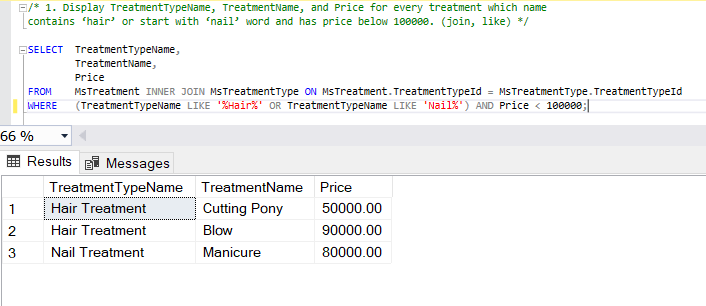
1. Display TreatmentTypeName, TreatmentName, and Price for every treatment which name contains ‘hair’ or start with ‘nail’ word and has price below 100000.

SELECT TreatmentTypeName,

TreatmentName,

Price

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

WHERE (TreatmentTypeName LIKE '%Hair%' OR TreatmentTypeName LIKE 'Nail%') AND 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.

SELECT DISTINCT StaffName,

CASE CHARINDEX(' ',MsStaff.StaffName) WHEN 0 THEN CONCAT(StaffName,'@oosalon.com')

ELSE LOWER(CONCAT(LEFT(StaffName,1),LTRIM(RIGHT(StaffName,CHARINDEX(' ',REVERSE(StaffName)))),'@oosalon.com'))

END AS StaffEmail

FROM MsStaff INNER JOIN HeaderSalonServices ON MsStaff.StaffId = HeaderSalonServices.StaffId

WHERE DATENAME(WEEKDAY,TransactionDate) LIKE '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.

SELECT REPLACE(TransactionId, 'TR', 'Trans') AS [New Transaction Id],

TransactionId AS [Old Transaction Day],

TransactionDate,

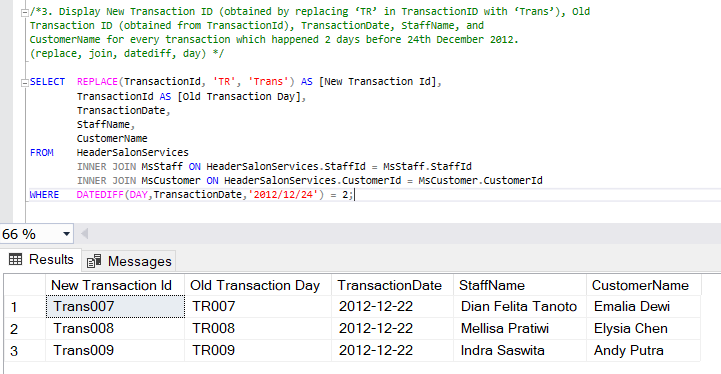
StaffName,

CustomerName

FROM HeaderSalonServices

INNER JOIN MsStaff ON HeaderSalonServices.StaffId = MsStaff.StaffId

INNER JOIN MsCustomer ON HeaderSalonServices.CustomerId = MsCustomer.CustomerId

WHERE DATEDIFF(DAY,TransactionDate,'2012/12/24') = 2;

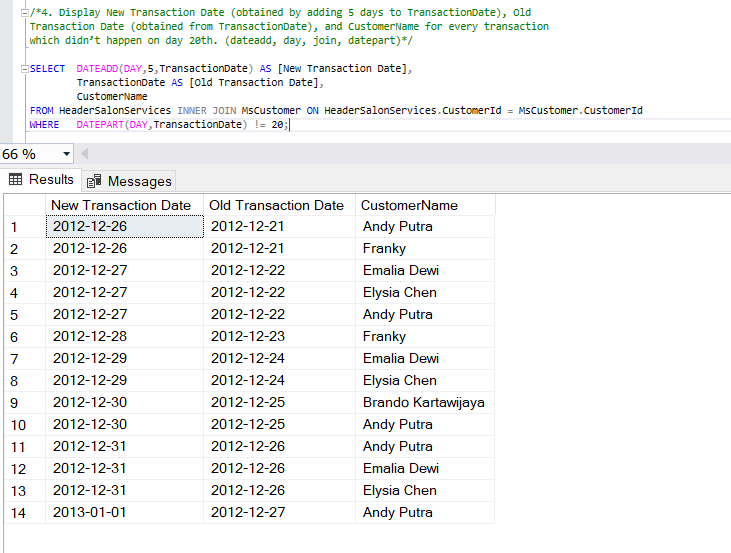
4. Display New Transaction Date (obtained by adding 5 days to TransactionDate), Old Transaction Date (obtained from TransactionDate), and CustomerName for every transaction which didn’t happen on day 20th .

