

SQL

STRUCTURED
QUERY
LANGUAGE

SQL-IN-JEC
QUICK LINKS
QUERIES
FUNCTIONS
CONSTRAINTS
STRUCTURED QUER
VIEWS
SQL SERVER
INDEXES
MYSQL
BASICS

Project Presentation - Week 2 -

Andrei Pavel
Trainee Consultant

SQL Project - week 2

Exercise 1 - Northwind Queries

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

Query

```
SELECT c.CustomerID,  
       c.CompanyName AS "Company Name",  
       c.Address  
FROM Customers c  
WHERE c.City IN ('Paris', 'London')
```

1.2 List all products stored in bottles.

Query

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

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

Query

```
SELECT p.ProductID AS "Product ID",  
       p.ProductName AS "Product Name",  
       s.CompanyName AS "Company Name",  
       s.Country AS "Country"  
FROM Products p  
JOIN Suppliers s ON p.SupplierID = s.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.

Query

```
SELECT c.CategoryName AS "Category",  
       COUNT(*) AS "Number of products in Category"  
FROM Products p  
JOIN Categories c ON p.CategoryID = c.CategoryID  
GROUP BY c.CategoryName  
ORDER BY COUNT(*) 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.

Query

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

Query

```
SELECT ROUND(SUM((od.UnitPrice-od.UnitPrice*od.Discount)*od.Quantity),2) AS "Sales Total",
       r.RegionDescription AS "Region"
FROM Region r
JOIN Territories t ON r.RegionID = t.RegionID
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 r.RegionID, r.RegionDescription
HAVING SUM((od.UnitPrice-od.UnitPrice*od.Discount)*od.Quantity) > 1000000
ORDER BY "Sales Total" DESC
```

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

Query

```
SELECT COUNT(*) AS "Number of Orders With Freight > 100.00",
       o.ShipCountry AS "Ship Country"
FROM Orders o
WHERE o.Freight > 100.00 AND o.ShipCountry IN ('USA', 'UK')
GROUP BY o.ShipCountry
```

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

Query

```
SELECT od.OrderID AS "Order Number",
       ROUND((od.UnitPrice*od.Discount*od.Quantity),2) AS "Highest Discount Amount"
FROM [Order Details] od
WHERE od.UnitPrice*od.Discount*od.Quantity =
      (SELECT MAX(orr.UnitPrice*orr.Discount*orr.Quantity) FROM [Order Details] orr)
ORDER BY "Highest Discount Amount" DESC
```

Exercise 2 - Create Spartans Table

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 sep and include University attended, course taken and mark achieved.

Add any other columns you feel would be appropriate.

IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.

Query

```
IF OBJECT_ID('dbo.Spartans', 'u') IS NOT NULL
DROP TABLE dbo.Spartans

CREATE TABLE Spartans(
    spartan_id INT IDENTITY PRIMARY KEY,
    title VARCHAR(30),
    first_name VARCHAR(30),
    last_name VARCHAR(30),
    university VARCHAR(30),
    course_taken VARCHAR(30),
    final_mark CHAR(2)
)
```

2.2 Write SQL statements to add the details of the Spartans in your course to the table you have created.

Query

```
INSERT INTO Spartans
VALUES
('Junior Consultant', 'Ahmed Abdul', 'Rahman', 'UCO', 'Software Engineering', 'A'),
('Junior Consultant', 'Alex', 'Ng', 'UCO', 'Software Engineering', 'B'),
('Junior Consultant', 'Asakar', 'Hussain', 'UCB', 'Marketing', 'A'),
('Junior Consultant', 'Ben', 'Middlehurst', 'UCB', 'Law', 'B'),
('Junior Consultant', 'Benjamin', 'Balls', 'UCB', 'Business', 'C'),
('Junior Consultant', 'Daniel', 'Alldritt', 'UCO', 'Marketing', 'A'),
('Junior Consultant', 'Gregory', 'Spratt', 'UCO', 'Law', 'C'),
('Junior Consultant', 'Ismail', 'Kadir', 'UAC', 'Software Engineering', 'C'),
('Junior Consultant', 'James', 'Fletcher', 'UCO', 'Software Engineering', 'B'),
('Junior Consultant', 'Josh', 'Weeden', 'UAC', 'Computer Science', 'A'),
('Junior Consultant', 'Jamie', 'Hammond', 'UCO', 'Computer Science', 'B'),
('Junior Consultant', 'Nathan', 'Johnston', 'UCO', 'Software Engineering', 'A'),
('Junior Consultant', 'Rashawn', 'Henry', 'UCO', 'Law', 'C'),
('Junior Consultant', 'Sidhant', 'Khosla', 'UFB', 'International Affairs', 'B'),
('Junior Consultant', 'Timin', 'Rickaby', 'UFB', 'International Affairs', 'A'),
('Junior Consultant', 'Yusuf', 'Uddin', 'UCO', 'Software Engineering', 'B'),
('Junior Consultant', 'Andrei', 'Pavel', 'RAU', 'Computer Science', 'A')
```

