Instructor:

- Mr. Nazeef Ul Haq (Lab)

Learning Objectives:

- Understanding of Group functions, Having clause, Column and Table Alias.

Helping Material:

1. AGGREGATE FUNCTIONS:

Aggregate functions allow you to perform a calculation on a set of values to return a single scalar value. The most common aggregate functions are:

- AVG
- MIN
- SUM
- COUNT
- STDEV
- STDEVP
- VAR
- VARP
- MAX

Note: All the Group function except COUNT ignore null values

2. THE GROUP BY CLAUSE

The SQL GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

For example:

```
SELECT SupplierID, AVG(UnitPrice)
FROM Products
GROUP BY SupplierID
```

3. THE HAVING CLAUSE

The WHERE clause is a row filter, the HAVING clause is a group filter. Only groups for which the HAVING predicate evaluates to TRUE are returned by the HAVING phase to the next logical query processing phase. Groups for which the predicate evaluates to FALSE or UNKNOWN are discarded. Because the HAVING clause is processed after the rows have been grouped, you can refer to aggregate functions in the logical expression.

For example, in the query below, the HAVING clause has the logical expression COUNT(*) > 1, meaning that the HAVING phase filters only groups (Title) with more than one row. The following fragment of query will help to retrieve required data:

```
SELECT Title, COUNT(*)
FROM Employees
```

```
GROUP BY Title
HAVING COUNT(*) >1
```

4. ALIASING TABLES AND COLUMNS

Aliases provide database administrators, as well as other database users, with the ability to reduce the amount of code required for a query, and to make queries simpler to understand. In addition, aliasing can be used as an obfuscation technique to protect the real names of database fields.

4.1. ALIASING COLUMNS

```
SELECT column_name AS alias_name FROM table_name;
```

4.2. ALIASING TABLES

```
SELECT column_name(s) FROM table_name AS alias_name;
```

5. FOOD FOR BRAIN

Here is some extra food for your brain which can help in further assignments:

- YEAR function is used to extract year from a date
- MONTH function is used to extract year from a date
- DAY function is used to extract day from a date.

Lab Tasks:

- Perform all the group function on Northwind Schema.
- Perform all the group function using HAVING clause on Northwind Schema
- Apply aliasing syntax on arbitrary column on Northwind Schema.
- **Bonus:** Find an alternate method to alias tables and columns in SQL Server.

Home Tasks:

All the given tasks should be performed in the context of Northwind Schema.

1. List name of all the products whose price is above average.
2. Write a query to generate report showing date wise orders shipped.
3. List name of all countries from where two or more suppliers belong to.
4. Write a query to generate report showing month wise orders delayed shipped. Your output should look like this

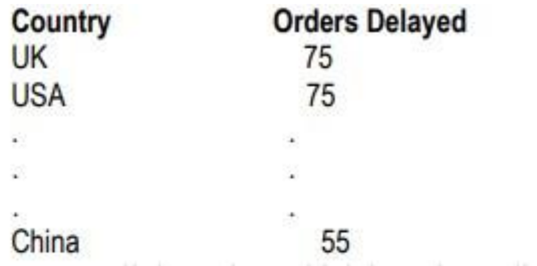
Month Number	**Orders Delayed**
1	75
2	75
.	.
.	.
.	.
12	55

5. Report all the orders which have been discounted. Your result should show the total discount against each order. Output should look like this

OrderID	**Discount**
10250	0.300000011920929
10251	0.100000001490116

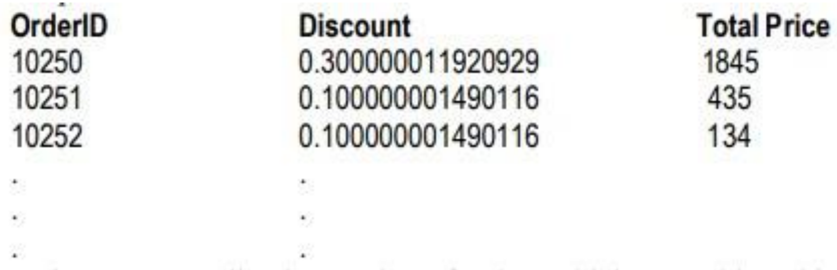
10252 0.100000001490116
. .
. .
. .

