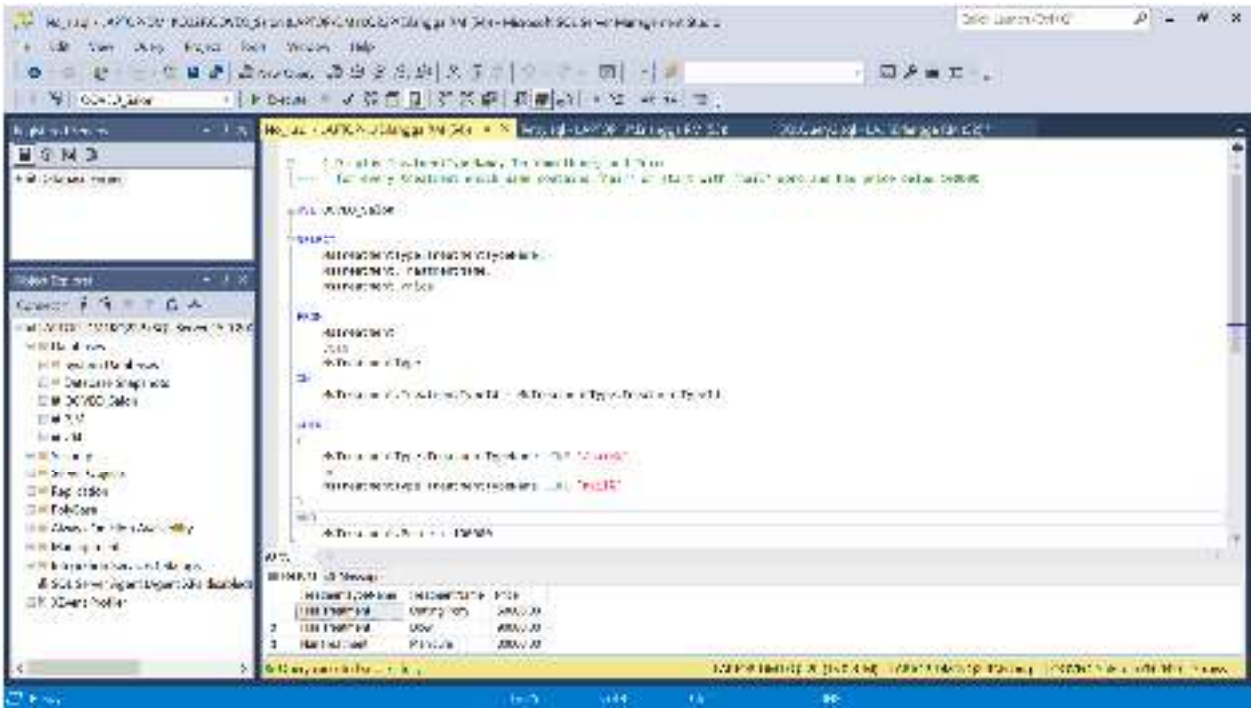
Nama : Erlangga Rizal Mahendra
NIM : 2440101704

Mo_6

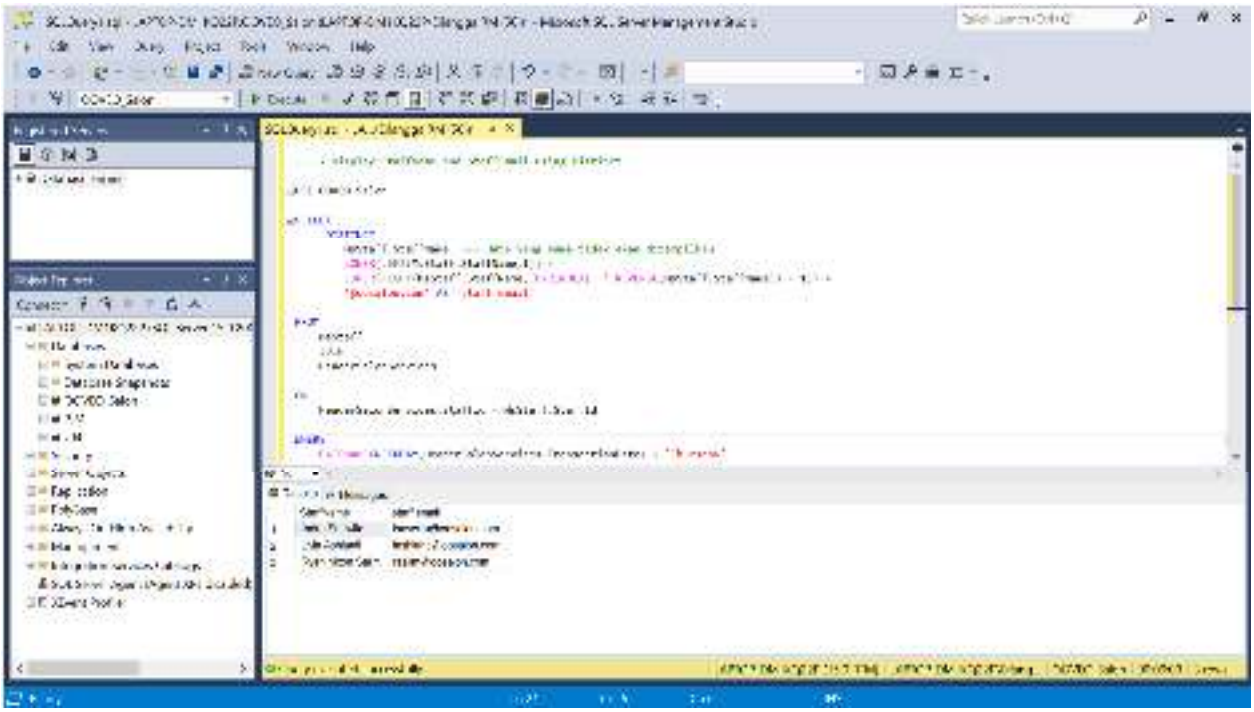
1. Display TreatmentTypeName, TreatmentName, and Price for every treatment which name contains ‘hair’ or start with ‘nail’ word and has price below 100000.
