Email:Radhika.chandra061@gmail.com

LinkedIn: Radhika G. | LinkedIn

Product Performance Analysis using SQL and Power BI

Radhika Gonnabattula

Product performance Analysis using SQL and Power BI

Author: Radhika G.

Tools & Technologies Used:

- Microsoft SQL Server
- Power BI
- DAX
- CSV/Excel
- Data Modelling (Star Schema)

Date: July 2025

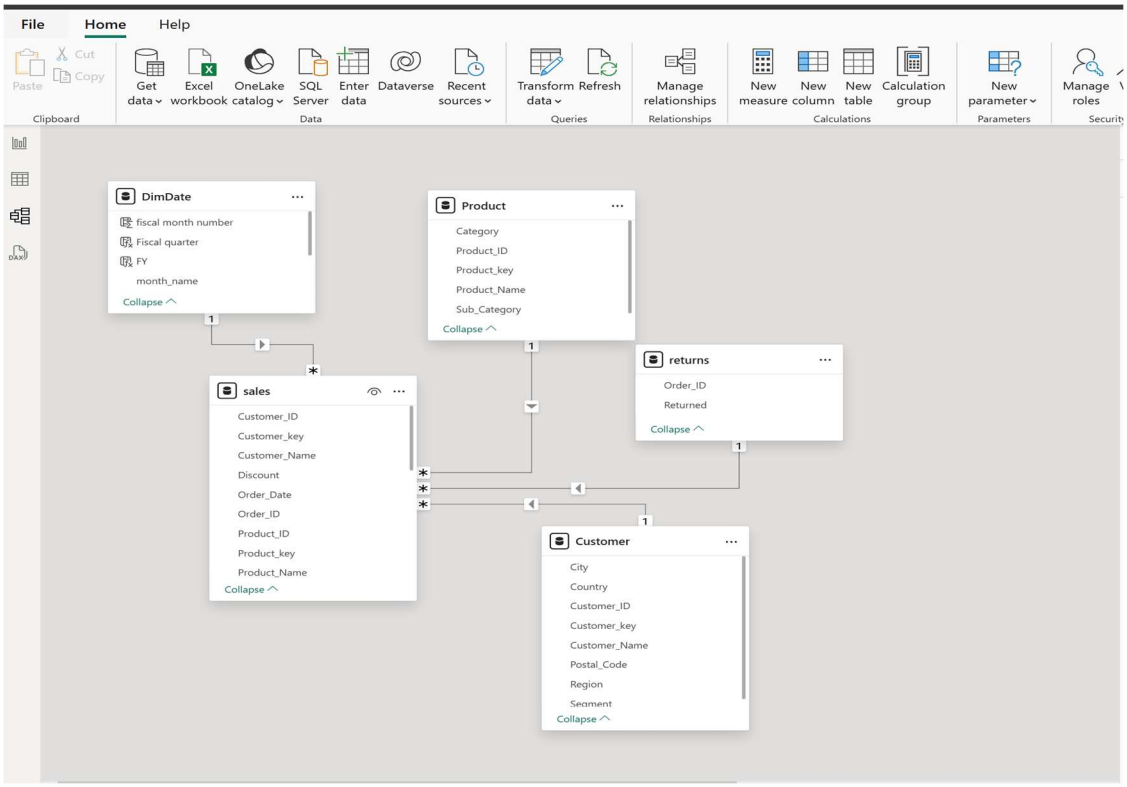
Project Overview

This project involved analysing 4 years of Superstore sales data to identify high-performing products, categories, and trends. SQL was used for data extraction and logic, and Power BI was used to create an interactive dashboard with dynamic KPIs, YoY Growth, and business insights.

Dataset Description

- **Source**: Sample Superstore Dataset
- **Tables Used**: Sales, Products, Returns, Region, Date (dim table)
- **Fields Analysed**: Sales, Profit, Category, Sub-Category, Product_ID, Product_name, Region, Customer_ID, Order Date

Here is how it is modelled:



