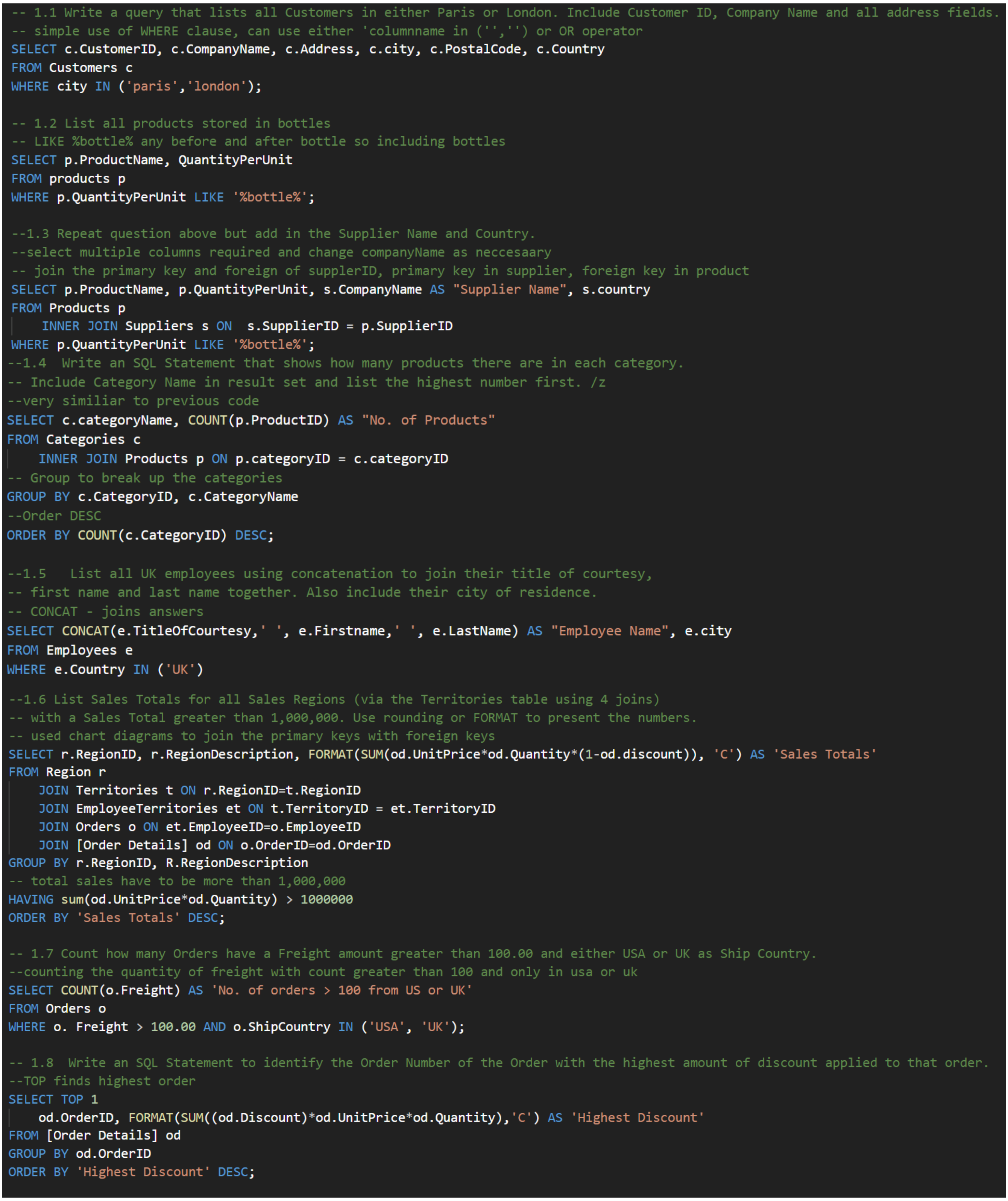
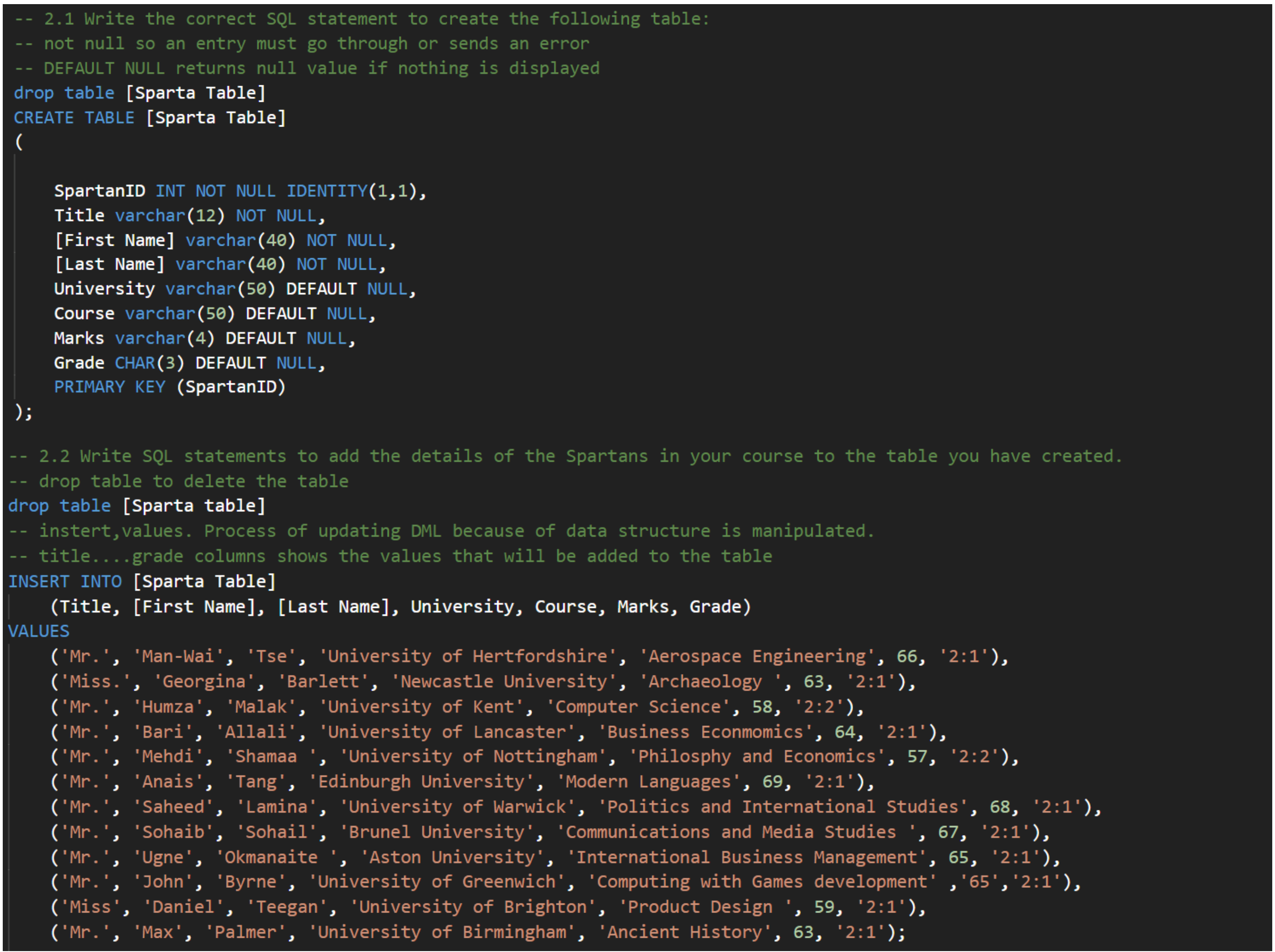
Monday 29th June 2020