(join, like)

```
USE OOVEO_Salon

SELECT
	MsTreatmentType.TreatmentTypeName,
	MsTreatment.TreatmentName,
	MsTreatment.Price

FROM
	MsTreatment
	JOIN
	MsTreatmentType
ON
	MsTreatment.TreatmentTypeId = MsTreatmentType.TreatmentTypeId

WHERE
(
	MsTreatmentType.TreatmentTypeName LIKE '%hair%'
	or
	MsTreatmentType.TreatmentTypeName LIKE 'nail%'
)
AND
	MsTreatment.Price < 100000
```



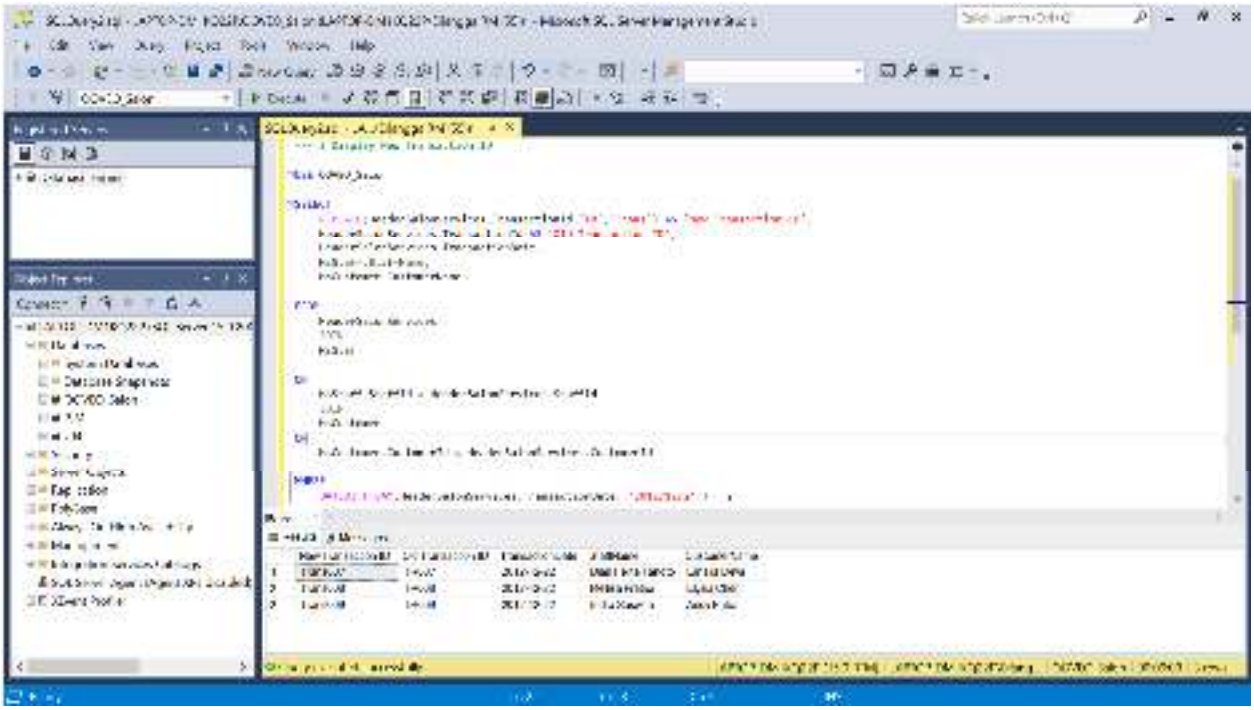
2. Display StaffName and StaffEmail (obtained from the first character of staff's name in lowercase format and followed with last word of staff's name and '@oosalon.com' word) for every staff who handle transaction on Thursday.The duplicated data must be displayed only once.