SQL queries and output

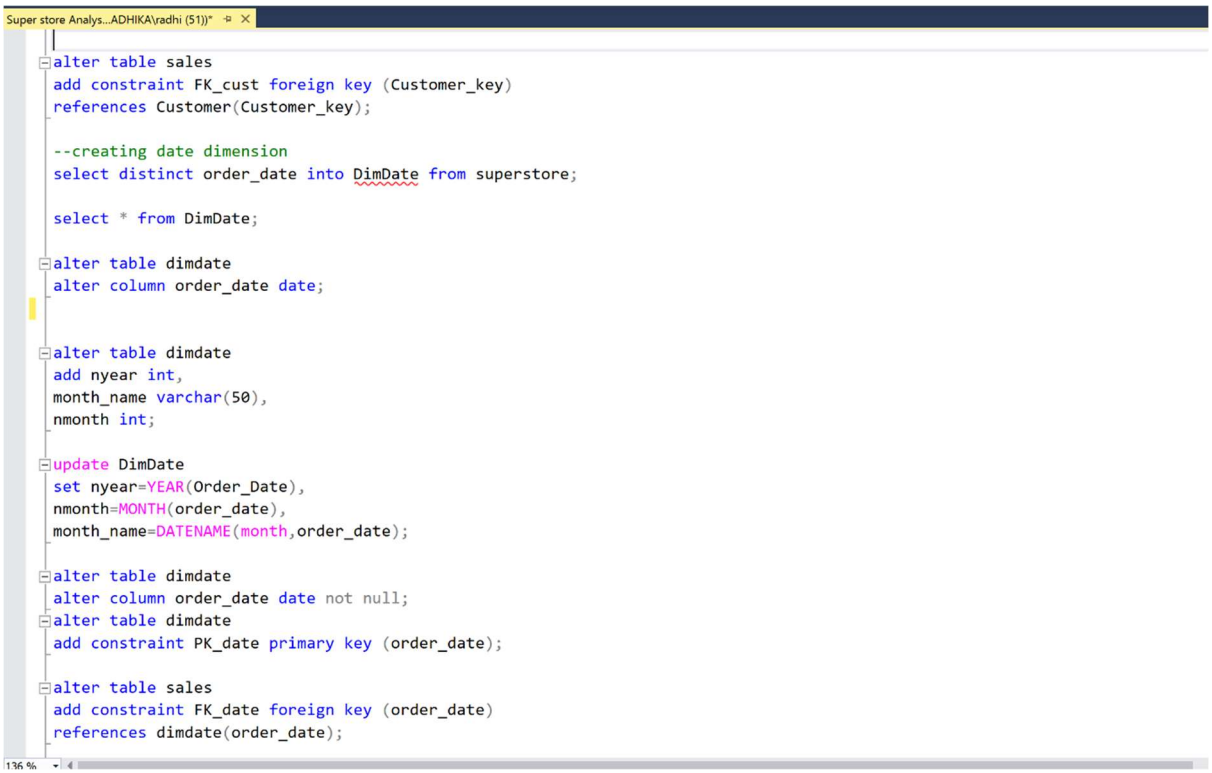
I have imported super store flat file in superstore database. I have created different tables like sales, products, customers and dim date from existing superstore flat file.

```
use superstore;
--creating product table
select distinct Product_ID,Product_Name,Category,Sub_Category
into Product
from superstore;

alter table product
add Product_key int identity(1,1);
select * from Product;
alter table product
add constraint PK_Prod Primary key (product_key);

--creating customer table
select distinct Customer_ID,Customer_Name,Segment,City,Country,Region,Postal_Code
into Customer
from superstore;
alter table Customer
add Customer_key int identity(1,1) Primary Key;

select * from Customer;

--Creating sales table
select Order_ID,Order_Date,Customer_ID,Customer_Name,Ship_Date,Ship_Mode,
Product_ID,Product_Name,Sales,Quantity,Discount,Profit
into sales
from superstore;

alter table sales
add sales_key int identity(1,1) Primary key;
```

Data types corrected. Created surrogate key as present keys have repeated entries.