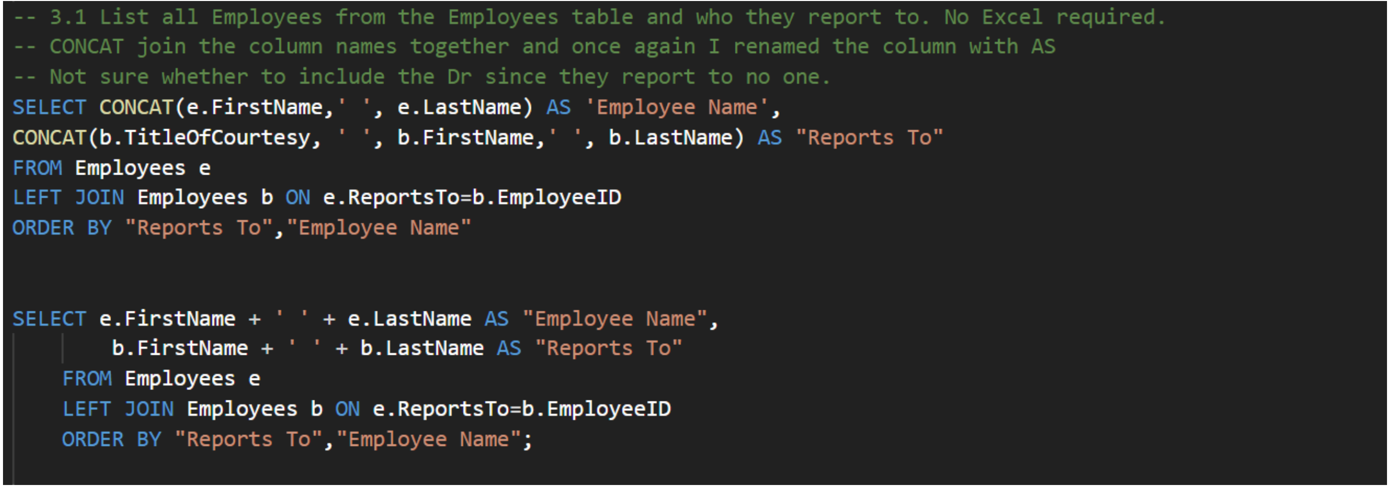
SQL Project



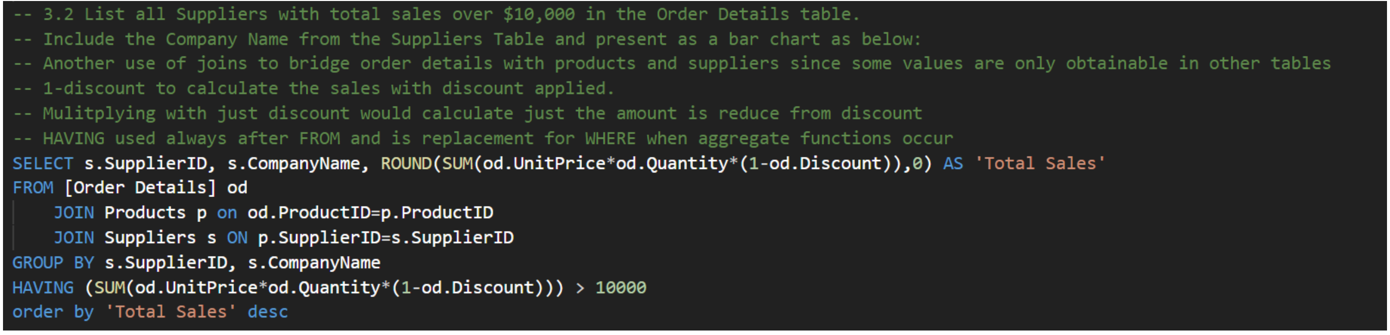
```sql
-- 1.1 Write a query that lists all Customers in either Paris or London. Include Customer ID, Company Name and all address fields.
