Northwind
DSLS 2023

MINI PROJECT

~ Data Engineering ~

Written By
Handhika Yanuar P

+6281225900513
handhikayp@gmail.com
https://handhikayp.medium.com/

Mini Project DE

Latar Belakang Projek

Tools dan Dataset

Tools : Microsoft SQL Server

Dataset

Pengerjaan

Bagian 1 Troubleshooting

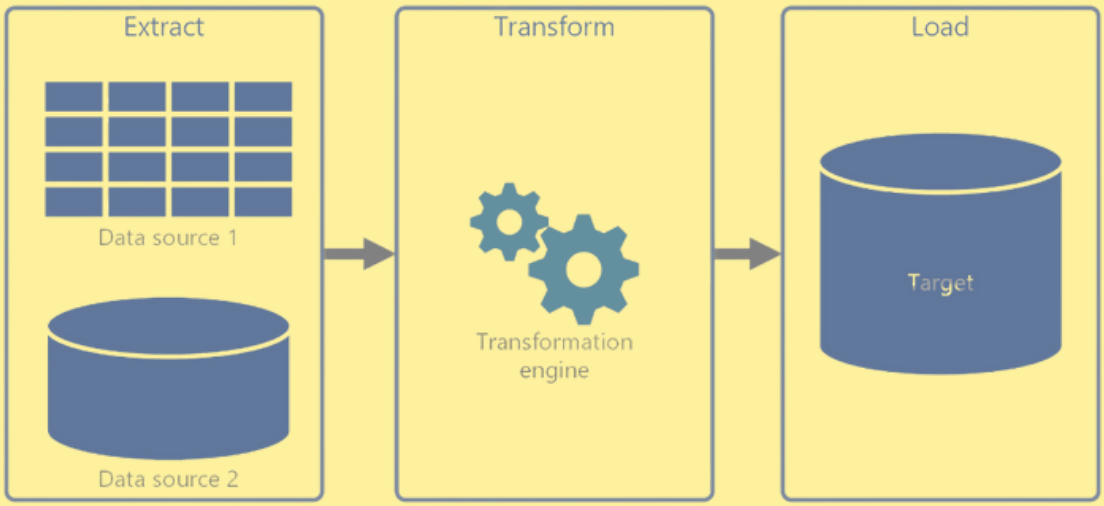
Bagian 2 Intermediate Queries

Bagian 3 Case Study

a. Product Analysis
b. Customer Analysis
c. Cohort Analysis

PENDAHULUAN

Tanggung jawab utama seorang data consultant adalah membantu perusahaan membuat keputusan berdasarkan data dan mengoptimalkan penggunaan data. Untuk mewujudkan tanggung jawab tersebut, langkah awal yang harus dikuasai oleh data consultant yaitu mampu mengintegrasikan data menggunakan metode ETL.



Sumber : microsoftlearn

ETL(Extract, Transform, Load) merupakan proses integrasi data meliputi ekstrak data dari berbagai sumber, berlanjut kepada transformasi data (proses agregasi, pembulatan, perhitungan, mapping, string function, dll), dan diakhiri dengan load yaitu data disimpan ke dalam sistem data warehouse. Data-data yang semula berasal dari berbagai sumber ini kemudian diolah menjadi data tunggal konsisten sehingga mudah dalam melakukan analisis. Oleh karena itu, kali ini kalian diminta untuk berlatih menjadi data consultant yang sesungguhnya.

handhikayp
Researcher

Jenis SQL yang akan digunakan pada projek ini adalah Microsoft SQL Server, SQL keluaran dari Microsoft ini sering digunakan perusahaan untuk mengakses big data. Untuk dapat menjalankan Microsoft SQL Server di laptop Anda, silakan ikuti panduan berikut.

Dataset yang digunakan adalah sampel data dari Microsoft, Northwind. Database Northwind ini menggambarkan database milik suatu perusahaan fiktif yang bernama Northwind Traders dimana perusahaan ini bergerak dalam bidang eksport import makanan. Dataset ini terdiri dari 13 tabel yakni :

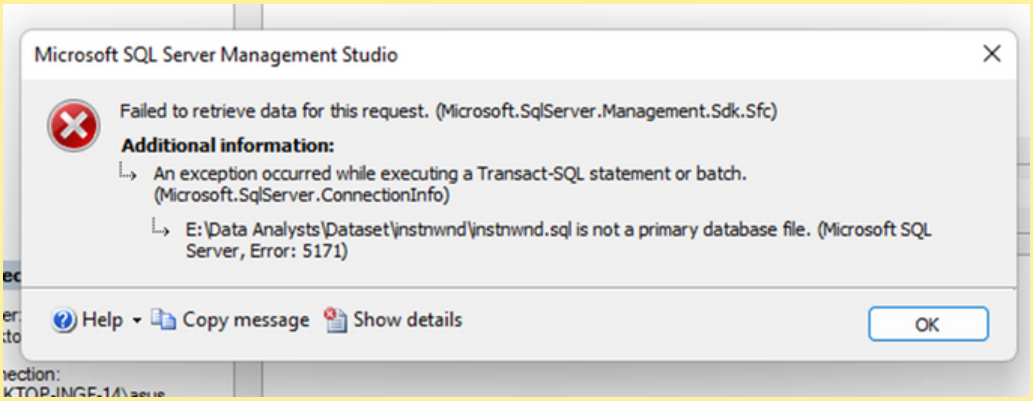
- Categories
- CustomerCustomerDemo
- CustomerDemographics
- Customers
- Employees
- EmployeeTerritories
- OrderDetails
- Orders
- Products
- Region
- Shippers
- Suppliers
- Territories

Bagian-1

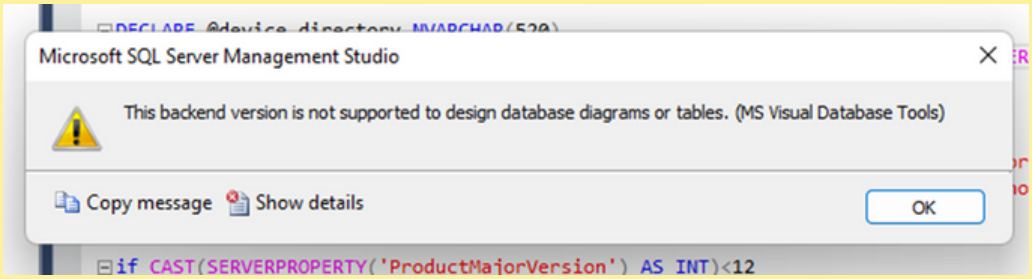
Troubleshooting

TROUBLESHOOTING

Proses pengerjaan mini project Data Engineer bagi saya yang bukan background matematika dan statistik cukup sulit. Dimana, peserta dituntut untuk membuat project baik dari intermediate query dan case study analysis. Meskipun demikian, saya cukup bisa melakukan penyusunan dokumentasi dari tools yang digunakan. Jika mengikuti tutorial yang diberikan, maka hasil akhirnya adalah data tidak dapat diload dan file tidak dapat diimport.



Kendala tersebut sebenarnya terjadi karena adanya kesalahan pemilihan versi SQL Server Management Studio (SSMS) yang diberikan panitia yaitu versi 18.



---

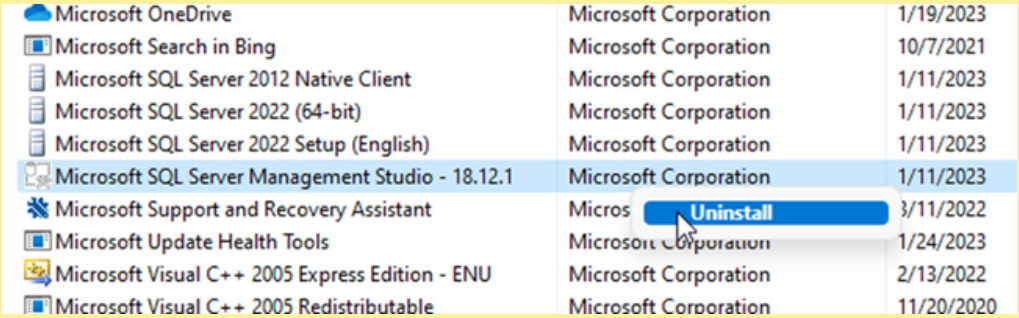
handhikayp

Researcher

TROUBLESHOOTING

Mengingat semua peserta tidak semua dari background data, seharusnya hal ini sudah dipertimbangkan (izin). Meskipun sebenarnya sudah dijelaskan dalam group terkait perbaikannya, penulis laporan tertarik untuk mengulik langkah instalasi dari aplikasi SSMS yang dapat digunakan.

1. Uninstall terlebih dahulu versi 18 yang tidak dapat digunakan

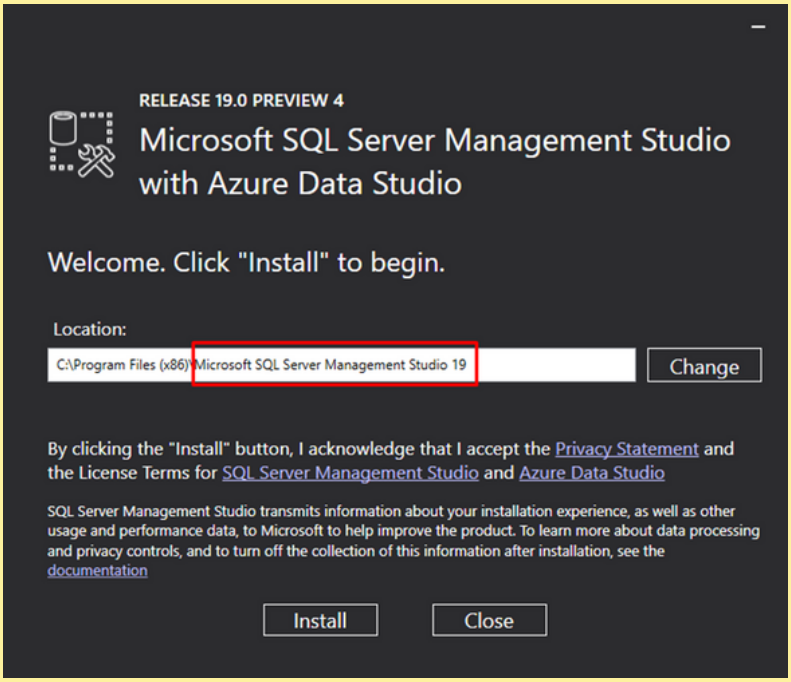


2. Unduh versi 19 pada link berikut. [LINK]

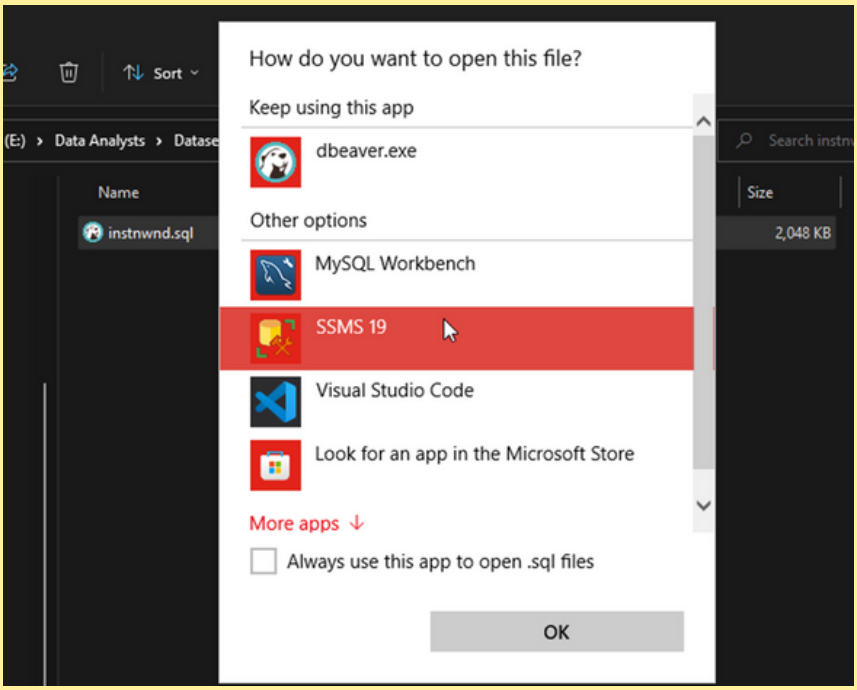
handhikayp
Researcher

TROUBLESHOOTING

3. Tentukan lokasi instalasi SSMS 19



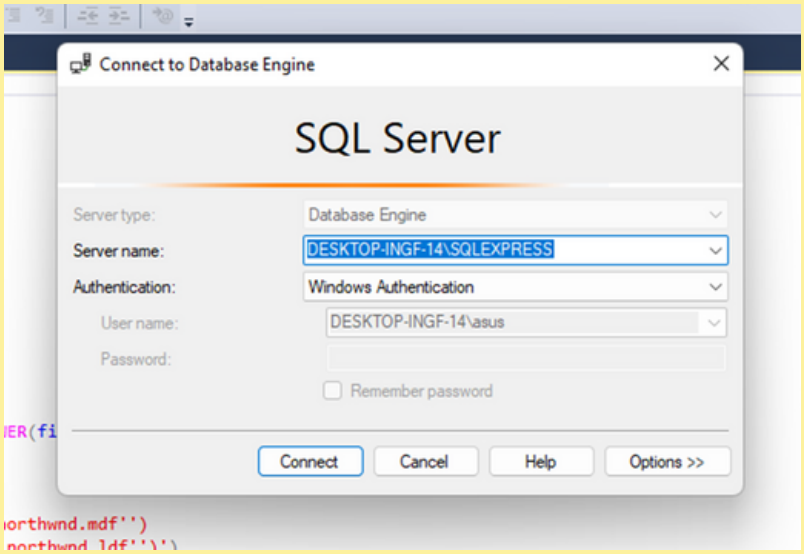
4. Setelah instalasi selesai, buka file northwind yang telah diberikan panitia [LINK] menggunakan SSMS 19 yang baru diinstall.



