployeeTerritories et ON t.TerritoryID = et.TerritoryID
JOIN Employees e ON et.EmployeeID = e.EmployeeID
JOIN Orders o ON e.EmployeeID = o.EmployeeID
JOIN [Order Details] od ON o.OrderID = od.OrderID
GROUP BY t.RegionID
HAVING SUM(od.UnitPrice * od.Quantity) > 1000000
ORDER BY "Total Sales" DESC;
```

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

```
SELECT COUNT(o.freight) AS "Freight"
FROM Orders o
WHERE o.ShipCountry
IN ('UK', 'USA')
AND Freight > 100;
```

**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 OrderID AS 'Order ID',
FORMAT((UnitPrice * Quantity) * Discount,'C') AS 'Discount Amount'
FROM [Order Details]
ORDER BY [Discount Amount] DESC;
```

**2.1 Create an SQL table which contains the details of students from different Universities**

```

CREATE TABLE [StudentTable] (
    [StudentID] INTEGER NOT NULL IDENTITY(1, 1),
    [Title] VARCHAR(255) NULL,
    [FirstName] VARCHAR(255) NULL,
    [Surname] VARCHAR(255) NULL,
    [University] VARCHAR(255) NULL,
    [Course] VARCHAR(255) NULL,
    [Mark] VARCHAR(255) NULL,
    PRIMARY KEY ([SpartansID])
);

INSERT INTO StudentTable ([Title],[FirstName],[Surname],[University],[Course],[Mark])
VALUES
('Mr.','Humza','Malak','University of Kent','Computer Science','2:2'),
('Mr.','Ibrahim','Bocus','University of Leicester','Mechanical Engineering','2:1'),
('Mr.','Bari','Allali','Lancaster University','Business Economics','2:1'),
('Mr.','Mehdi','El-Chamaa','University of Nottingham','Philosophy and Economics','2:2'),
('Miss.','Anais','Tang','Edinburgh University','Modern Languages','2:1'),
('Mr.','Agbo','Lamina','University of Warwick','Politics and International Studies','2:1'),
('Mr.','Marcus','Tse','University of Hertfordshire','Aerospace Engineering ','2:1'),
('Mr.','Sohaib','Sohail','Brunel University London','Communications and Media Studies ','2:1'),
('Miss.','Ugne','Okmanaitė','Aston University','International Business & Management','2:1'),
('Mr.','Daniel','Teegan','University of Brighton','Product Design','2:2'),
('Mr.','Andrew','Osbourne','University of Portsmouth','Biomedical Science','2:1');

```

### 3.1 List all Employees from the Employees table and who they report to. No Excel required.

```

SELECT
CONCAT (ee.FirstName, ' ', ee.LastName) AS "Worker",
CONCAT (e.FirstName, ' ', e.LastName) AS "Manager"
FROM Employees e, Employees ee
WHERE e.EmployeeID = ee.ReportsTo
ORDER BY "Manager"

```

### 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.

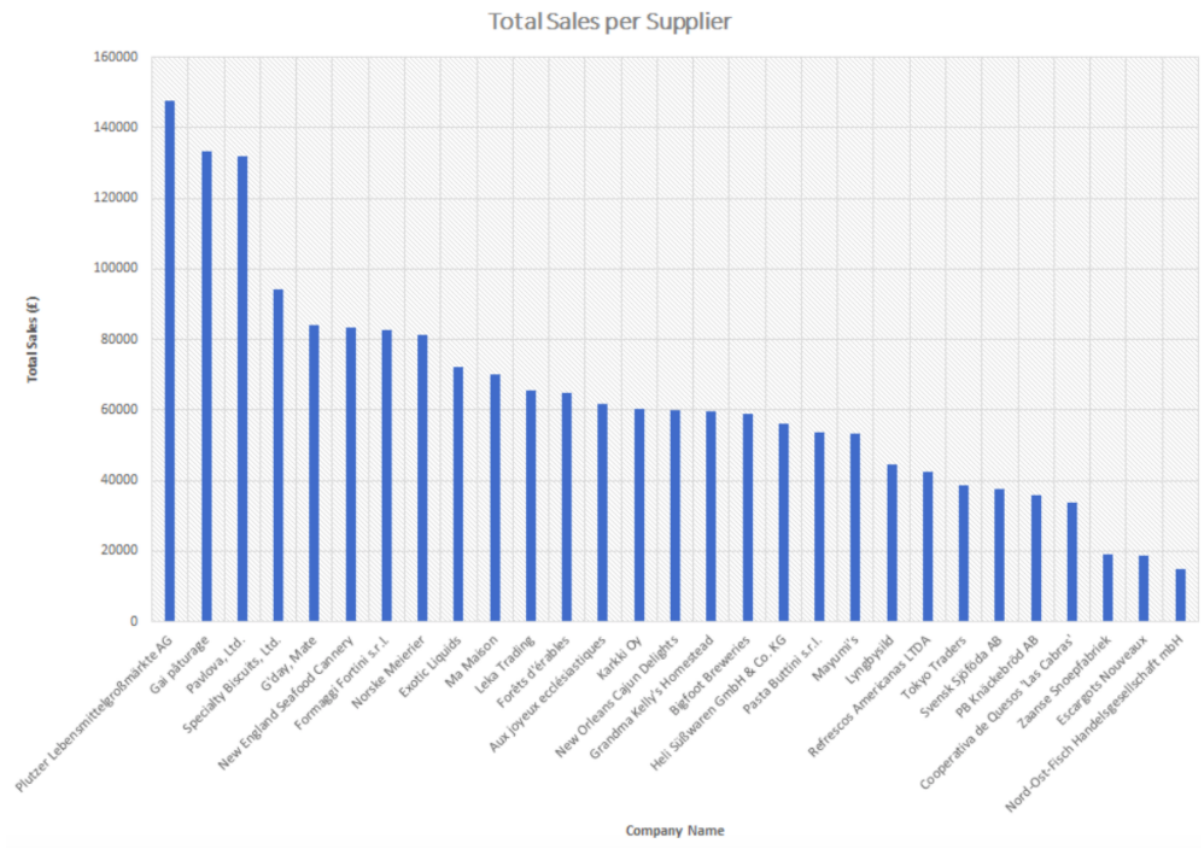
```

SELECT
    s.CompanyName AS "Supplier"
, ROUND(SUM((od.UnitPrice * od.Quantity) * (1.00-Discout)),2) AS "Total Sales"
FROM [Order Details] od
JOIN Products p ON od.ProductID = p.ProductID
JOIN Suppliers s ON p.SupplierID = s.SupplierID
GROUP BY s.CompanyName
HAVING ROUND(SUM((od.UnitPrice * od.Quantity) * (1.00-Discout)),2) > 10000
ORDER BY 'Total Sales'

