Source of data: **AdventureWorks**

Task 1
Data Source: SalesOrderHeader
Define the following queries:
1.1 Specify the years in which the orders were registered in the database.
1.2 Create a list of orders placed in the first year of order registration (ID, Year, Order Amount).
1.3 Create a list of orders placed in May in individual years (year, month, ID, Order amount)

1.1. SQL query + fragment of the result (4 records from ?)
Table 1.1 Fragment of the query results for task 1.1

Lata
2012
2013
…

Rec: 4/4

Solution:

```
SELECT DISTINCT  YEAR([OrderDate]) AS "Years"
FROM [Sales].[SalesOrderHeader]
ORDER BY Years;
```

Years
2011
2012
2013
2014

I had 4 records after running the query as presented above.

1.2. SQL query + fragment of the result (4 records from ?)
Table 1.2 Fragment of the query results for task 1.2

Identifier	Year	Amount
45266	2012	27605.63
45267	2012	3899.68
45268	2012	944.62
45269	2012	2280.14

Rec.: 4/4

Solution:

```
SELECT DISTINCT [SalesOrderID]  AS "Identifier" ,YEAR([OrderDate]) AS "Year",
ROUND([TotalDue]*1,2)  AS "Amount"
FROM [Sales].[SalesOrderHeader]
WHERE  [SalesOrderID] BETWEEN 45266 AND 45269
   ORDER BY Year;
```

Identifier	Year	Amount
45266	2012	27605.63
45267	2012	3899.68

45268	2012	944.62
45269	2012	2280.14

I had 4 records after running the query as presented above.

1.3. SQL query + fragment of the result (4 records from ?)

Table 1.3 Fragment of the query results for task 1.3

Year	Month	Identifier	Amount
2012	5	46685	2410.63
2012	5	46686	865.20
2013	5	50775	2264.25
2013	5	50776	1105.48

Rec: 4/4

Solution:

```
SELECT YEAR([OrderDate]) AS "Year", MONTH([OrderDate]) AS "Month", [SalesOrderID]  AS
"Identifier",  ROUND([TotalDue],2)   AS "Amount"
       FROM Sales.SalesOrderHeader
       WHERE MONTH([OrderDate]) = 5  AND [SalesOrderID] BETWEEN 46685 AND 50776
```

Year	Month	Identifier	Amount
2012	5	46685	2410.63
2012	5	46686	865.20
2013	5	50775	2264.25
2013	5	50776	1105.48

I had 4 records after running the query as presented above.

Task 2

2.1. Create a list of customers with more than 25 orders (use CTE). An example of the query result is presented in Table 2.1 below:

Table 2.1. Fragment of the query results for task 2.1

CustomerId	Last name, First name	Number of orders
11091	Perez, Dalton	28
11176	Roberts, Mason	28
11185	Henderson, Ashley	27
11200	Griffin, Jason	27
…	…	…

Rec.: 4/13

Solution:

```
SELECT S.CustomerID , P.LastName +
', ' + P.FirstName as "LastName,
FirstName", COUNT(S.SalesOrderID) as
"Number of orders"
      FROM
[Sales].[SalesOrderHeader] as S
       JOIN Person.Person P ON S.CustomerID = P.BusinessEntityID
       GROUP BY S.CustomerID, P.LastName, P.FirstName
       HAVING COUNT(S.SalesOrderID) > 25
       ORDER BY CustomerID ASC
```

CustomerID	LastName, FirstName	Number of orders
11091	Taylor, Jennifer	28
11176	Miller, Morgan	28
11185	Jackson, Morgan	27
11200	Lewis, Morgan	27

11223	Moore, Isabella	27
11262	Wilson, Natalie	27
11276	Martin, Natalie	27
11277	Vazquez, Ruben	27
11287	Carlson, Ruben	27
11300	Shen, Marshall	27
11330	Lewis, Grace	27
11331	Lee, Grace	27
11711	Jones, Brianna	27

I had 13 records after running the query as presented above.