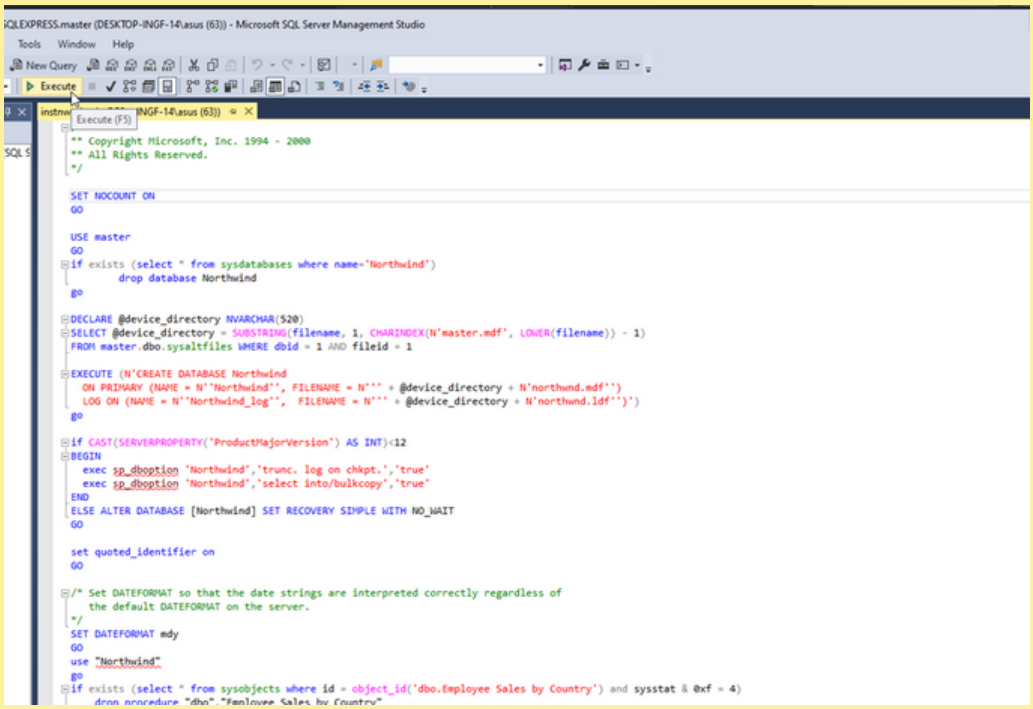
handhikayp
Researcher

TROUBLESHOOTING

5. Aplikasi akan terbuka dan tampilan window baru akan muncul, langsung pilih `Connect` saja.



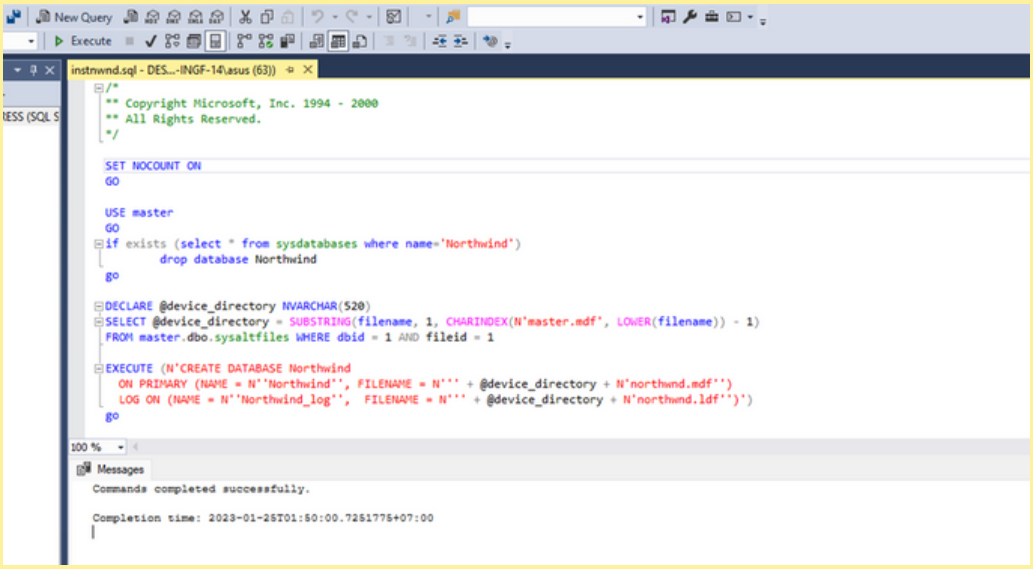
4. File yang diberikan sebenarnya adalah file SQL, jadi setelah file dihubungakan dengan SPSS, maka query SQL akan ditampilkan. Supaya database northwind muncul, tekan Execute (F5)



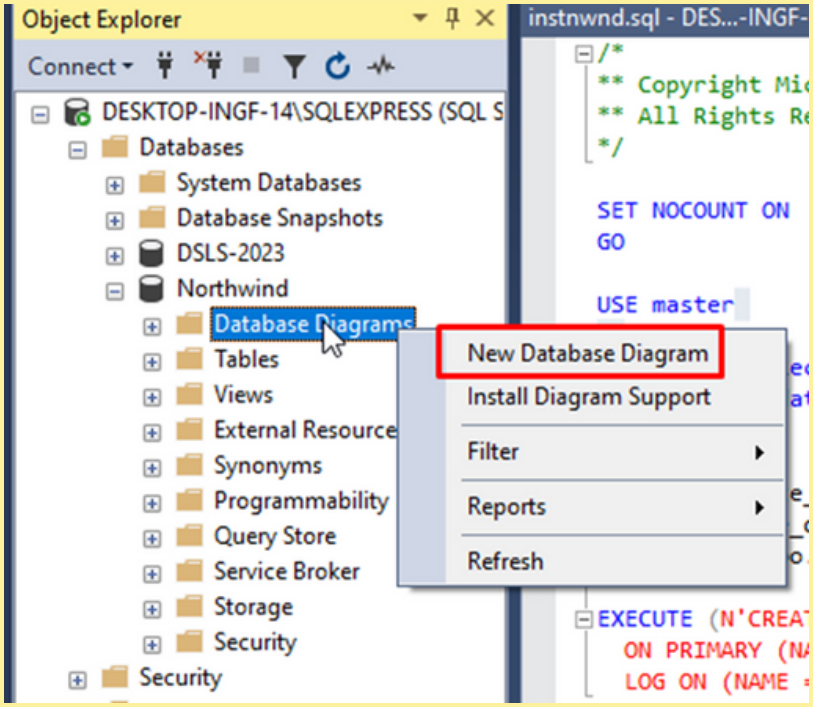
handhikayp
Researcher

TROUBLESHOOTING

7. Maka proses running querys SQL akan berjalan tanpa kendala.



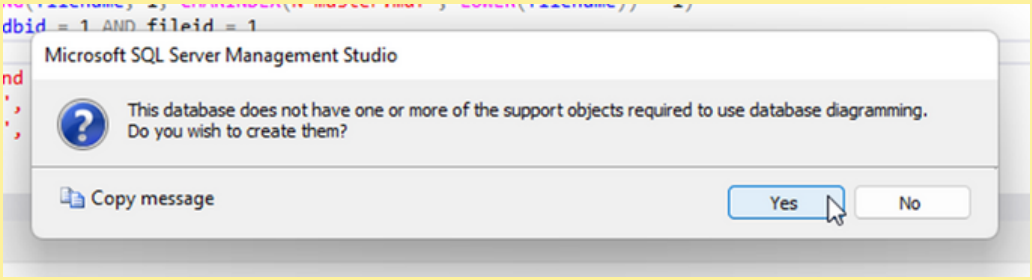
8. Database Northwind akan muncul pada sebelah kiri (Objcet Explorer). Expand pilihan Northwind dan pada Database Diagrams, klik kanan untuk membuat diagram baru.



