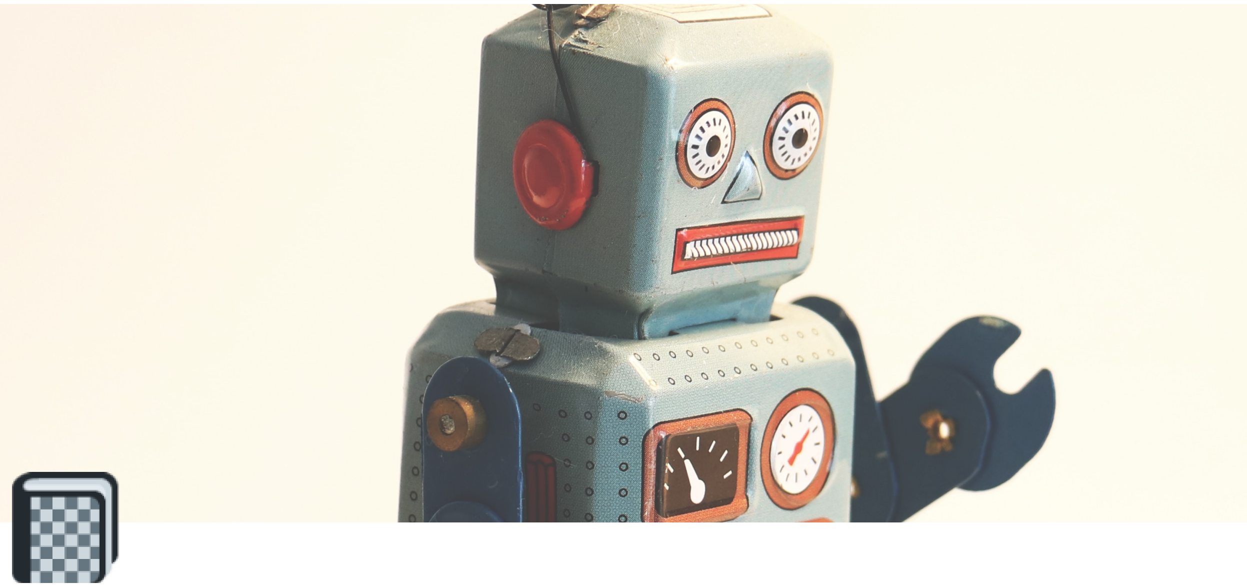


SQL Project

🔢This document contains the code for an SQL project I created during the second
week of my Sparta Global training.

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, c.Address, c.City, c.PostalCode,
c.Country
FROM Customers c
    WHERE c.City IN ('paris','london');
```

Answer

```
SELECT CustomerID AS “Customer ID”, CompanyName AS “Customer Name”,
Address + ‘, ‘ + City + ‘, ‘ + PostalCode + ‘, ‘ + Country AS “Address”
FROM Customers
  WHERE City IN ('Paris', 'London');
```

1.2

```
List all products stored in bottles
SELECT p.ProductName, p.QuantityPerUnit, s.CompanyName AS "Supplier Name", s.country
FROM Products p
    INNER JOIN Suppliers s ON  s.SupplierID = p.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottle%';
```

Answer

```
SELECT ProductName, QuantityPerUnit
FROM Products
  WHERE QuantityPerUnit LIKE '%Bottle%';
```

1.3

```
-- Repeat question above but add in the Supplier Name and Country.
SELECT p.ProductName, p.QuantityPerUnit, s.CompanyName AS "Supplier Name", s.country
FROM Products p
    INNER JOIN Suppliers s ON  s.SupplierID = p.SupplierID
WHERE p.QuantityPerUnit LIKE '%bottle%';
```

```
SELECT ProductName, QuantityPerUnit, CompanyName, Country
  FROM Products p
        INNER JOIN Suppliers s ON p.SupplierID = s.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 c.categoryName, COUNT(p.ProductID) AS "No. of Products"
FROM Categories c
    INNER JOIN Products p ON p.categoryID = c.categoryID
-- Group to break up the categories
GROUP BY c.CategoryID, c.CategoryName
--Order DESC
ORDER BY COUNT(c.CategoryID) DESC;
```

Answer

```
SELECT  c.CategoryName "Category Name", COUNT(*) as "No of Products"
FROM Products p
      INNER 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.