2.2. Determine what factors affect the number of orders. An example of the query result is presented in Table 2.2 below.
Data Source: SalesOrderHeaderSalesReason, Sales.SalesReason

Table 2.2 Fragment of the query results for task 2.2

Factor	Orders
Price	17473
On Promotion	3515
Manufacturer	1746
…	…

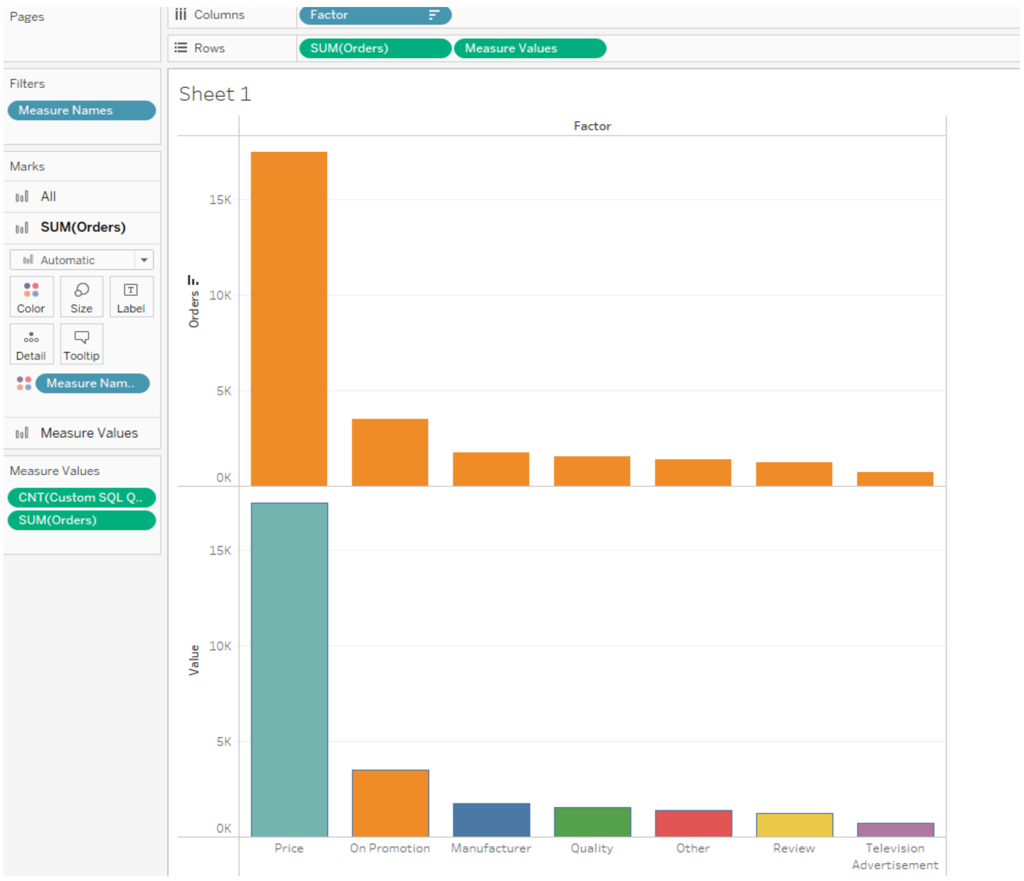
Rec.: 3/7

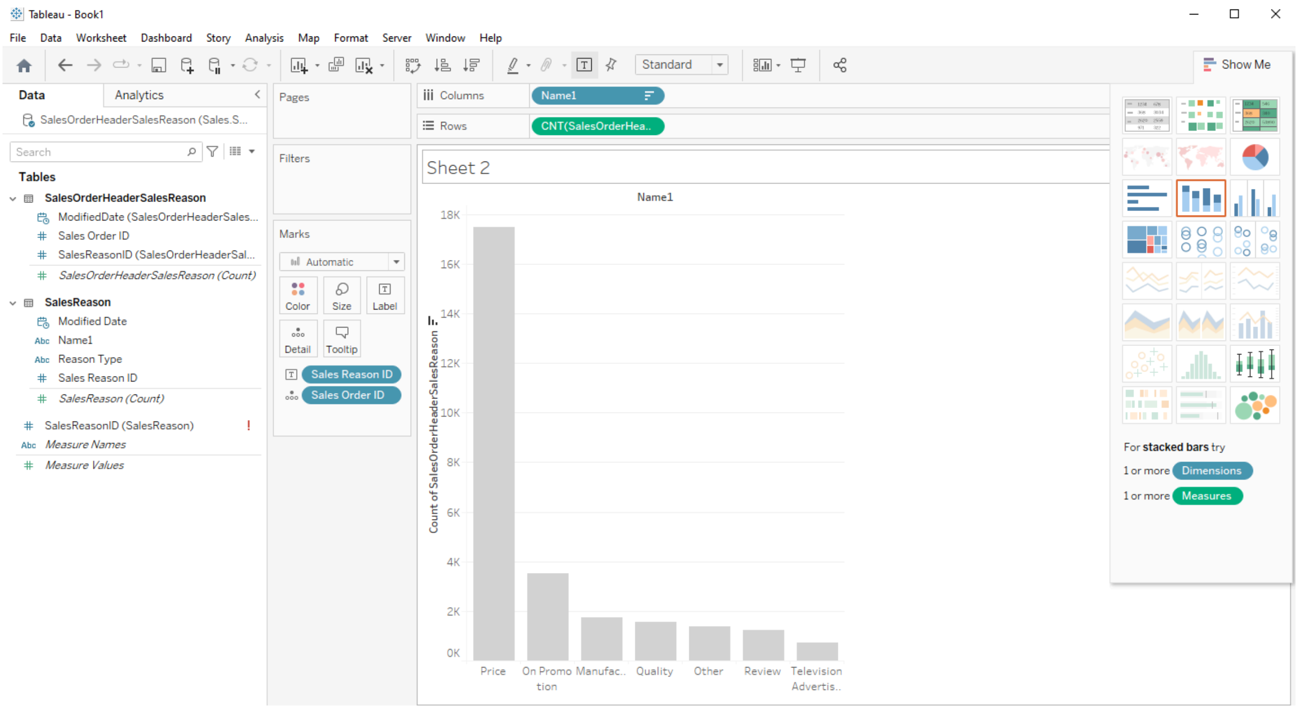
2.1 SQL query + fragment of the result (4 records from ?)
Solution:

```
SELECT r.Name [Factor], Count(SalesOrderID) [Orders] FROM Sales.SalesOrderHeaderSalesReason
s
INNER JOIN Sales.SalesReason r ON s.SalesReasonID = r.SalesReasonID
GROUP BY r.Name
   ORDER BY [Orders] DESC;
```