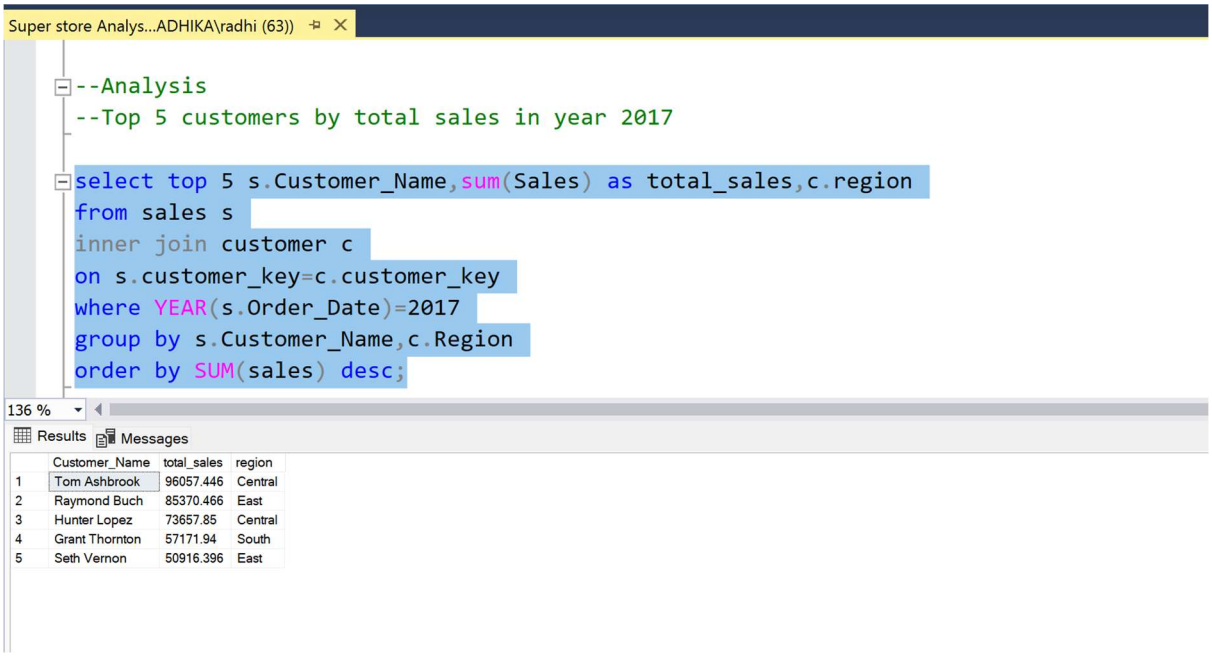
```
alter table sales
alter column order_date date;
alter table sales
alter column ship_date date;
alter table sales
add Product_key int;
alter table sales
add Customer_key int;

update s
set s.product_key=p.Product_key
from sales s
inner join Product p
on s.Product_ID=p.Product_ID
and s.Product_Name=p.Product_Name;

update s
set s.customer_key=c.customer_key
from sales s
inner join Customer c
on s.Customer_ID=c.Customer_ID
and s.Customer_Name=c.Customer_Name;

alter table sales
add constraint FK_prod foreign key (Product_key)
references product(product_key);

alter table sales
add constraint FK_cust foreign key (Customer_key)
references Customer(Customer_key);
```

Created date dimension table and created star schema using foreign keys for all tables



```
alter table sales
add constraint FK_cust foreign key (Customer_key)
references Customer(Customer_key);

--creating date dimension
select distinct order_date into DimDate from superstore;

select * from DimDate;

alter table dimdate
alter column order_date date;


alter table dimdate
add nyear int,
month_name varchar(50),
nmonth int;

update DimDate
set nyear=YEAR(Order_Date),
nmonth=MONTH(order_date),
month_name=DATENAME(month,order_date);

alter table dimdate
alter column order_date date not null;
alter table dimdate
add constraint PK_date primary key (order_date);

alter table sales
add constraint FK_date foreign key (order_date)
references dimdate(order_date);
```

Top 5 customers by total sales in the year 2017

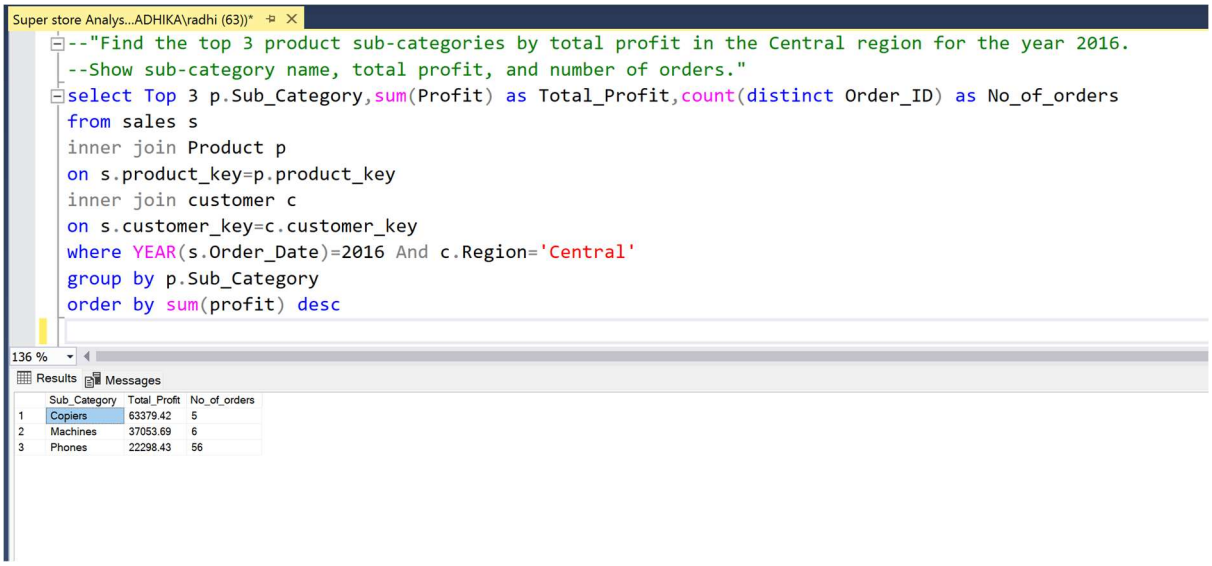


```
--Analysis
--Top 5 customers by total sales in year 2017

select top 5 s.Customer_Name,sum(Sales) as total_sales,c.region
from sales s
inner join customer c
on s.customer_key=c.customer_key
where YEAR(s.Order_Date)=2017
group by s.Customer_Name,c.Region
order by SUM(sales) desc;
```