-- simple use of WHERE clause, can use either 'columnname in ('','') or OR operator
SELECT c.CustomerID, c.CompanyName, c.Address, c.city, c.PostalCode, c.Country
FROM Customers c
WHERE city IN ('paris','london');

-- 1.2 List all products stored in bottles
-- LIKE %bottle% any before and after bottle so including bottles
SELECT p.ProductName, QuantityPerUnit
FROM products p
WHERE p.QuantityPerUnit LIKE '%bottle%';

--1.3 Repeat question above but add in the Supplier Name and Country.
--select multiple columns required and change companyName as neccesaary
-- join the primary key and foreign of supplerID, primary key in supplier, foreign key in product
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%';
--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. /z
--very similiar to previous code
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;

--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.
-- CONCAT - joins answers
SELECT CONCAT(e.TitleOfCourtesy,' ', e.Firstname,' ', e.LastName) AS "Employee Name", e.city
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.
-- used chart diagrams to join the primary keys with foreign keys
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;

-- 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');

-- 1.8  Write an SQL Statement to identify the Order Number of the Order with the highest amount of discount applied to that order.
--TOP finds highest 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;
```



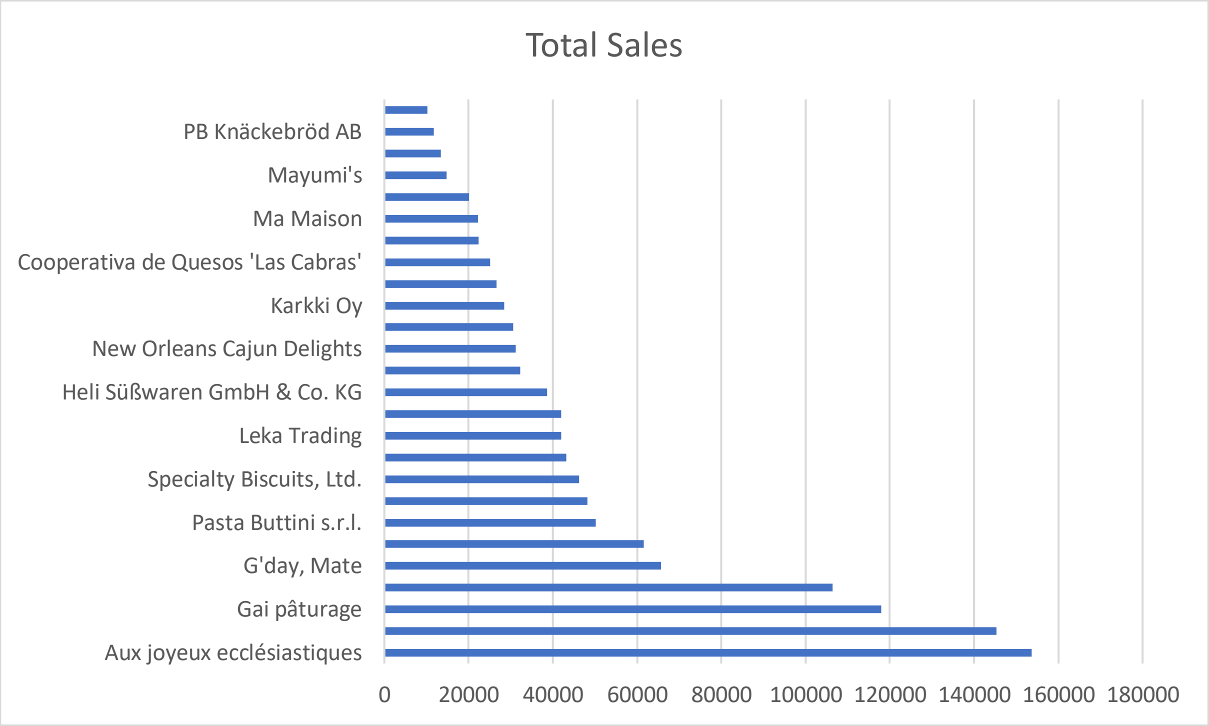
```
-- 2.1 Write the correct SQL statement to create the following table:
-- not null so an entry must go through or sends an error
-- DEFAULT NULL returns null value if nothing is displayed
drop table [Sparta Table]
CREATE TABLE [Sparta Table]
(

    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)
);

-- 2.2 Write SQL statements to add the details of the Spartans in your course to the table you have created.