Factor	Orders
Price	17473
On Promotion	3515
Manufacturer	1746
Quality	1551
Other	1395
Review	1245
Television Advertisement	722

I had 7 records after running the query as presented above.
2.2 Tableau - the same result in graphical form
Solution:





Task 3

Define a query that determines the sales made by employees to individual customers in the years recorded in the database. An example of the query result is presented in Table 3 below:

Table 3 Fragment of the query results for task 3

SalesPersonID	CustomerID	2011	2012	2013	2014
274	30075	Lack	Lack	Lack	29524.05
274	30096	Lack	26305.46	Lack	Lack
275	29486	Lack	Lack	151107.24	53531.92
275	29487	37621.78	22003.89	Lack	Lack
…	…	…	…	…	…

Rec.: 4/? 860

3.1 SQL query + fragment of the result (4 records from ?)

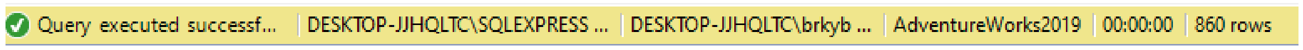
Solution:

```
SELECT * FROM (
SELECT SalesPersonId, CustomerID, TotalDue, YEAR(OrderDate) AS "Year"
FROM Sales.SalesOrderHeader
WHERE SalesPersonId IS NOT NULL
GROUP BY CustomerID, SalesPersonId, TotalDue, YEAR(OrderDate)
) As SalesResult
PIVOT (
SUM([TotalDue])
FOR [Year] in ([2011], [2012], [2013], [2014])
) AS P
 ORDER BY P.SalesPersonID;
```

SalesPersonId	CustomerID	2011	2012	2013	2014
274	29491	NULL	37625.4303	NULL	NULL
274	29493	2417.4793	NULL	NULL	NULL
274	29514	NULL	NULL	3931.9148	NULL
274	29523	NULL	NULL	NULL	38762.014
274	29576	NULL	60.1358	NULL	NULL
274	29579	NULL	NULL	NULL	39933.1824
274	29604	NULL	729.6412	NULL	NULL
274	29605	NULL	33206.0223	NULL	NULL
274	29616	NULL	83549.5837	NULL	71861.7017
274	29617	NULL	NULL	224100.514	NULL
274	29623	NULL	NULL	2194.914	NULL
274	29650	NULL	NULL	93.5417	NULL
274	29666	NULL	NULL	17990.9615	NULL
274	29669	NULL	4460.0736	NULL	NULL
274	29671	NULL	NULL	NULL	13290.1645
274	29675	NULL	4792.247	NULL	NULL