(distinct, lower, left, reverse, left, charindex, join, datename, weekday, like)

```
USE OOVEO_Salon

SELECT
        DISTINCT
              MsStaff.StaffName, --- Data yang sama tidak akan ditampilkan
              LOWER(LEFT(MsStaff.StaffName,1)) +
              LOWER(RIGHT(MsStaff.StaffName,CHARINDEX(' ',REVERSE(MsStaff.StaffName)) -
1)) +
              '@oosalon.com' AS 'staff email'

FROM
       MsStaff
       JOIN
       HeaderSalonServices

ON
       HeaderSalonServices.StaffId = MsStaff.StaffId

WHERE
       DATENAME(WEEKDAY,HeaderSalonServices.TransactionDate) = 'Thursday'
```



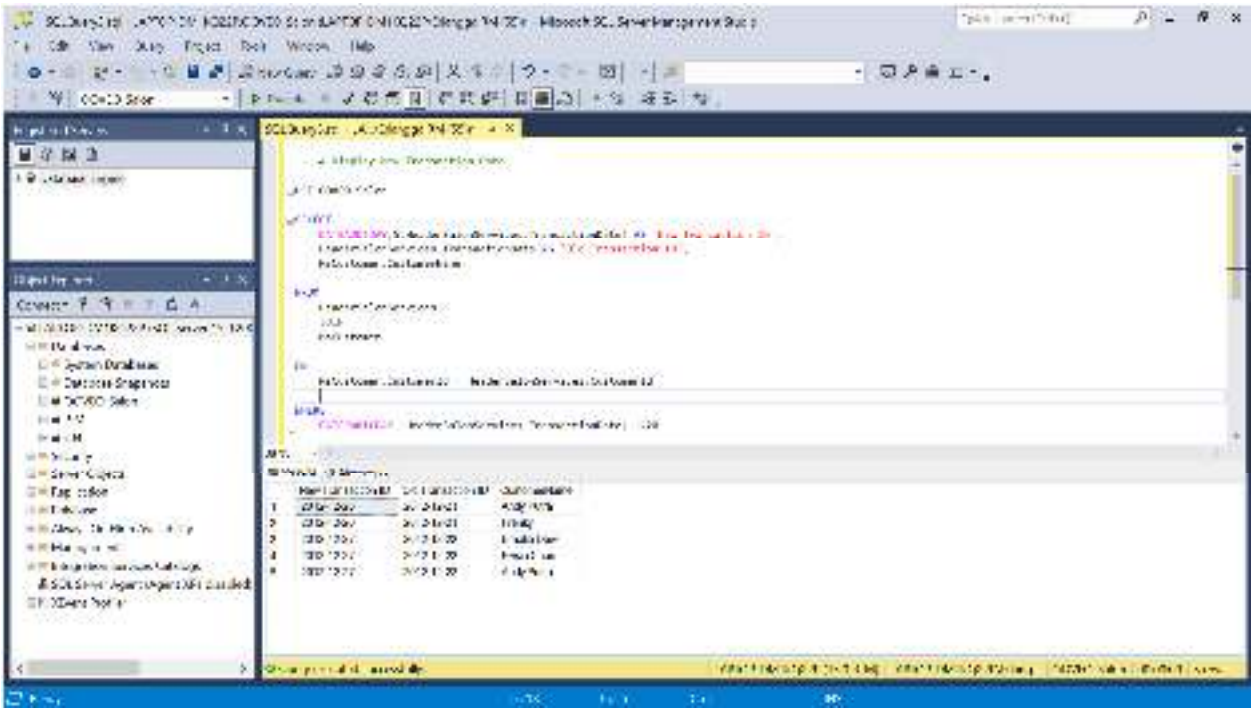
3. Display New Transaction ID (obtained by replacing 'TR' in TransactionID with 'Trans'), Old Transaction ID (obtained from TransactionId), TransactionDate, StaffName, and CustomerName for every transaction which happened 2 days before 24th December 2012.
(**replace**, **join**, **datediff**, **day**)

```
USE OOVEO_Salon

SELECT
	REPLACE(HeaderSalonServices.TransactionId,'TR','Trans') AS 'New Transaction ID',
	HeaderSalonServices.TransactionId AS 'Old Transaction ID',
	HeaderSalonServices.TransactionDate,
	MsStaff.StaffName,
	MsCustomer.CustomerName

FROM
	HeaderSalonServices
	JOIN
	MsStaff

ON
	MsStaff.StaffId = HeaderSalonServices.StaffId
	JOIN
	MsCustomer
ON
	MsCustomer.CustomerId = HeaderSalonServices.CustomerId

WHERE
DATEDIFF(DAY,HeaderSalonServices.TransactionDate, '2012/12/24') = 2
```



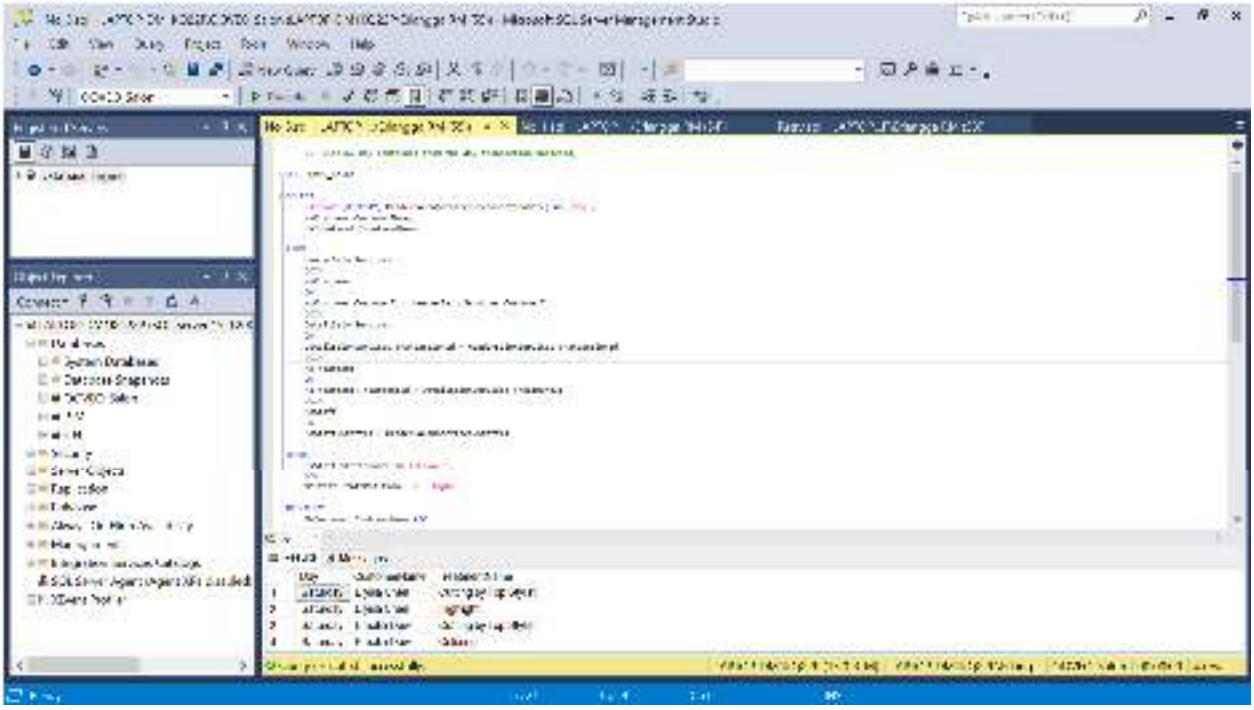
4. Display New Transaction Date (obtained by adding 5 days to TransactionDate), Old Transaction Date (obtained from TransactionDate), and CustomerName for every transaction which didn't happen on day 20th.