SELECT CONCAT(e.TitleOfCourtesy,' ', e.Firstname,' ', e.LastName)
AS "Employee Name", e.city
FROM Employees e
WHERE e.Country IN ('UK')
```

Answer

```
SELECT TitleOfCourtesy + ' ' + FirstName + ' ' + LastName As Employee, City
  FROM Employees
  WHERE 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.
SELECT r.RegionID, r.RegionDescription,
FORMAT(SUM(od.UnitPrice*od.Quantity*(1-od.discount)), 'C')
AS 'Sales Totals'
FROM Region r
    JOIN Territories t ON r.RegionID=t.RegionID
    JOIN EmployeeTerritories et ON t.TerritoryID = et.TerritoryID
    JOIN Orders o ON et.EmployeeID=o.EmployeeID
    JOIN [Order Details] od ON o.OrderID=od.OrderID
GROUP BY r.RegionID, R.RegionDescription
-- total sales have to be more than 1,000,000
HAVING sum(od.UnitPrice*od.Quantity) > 1000000
ORDER BY 'Sales Totals' DESC;
```

Answer

```
SELECT r.RegionID, r.RegionDescription AS Region,
FORMAT(SUM((UnitPrice * Quantity) * (1-Discount)),'C')
AS "Sales Total by Region"
    FROM Orders AS o
      INNER JOIN [Order Details] AS od ON od.OrderID = o.OrderID
      INNER JOIN EmployeeTerritories AS et ON o.EmployeeID = et.EmployeeID
      INNER JOIN Territories AS t ON et.TerritoryID = t.TerritoryID
      INNER JOIN Region AS r ON t.RegionID = r.RegionID
    GROUP BY r.RegionDescription, r.RegionID
    HAVING SUM((UnitPrice * Quantity) * (1-Discount)) > 1000000
    ORDER BY "Sales Total by Region" DESC;
```

1.7

```
-- 1.7 Count how many Orders have a Freight amount greater than 100.00
-- and either USA or UK as Ship Country.
-- counting the quantity of freight with count greater than 100 and
-- only in usa or uk
SELECT COUNT(o.Freight) AS 'No. of orders > 100 from US or UK'
FROM Orders o
WHERE o. Freight > 100.00 AND o.ShipCountry IN ('USA', 'UK');
```

Answer

```
SELECT COUNT(*) AS 'No of Orders >100 from US or UK'
  FROM Orders
  WHERE Freight>100 AND ShipCountry IN ('USA','UK');
```

1.8

```
-- 1.8  Write an SQL Statement to identify the Order Number of the Order
-- with the highest amount of discount applied to that order.
SELECT TOP 1
    od.OrderID, FORMAT(SUM((od.Discount)*od.UnitPrice*od.Quantity),'C')
AS 'Highest Discount'
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY 'Highest Discount' DESC;
```

Answer

```
SELECT Orderid AS "Order ID",
    FORMAT((UnitPrice * Quantity) * Discount,'C') AS "Total Discount"
    FROM [Order Details]
    WHERE ((UnitPrice * Quantity) * Discount) =
    (
SELECT MAX(
       (UnitPrice * Quantity) * Discount) -- AS 'Discount Amount'
    FROM [Order Details]
    )
    ;
```

2.1

```
-- 2.1 Write the correct SQL statement to create the following table:
drop table [Sparta Table]
CREATE TABLE [Sparta Table]
(

    SpartanID INT NOT NULL IDENTITY(1,1),
    Title varchar(12) NOT NULL,
    [First Name] varchar(40) NOT NULL,
    [Last Name] varchar(40) NOT NULL,
    University varchar(50) DEFAULT NULL,
    Course varchar(50) DEFAULT NULL,
    Marks varchar(4) DEFAULT NULL,
    Grade CHAR(3) DEFAULT NULL,
    PRIMARY KEY (SpartanID)
);
```

Answer

```
DROP TABLE Spartans;

CREATE TABLE Spartans
(
  EmployeeID int IDENTITY(1,1) NOT NULL PRIMARY KEY,
  Title varchar(20) NOT NULL,
  FirstName varchar(50) NOT NULL,
  LastName varchar(50) NOT NULL,
  SpartaCourse varchar(50) NOT NULL,
  University varchar(50),
  UniCourse varchar(50),
  UniMark varchar(30),
  StartDate date
  );
```

2.2