-- drop table to delete the table
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');
```

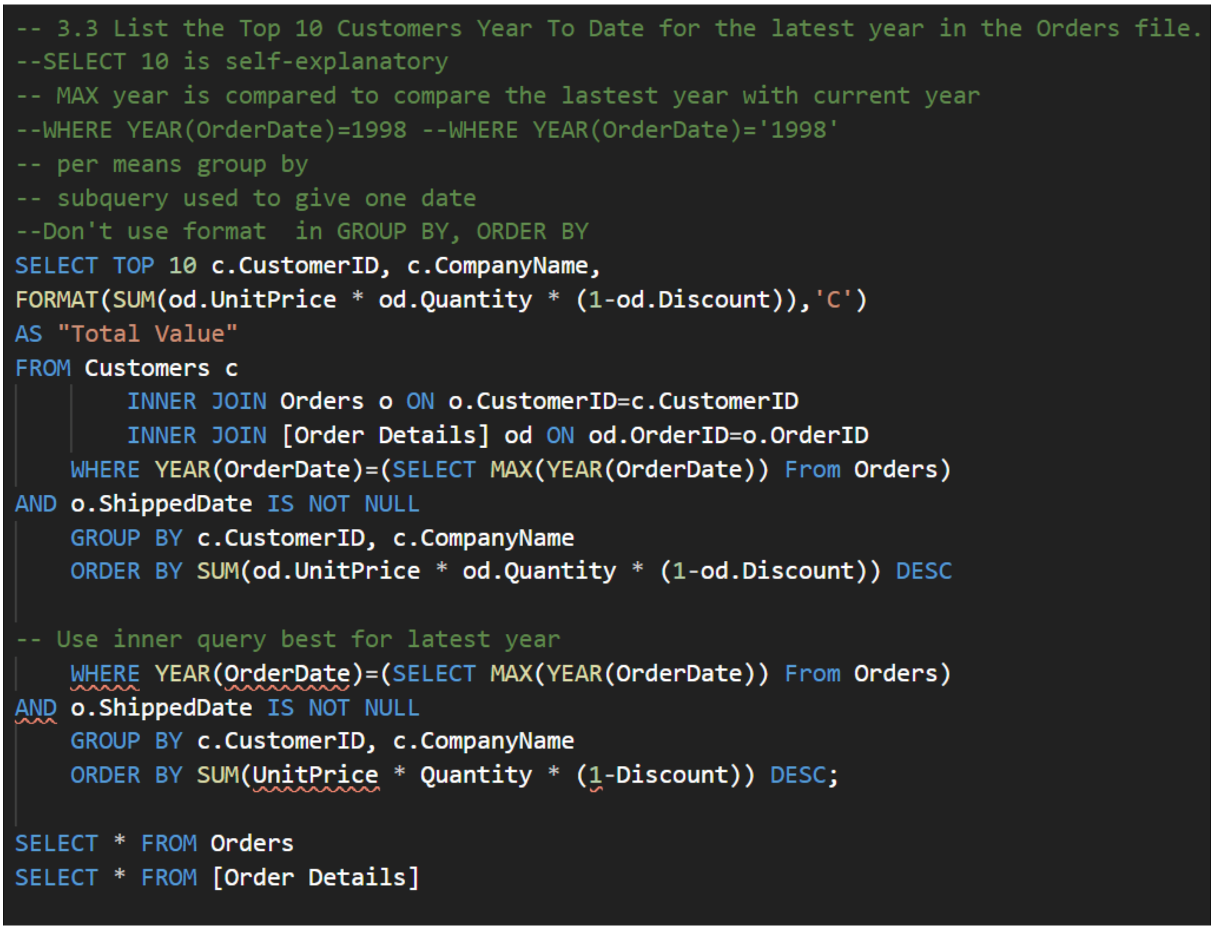


```
-- 3.1 List all Employees from the Employees table and who they report to. No Excel required.