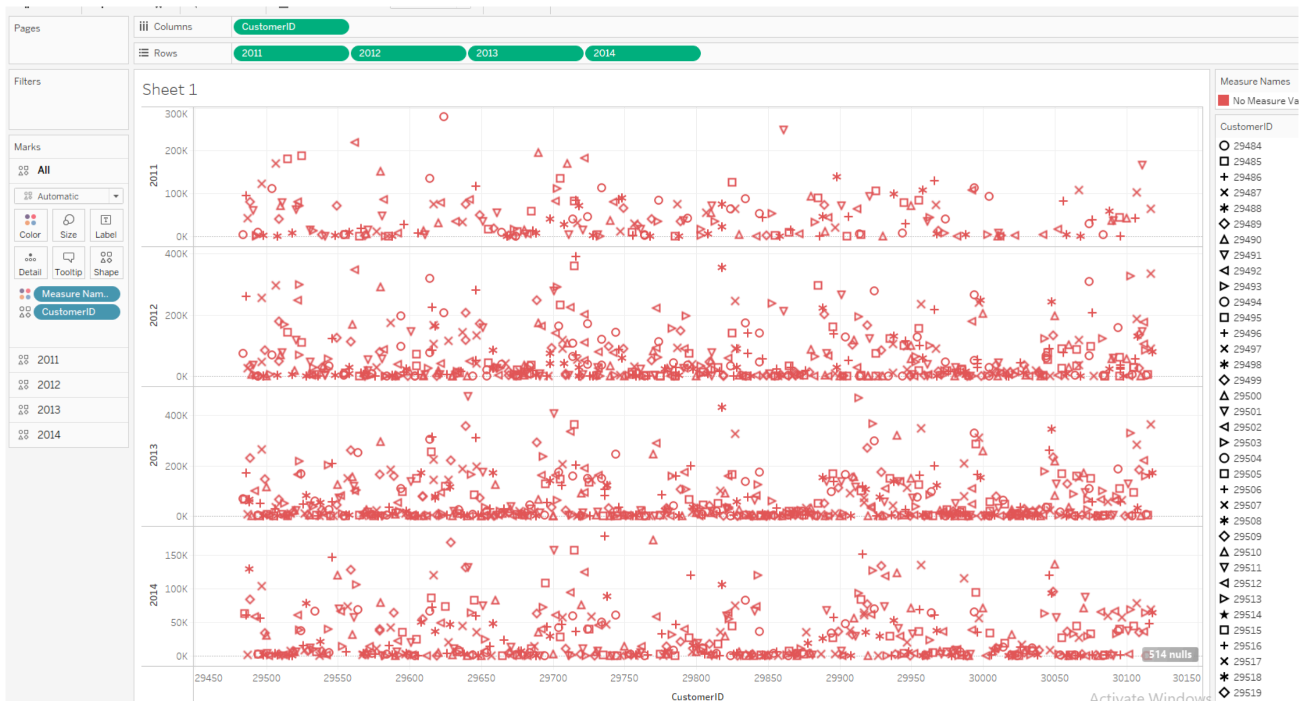
274	29680	NULL	3047.9964	NULL	NULL
274	29691	NULL	236.3547	NULL	NULL
274	29707	NULL	90167.3302	NULL	NULL
274	29716	NULL	68918.2404	NULL	NULL
274	29719	NULL	NULL	1094.4518	NULL
274	29722	NULL	51204.0606	95426.0186	NULL

There was 860 row in table as represented below.


Query executed successf... | DESKTOP-JJHQLTC\SQLEXPRESS ... | DESKTOP-JJHQLTC\brkyb ... | AdventureWorks2019 | 00:00:00 | 860 rows

3.2 Tableau - the same result in graphical form

Solution:



Presented in a different shapes.

Task 4

Create a pivot table that shows:

4.1 The average annual amount of purchases made by customers in 2013-2014 using the PIVOT operator.

4.2 The average annual amount of purchases made by customers in 2013-2014 without the PIVOT operator.

Table 4 **Fragment of the query results for task 4**

Name	CustomerID	2013	2014
Achong, Gustavo	29484	30937.91	NULL

Abel, Catherine	29485	28773.45	27670.88
Abercrombie, Kim	29486	37776.81	26765.96
Acevedo, Humberto	29487	2461.74	465.15

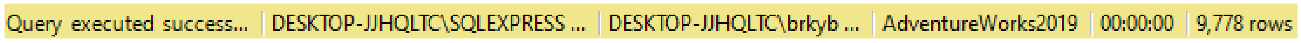
Rec: 4/?