```
-- 2.2 Write SQL statements to add the details of the Spartans in your
-- course to the table you have created
drop table [Sparta table]
-- instert,values. Process of updating DML because of data structure
-- is manipulated.
-- title....grade columns shows the values that will be added to the
-- table
INSERT INTO [Sparta Table]
    (Title, [First Name], [Last Name], University, Course, Marks, Grade)
VALUES
    ('Mr.', 'Man-Wai', 'Tse', 'University of Hertfordshire',
    'Aerospace Engineering', 66, '2:1'),
    ('Miss.', 'Georgina', 'Barlett', 'Newcastle University',
    'Archaeology ', 63, '2:1'),
    ('Mr.', 'Humza', 'Malak', 'University of Kent', 'Computer Science',
     58, '2:2'),
    ('Mr.', 'Bari', 'Allali', 'University of Lancaster', 'Business
    Econmomics', 64, '2:1'),
    ('Mr.', 'Mehdi', 'Shamaa ', 'University of Nottingham',
    'Philosphy and Economics', 57, '2:2'),
    ('Mr.', 'Anais', 'Tang', 'Edinburgh University', 'Modern Languages',
     69, '2:1'),
    ('Mr.', 'Saheed', 'Lamina', 'University of Warwick', 'Politics and
    International Studies', 68, '2:1'),
    ('Mr.', 'Sohaib', 'Sohail', 'Brunel University', 'Communications
    and Media Studies ', 67, '2:1'),
    ('Mr.', 'Ugne', 'Okmanaite ', 'Aston University', 'International
    Business Management', 65, '2:1'),
    ('Mr.', 'John', 'Byrne', 'University of Greenwich', 'Computing with
     Games development' ,'65','2:1'),
    ('Miss', 'Daniel', 'Teegan', 'University of Brighton', 'Product
    Design ', 59, '2:1'),
    ('Mr.', 'Max', 'Palmer', 'University of Birmingham', 'Ancient
    History', 63, '2:1');
```

Answer - Example

```
INSERT INTO Spartans
(Title, FirstName, LastName, SpartaCourse,University,UniCourse,UniMark,StartDate)
VALUES
(‘Miss’, ‘Giverny’, ‘Wilson-Martin’,’BA-Test’,’Staffordshire’,’Games Design’,
’2:2’,’2014-02-10’);
```

```
INSERT INTO Spartans
(Title, FirstName, LastName, SpartaCourse,University,UniCourse,UniMark,StartDate)
VALUES
(‘Mr’, ‘Rob’, ‘Whitehouse’,’SDET’,’Sheffield’,’Physics’,’2:1’,’2015-01-01’);
```

3.1

```
-- 3.1 List all Employees from the Employees table and who they report
-- to. No Excel required.
SELECT CONCAT(e.FirstName,' ', e.LastName) AS 'Employee Name',
CONCAT(b.TitleOfCourtesy, ' ', b.FirstName,' ', b.LastName) AS "Reports
To"
FROM Employees e
LEFT JOIN Employees b ON e.ReportsTo=b.EmployeeID
ORDER BY "Reports To","Employee Name"
```

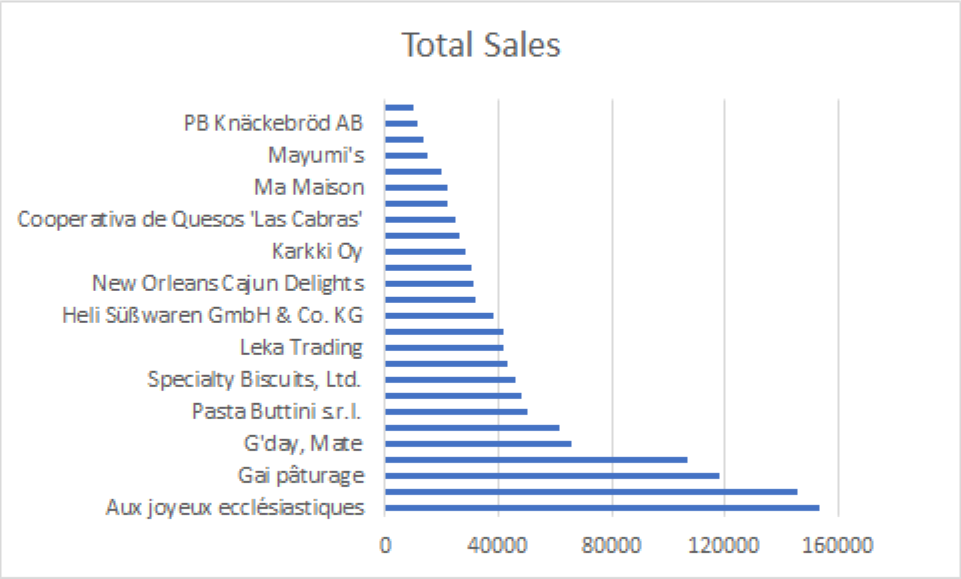
Answer

```
SELECT e.FirstName + ' ' + e.LastName AS "Employee Name",
    b.FirstName + ' ' + b.LastName AS "Reports To"
  FROM Employees e
  LEFT JOIN Employees b ON e.ReportsTo=b.EmployeeID
  ORDER BY "Reports To","Employee Name";
```

3.2

```
-- 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:

