SQL project

This is documentation demonstrates the knowledge gained in the aspects of SQL by completing the set exercise listed in the project. Each exercise will be stated following an SQL query to achieve the goal and will be using the Northwind database.

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.

```
--1.1
SELECT c.CustomerID, c.CompanyName, c.Address, c.City, c.Region, c.PostalCode
FROM Customers c
WHERE c.City = 'Paris' OR c.City ='London'
```

1.2 List all products stored in bottles.

```
--1.2
SELECT * FROM Products
SELECT * FROM Products p WHERE QuantityPerUnit LIKE '%bottle%'
```

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

```
--1.3
SELECT p.*,  s.CompanyName, s.Country
FROM Products p
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.

```
--1.4
SELECT p.CategoryID AS "Category", c.CategoryName, COUNT(p.ProductName) AS "Number of Products"
FROM Products p INNER JOIN Categories c
ON p.CategoryID=c.CategoryID
GROUP BY p.CategoryID, c.CategoryName
ORDER BY "Number of Products" 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.

```
--1.5
SELECT * FROM Employees
SELECT CONCAT(e.TitleOfCourtesy , ' ', e.FirstName, ' ', e.LastName) AS "Name",
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.

```
--1.6 SP_HELP Orders
SELECT  r.RegionDescription AS "Regions",ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) AS "Sales Total"
FROM [Order Details] od
INNER JOIN Orders o ON od.OrderID=o.OrderID
INNER JOIN EmployeeTerritories et ON  o.EmployeeID = et.EmployeeID
INNER JOIN Territories t ON et.TerritoryID = t.TerritoryID
INNER JOIN Region r ON t.RegionID = r.RegionID
GROUP BY r.RegionDescription
HAVING ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) > 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.

```
SELECT COUNT(*) AS "Total orders with Frieght Over 100.00 with Ship Country as USA 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(value) of discount applied to that order.

```
--1.8
SELECT TOP 1 od.OrderID
FROM [Order Details] od
GROUP BY od.OrderID
ORDER BY SUM(od.UnitPrice*od.Quantity) 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 separate columns, and include University attended, course taken, and mark achieved. Add any other columns you feel would be appropriate.

```
--Exercise 2 - Ceate Sparetans Table
USE jDelaCruz_db
DROP TABLE Spartans

CREATE TABLE Spartans(
Spartan_id INT IDENTITY(1,1) PRIMARY KEY,
title VARCHAR(6),
forename VARCHAR(20),
surename VARCHAR(20),
university_name VARCHAR(50),
course_name VARCHAR(50),
mark VARCHAR(5)
)
```

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

```
-- 2.2
INSERT INTO Spartans VALUES ('Mr.','Jian','Dela Cruz','University of Reading','Computer Science','2:2'),
('Mr.','Adrian','Wong','University of Southampton','Computer Science','2:1'),
('Mr.','Alex','Lynch','University of Brighton','Software Engineering','2:2'),
('Mr.','Alexander','Legon','University of Northampton','Electronics Engineering','2:1'),
('Mr.','George','Repole','University of Surrey','Computer Science','2:2'),
('Mr.','Karim','Wohler','University of Warwick','Mathematics','2:1'),
('Mr.','Sotiris','Loizou','University of Birmingham','Computer Science','2:2'),
('Mr.','Thomas','Canfield','University of Loughbrough','Engineering','2:1'),
('Mr.','Thomas','Kirkwood','University of Glosgow','Engineering','2:2'),
('Mr.','Hoa','Chang','University College London ','Chemistry','2:1');
```

Exercise 3 – Northwind Data Analysis linked to Excel

3.1 List all Employees from the Employees table and who they report to. No Excel required. Please mention the Employee Names and the ReportTo names.

```
--3.1
SELECT CONCAT(e.FirstName, ' ', e.LastName) AS "Name",
CONCAT(e2.FirstName, ' ', e2.LastName) AS "Report to"
FROM Employees e
LEFT JOIN Employees e2 ON e2.EmployeeID=e.ReportsTo
ORDER BY "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:

```
--3.2
SELECT s.CompanyName As "Company Name",ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) 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 "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. No Excel required.

```
--3.3
SELECT TOP 10 c.CompanyName, ROUND(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),2) as "Total Spent"
FROM [Order Details] od
INNER JOIN Orders o ON od.OrderID=o.OrderID
INNER JOIN Customers c ON o.CustomerID=c.CustomerID
WHERE YEAR(o.ShippedDate) IN (SELECT YEAR(MAX(o.ShippedDate)) FROM Orders o) AND o.ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY "Total Spent" DESC
```

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

```
--3.4
SELECT * FROM Orders o
SELECT CONCAT(MONTH(o.OrderDate), '/', YEAR(o.OrderDate)) AS "Month",
ROUND(AVG(CAST(DATEDIFF(d,OrderDate,ShippedDate) AS FLOAT)), 2) AS "Average ship time by month"
FROM Orders o
GROUP BY MONTH(o.OrderDate),YEAR(o.OrderDate)
ORDER BY YEAR(o.OrderDate)
```