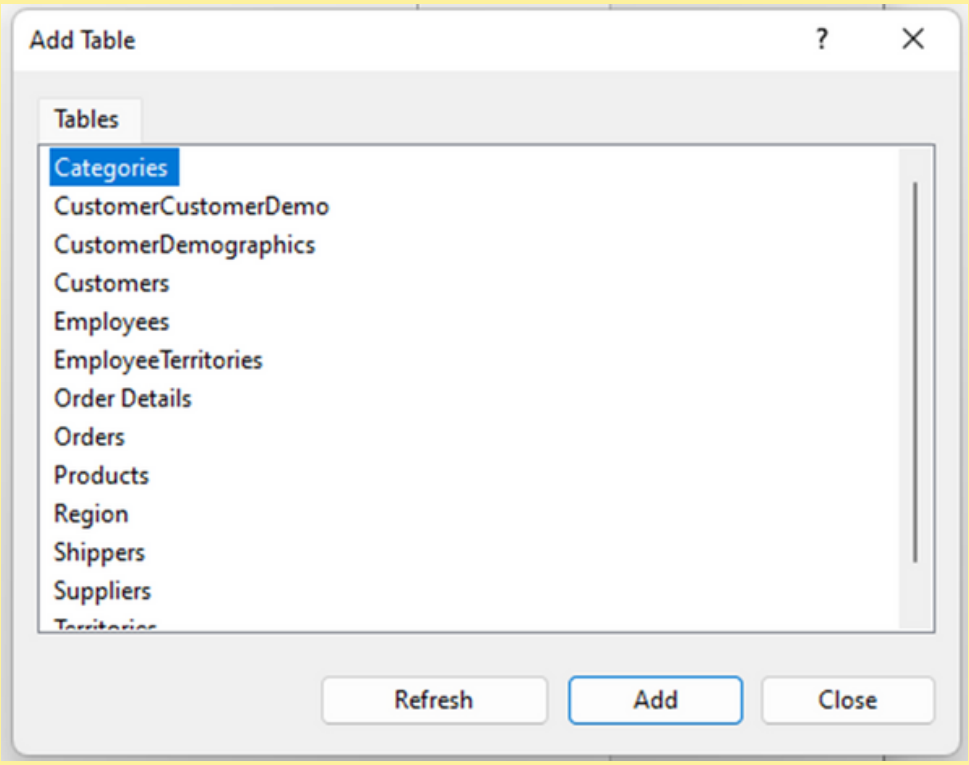
handhikayp
Researcher

TROUBLESHOOTING

9. Akan muncul window baru tentang pemberitahuan pembuatan database baru. Klik Yes.

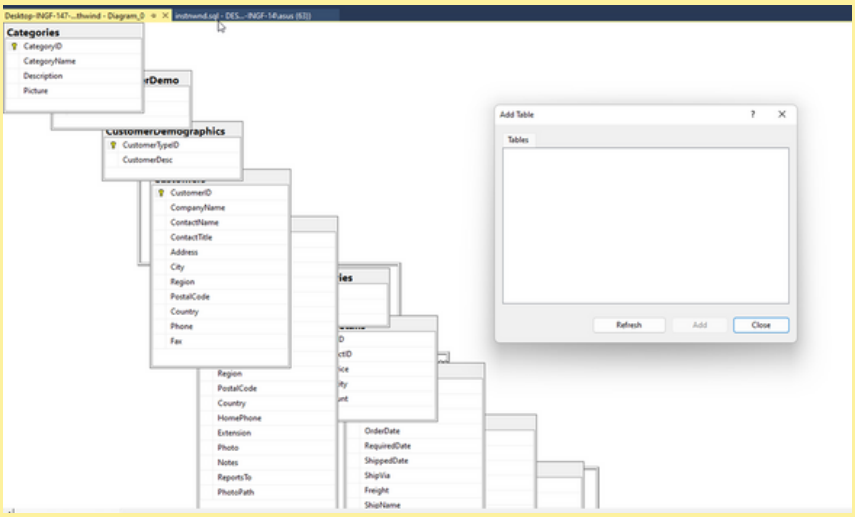


10. Disini kita bisa menentukan table apa saja yang hendak dimasukkan. Pada contoh ini kita bisa memasukkan semua data.

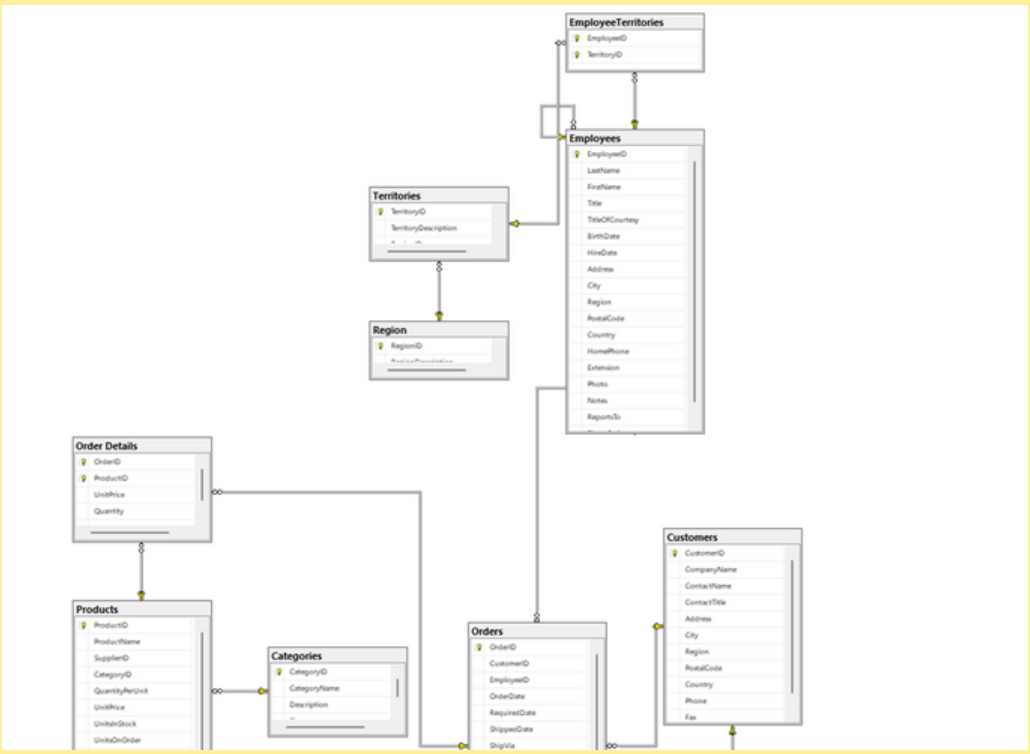


TROUBLESHOOTING

10. Awalnya, semua tabel akan terlihat berantakan sebagai berikut.



11. Tetapi dengan penataan, tabel dapat dirapihkan untuk mengetahui korelasi antar data.



Dari sini proses troubleshooting selesai dan dapat dilanjutkan ke bagian selanjutnya

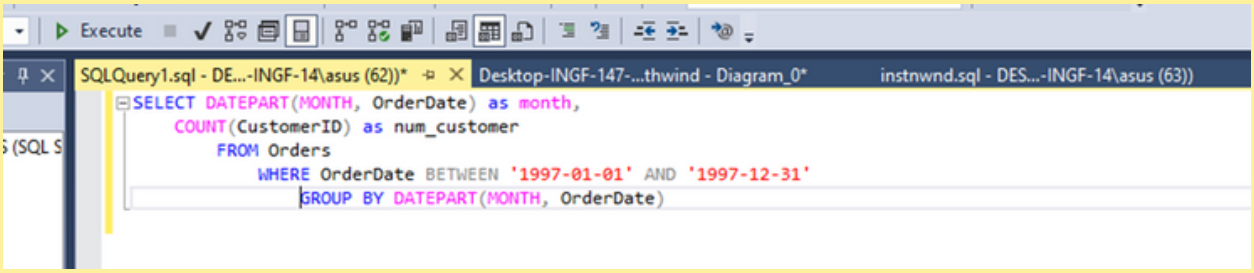
handhikayp
Researcher

Bagian-2

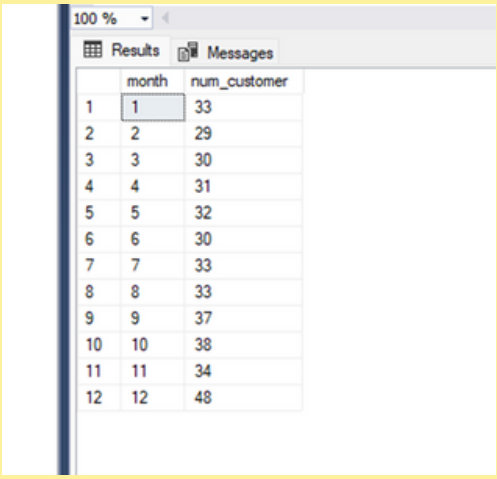
Intermediate Queries

Tulis query untuk mendapatkan jumlah customer tiap bulan yang melakukan order pada tahun 1997.

Input



Output



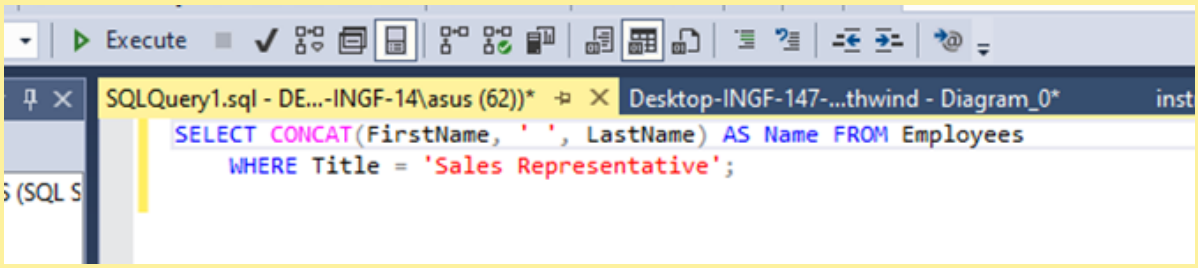
	month	num_customer
1	1	33
2	2	29
3	3	30
4	4	31
5	5	32
6	6	30
7	7	33
8	8	33
9	9	37
10	10	38
11	11	34
12	12	48

INTERMEDIATE QUERIES

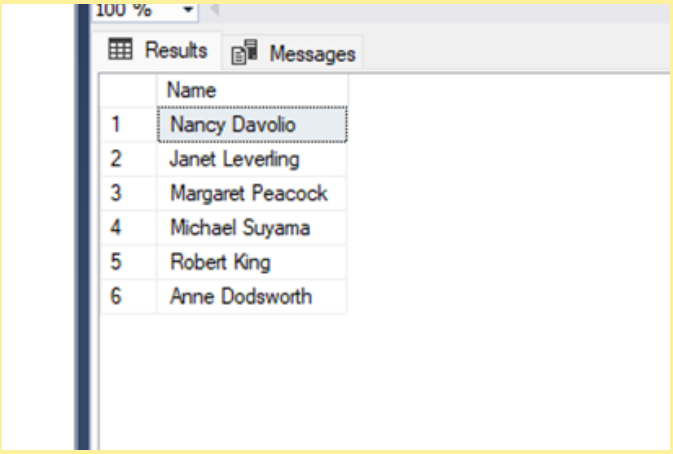
handhikayp
Researcher

Tulis query untuk mendapatkan nama employee yang termasuk Sales Representative.

Input



Output



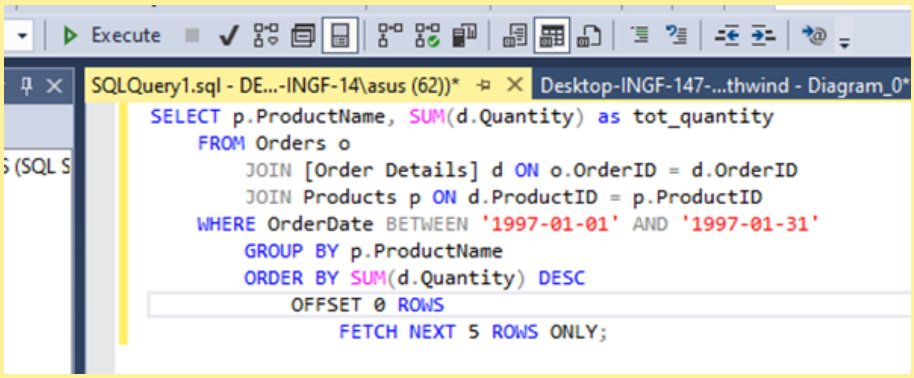
INTERMEDIATE QUERIES

2

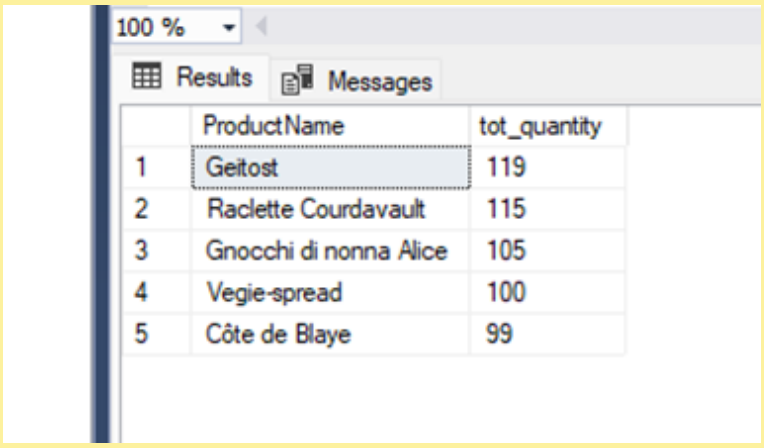
handhikayp
Researcher

Tulis query untuk mendapatkan top 5 nama produk yang quantitynya paling banyak diorder pada bulan Januari 1997.

Input



Output



	ProductName	tot_quantity
1	Geitost	119
2	Raclette Courdavault	115
3	Gnocchi di nonna Alice	105
4	Vegie-spread	100
5	Côte de Blaye	99

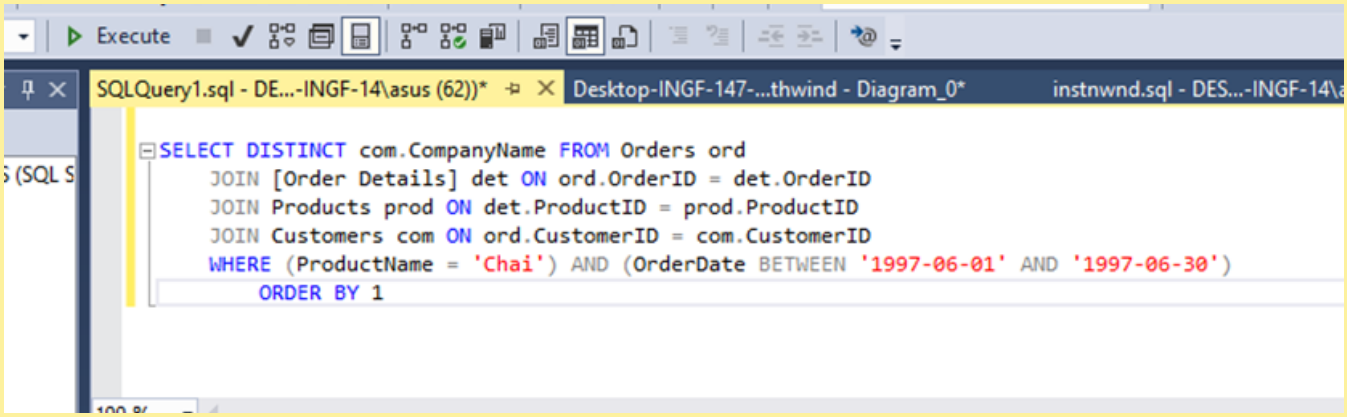
INTERMEDIATE QUERIES

3

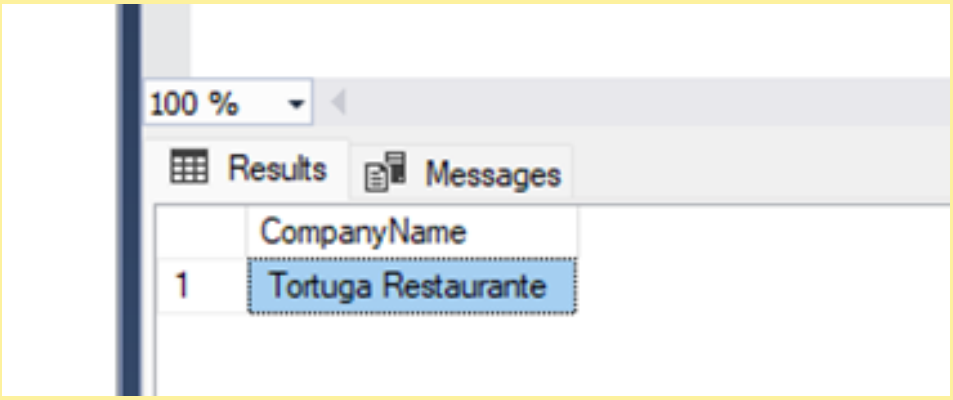
handhikayp
Researcher

Tulis query untuk mendapatkan nama company yang melakukan order Chai pada bulan Juni 1997.