(dateadd, day, join, datepart)

```
USE OOVEO_Salon

SELECT
	DATEADD(DAY,5,HeaderSalonServices.TransactionDate) AS 'New Transaction ID',
	HeaderSalonServices.TransactionDate AS 'Old Transaction ID',
	MsCustomer.CustomerName

FROM
	HeaderSalonServices
	JOIN
	MsCustomer

ON
	MsCustomer.CustomerId = HeaderSalonServices.CustomerId

WHERE
	DATEPART(DAY, HeaderSalonServices.TransactionDate) !=20
```



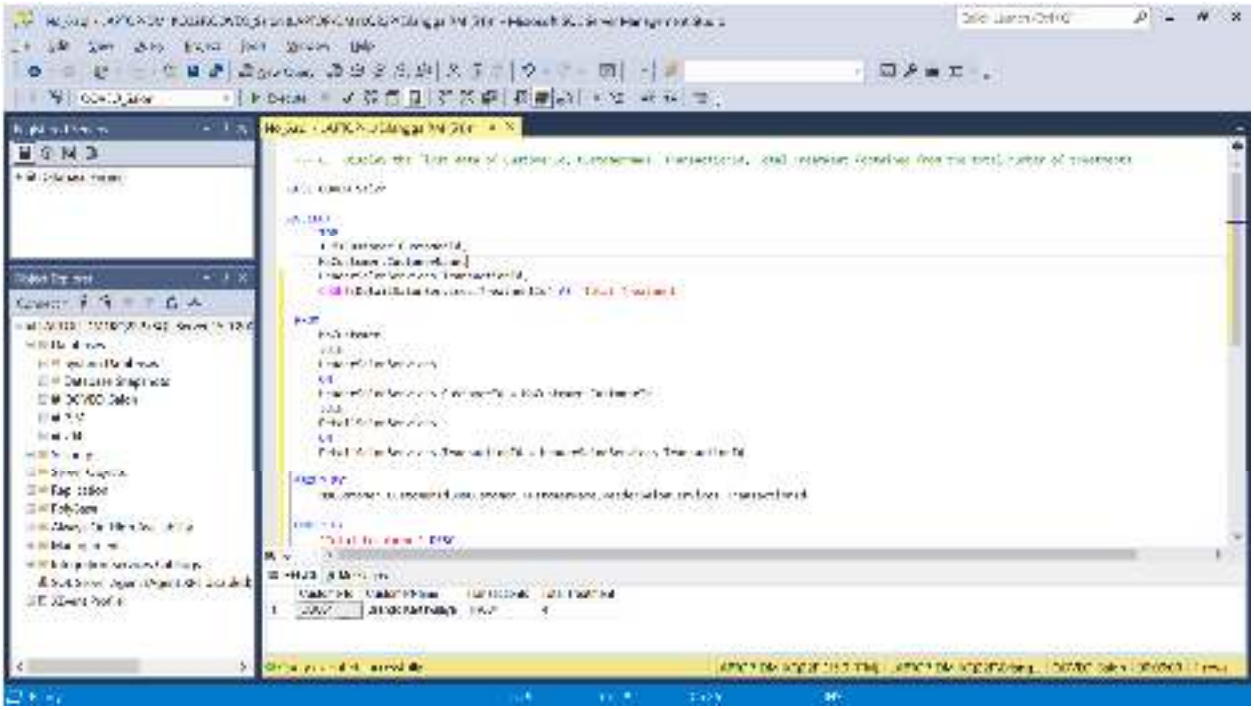
5. Display Day (obtained from the day transaction happened), CustomerName, and TreatmentName for every Customer who was handled by female staff that has position name begin with 'TOP' word. Then order the data based on CustomerName in ascending format. (**datename**, **weekday**, **join**, **in**, **like**, **order by**)

```
USE OOVEO_Salon

SELECT
	DATENAME(WEEKDAY, HeaderSalonServices.TransactionDate) AS 'Day',
	MsCustomer.CustomerName,
	MsTreatment.TreatmentName

FROM
	HeaderSalonServices
	JOIN
	MsCustomer
	ON
	MsCustomer.CustomerId = HeaderSalonServices.CustomerId
	JOIN
	DetailSalonServices
	ON
	DetailSalonServices.TransactionId = HeaderSalonServices.TransactionId
	JOIN
	MsTreatment
	ON
	MsTreatment.TreatmentId = DetailSalonServices.TreatmentId
	JOIN
	MsStaff
	ON
	MsStaff.StaffId = HeaderSalonServices.StaffId

WHERE
	MsStaff.StaffGender IN ('Female')