6. Write a query to list the number of orders which were shipped in the cities of USA in 1997. Show the number of order against each city.
7. Write a query to generate report showing country wise orders delayed shipped. Your output should look like this:



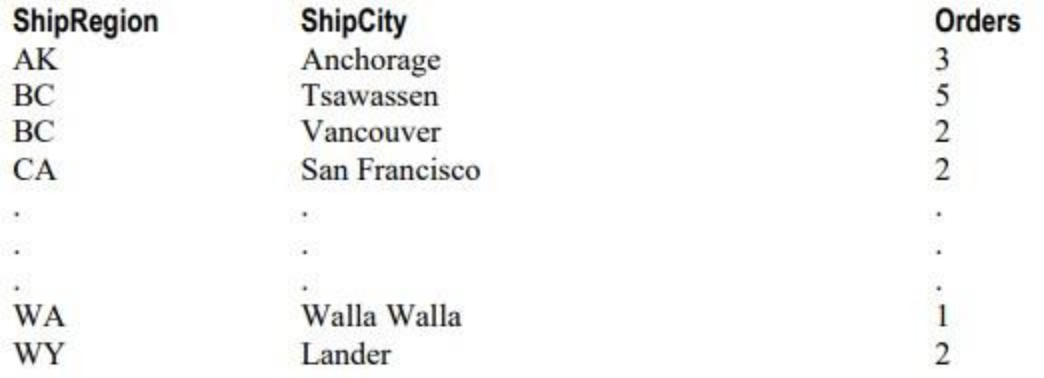
Country	Orders Delayed
UK	75
USA	75
.	.
.	.
.	.
China	55

8. Report all the orders which have been discounted with total price of order. Your result should show the total discount against each order. Output should look like this:



OrderID	Discount	Total Price
10250	0.300000011920929	1845
10251	0.100000001490116	435
10252	0.100000001490116	134
.	.	
.	.	
.	.	

9. Write a query to list the number of orders which were shipped in the cities of each region in 1997. Show the number of order against each city. Your results should look like this:



ShipRegion	ShipCity	Orders
AK	Anchorage	3
BC	Tsawassen	5
BC	Vancouver	2
CA	San Francisco	2
.	.	.
.	.	.
.	.	.
WA	Walla Walla	1
WY	Lander	2

HackerRank Tasks:

1. https://www.hackerrank.com/challenges/average-population/problem?isFullScreen=true
2. https://www.hackerrank.com/challenges/revising-aggregations-the-average-function/problem?isFullScreen=true
3. https://www.hackerrank.com/challenges/revising-aggregations-sum/problem?isFullScreen=true
4. https://www.hackerrank.com/challenges/revising-aggregations-the-count-function/problem?isFullScreen=true
5. https://www.hackerrank.com/challenges/weather-observation-station-6/problem?isFullScreen=true
6. https://www.hackerrank.com/challenges/weather-observation-station-7/problem?isFullScreen=true
7. https://www.hackerrank.com/challenges/weather-observation-station-8/problem?isFullScreen=true
8. https://www.hackerrank.com/challenges/weather-observation-station-9/problem?isFullScreen=true

9. https://www.hackerrank.com/challenges/weather-observation-station-10/problem?isFullScreen=true
10. https://www.hackerrank.com/challenges/weather-observation-station-11/problem?isFullScreen=true
11. https://www.hackerrank.com/challenges/weather-observation-station-12/problem?isFullScreen=true
12. https://www.hackerrank.com/challenges/more-than-75-marks/problem?isFullScreen=true
13. https://www.hackerrank.com/challenges/population-density-difference/problem?isFullScreen=true
14. https://www.hackerrank.com/challenges/the-blunder/problem?isFullScreen=true
15. https://www.hackerrank.com/challenges/earnings-of-employees/problem?isFullScreen=true
16. https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true
17. https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true

What to Submit:

You are required to submit the following files.

- Dbscript.sql
- Pdf report.