Name: Wahdel Woodhouse
Sparta Global

EXERCISE 1

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

```
7   SELECT c.CustomerID, c.CompanyName, c.Address, c.PostalCode, c.City, c.Country
8   FROM Customers c
9   WHERE c.City = 'Paris' OR c.City = 'London';
```

1.2 List all products stored in bottles.

```
13  SELECT p.ProductID, p.ProductName, p.QuantityPerUnit
14  FROM Products p
15  WHERE CHARINDEX('bottle', p.QuantityPerUnit) != 0;
```

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

```
19  SELECT p.ProductID, p.ProductName, p.QuantityPerUnit, s.CompanyName, s.Country
20  FROM Products p
21  INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
22  WHERE CHARINDEX('bottle', p.QuantityPerUnit) != 0;
```

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.

```
26  SELECT c.CategoryName, COUNT(p.ProductID) AS "Number of Products in Category"
27  FROM Categories c
28  INNER JOIN Products p On c.CategoryID = p.CategoryID
29  GROUP BY c.CategoryName
30  ORDER BY "Number of Products in Category" 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.

```
34  SELECT CONCAT(e.TitleOfCourtesy, ' ', e.FirstName, ' ', e.LastName) AS "Full Name with Title", e.City
35  FROM Employees e
36  WHERE e.Country = 'UK';
```

1.6 List Sales Totals for all Sales Regions with a Sales Total greater than 1,000,000. Use rounding or FORMAT to present the numbers.

```
40  SELECT t.RegionID,
41      FORMAT(SUM((1-od.Discount)*od.Quantity*od.UnitPrice), 'C', 'en-gb') AS "Sales Total Per Region"
42  FROM Territories t
43  INNER JOIN EmployeeTerritories et ON t.TerritoryID = et.TerritoryID
44  INNER JOIN Employees e ON et.EmployeeID = e.EmployeeID
45  INNER JOIN Orders o ON e.EmployeeID = o.EmployeeID
46  INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
47  GROUP BY t.RegionID
48  HAVING SUM((1-od.Discount)*od.Quantity*od.UnitPrice) > 1000000;
```

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

```
52  SELECT COUNT(*) AS "Has more than 100 Freights From UK/USA"
53  FROM Orders o
54  WHERE o.Freight > 100 AND (o.ShipCountry='UK' OR o.ShipCountry='USA');
```

Name: Wahdel Woodhouse
Sparta Global

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.

```sql
SELECT TOP 1 od.OrderID,
    FORMAT(SUM(od.UnitPrice*od.Quantity*od.Discount), 'C', 'en-gb') AS "Discount Value"
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY SUM(od.UnitPrice*od.Quantity*od.Discount) DESC;
```

EXERCISE 2

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.

```sql
CREATE TABLE spartans_table
(
    spartan_id INT IDENTITY(1, 1) PRIMARY KEY,
    separate_title VARCHAR(4) NOT NULL,
    first_name VARCHAR(10) NOT NULL,
    last_name VARCHAR(20) NOT NULL,
    university_attended VARCHAR(30) NOT NULL,
    course_taken VARCHAR(25) NOT NULL,
    marks_achieved INT NOT NULL
);
```

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

```sql
INSERT INTO spartans_table
(separate_title, first_name, last_name, university_attended, course_taken, marks_achieved)
VALUES
('Mr.', 'Wahdel', 'Woodhouse', 'Royal Holloway', 'Cyber Security', 70),
('Mr.', 'Aaron', 'Banjoko', 'Brunel', 'Artificial Intelligence', 70),
('Mr.', 'Bradley', 'Williams', 'University of Birmingham', 'Software Engineering', 70),
('Mr.', 'Kurtis', 'Hanson', 'Coventry', 'Software Design', 70),
('Mr.', 'Joel', 'Fright', 'University College London', 'Robotics', 70),
('Mr.', 'Dominic', 'Cogan-Tucker', 'Oxford', 'Physics', 70),
('Mr.', 'Malik', 'Shams', 'Cambridge', 'Electrical Engineering', 70);
```

EXERCISE 3

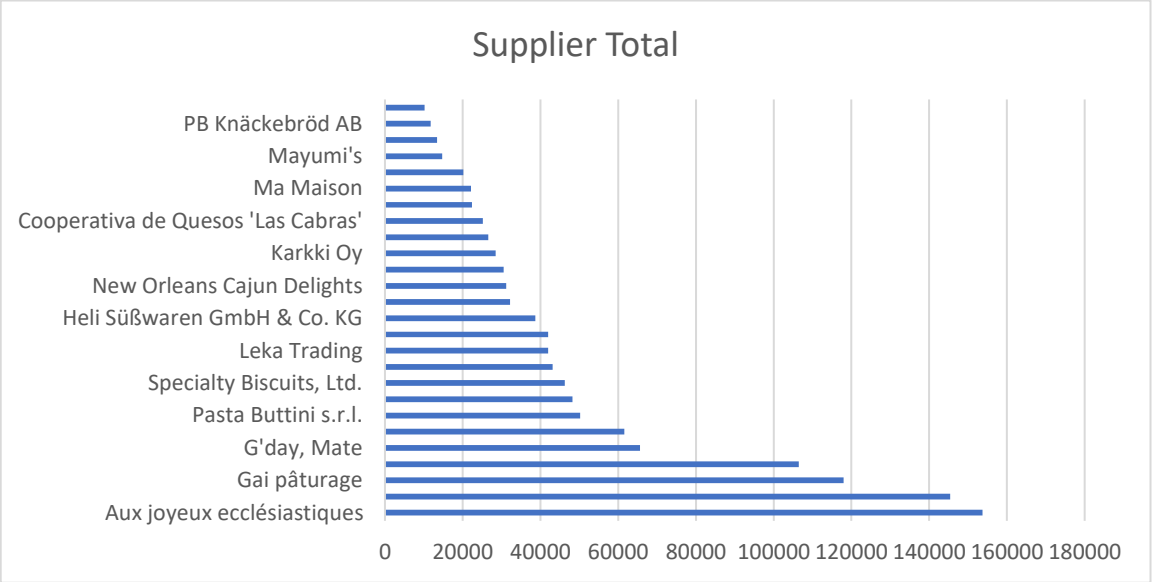
3.1 List all Employees from the Employees table and who they report to. Include Employee names and ReportTo names.

```sql
SELECT CONCAT(e1.FirstName, ' ', e1.LastName) AS "Employee Name",
    e2.FirstName + ' ' + e2.LastName AS "Reports to"
FROM Employees e1
LEFT JOIN Employees e2 ON e1.ReportsTo = e2.EmployeeID;
```

Name: Wahdel Woodhouse
Sparta Global

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.CompanyName, SUM((1-od.Discount)*od.Quantity*od.UnitPrice) 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((1-od.Discount)*od.Quantity*od.UnitPrice) > 10000
ORDER BY "Supplier Total" DESC;
```



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

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

Name: Wahdel Woodhouse
Sparta Global

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

```
132    SELECT CONCAT(MONTH(o.OrderDate), '-', YEAR(o.OrderDate)) AS "Month Shipped",
133        AVG(DATEDIFF(dd, o.OrderDate, o.ShippedDate)) AS "Average Ship Time By Month"
134    FROM Orders o
135    GROUP BY YEAR(o.OrderDate), MONTH(o.OrderDate)
136    ORDER BY YEAR(o.OrderDate), MONTH(o.OrderDate);
```