SELECT DATEADD(DAY,5,TransactionDate) AS [New Transaction Date],

TransactionDate AS [Old Transaction Date],

CustomerName

FROM HeaderSalonServices INNER JOIN MsCustomer ON HeaderSalonServices.CustomerId = MsCustomer.CustomerId

WHERE DATEPART(DAY,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.

SELECT DATENAME(WEEKDAY,TransactionDate) AS Day,

CustomerName,

TreatmentName

FROM HeaderSalonServices

INNER JOIN MsCustomer ON HeaderSalonServices.CustomerId = MsCustomer.CustomerId

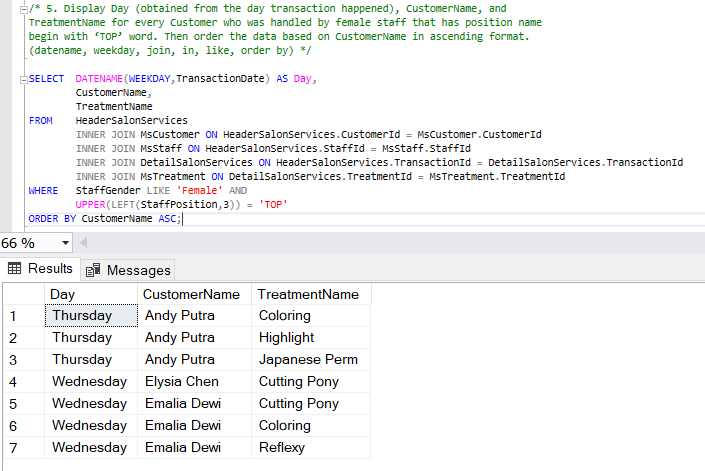
INNER JOIN MsStaff ON HeaderSalonServices.StaffId = MsStaff.StaffId

INNER JOIN DetailSalonServices ON HeaderSalonServices.TransactionId = DetailSalonServices.TransactionId

INNER JOIN MsTreatment ON DetailSalonServices.TreatmentId = MsTreatment.TreatmentId

WHERE StaffGender LIKE 'Female' AND

UPPER(LEFT(StaffPosition,3)) = 'TOP'

ORDER BY 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.

SELECT TOP 1 MsCustomer.CustomerId,CustomerName,HeaderSalonServices.TransactionId,

COUNT(MsTreatment.TreatmentId) AS [Total Treatment]

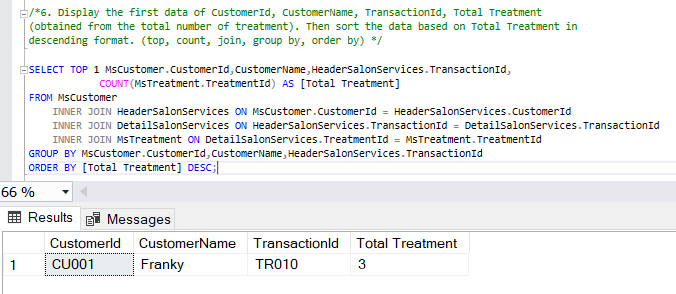
FROM MsCustomer

INNER JOIN HeaderSalonServices ON MsCustomer.CustomerId = HeaderSalonServices.CustomerId

INNER JOIN DetailSalonServices ON HeaderSalonServices.TransactionId = DetailSalonServices.TransactionId

INNER JOIN MsTreatment ON DetailSalonServices.TreatmentId = MsTreatment.TreatmentId

GROUP BY MsCustomer.CustomerId,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.

SELECT MsCustomer.CustomerId,HeaderSalonServices.TransactionId,CustomerName,

SUM(MsTreatment.Price) AS [Total Price]

FROM MsCustomer

INNER JOIN HeaderSalonServices ON MsCustomer.CustomerId = HeaderSalonServices.CustomerId

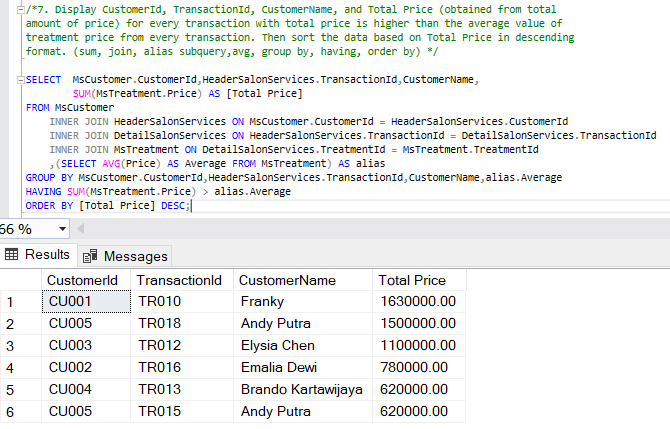
INNER JOIN DetailSalonServices ON HeaderSalonServices.TransactionId = DetailSalonServices.TransactionId