	AND
	MsStaff.StaffPosition LIKE 'top%'

ORDER BY
	MsCustomer.CustomerName ASC
```



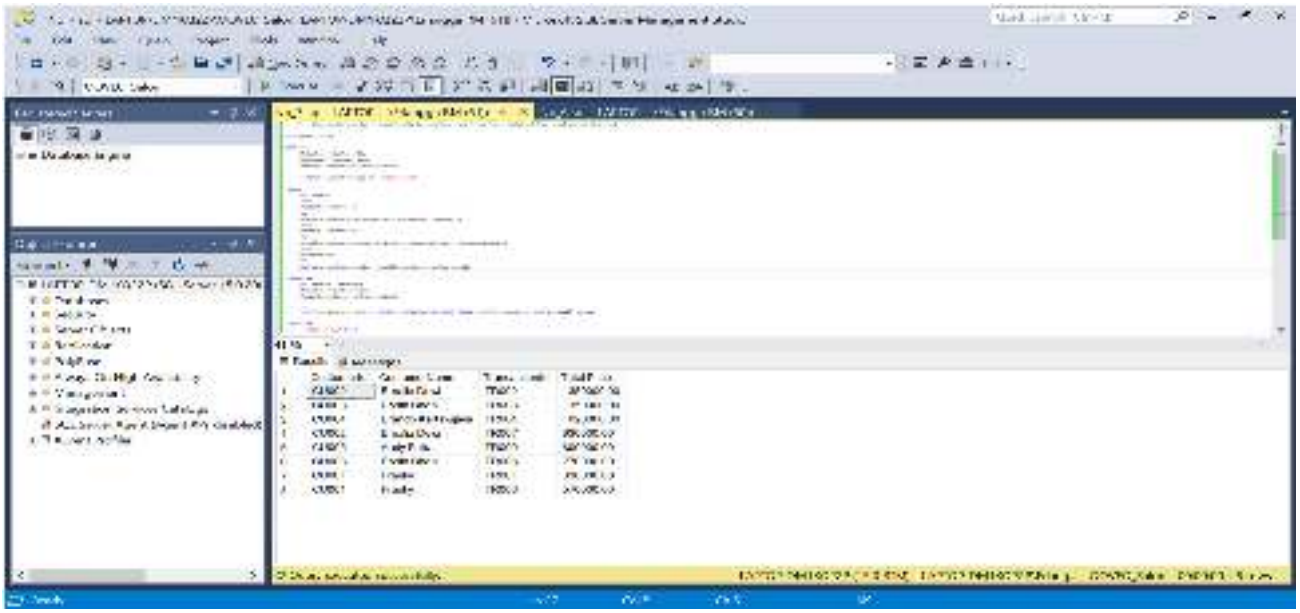
6. Display the first data of CustomerId, CustomerName, TransactionId, Total Treatment (obtained from the total number of treatment). Then sort the data based on Total Treatment in descending format.
   (**top**, **count**, **join**, **group by**, **order by**)

```
USE OOVEO_Salon

SELECT
	TOP
	1 MsCustomer.CustomerId,
	MsCustomer.CustomerName,
	HeaderSalonServices.TransactionId,
	COUNT(DetailSalonServices.TreatmentId) AS 'Total Treatment'

FROM
	MsCustomer
	JOIN
	HeaderSalonServices
	ON
	HeaderSalonServices.CustomerId = MsCustomer.CustomerId
	JOIN
	DetailSalonServices
	ON
	DetailSalonServices.TransactionId = HeaderSalonServices.TransactionId

GROUP BY
	MsCustomer.CustomerId,MsCustomer.CustomerName,HeaderSalonServices.TransactionId

ORDER BY
	'Total treatment' DESC
```



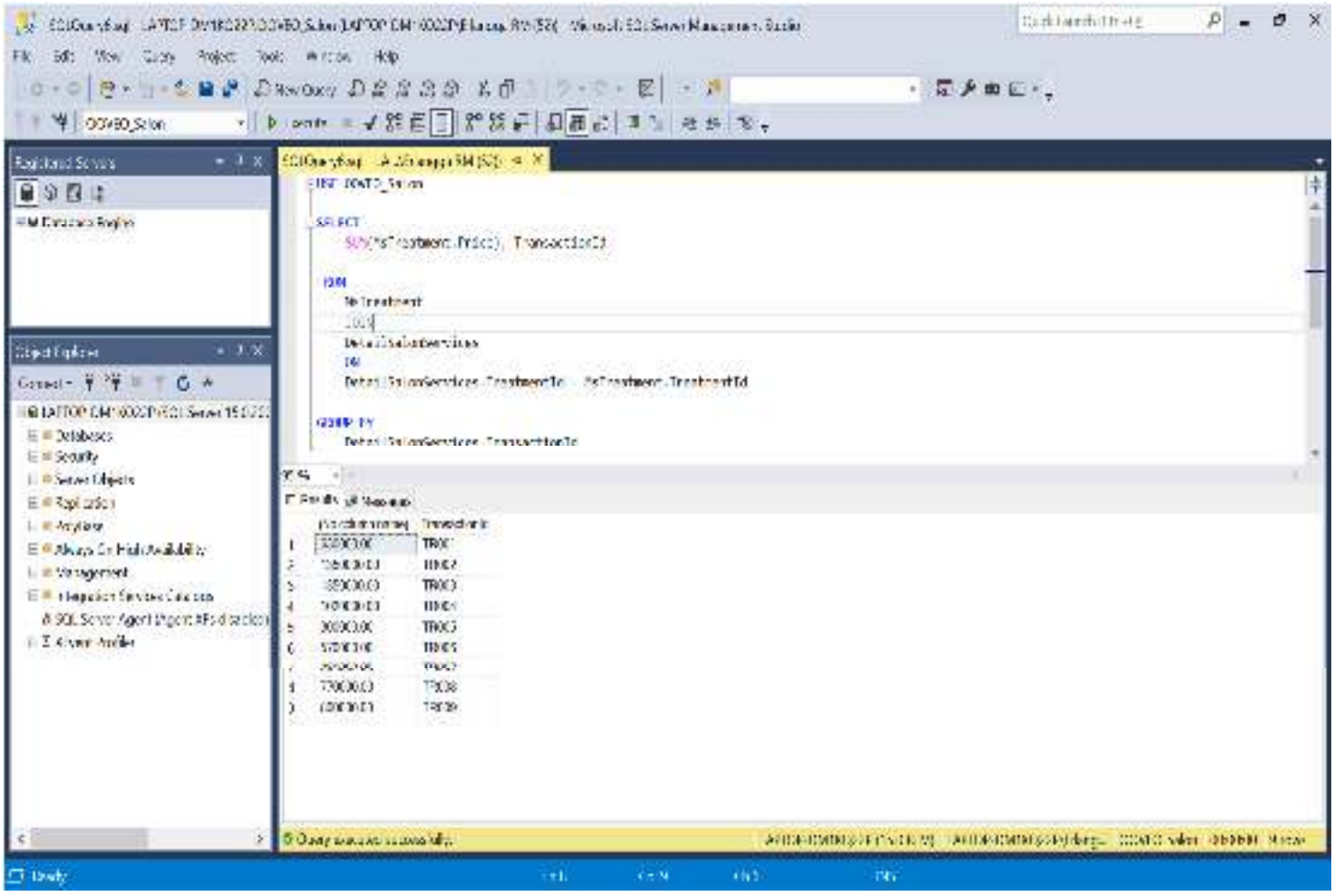
7. Display CustomerId, TransactionId, CustomerName, and Total Price (obtained from total amount of price) for every transaction with total price is higher than the average value of treatment price from every transaction. Then sort the data based on Total Price in descending format.
   (sum, join, alias subquery,avg, group by, having, order by)

```
