Practice SQL

Use the "northwind_onlinedemo.txt" database with https://sqliteonline.com/ as indicated in the tutorial 1 and formulate the SQL queries by yourself to answer the following questions. Don't hesitate to Google your SQL question to find help. To obtain the corrections, email your answers to damien.dupre@dcu.ie.

Questions:

1. Retrieve all columns in the Region table.

```sql
SELECT *
FROM region;
```

2. Select the FirstName and LastName columns from the Employees table.

```sql
SELECT FirstName, LastName
FROM employees;
```

3. Select the FirstName and LastName columns from the Employees table. Sort by LastName.

```sql
SELECT FirstName, LastName
FROM employees
ORDER BY LastName;
```

4. Create a report showing Northwind's orders sorted by Freight from most expensive to cheapest. Show OrderID, OrderDate, ShippedDate, CustomerID, and Freight.

```sql
SELECT OrderID, OrderDate, ShippedDate, CustomerID, Freight
FROM orders
ORDER BY Freight DESC;
```

5. Create a report showing the title and the first and last name of all sales representatives.

```sql
SELECT Title, FirstName, LastName
FROM employees
WHERE Title = 'Sales Representative';
```

6. Create a report showing the first and last names of all employees who have a region corresponding to 'WA'.

```sql
SELECT FirstName, LastName
FROM employees
WHERE Region = 'WA';
```

7. Create a report showing the first and last name of all employees whose last names start with a letter in the last half of the alphabet. Sort by LastName in descending order.

```sql
SELECT FirstName, LastName
FROM employees
WHERE LastName < "M"
ORDER BY LastName DESC;
/*note the type of quotation mark doesn't influence the query (simple ' or double ")*/
```

8. Create a report showing the first and last name of all sales representatives who are from Seattle or Redmond.

```sql
SELECT FirstName, LastName, City, Title
FROM employees
WHERE City = 'Seattle' OR City = 'Redmond' AND Title = 'Sales Representative';
```

9. Create a report that shows the company name, contact title, city and country of all customers in Mexico or in any city in Spain except Madrid.

```sql
SELECT CompanyName, ContactTitle, City, Country
FROM customers
WHERE Country = 'Spain' OR Country = 'Mexico' AND NOT City = 'Madrid';
/*or with the symbol not equal <>
but this symbol can change according the different SQL versions*/
SELECT CompanyName, ContactTitle, City, Country
FROM customers
WHERE Country = 'Spain' OR Country = 'Mexico' AND City <> 'Madrid';
```

10. If the cost of freight is greater than or equal to 500.00, it will now be taxed by 10%. Create a report that shows the order id, freight cost, freight cost with this tax for all orders of 500 or more.

```sql
SELECT OrderID, Freight, Freight * 1.1 AS FreightWithTax
FROM orders
WHERE Freight >= 500;
```

11. Find the Total Number of Units Ordered of Product ID 3.

```sql
SELECT ProductID, SUM(Quantity) AS TotalUnitsOrdered
FROM order_details
WHERE ProductID = 3;
```

12. Retrieve the number of employees in each city.

```sql
SELECT City, COUNT(EmployeeID) AS NumberEmployees
FROM employees
GROUP BY City;
```

13. Create a report showing employee orders.

```sql
SELECT FirstName, LastName, OrderID
FROM  orders
LEFT JOIN employees ON orders.EmployeeID = employees.EmployeeID;
/*also works by indicating from which table the variables have been selected
but this precision is necessary only if the same select variables appear in multiple tables*/
SELECT employees.FirstName, employees.LastName, orders.OrderID
FROM  orders
LEFT JOIN employees ON orders.EmployeeID = employees.EmployeeID;
```

14. Create a report showing the Order ID, the name of the company that placed the order, and the first and last name of the associated employee. Only show orders placed after January 1, 1998 that shipped after they were required. Sort by Company Name.

```sql
SELECT OrderID, CompanyName, FirstName, LastName, OrderDate
FROM orders
INNER JOIN customers ON orders.CustomerID = customers.CustomerID
INNER JOIN employees ON orders.EmployeeID = employees.EmployeeID
WHERE RequiredDate < ShippedDate AND OrderDate > '1998-01-01'
ORDER BY CompanyName;
```

15. Create a report that shows the total quantity of products (from the order_details table) ordered. Only

show records for products.

```
/*here ProductID is selected but appears in both table products and order_details
therefore indicating the origin table in select is necessary*/
SELECT products.ProductName, order_details.ProductID, SUM(order_details.Quantity) AS TotalQuantity
FROM order_details
INNER JOIN products ON order_details.ProductID = products.ProductID
GROUP BY products.ProductName;
```