Exercise 3 - Northwind Data Analysis linked to Excel

3.1 List all Employees from the Employees table and who they report to.

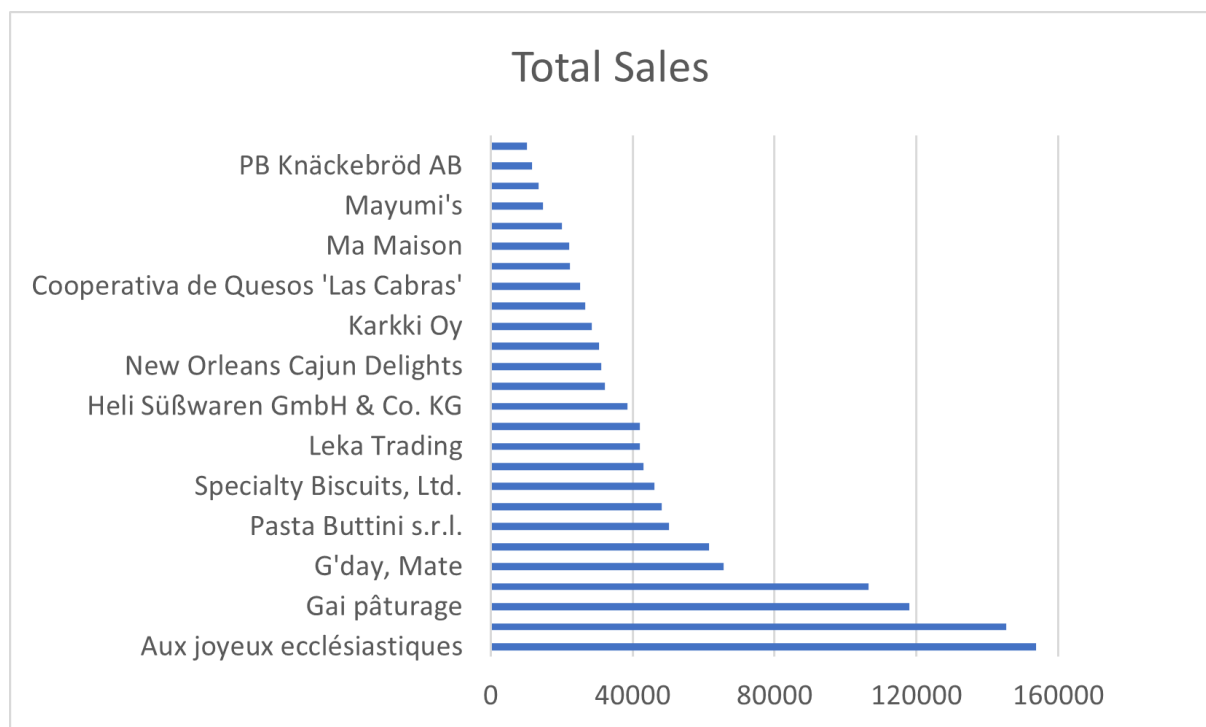
Query

```
SELECT CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "Employee",
       CONCAT(rep.TitleOfCourtesy, ' ', rep.FirstName, ' ', rep.LastName) AS "Reports to"
FROM Employees e
LEFT JOIN Employees rep ON e.ReportsTo = rep.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:

Query

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



3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped

Query

```
SELECT TOP 10 c.CompanyName AS "Company Name",
       ROUND(SUM(od.Quantity*(od.UnitPrice-od.UnitPrice*od.Discount)),2) AS "Total Value Of Orders Shipped"
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN [Order Details] od ON o.OrderID = od.OrderID
WHERE YEAR(o.OrderDate)=(SELECT MAX(YEAR(oo.OrderDate)) FROM Orders oo)
GROUP BY c.CompanyName
ORDER BY "Total value of orders shipped" DESC
```

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

Query

```
SELECT AVG(CAST(DATEDIFF(DAY,o.OrderDate,o.ShippedDate) AS DECIMAL(10,2))) AS "Average Ship Time",  
       FORMAT(o.OrderDate,'MMM-yyyy') AS "Date"  
FROM Orders o  
GROUP BY FORMAT(o.OrderDate,'MMM-yyyy'),  
         DATEPART(YEAR, o.OrderDate),  
         DATEPART(MONTH, o.OrderDate)  
ORDER BY DATEPART(YEAR, o.OrderDate),  
         DATEPART(MONTH, o.OrderDate)
```