USE OOVEO_Salon

SELECT
	MsCustomer.CustomerId,
	MsCustomer.CustomerName,
	HeaderSalonServices.TransactionId,

	SUM(MsTreatment.Price) AS 'Total Price'

FROM
	MsCustomer
	JOIN
	HeaderSalonServices
	ON
	HeaderSalonServices.CustomerId = MsCustomer.CustomerId
	JOIN
	DetailSalonServices
	ON
	DetailSalonServices.TransactionId = HeaderSalonServices.TransactionId
	JOIN
	MsTreatment
	ON
	MsTreatment.TreatmentId = DetailSalonServices.TreatmentId

GROUP BY
	MsCustomer.CustomerId,
	MsCustomer.CustomerName,
	HeaderSalonServices.TransactionId

HAVING
	SUM(MsTreatment.Price) > (SELECT SUM(av.Hasil) FROM (SELECT AVG(MsTreatment.Price)
AS Hasil) AS av)

ORDER BY
	 'Total Price' DESC
```



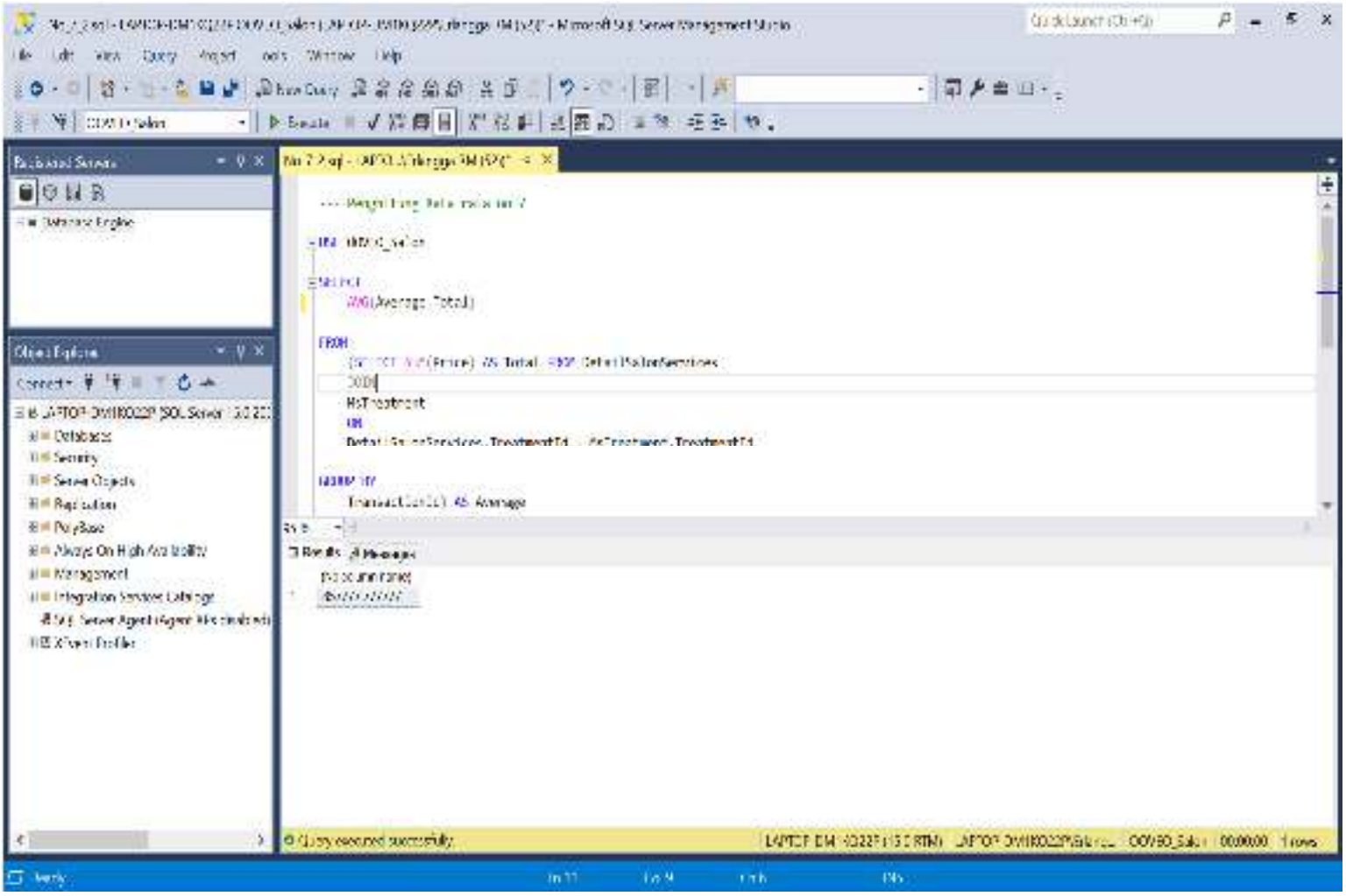
7.1 Menghitung total masing-masing baris

```
USE     OOVEO_Salon

SELECT
        SUM(MsTreatment.Price), TransactionId

FROM
        MsTreatment
        JOIN
        DetailSalonServices
        ON
        DetailSalonServices.TreatmentId = MsTreatment.TreatmentId

GROUP BY
        DetailSalonServices.TransactionId
```



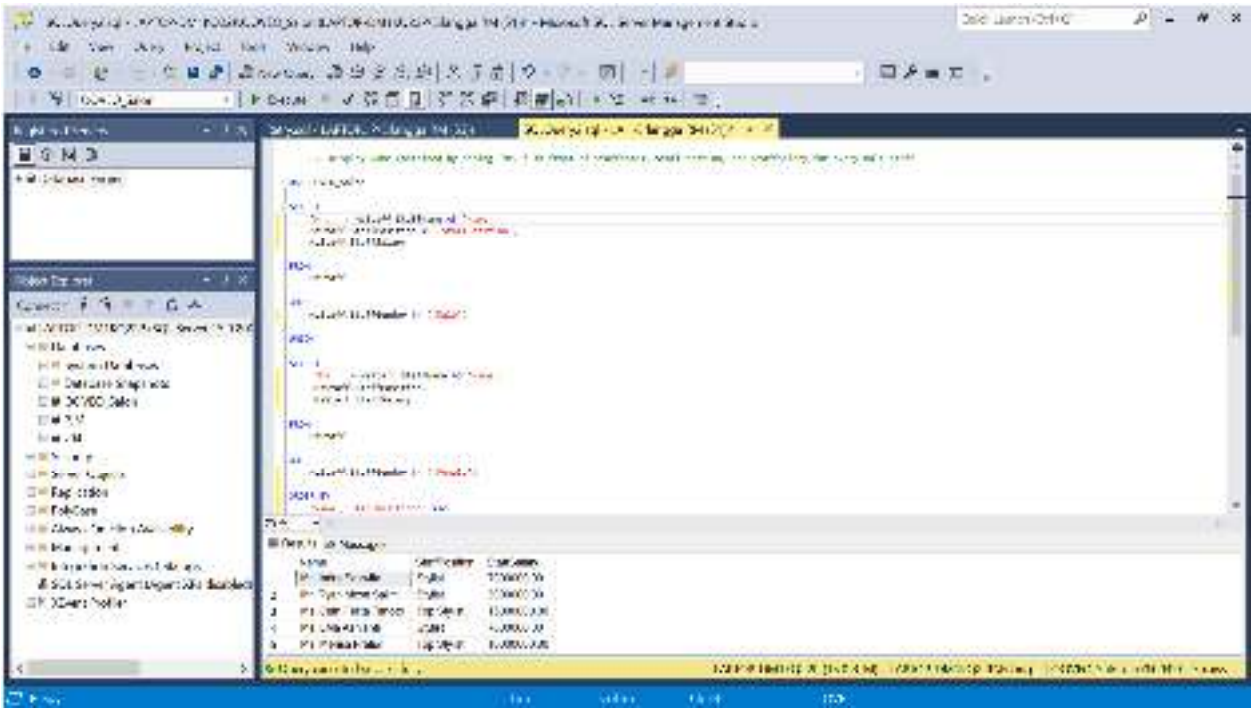
7.2 Menghitung rata-rata

```
USE OOVEO_Salon

SELECT
	AVG(Average.Total)

FROM
	(SELECT SUM(Price) AS Total FROM DetailSalonServices
	JOIN
	MsTreatment
	ON
	DetailSalonServices.TreatmentId = MsTreatment.TreatmentId

GROUP BY