Input



Output



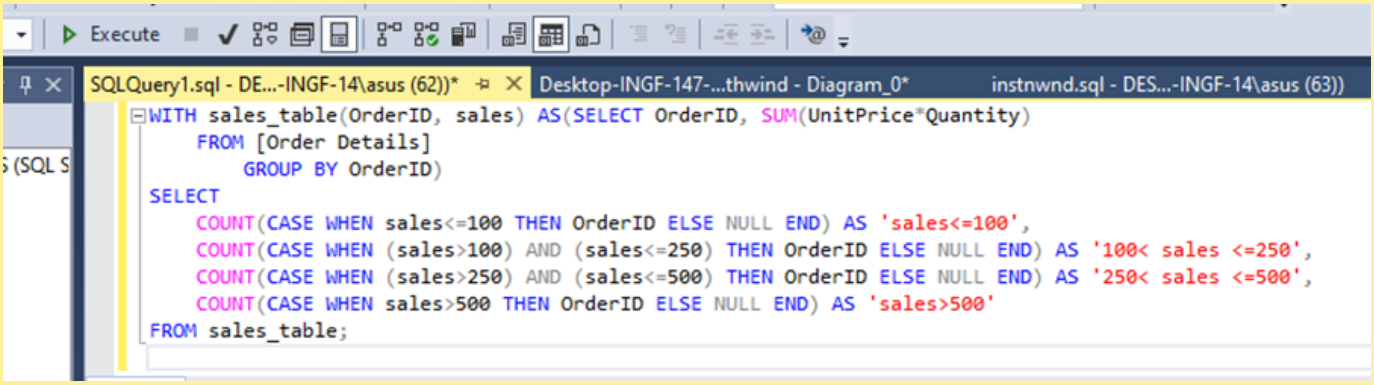
INTERMEDIATE QUERIES

4

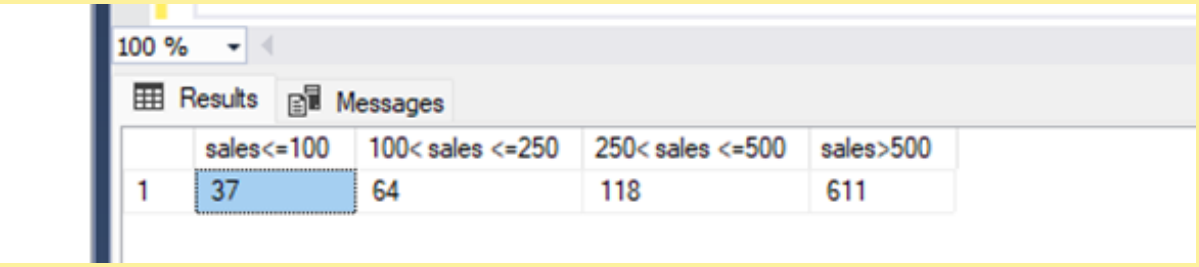
handhikayp
Researcher

Tulis query untuk mendapatkan jumlah OrderID yang pernah melakukan sales (unit_price dikali quantity) <=100, 100<x<=250, 250<x<=500, dan >500.

Input



Output



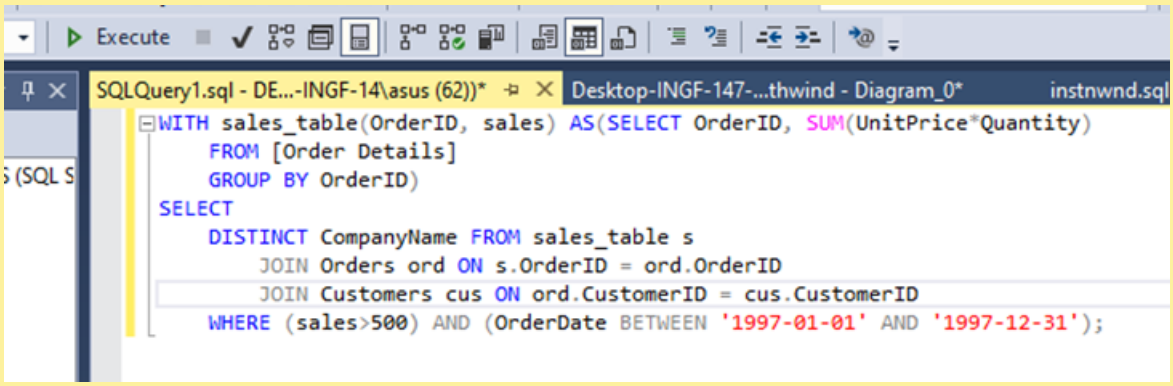
INTERMEDIATE QUERIES

5

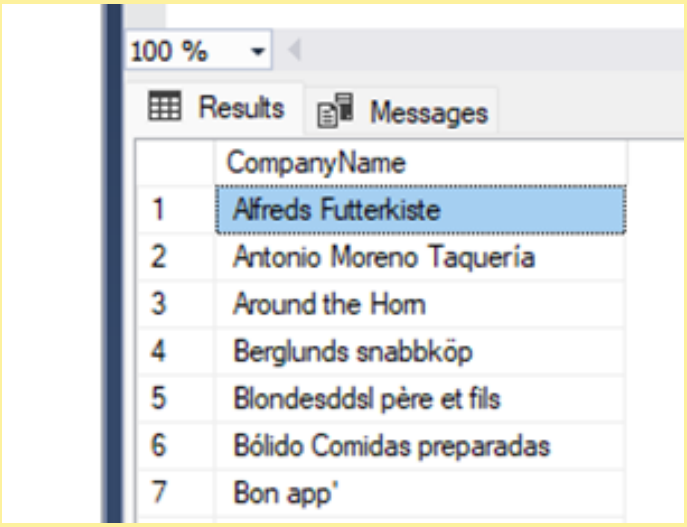
handhikayp
Researcher

Tulis query untuk mendapatkan Company name yang melakukan sales di atas 500 pada tahun 1997.

Input



Output



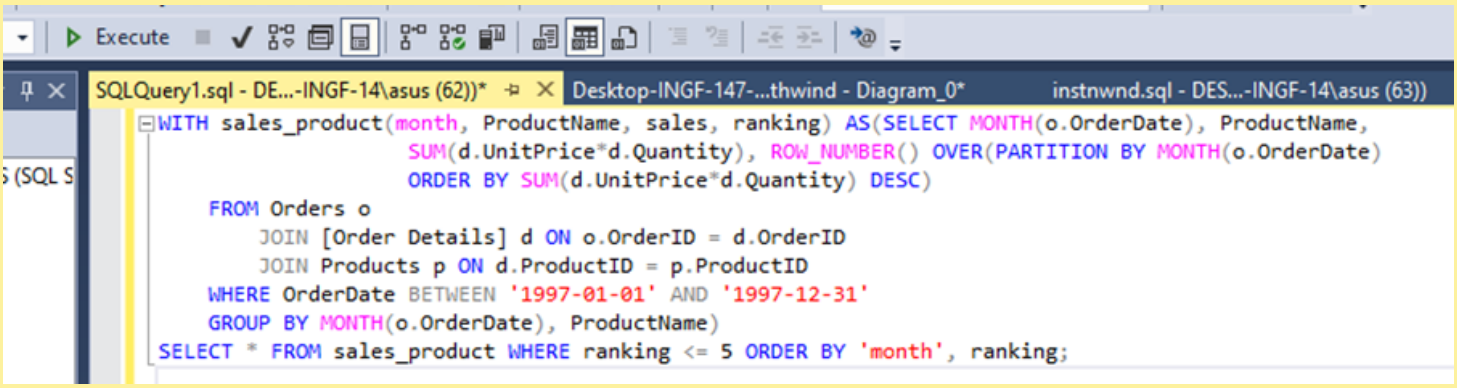
INTERMEDIATE QUERIES

6

handhikayp
Researcher

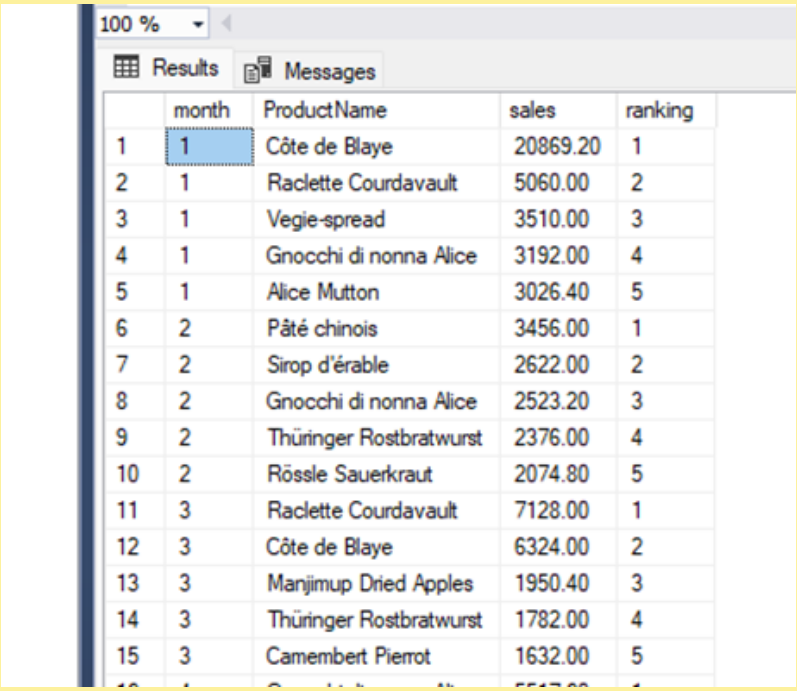
Tulis query untuk mendapatkan nama produk yang merupakan Top 5 sales tertinggi tiap bulan di tahun 1997.