-- CONCAT join the column names together and once again I renamed the column with AS
-- Not sure whether to include the Dr since they report to no one.
SELECT CONCAT(e.FirstName,' ', e.LastName) AS 'Employee 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"


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";

```

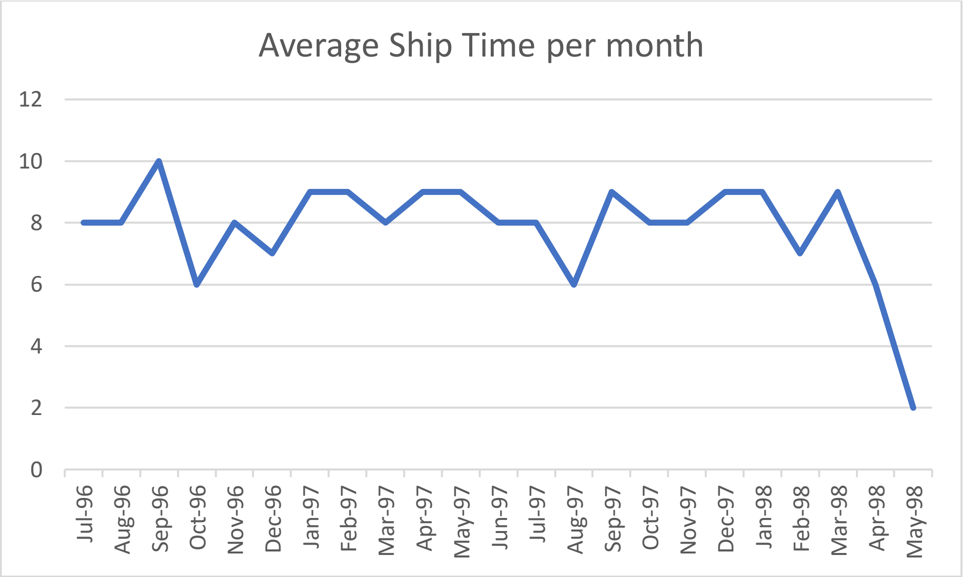


```
-- 3.2 List all Suppliers with total sales over $10,000 in the Order Details table.
-- Include the Company Name from the Suppliers Table and present as a bar chart as below:
-- Another use of joins to bridge order details with products and suppliers since some values are only obtainable in other tables
-- 1-discount to calculate the sales with discount applied.
-- Mulitplying with just discount would calculate just the amount is reduce from discount
-- HAVING used always after FROM and is replacement for WHERE when aggregate functions occur
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
```





```
-- 3.3 List the Top 10 Customers Year To Date for the latest year in the Orders file.
--SELECT 10 is self-explanatory
-- MAX year is compared to compare the lastest year with current year
--WHERE YEAR(OrderDate)=1998 --WHERE YEAR(OrderDate)='1998'
-- per means group by
-- subquery used to give one date
--Don't use format  in GROUP BY, ORDER BY
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

-- Use inner query best for latest year
    WHERE YEAR(OrderDate)=(SELECT MAX(YEAR(OrderDate)) From Orders)
AND o.ShippedDate IS NOT NULL
    GROUP BY c.CustomerID, c.CompanyName
    ORDER BY SUM(UnitPrice * Quantity * (1-Discount)) DESC;

SELECT * FROM Orders
SELECT * FROM [Order Details]
```


Average Ship Time per month
12
10
8
6
4
2
0
Jul-96
Aug-96
Sep-96
Oct-96
Nov-96
Dec-96
Jan-97
Feb-97
Mar-97
Apr-97
May-97
Jun-97
Jul-97
Aug-97
Sep-97
Oct-97
Nov-97
Dec-97
Jan-98
Feb-98
Mar-98
Apr-98
May-98