	TransactionId) AS Average
```



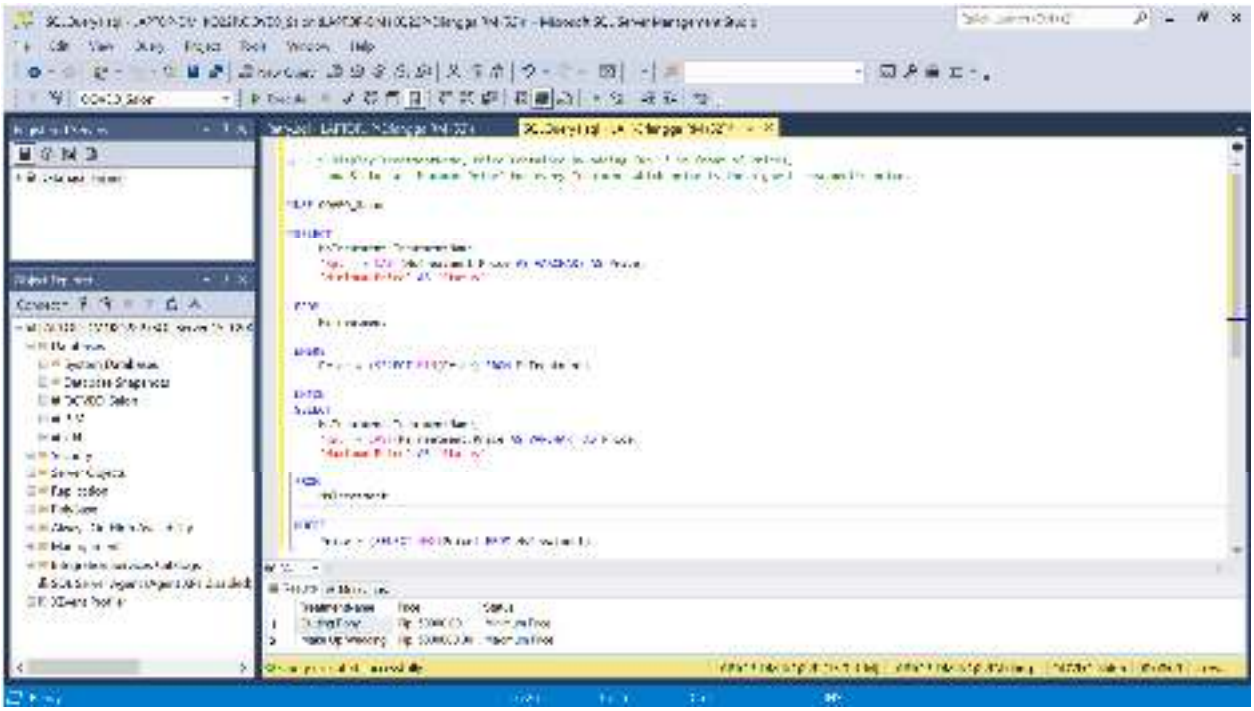
8. Display Name (obtained by adding 'Mr. ' in front of StaffName), StaffPosition, and StaffSalary for every male staff. The **combine** it with Name (obtained by adding 'Ms. ' in front of StaffName), StaffPosition, and StaffSalary for every female staff. Then sort the data based on Name and StaffPosition in ascending format.
(union, order by)

```
USE OOVEO_Salon

SELECT
	'Mr. ' + MsStaff.StaffName AS 'Name',
	MsStaff.StaffPosition AS 'StaffPosition',
	MsStaff.StaffSalary

FROM
	MsStaff

WHERE
	MsStaff.StaffGender IN ('Male')

UNION

SELECT
	 'Ms. ' + MsStaff.StaffName AS 'Name',
	 MsStaff.StaffPosition,
	 MsStaff.StaffSalary

FROM
	MsStaff

WHERE
	MsStaff.StaffGender IN ('Female')

ORDER BY
	'Name','StaffPosition'  asc
```



9. Display TreatmentName, Price (obtained by adding 'Rp. ' in front of Price), and Status as 'Maximum Price' for every Treatment which price is the highest treatment's price. Then **combine** it with TreatmentName, Price (obtained by adding 'Rp. ' in front of Price), and Status as 'Minimum Price' for every Treatment which price is the lowest treatment's price.
(cast, max, alias subquery, union, min)

```
USE OOVEO_Salon

SELECT
	MsTreatment.TreatmentName,
	'Rp. ' + CAST(MsTreatment.Price AS VARCHAR) AS Price,