Input



```
WITH sales_product(month, ProductName, sales, ranking) AS(SELECT MONTH(o.OrderDate), ProductName,
                    SUM(d.UnitPrice*d.Quantity), ROW_NUMBER() OVER(PARTITION BY MONTH(o.OrderDate)
                    ORDER BY SUM(d.UnitPrice*d.Quantity) DESC)
    FROM Orders o
        JOIN [Order Details] d ON o.OrderID = d.OrderID
        JOIN Products p ON d.ProductID = p.ProductID
    WHERE OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
    GROUP BY MONTH(o.OrderDate), ProductName)
SELECT * FROM sales_product WHERE ranking <= 5 ORDER BY 'month', ranking;
```

Output



Results | Messages

	month	ProductName	sales	ranking
1	1	Côte de Blaye	20869.20	1
2	1	Raclette Courdavault	5060.00	2
3	1	Vegie-spread	3510.00	3
4	1	Gnocchi di nonna Alice	3192.00	4
5	1	Alice Mutton	3026.40	5
6	2	Pâté chinois	3456.00	1
7	2	Sirop d'érable	2622.00	2
8	2	Gnocchi di nonna Alice	2523.20	3
9	2	Thüringer Rostbratwurst	2376.00	4
10	2	Rössle Sauerkraut	2074.80	5
11	3	Raclette Courdavault	7128.00	1
12	3	Côte de Blaye	6324.00	2
13	3	Manjimup Dried Apples	1950.40	3
14	3	Thüringer Rostbratwurst	1782.00	4
15	3	Camembert Pierrot	1632.00	5

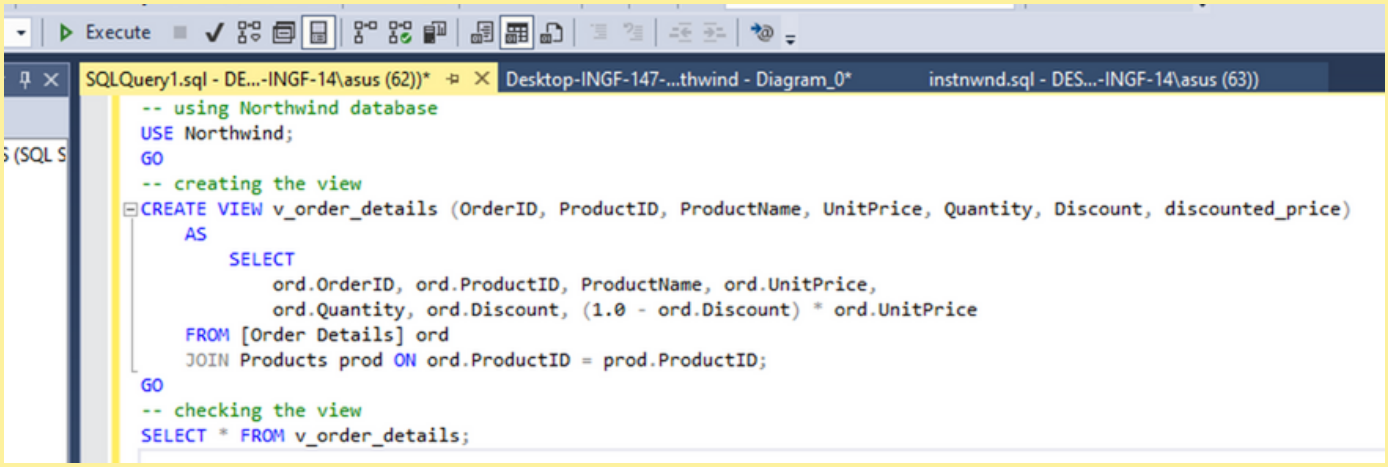
INTERMEDIATE QUERIES

7

handhikayp
Researcher

Buatlah view untuk melihat Order Details yang berisi OrderID, ProductID, ProductName, UnitPrice, Quantity, Discount, Harga setelah diskon.

Input



Output

	OrderID	ProductID	ProductName	UnitPrice	Quantity	Discount	discounted_price
1	10248	11	Queso Cabrales	14.00	12	0	14
2	10248	42	Singaporean Hokkien Fried Mee	9.80	10	0	9.8
3	10248	72	Mozzarella di Giovanni	34.80	5	0	34.8
4	10249	14	Tofu	18.60	9	0	18.6
5	10249	51	Manjimup Dried Apples	42.40	40	0	42.4
6	10250	41	Jack's New England Clam Chowder	7.70	10	0	7.7
7	10250	51	Manjimup Dried Apples	42.40	35	0.15	36.04
8	10250	65	Louisiana Fiery Hot Pepper Sauce	16.80	15	0.15	14.28
9	10251	22	Gustaf's Knäckebröd	16.80	6	0.05	15.96
10	10251	57	Ravioli Angelo	15.60	15	0.05	14.82
11	10251	65	Louisiana Fiery Hot Pepper Sauce	16.80	20	0	16.8
12	10252	20	Sir Rodney's Marmalade	64.80	40	0.05	61.56
13	10252	33	Geitost	2.00	25	0.05	1.9
14	10252	60	Camembert Pierrot	27.20	40	0	27.2
15	10253	31	Gorgonzola Telino	10.00	20	0	10
16	10253	39	Chartreuse verte	14.40	42	0	14.4
17	10253	49	Maxilaku	16.00	40	0	16
18	10254	24	Guaraná Fantástica	3.60	15	0.15	3.06

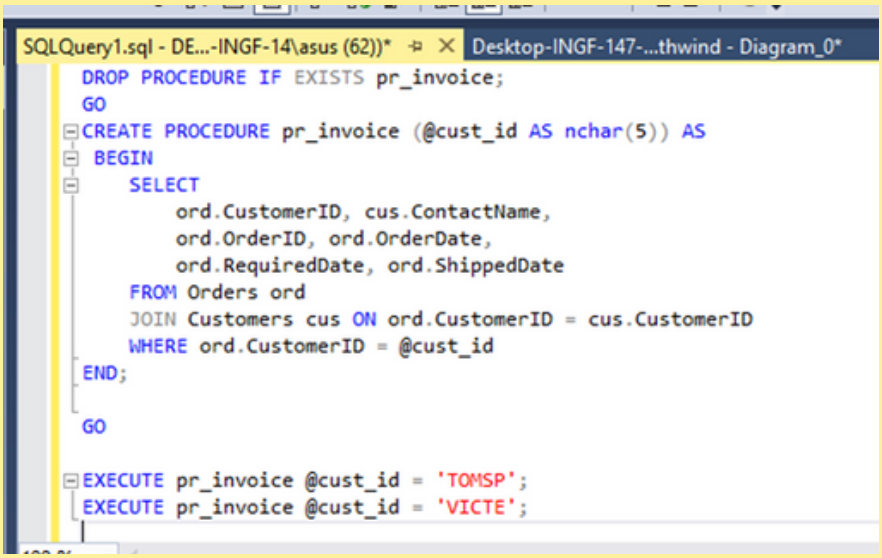
INTERMEDIATE QUERIES

8

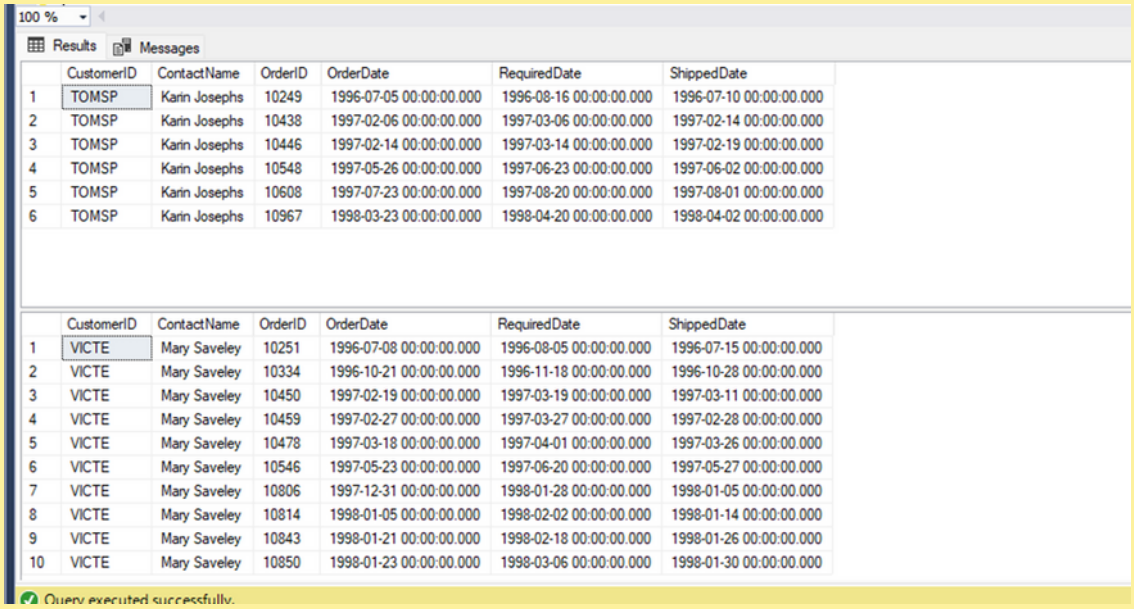
handhikayp
Researcher

Buatlah procedure Invoice untuk memanggil CustomerID, CustomerName, OrderID, OrderDate, RequiredDate, ShippedDate jika terdapat inputan CustomerID tertentu.

Input



Output



100 %

Results | Messages

	CustomerID	ContactName	OrderID	OrderDate	RequiredDate	ShippedDate
1	TOMSP	Karin Josephs	10249	1996-07-05 00:00:00.000	1996-08-16 00:00:00.000	1996-07-10 00:00:00.000
2	TOMSP	Karin Josephs	10438	1997-02-06 00:00:00.000	1997-03-06 00:00:00.000	1997-02-14 00:00:00.000
3	TOMSP	Karin Josephs	10446	1997-02-14 00:00:00.000	1997-03-14 00:00:00.000	1997-02-19 00:00:00.000
4	TOMSP	Karin Josephs	10548	1997-05-26 00:00:00.000	1997-06-23 00:00:00.000	1997-06-02 00:00:00.000
5	TOMSP	Karin Josephs	10608	1997-07-23 00:00:00.000	1997-08-20 00:00:00.000	1997-08-01 00:00:00.000
6	TOMSP	Karin Josephs	10967	1998-03-23 00:00:00.000	1998-04-20 00:00:00.000	1998-04-02 00:00:00.000

	CustomerID	ContactName	OrderID	OrderDate	RequiredDate	ShippedDate
1	VICTE	Mary Saveley	10251	1996-07-08 00:00:00.000	1996-08-05 00:00:00.000	1996-07-15 00:00:00.000
2	VICTE	Mary Saveley	10334	1996-10-21 00:00:00.000	1996-11-18 00:00:00.000	1996-10-28 00:00:00.000
3	VICTE	Mary Saveley	10450	1997-02-19 00:00:00.000	1997-03-19 00:00:00.000	1997-03-11 00:00:00.000
4	VICTE	Mary Saveley	10459	1997-02-27 00:00:00.000	1997-03-27 00:00:00.000	1997-02-28 00:00:00.000
5	VICTE	Mary Saveley	10478	1997-03-18 00:00:00.000	1997-04-01 00:00:00.000	1997-03-26 00:00:00.000
6	VICTE	Mary Saveley	10546	1997-05-23 00:00:00.000	1997-06-20 00:00:00.000	1997-05-27 00:00:00.000
7	VICTE	Mary Saveley	10806	1997-12-31 00:00:00.000	1998-01-28 00:00:00.000	1998-01-05 00:00:00.000
8	VICTE	Mary Saveley	10814	1998-01-05 00:00:00.000	1998-02-02 00:00:00.000	1998-01-14 00:00:00.000
9	VICTE	Mary Saveley	10843	1998-01-21 00:00:00.000	1998-02-18 00:00:00.000	1998-01-26 00:00:00.000
10	VICTE	Mary Saveley	10850	1998-01-23 00:00:00.000	1998-03-06 00:00:00.000	1998-01-30 00:00:00.000

Query executed successfully.

INTERMEDIATE QUERIES

9

handhikayp
Researcher

Bagian-3

Case Study

a. Product Analysis
b. Customer Analysis
c. Cohort Analysis

a. Product Analysis

Berikut adalah source code lengkap dari case study product analysis yang digunakan.