```



Bar chart of the above query



### 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.

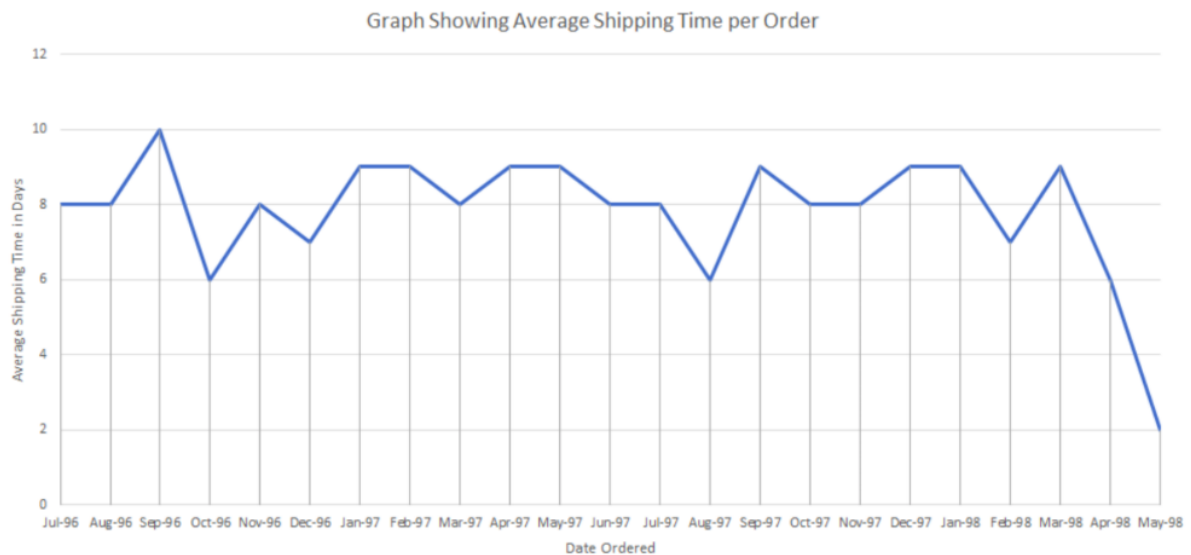
```
SELECT TOP 10 c.CompanyName
, ROUND(SUM((od.UnitPrice * od.Quantity) * (1.00-Discount)),2) AS "Total Sales"
FROM [Order Details]od
JOIN Orders o ON od.OrderID = o.OrderID
JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY o.ShippedDate, c.CompanyName
HAVING YEAR(MAX(o.ShippedDate)) = YEAR(o.ShippedDate)
ORDER BY "Total Sales" DESC
```

### 3.4 Plot the Average Ship Time by month for all data in the Orders Table using a line chart as below.

```
SELECT
FORMAT(o.ShippedDate, 'MMM-yy') AS "Shipping Month",
AVG(DATEDIFF(DAY, o.OrderDate, o.ShippedDate)) AS "Average Ship Time"
FROM Orders o
WHERE o.ShippedDate IS NOT NULL
GROUP BY
YEAR(o.ShippedDate),
MONTH(o.ShippedDate),
FORMAT(o.ShippedDate, 'MMM-yy')
ORDER BY
YEAR(o.ShippedDate),
MONTH(o.ShippedDate);
```



## Bar chart of the above query



### In conclusion:

This was a presentation of my SQL project. After having spent only a few days learning this language, I really appreciate how powerful it can be as a tool to manipulate and retrieve data from databases. I have learned how to present information with precision, how to use aggregate functions and operators to create complex queries as well as represent my results on external charts such as the ones displayed on this project.