INNER JOIN MsTreatment ON DetailSalonServices.TreatmentId = MsTreatment.TreatmentId

,(SELECT AVG(Price) AS Average FROM MsTreatment) AS alias

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

HAVING SUM(MsTreatment.Price) > alias.Average

ORDER BY [Total Price] DESC;

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.

SELECT CONCAT('Mr. ',StaffName) AS [Name],

StaffPosition,

StaffSalary

FROM MsStaff

WHERE StaffGender LIKE 'Male'

UNION

SELECT CONCAT('Ms. ',StaffName) AS [Name],

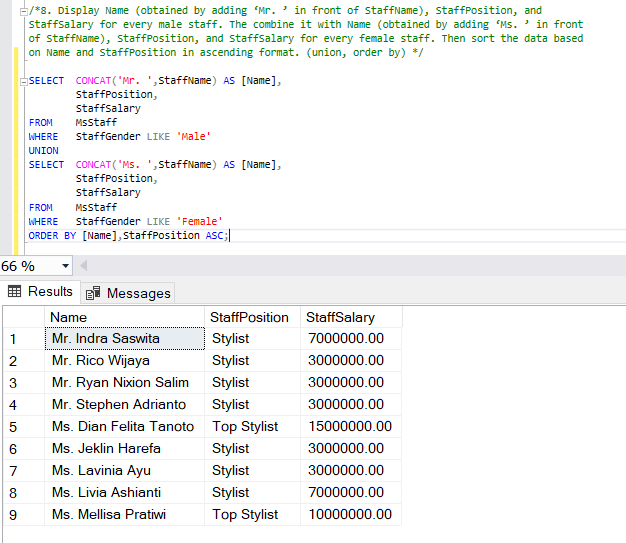
StaffPosition,

StaffSalary

FROM MsStaff

WHERE StaffGender LIKE '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.

SELECT TreatmentName,

CONCAT('Rp. ',Price) AS Price,

'Maximum Price' AS [Status]

FROM MsTreatment, (SELECT MAX(Price) AS Maximum FROM MsTreatment) AS alias

GROUP BY TreatmentName,Price,alias.Maximum

HAVING Price = alias.Maximum

UNION

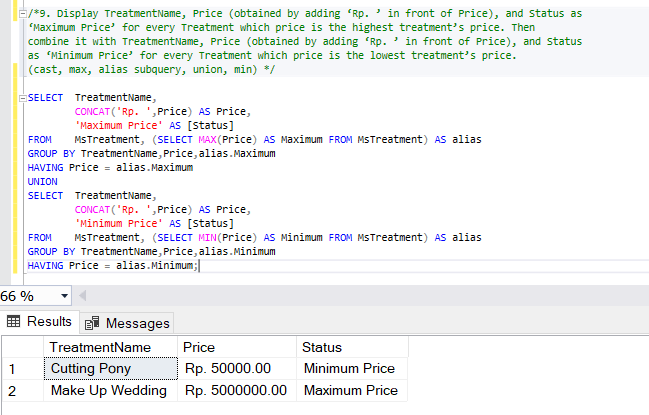
SELECT TreatmentName,

CONCAT('Rp. ',Price) AS Price,

'Minimum Price' AS [Status]

FROM MsTreatment, (SELECT MIN(Price) AS Minimum FROM MsTreatment) AS alias

GROUP BY TreatmentName,Price,alias.Minimum

HAVING Price = alias.Minimum;

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

SELECT CustomerName AS [Longest Name of Staff and Customer],

LEN(CustomerName) AS [Length of name],

'Customer' AS [Status]

FROM MsCustomer, (SELECT MAX(LEN(CustomerName)) AS Maximum FROM MsCustomer) AS alias

GROUP BY CustomerName,alias.Maximum

HAVING LEN(CustomerName) = alias.Maximum

UNION

SELECT StaffName AS [Longest Name of Staff and Customer],

LEN(StaffName) AS [Length of name],

'Staff' AS [Status]

FROM MsStaff, (SELECT MAX(LEN(StaffName)) AS Maximum FROM MsStaff) AS alias

GROUP BY StaffName,alias.Maximum

HAVING LEN(StaffName) = alias.Maximum;