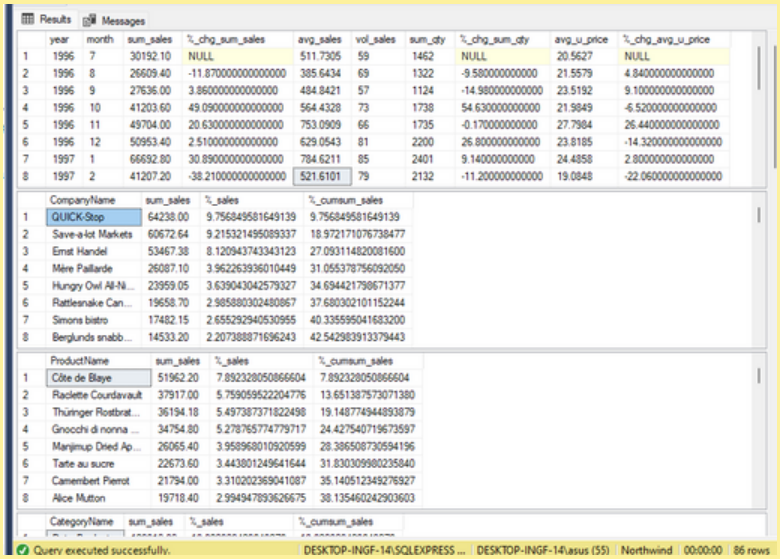
```
-- Product Analysis -- Western
use Northwind;

-- 1. PRODUCT ANALYSIS
-- overall picture of sales
SELECT

 YEAR(ord.OrderDate) AS 'year'
 , MONTH(ord.OrderDate) AS 'month'
 , SUM(od.UnitPrice * od.Quantity) AS 'sum_sales'
 , ROUND(100.0*(SUM(od.UnitPrice * od.Quantity) - LAG(SUM(od.UnitPrice * od.Quantity), 1) OVER(ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate))) /
  LAG(SUM(od.UnitPrice * od.Quantity), 1) OVER(ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate)), 2) AS '%_chg_sum_sales'

 , AVG(od.UnitPrice * od.Quantity) AS 'avg_sales'
 , COUNT(od.UnitPrice * od.Quantity) AS 'vol_sales'

 , SUM(od.Quantity) AS 'sum_qty'
 , ROUND(100.0*(SUM(od.Quantity) - LAG(SUM(od.Quantity), 1) OVER(ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate))) /
  LAG(SUM(od.Quantity), 1) OVER(ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate)), 2) AS '%_chg_sum_qty'

 , AVG(od.UnitPrice) AS 'avg_u_price'
 , ROUND(100.0*(AVG(od.UnitPrice) - LAG(AVG(od.UnitPrice), 1) OVER(ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate))) /
  LAG(AVG(od.UnitPrice), 1) OVER(ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate)), 2) AS '%_chg_avg_u_price'

FROM Orders ord
JOIN [Order Details] od ON ord.OrderID = od.OrderID
GROUP BY YEAR(ord.OrderDate), MONTH(ord.OrderDate)
ORDER BY YEAR(ord.OrderDate), MONTH(ord.OrderDate)
;

-- because only in 1997 that capture the full date -> focusing analysis in 1997 only
-- pareto customer's company on sales in 1997
WITH company_sales AS
 (SELECT
  c.CompanyName
  , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN Customers c ON ord.CustomerID = c.CustomerID
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 WHERE ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY c.CompanyName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM company_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM company_sales)
as '%_cumsum_sales'
FROM company_sales
;

-- pareto product name on sales
WITH product_sales AS
 (SELECT
  prod.ProductName
  , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Products prod ON od.ProductID = prod.ProductID
 WHERE ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY prod.ProductName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM product_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM product_sales)
as '%_cumsum_sales'
FROM product_sales
;

-- pareto category name on sales
WITH category_sales AS
 (SELECT
  cat.CategoryName
  , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Products prod ON od.ProductID = prod.ProductID
 JOIN Categories cat ON prod.CategoryID = cat.CategoryID
 WHERE ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY cat.CategoryName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM category_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM category_sales)
as '%_cumsum_sales'
FROM category_sales
;
```

```
-- pareto sales's teritory on sales
WITH teritory_sales AS
 (SELECT
 t.TerritoryDescription
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Employees e ON e.EmployeeID = ord.EmployeeID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 WHERE ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY t.TerritoryDescription)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM teritory_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM teritory_sales)
as '%_cumsum_sales'
FROM teritory_sales
;

-- pareto sales's region on sales
WITH region_sales AS
 (SELECT
 r.RegionDescription
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Employees e ON e.EmployeeID = ord.EmployeeID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY r.RegionDescription)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM region_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM region_sales) as
'%_cumsum_sales'
FROM region_sales
;

-- pareto territory name on sales in Western region
-- territory in Western with the highest sales
WITH territory_sales AS
 (SELECT
 t.TerritoryDescription
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Employees e ON e.EmployeeID = ord.EmployeeID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE (ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31') AND (RegionDescription = 'Western')
 GROUP BY t.TerritoryDescription)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM territory_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM territory_sales)
as '%_cumsum_sales'
FROM territory_sales
;


-- pareto category name on sales in Western region
-- category name in Western with the highest sales
WITH category_sales AS
 (SELECT
 cat.CategoryName
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Products prod ON od.ProductID = prod.ProductID
 JOIN Categories cat ON prod.CategoryID = cat.CategoryID
 JOIN Employees e ON e.EmployeeID = ord.EmployeeID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE (ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31') AND (RegionDescription = 'Western')
 GROUP BY cat.CategoryName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM category_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM category_sales)
as '%_cumsum_sales'
FROM category_sales
;
```

```
-- pareto product name on sales in Western region
-- product name in Western with the highest sales
WITH product_sales AS
 (SELECT
 prod.ProductName
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Products prod ON od.ProductID = prod.ProductID
 JOIN Categories cat ON prod.CategoryID = cat.CategoryID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE (ord.OrderDate BETWEEN '1997-01-01' AND '1997-12-31') AND (RegionDescription = 'Western')
 GROUP BY prod.ProductName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM product_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM product_sales) as
'%_cumsum_sales'
FROM product_sales
;


-----
-----
-- Analyze the spike phenomena on Nov 97 to Apr 98 period

-- pareto sales's region on sales on Nov 97 to Apr 98 period
WITH region_sales AS
 (SELECT
 r.RegionDescription
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE ord.OrderDate BETWEEN '1997-11-01' AND '1998-04-30'
 GROUP BY r.RegionDescription)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM region_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM region_sales) as
'%_cumsum_sales'
FROM region_sales
;


-- pareto territory name on sales in Western region
-- territory in Western with the highest sales on Nov 97 to Apr 98 period
WITH territory_sales AS
 (SELECT
 t.TerritoryDescription
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Employees e ON e.EmployeeID = ord.EmployeeID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE (ord.OrderDate BETWEEN '1997-11-01' AND '1998-04-30') AND (RegionDescription = 'Western')
 GROUP BY t.TerritoryDescription)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM territory_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM territory_sales) as
'%_cumsum_sales'
FROM territory_sales
;

-- pareto category name on sales in Western region
-- category name in Western with the highest sales on Nov 97 to Apr 98 period
WITH category_sales AS
 (SELECT
 cat.CategoryName
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Products prod ON od.ProductID = prod.ProductID
 JOIN Categories cat ON prod.CategoryID = cat.CategoryID
 JOIN Employees e ON e.EmployeeID = ord.EmployeeID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE (ord.OrderDate BETWEEN '1997-11-01' AND '1998-04-30') AND (RegionDescription = 'Western')
 GROUP BY cat.CategoryName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM category_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM category_sales) as
'%_cumsum_sales'
FROM category_sales
;
```

```
-- pareto category name on sales in Western region
-- product name in Western with the highest sales on Nov 97 to Apr 98 period
WITH product_sales AS
 (SELECT
 prod.ProductName
 , SUM(od.UnitPrice*od.Quantity) as 'sum_sales'
 FROM Orders ord
 JOIN [Order Details] od ON ord.OrderID = od.OrderID
 JOIN Products prod ON od.ProductID = prod.ProductID
 JOIN Categories cat ON prod.CategoryID = cat.CategoryID
 JOIN EmployeeTerritories et ON et.EmployeeID = ord.EmployeeID
 JOIN Territories t ON et.TerritoryID = t.TerritoryID
 JOIN Region r ON r.RegionID = t.RegionID
 WHERE (ord.OrderDate BETWEEN '1997-11-01' AND '1998-04-30') AND (RegionDescription = 'Western')
 GROUP BY prod.ProductName)

SELECT
 *
 , 100.0 * sum_sales / (SELECT SUM(sum_sales) FROM product_sales) AS '%_sales'
 , 100.0 * SUM(sum_sales) OVER(ORDER BY sum_sales DESC ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) / (SELECT SUM(sum_sales) FROM product_sales) as
'%_cumsum_sales'
FROM product_sales
;
```

Apabila di jalankan source code di atas akan memberikan output berupa tabel-tabel yang berkorelasi seperti berikut beberapa dari tabel tersebut.



Results Messages

	year	month	sum_sales	%_chg_sum_sales	avg_sales	vol_sales	sum_qty	%_chg_sum_qty	avg_u_price	%_chg_avg_u_price
1	1996	7	30192.10	NULL	511.7305	59	1462	NULL	20.5627	NULL
2	1996	8	26609.40	-11.870000000000000	385.6434	69	1322	-9.580000000000	21.5579	4.840000000000000
3	1996	9	27636.00	3.860000000000000	484.8421	57	1124	-14.980000000000	23.5192	9.100000000000000
4	1996	10	41203.60	49.090000000000000	564.4328	73	1738	54.630000000000	21.9849	-6.520000000000000
5	1996	11	49704.00	20.630000000000000	753.0909	66	1735	-0.170000000000	27.7984	26.440000000000000
6	1996	12	50953.40	2.510000000000000	629.0543	81	2200	26.800000000000	23.8185	-14.320000000000000
7	1997	1	66692.80	30.890000000000000	784.6211	85	2401	9.140000000000	24.4858	2.800000000000000
8	1997	2	41207.20	-38.210000000000000	521.6101	79	2132	-11.200000000000	19.0848	-22.060000000000000

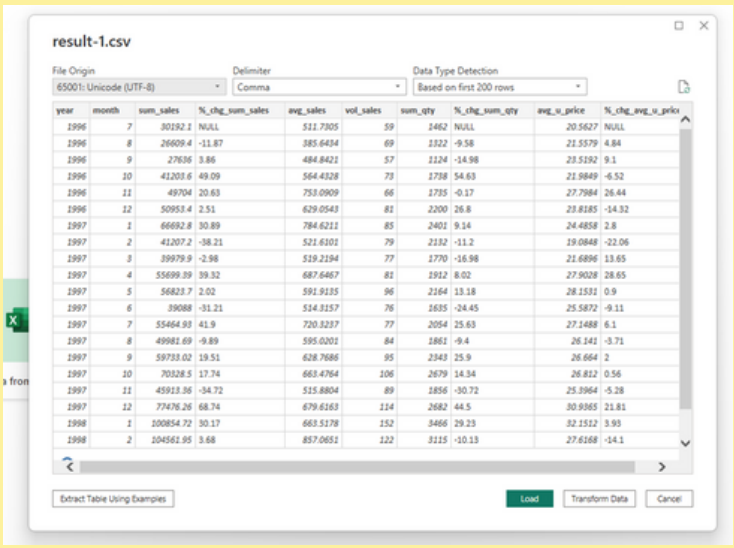
	CompanyName	sum_sales	%_sales	%_cumsum_sales
1	QUICK-Stop	64238.00	9.756849581649139	9.756849581649139
2	Save-a-lot Markets	60672.64	9.215321495089337	18.972171076738477
3	Ernst Handel	53467.38	8.120943743943123	27.093114820081600
4	Mère Paillarde	26087.10	3.962263936010449	31.055378756092050
5	Hungry Owl All-Ni...	23959.05	3.639043042579327	34.694421798671377
6	Rattlesnake Can...	19658.70	2.985880302480867	37.680302101152244
7	Simons bistro	17482.15	2.655292940530955	40.335595041683200
8	Berglunds snabb...	14533.20	2.207388871696243	42.542983913379443

	ProductName	sum_sales	%_sales	%_cumsum_sales
1	Côte de Blaye	51962.20	7.892328050866604	7.892328050866604
2	Raclette Courdavault	37917.00	5.759059522204776	13.651387573071380
3	Thüringer Rostbrat...	36194.18	5.497387371822498	19.148774944893879
4	Gnocchi di nonna ...	34754.80	5.278765774779717	24.427540719673597
5	Manjimup Dried Ap...	26065.40	3.958968010920599	28.386508730594196
6	Tarte au sucre	22673.60	3.443801249641644	31.830309980235840
7	Camembert Pierrot	21794.00	3.310202369041087	35.140512349276927
8	Alice Mutton	19718.40	2.994947893626675	38.135460242903603

	CategoryName	sum_sales	%_sales	%_cumsum_sales

Query executed successfully. DESKTOP-INGF-14\SQLEXPRESS ... DESKTOP-INGF-14\asus (55) Northwind 00:00:00 86 rows

Keseluruhan tabel yang dihasilkan sebenarnya berkorelasi satu sama lain, tetapi untuk mempermudah analisis, hanya beberapa tabel saja yang digunakan, salah satunya adalah tabel pertama. Tabel tersebut kemudian dijadikan inputan pada Power BI untuk memperoleh analisis lanjutan.