SELECT s.SupplierID, s.CompanyName, ROUND(SUM(od.UnitPrice*od.Quantity*
(1-od.Discount)),0) 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.SupplierID, s.CompanyName
HAVING (SUM(od.UnitPrice*od.Quantity*(1-od.Discount))) > 10000
order by 'Total Sales' desc
```



Answer

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

3.3

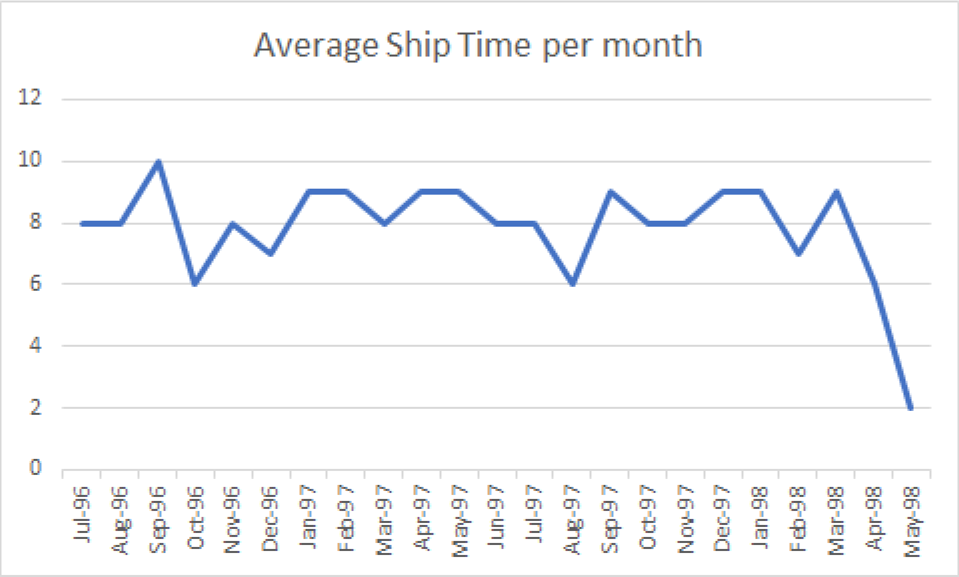
```
-- 3.3 List the Top 10 Customers Year To Date for the latest year in
-- the Orders file. Based on total value of orders shipped. No Excel
-- required.
SELECT TOP 10 c.CustomerID, c.CompanyName,
FORMAT(SUM(od.UnitPrice * od.Quantity * (1-od.Discount)),'C')
AS "Total Value"
FROM Customers c
    INNER JOIN Orders o ON o.CustomerID=c.CustomerID
    INNER JOIN [Order Details] od ON od.OrderID=o.OrderID
  WHERE YEAR(OrderDate)=(SELECT MAX(YEAR(OrderDate)) From Orders)
AND o.ShippedDate IS NOT NULL
    GROUP BY c.CustomerID, c.CompanyName
    ORDER BY SUM(od.UnitPrice * od.Quantity * (1-od.Discount)) DESC
```

Answer

```
SELECT TOP 10 c.CustomerID AS "Customer ID", c.CompanyName As "Company",
FORMAT(SUM(UnitPrice * Quantity * (1-Discount)),'C')
AS "YTD Sales"
FROM Customers c
    INNER JOIN Orders o ON o.CustomerID=c.CustomerID
    INNER JOIN [Order Details] od ON od.OrderID=o.OrderID
  WHERE YEAR(OrderDate)=(SELECT MAX(YEAR(OrderDate)) From Orders)
  --WHERE YEAR(OrderDate)=1998 --WHERE YEAR(OrderDate)='1998'
AND o.ShippedDate IS NOT NULL
  GROUP BY c.CustomerID, c.CompanyName
  ORDER BY SUM(UnitPrice * Quantity * (1-Discount)) DESC;
```

3.4

```
-- 3.4 Plot the Average Ship Time by month for all data in the Orders
-- Table using a line chart as below.
SELECT AVG(sq1.[Time taken to ship orders]) AS "Average Ship Time per
month"
, sq1.[Month and year of delivery] "Month and year of delivery"
FROM (SELECT (DATEDIFF(d, o.OrderDate, o.ShippedDate)) AS "Time taken
to ship orders",
Format(o.orderdate, 'yy-MM') AS "Month and year of delivery"
FROM Orders o) sq1
GROUP BY "Month and year of delivery"
ORDER BY "Month and year of delivery";
```



Answer

```
SELECT MONTH(OrderDate) Month, YEAR(OrderDate) Year,
AVG(CAST(DATEDIFF(d, OrderDate, ShippedDate) As DECIMAL(10,2))) As ShipTime
  FROM orders
  WHERE ShippedDate IS NOT NULL
  GROUP BY YEAR(OrderDate),MONTH(OrderDate)
  ORDER BY Year ASC, Month ASC
```