4.3 SQL query + fragment of the result (4 records from 9778)

Solution:

```
SELECT * FROM (
SELECT n.FirstName+',' +n.LastName [Name],CustomerID, TotalDue, YEAR(OrderDate) AS "Year"
FROM Sales.SalesOrderHeader
INNER JOIN Person.Person n ON Sales.SalesOrderHeader.CustomerID = n.BusinessEntityID
GROUP BY CustomerID, TotalDue, YEAR(OrderDate), n.FirstName,n.LastName
) SalesResult
PIVOT (
AVG([TotalDue])
FOR [Year] in ([2013], [2014])
   ) AS N;
```

Name	CustomerID	2013	2014
Mary,Young	11000	2679.0725	NULL
Amber,Young	11001	2674.0227	650.8008
Courtney,Young	11002	2604.5126	NULL
Jenna,Young	11003	2618.4632	NULL
Chloe,Harris	11004	2649.801	NULL
Joe,Madan	11005	2622.3529	NULL
Chloe,Martin	11006	2607.2696	NULL
Marshall,Xu	11007	2658.083	NULL
Joe,Srini	11008	2614.0543	NULL
Joe,Prasad	11009	2605.7778	NULL
Joe,Schmidt	11010	2590.1476	NULL
Joe,Rana	11011	2615.0101	NULL
Chloe,Thompson	11012	82.8529	6.9394
Joe,Raman	11013	43.0729	82.8529
Chloe,Garcia	11014	76.4936	NULL
Joe,Subram	11015	2763.5719	NULL



```
Query executed success... | DESKTOP-JJHQLTC\SQLEXPRESS ... | DESKTOP-JJHQLTC\brkyb ... | AdventureWorks2019 | 00:00:00 | 9,778 rows
```

4.4 SQL query + fragment of the result (4 records from 9778)

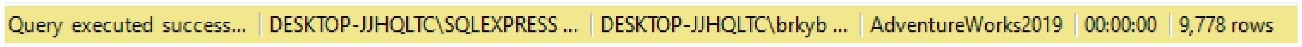
Solution:

```
SELECT f.FirstName+',' + f.LastName [Name], CustomerID,
AVG(CASE WHEN YEAR(OrderDate) = (2013) THEN [TotalDue] END) "2013",
AVG(CASE WHEN YEAR(OrderDate) = (2014) THEN [TotalDue] END) "2014"
FROM Sales.SalesOrderHeader
INNER JOIN Person.Person f ON [Sales].[SalesOrderHeader].CustomerID = f.BusinessEntityID
GROUP BY CustomerID, f.FirstName, f.LastName
       ORDER BY CustomerID;
```

Name	CustomerID	2013	2014
Mary,Young	11000	2679.0725	NULL
Amber,Young	11001	2674.0227	650.8008

Courtney,Young	11002	2604.5126	NULL
Jenna,Young	11003	2618.4632	NULL
Chloe,Harris	11004	2649.801	NULL
Joe,Madan	11005	2622.3529	NULL
Chloe,Martin	11006	2607.2696	NULL
Marshall,Xu	11007	2658.083	NULL
Joe,Srini	11008	2614.0543	NULL
Joe,Prasad	11009	2605.7778	NULL
Joe,Schmidt	11010	2590.1476	NULL
Joe,Rana	11011	2615.0101	NULL
Chloe,Thompson	11012	82.8529	6.9394
Joe,Raman	11013	43.0729	82.8529
Chloe,Garcia	11014	76.4936	NULL
Joe,Subram	11015	2763.5719	NULL

There was 9.778 rows in the table as represented below.



Query executed success... | DESKTOP-JJHQLTC\SQLEXPRESS ... | DESKTOP-JJHQLTC\brkyb ... | AdventureWorks2019 | 00:00:00 | 9,778 rows

Conclusions:

Use this section to provide your conclusions:

First of all we used AdventureWorks2019 library in the MsSQL to take the data from.

*In conclusion, what I learned in this assignment is that sorting the tables using whether DESC (largest to smaller) or ASC (smaller to largest),SELECT DISTINCT statement to return only distinct (different) values. Sometimes there are duplicated values which we do now want to present in a table, shift comma using (*x), review ORDER BY and QUERY BY, Common table expression (CTE) (defines temporary named result set that available temporarily in the execution scope of a statement), finding specific date or value in the library table using BETWEEN _ AND_ command, AVG statement in using CASE statement , review ROUND, MONTH, COUNT, İNNER JOİN and using PİVOT operator(Pivot operator converts each row in the aggregated result set into corresponding columns in the output set). Moreover, even though I didn't have a experience about the Tableau app, I experienced basic leading data visualization tool used for data analysis and tried to represent them in a SHEET as the professor showed during the lecture.*