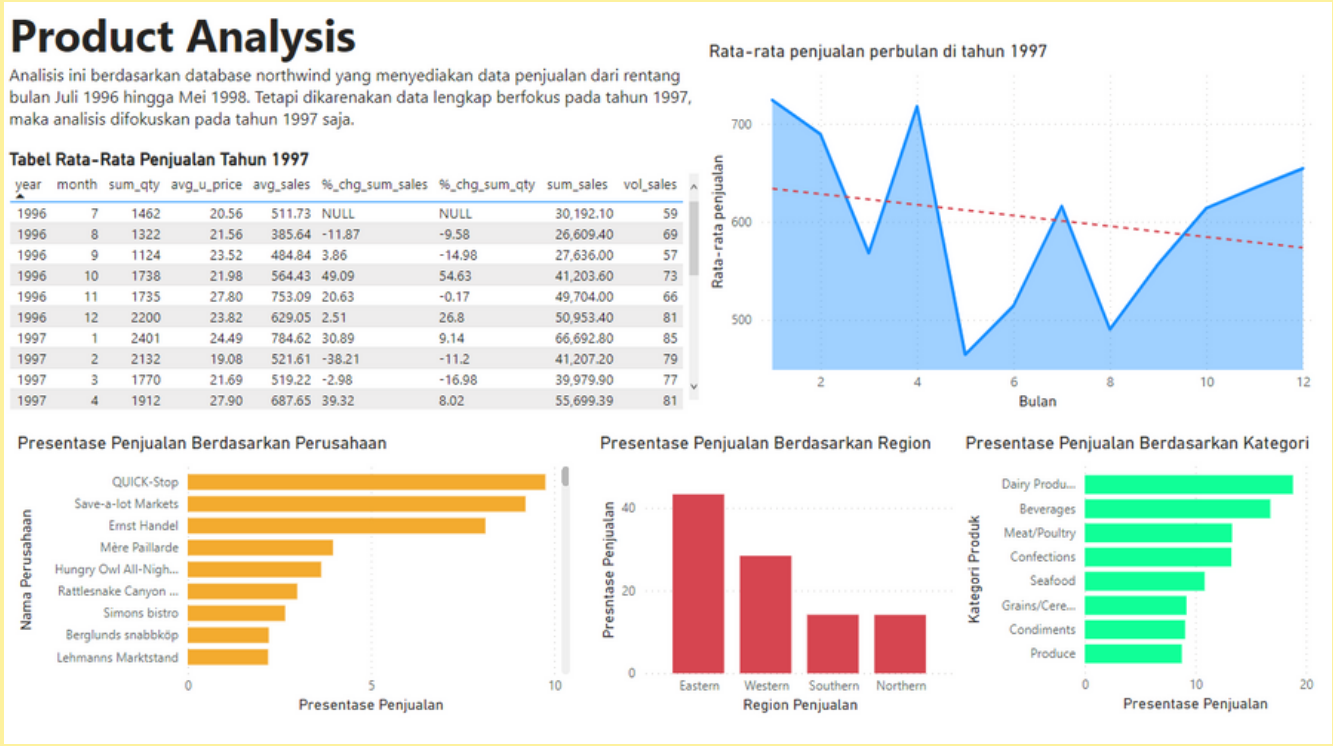
result-1.csv

File Origin: 65001: Unicode (UTF-8) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

year	month	sum_sales	%_chg_sum_sales	avg_sales	vol_sales	sum_qty	%_chg_sum_qty	avg_u_price	%_chg_avg_u_price
1996	7	30192.1	NULL	511.7305	59	1462	NULL	20.5627	NULL
1996	8	26609.4	-11.87	385.6434	69	1322	-9.58	21.5579	4.84
1996	9	27636	3.86	484.8421	57	1124	-14.98	23.5192	9.1
1996	10	41203.6	49.09	564.4328	73	1738	54.63	21.9849	-6.52
1996	11	49704	20.63	753.0909	66	1735	-0.17	27.7984	26.44
1996	12	50953.4	2.51	629.0543	81	2200	26.8	23.8185	-14.32
1997	1	66692.8	30.89	784.6211	85	2401	9.14	24.4858	2.8
1997	2	41207.2	-38.21	521.6101	79	2132	-11.2	19.0848	-22.06
1997	3	39979.9	-2.98	519.2194	77	1770	-16.98	21.6896	13.65
1997	4	55699.39	39.32	687.6467	81	1912	8.02	27.9028	28.65
1997	5	56823.7	2.02	591.9135	96	2164	13.18	28.1531	0.9
1997	6	39088	-31.21	514.3157	76	1635	-24.45	25.5872	-9.11
1997	7	55464.93	41.9	720.3237	77	2054	25.63	27.1488	6.1
1997	8	49981.69	-9.89	595.0201	84	1861	-9.4	26.141	-3.71
1997	9	59733.02	19.51	628.7686	95	2343	25.9	26.664	2
1997	10	70328.5	17.74	663.4764	106	2679	14.34	26.812	0.56
1997	11	45913.36	-34.72	515.8804	89	1856	-30.72	25.3964	-5.28
1997	12	77476.26	68.74	679.6163	114	2682	44.5	30.9365	21.81
1998	1	100854.72	30.17	663.5178	152	3466	29.23	32.1512	3.93
1998	2	104561.95	3.68	857.0651	122	3115	-10.13	27.6168	-14.1

Extract Table Using Examples | Load | Transform Data | Cancel

Berikut adalah hasil tampilan grafik dari penggunaan aplikasi Power BI yang telah dilakukan.



Gambar tersebut menjelaskan beberapa hal seperti tabel rata-rata penjualan yang difokuskan pada tahun 1997. Terdapat juga garis trend-line yang menunjukkan trend ke bawah selama tahun 1997. Selain itu juga terdapat tampilan perusahaan dengan rata-rata penjualan tertinggi yaitu QUICK-Stop,. Juga terdapat region penjualan, dimana dari analisis Power BI dapat diketahui, bahwa region eastern memiliki rata-rata penjualan tertinggi. Terakhir, berdasarkan kategori, kategori Dairy Product, menjadi kategori paling banyak diperjualbelikan pada tahun 1997.

Pemahaman peneliti sangatlah minim terkait pemanfaatan SQL dan Power BI sehingga analisis yang dapat dilakukan hanya ini saja. Gambar lebih jelas akan ditampilkan nantinya atau melalui dashboard Power BI.

b. Customer Analysis

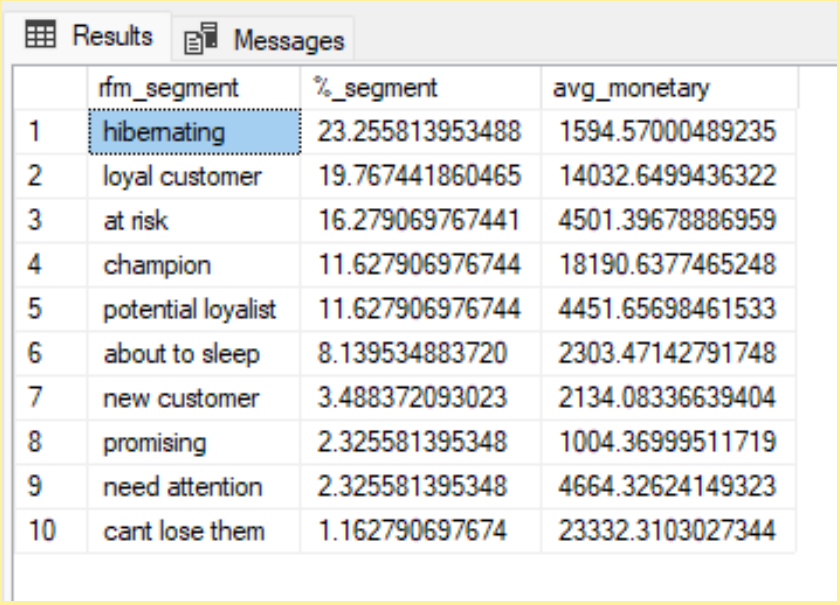
Berikut adalah source code lengkap dari case study customer analysis yang digunakan.

```
-- Customer

USE Northwind;

-- 1. rfm score for each customer id
WITH
base_rfm_score AS
 (SELECT
  o.CustomerID
  , COUNT(DISTINCT o.OrderID) as 'frequency_value'
  , DATEDIFF(DAY, MAX(o.OrderDate), '1998-01-07') as 'recency_value'
  , SUM((1-od.Discount)*(od.UnitPrice*od.Quantity)) as 'monetary_value'
  , NTILE(5) OVER(ORDER BY COUNT(DISTINCT o.OrderID) ASC) as 'frequency_score'
  , NTILE(5) OVER(ORDER BY DATEDIFF(DAY, MAX(o.OrderDate), '1998-01-07') DESC) as 'recency_score'
  , NTILE(5) OVER(ORDER BY SUM((1-od.Discount)*(od.UnitPrice*od.Quantity)) ASC) as
'monetary_score'
 FROM Orders o
 JOIN [Order Details] od ON o.OrderID = od.OrderID
 WHERE o.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY o.CustomerID),

-- 2. from the rfm score, group each customer to rfm segment
rfm_segment_table AS
 (SELECT
  *
  , (monetary_score + recency_score + frequency_score)/3 as 'rfm_score'
  , CASE WHEN (recency_score = 5) AND ((frequency_score=5) OR (frequency_score=4)) THEN
'champion'
  WHEN ((recency_score = 3) OR (recency_score = 4)) AND ((frequency_score=5) OR
(frequency_score=4)) THEN 'loyal customer'
  WHEN ((recency_score = 1) OR (recency_score = 2)) AND (frequency_score=5) THEN 'cant lose them'
  WHEN ((recency_score = 5) OR (recency_score = 4)) AND ((frequency_score=3) OR
(frequency_score=2)) THEN 'potential loyalist'
  WHEN (recency_score = 3) AND (frequency_score=3) THEN 'need attention'
  WHEN ((recency_score = 1) OR (recency_score = 2)) AND ((frequency_score=3) OR
(frequency_score=4)) THEN 'at risk'
  WHEN (recency_score = 5) AND (frequency_score=1) THEN 'new customer'
  WHEN (recency_score = 4) AND (frequency_score=1) THEN 'promising'
  WHEN (recency_score = 3) AND ((frequency_score=1) OR (frequency_score=2)) THEN 'about to sleep'
  ELSE 'hibernating' END AS 'rfm_segment'
 FROM base_rfm_score)

-- 3. for each customer segment, calcuate its proportion & avg monetary value
SELECT
 rfm_segment
 , 100.0*COUNT(CustomerID)/(SELECT COUNT(*) FROM rfm_segment_table) AS '%_segment'
 , AVG(monetary_value) AS 'avg_monetary'
FROM rfm_segment_table
GROUP BY rfm_segment
ORDER BY 100.0*COUNT(CustomerID)/(SELECT COUNT(*) FROM rfm_segment_table) DESC
;
```

Setelah dijalankan, hasil dari source code di atas adalah berupa analisis kustomer berdasarkan nilai segmentasi rfm. Sudah banyak sumber yang menjelaskan mengenai teknik analisis ini. Salah satunya dari situs *Barilliance*, disebutkan bahwa RFM analysis adalah teknik segmentasi perilaku pelanggan yang berbasiskan data, singkatan dari Recency, Frequency, Monetary value.

Ide dari analisis ini yaitu dengan mengelompokkan pelanggan berdasarkan kapan pembelian terakhir mereka, seberapa sering pembelian di masa lalunya, serta berapa banyak yang dihabiskan secara keseluruhan. Ketiga ukuran ini telah terbukti menjadi prediktor efektif dalam proses penawaran pemasaran.

Dari penjelasan tersebut, berikut adalah output yang dihasilkan setelah source code dijalankan.


Results Messages

	rfm_segment	%_segment	avg_monetary
1	hibernating	23.255813953488	1594.57000489235
2	loyal customer	19.767441860465	14032.6499436322
3	at risk	16.279069767441	4501.39678886959
4	champion	11.627906976744	18190.6377465248
5	potential loyalist	11.627906976744	4451.65698461533
6	about to sleep	8.139534883720	2303.47142791748
7	new customer	3.488372093023	2134.08336639404
8	promising	2.325581395348	1004.36999511719
9	need attention	2.325581395348	4664.32624149323
10	cant lose them	1.162790697674	23332.3103027344

Gambar pada halaman berikutnya berisikan analisis menggunakan Power BI dari data yang dihasilkan dari rfm segmentation yang dihasilkan sebelumnya. Pada data tersebut terdapat tiga jenis visualisasi. Pertama adalah tabel, yang merupakan output dari SMSS 19. Kemudian terdapat dua buah Treemap yang mengambarkan persebaran serta presentase dari data tersebut berdasarkan keseluruhan segmentasi.

Apabila dilihat dari presentase segmen, presentase terbesar dimiliki oleh pelanggan dengan sifat 'hibernating' dengan presentase sebesar 23.26% dan yang paling kecil adalah 'cant lose them' dengan presentase 1.16%. Disisi sebaliknya dari segi rata-rata monetary. nilai dari 'cant lose them' merupakan nilai terbesar dengan 23.33K. Sedangkan nilai terkecil adalah promising dengan hanya 1.004K. Menggunakan tools Power BI, kita dapat melihat satu persatu data dengan lebih mudah.