	'Minimum Price' AS 'Status'

FROM
	MsTreatment

WHERE
	Price = (SELECT MIN(Price) FROM MsTreatment)

UNION
SELECT
	MsTreatment.TreatmentName,
	'Rp. '+ CAST(MsTreatment.Price AS VARCHAR) AS Price,
	'Maximum Price' AS 'Status'

FROM
	MsTreatment

WHERE
	Price = (SELECT MAX(Price) FROM MsTreatment)
```



10. Display Longest Name of Staff and Customer (obtained from CustomerName), Length of Name (obtained from length of customer's name), Status as 'Customer' for every customer who has the longest name. Then **combine** it with Longest Name of Staff and Customer (obtained from StaffName), Length of Name (obtained from length of staff's name), Status as 'Staff' for every staff who has the longest name
(len, max, alias subquery, union)

```
USE OOVEO_Salon

SELECT
	MsCustomer.CustomerName AS 'Longest Name Of Staff and Customer',
	LEN(MsCustomer.CustomerName) AS 'Length Of Name',
	'Customer' AS 'Status'

FROM
	MsCustomer

WHERE
	LEN(MsCustomer.CustomerName) = (SELECT MAX(LEN(MsCustomer.CustomerName))
FROM
	MsCustomer)

UNION

SELECT
	MsStaff.StaffName AS 'Longest Name Of Staff and Customer',
	LEN(MsStaff.StaffName) AS 'Length Of Name',
	'Staff' As 'Status'

FROM
	MsStaff

WHERE
	LEN(MsStaff.StaffName) = (SELECT MAX(LEN(MsCustomer.CustomerName))
FROM
     MsCustomer)
```