Customer Analysis - RFM Segmentation

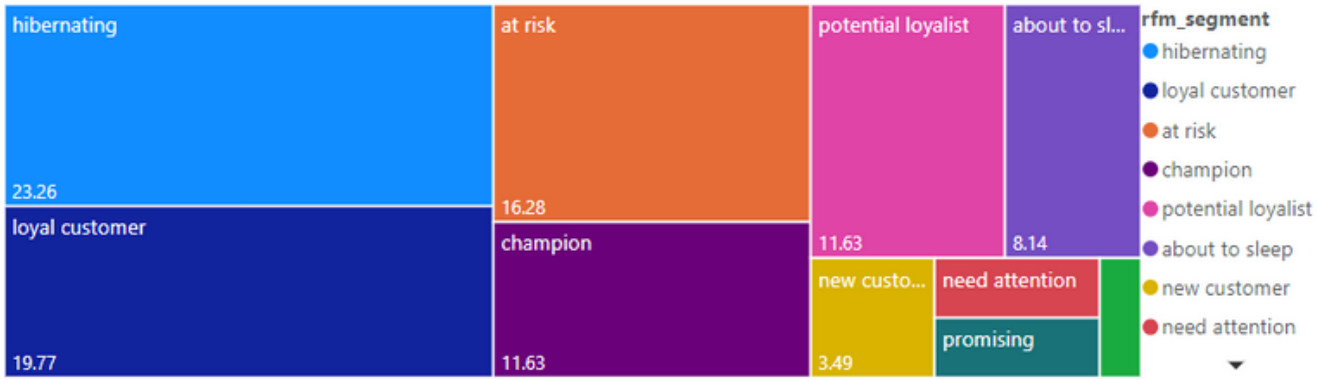
Analisis berfokus pada perlakuan segmenetasi pada data pelanggan. RFM analysis adalah teknik segmentasi perilaku pelanggan yang berbasiskan data, singkatan dari Recency, Frequency, Monetary value. Ide dari analisis ini yaitu dengan mengelompokkan pelanggan berdasarkan kapan pembelian terakhir mereka, seberapa sering pembelian di masa lalunya, serta berapa banyak yang dihabiskan secara keseluruhan.

Tabel RFM Segmentation

rfm_segment	%_segment	avg_monetary
hibernating	23.26	1,594.57
loyal customer	19.77	14,032.65
at risk	16.28	4,501.40
potential loyalist	11.63	4,451.66
champion	11.63	18,190.64
about to sleep	8.14	2,303.47
new customer	3.49	2,134.08
promising	2.33	1,004.37
need attention	2.33	4,664.33
cant lose them	1.16	23,332.31

Data pada halaman ini berisikan tabel dari RFM segmentation yang dihasilkan dari menjalankan program SQL yang dibuat. Gambaran treemap disebelah kanan menunjukkan presentase persebaran data dari masing-masing segmentation pelanggan.

Treemap Segmentation Presentation



Treemap Average Monetary

c. Cohort Analysis

Berikut adalah source code lengkap dari case study cohort analysis yang digunakan.

```
-- Cohort Analysis - User Retention (monthly) - 1997

USE Northwind;

WITH
-- 1. for each customer id, find their first time order (month)
first_buy AS(
 SELECT
  o.CustomerID
  , DATEPART(MONTH, MIN(o.OrderDate)) first_time_buy
 FROM Orders o
 WHERE o.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'
 GROUP BY o.CustomerID),

-- 2. for each customer id, find all of their time order (month)
next_purchase AS(
 SELECT
  o.CustomerID
  , DATEPART(MONTH, o.OrderDate) - first_time_buy AS buy_interval
 FROM Orders o
 JOIN first_buy f ON o.CustomerID = f.CustomerID
 WHERE o.OrderDate BETWEEN '1997-01-01' AND '1997-12-31'),

-- 3. for each first time buy (month), calculate the number of total distinct customer
initial_user AS(
 SELECT
  first_time_buy
  , COUNT(DISTINCT CustomerID) AS users
 FROM first_buy
 GROUP BY first_time_buy),

-- 4. calculate the retention for each first time buy (month) & for each buy interval (month)
retention AS(
 SELECT
  f.first_time_buy
  , buy_interval
  , COUNT(DISTINCT n.CustomerID) AS users_transacting
 FROM first_buy f
 JOIN next_purchase n ON f.CustomerID = n.CustomerID
 WHERE buy_interval IS NOT NULL
 GROUP BY f.first_time_buy, buy_interval)

-- 5. put it all together, convert the retention into percentage
SELECT
 r.first_time_buy,
 i.users ,
 r.buy_interval,
 r.users_transacting,
 100.0*r.users_transacting/i.users AS '%_user_transacting'
FROM retention r
LEFT JOIN initial_user i ON r.first_time_buy = i.first_time_buy
ORDER BY r.first_time_buy, r.buy_interval
;
```

Setelah dijalankan, hasil dari source code di atas adalah berupa analisis cohort, analisis ini merupakan analisis yang mengambil data dari perilaku pelanggan dari berbagai platform (E-Commerce, web, atau game). Analisis ini memecah pelanggan ke dalam beberapa grup supaya dapat dilakukan analisis lanjutan.

Analisis cohort dapat dijadikan ukuran keterlibatan pengguna dari waktu ke waktu. Melalui analisis ini dapat dilakukan pengukuran keterlibatan pengguna, apakah bersifat sementara karena adanya event tertentu atau sudah stabil.

Dari penjelasan tersebut, berikut adalah output yang dihasilkan setelah source code dijalankan. Data tersebut terdiri dari lima buah kolom, antara lain first_time_buy, users, buy_interval, user_transacting, %_user_transacting. Data ini kemudian dijadikan inputan Power BI untuk diperoleh insight yang dapat idimanfaatkan kedepannya.

	first_time_buy	users	buy_interval	users_transacting	%_user_transacting
1	1	27	0	27	100.000000000000
2	1	27	1	6	22.222222222222
3	1	27	2	6	22.222222222222
4	1	27	3	8	29.629629629629
5	1	27	4	7	25.925925925925
6	1	27	5	10	37.037037037037
7	1	27	6	10	37.037037037037
8	1	27	7	13	48.148148148148
9	1	27	8	11	40.740740740740
10	1	27	9	10	37.037037037037
11	1	27	10	14	51.851851851851
12	1	27	11	14	51.851851851851
13	2	15	0	15	100.000000000000
14	2	15	1	6	40.000000000000
15	2	15	2	3	20.000000000000
16	2	15	3	6	40.000000000000
17	2	15	4	6	40.000000000000
18	2	15	5	3	20.000000000000
19	2	15	6	5	33.333333333333
20	2	15	7	4	26.666666666666

Hampir serupa dengan sebelumnya, halaman berikutnya memuat analisis menggunakan Power BI dari data cohort analysis. Visualisasi yang digunakan pada cohort ini yaitu menggunakan tabel, heatmap, dan donut chart. Apabila dilihat terdapat tiga jenis utama, yaitu interval pembelian, pembelian pertama, dan presentase transaksi yang membagi pelanggan ke dalam delapan kluster. Dari segi interval pembelian, pengguna dengan kluster 27 memiliki bagian terbesar, kemudian dari segi pembelian pertama, kluster 3 memiliki bagian paling besar. Terakhir dari persentase transaksi, kluster 3 juga paling tinggi diikuti oleh kluster 27. Hal ini menunjukkan kluster 3 dan 27 merupakan pelanggan yang aktif berdasarkan data northwind.

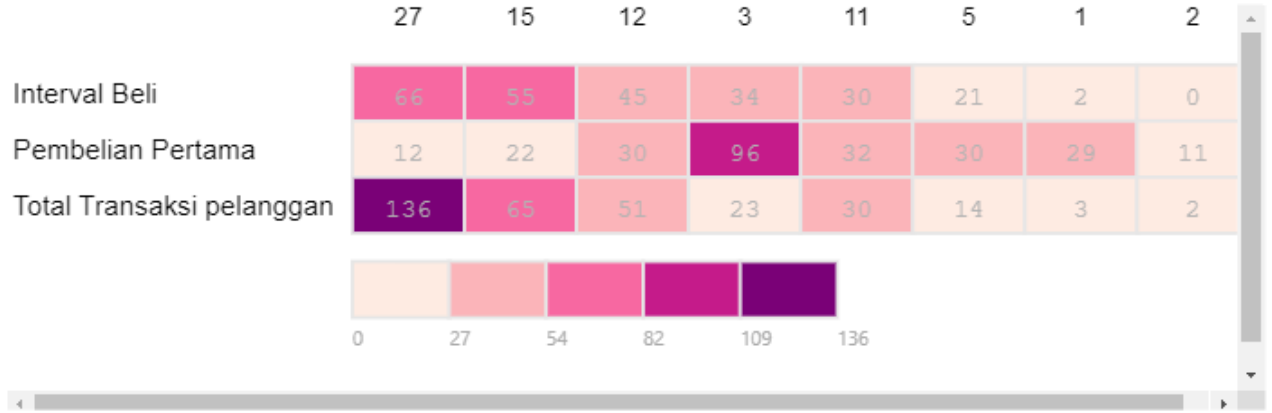
Cohort Analysis

Cohort Analysis merupakan analisis yang mengambil data dari perilaku pelanggan dari berbagai platform (E-Commerce, web, atau game). Analisis ini memecah pelanggan ke dalam beberapa grup supaya dapat dilakukan analisis lanjutan.

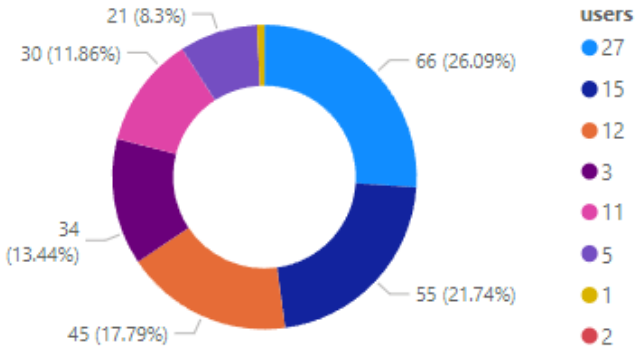
Tabel Output Analysis Cohort

users	first_time_buy	buy_interval	%_user_transacting	users_transacting
1	9	0	100.00	1
1	10	0	100.00	1
1	10	2	100.00	1
2	11	0	100.00	2
3	6	0	100.00	3
3	6	4	66.67	2
3	6	5	33.33	1
3	6	6	33.33	1
3	7	0	100.00	3
3	7	2	66.67	2
3	7	3	33.33	1
3	7	5	33.33	1

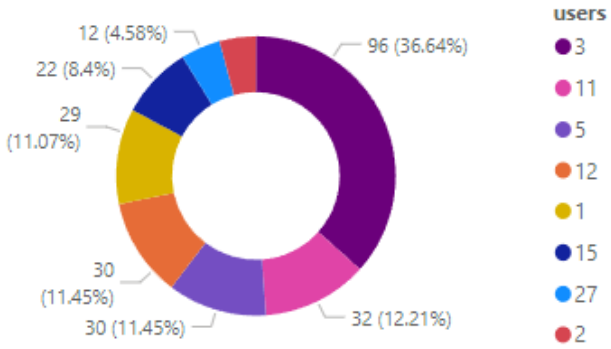
Heatmap klasifikasi pelanggan terhadap pembelian pertama, interval beli, dan transaksi



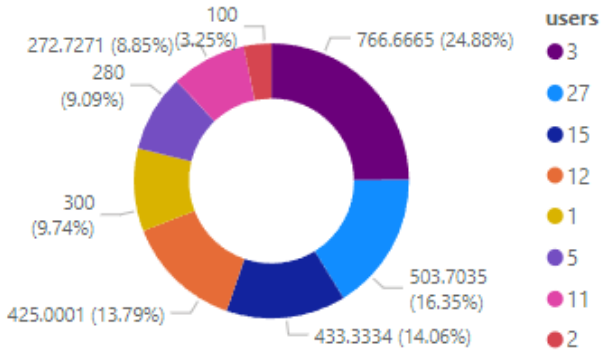
Interval Pembelian Berdasarkan Kluster Pengguna



Pembelian Pertama Berdasarkan Kluster Pengguna Pengguna



Presentase Transaksi Berdasarkan Kluster Pengguna



SEKIAN

handhikayp

Researcher