Results Messages

	Customer_Name	total_sales	region
1	Tom Ashbrook	96057.446	Central
2	Raymond Buch	85370.466	East
3	Hunter Lopez	73657.85	Central
4	Grant Thornton	57171.94	South
5	Seth Vernon	50916.396	East

Top 3 Product sub categories by total profit in central region for the year 2016

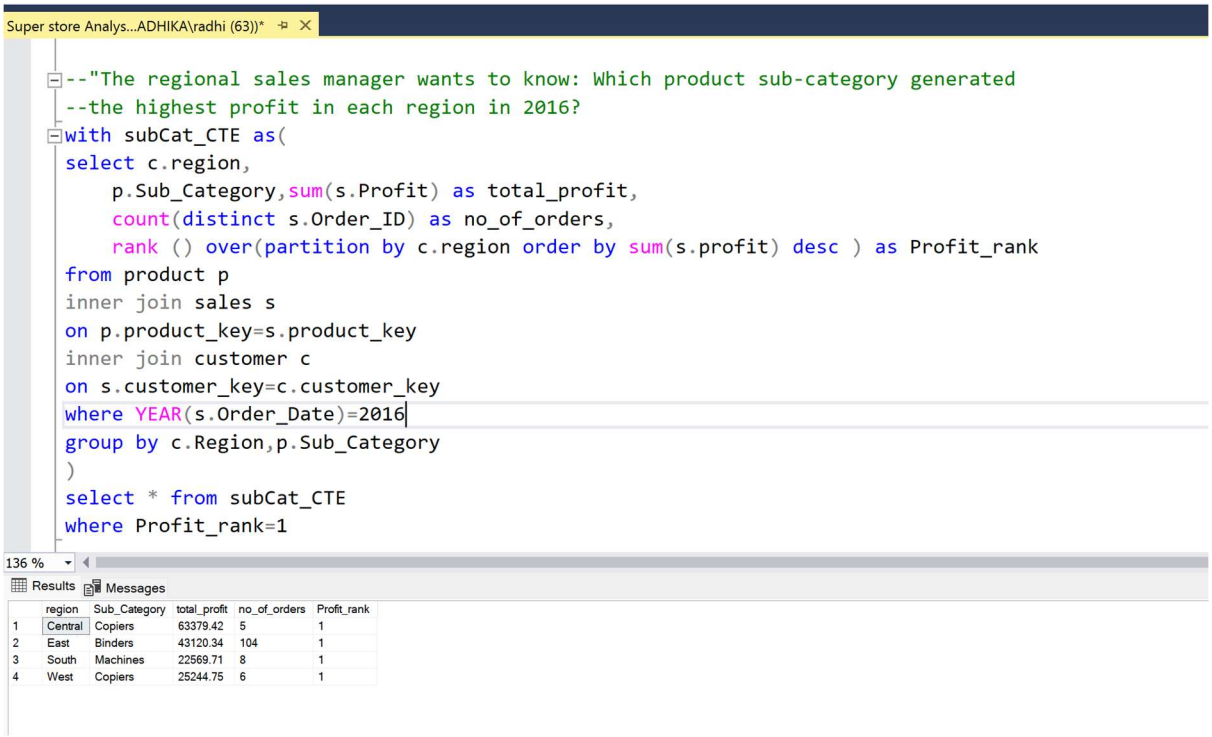


```
--"Find the top 3 product sub-categories by total profit in the Central region for the year 2016.
--Show sub-category name, total profit, and number of orders."
select Top 3 p.Sub_Category,sum(Profit) as Total_Profit,count(distinct Order_ID) as No_of_orders
from sales s
inner join Product p
on s.product_key=p.product_key
inner join customer c
on s.customer_key=c.customer_key
where YEAR(s.Order_Date)=2016 And c.Region='Central'
group by p.Sub_Category
order by sum(profit) desc
```

Results Messages

	Sub_Category	Total_Profit	No_of_orders
1	Copiers	63379.42	5
2	Machines	37053.69	6
3	Phones	22298.43	56

Which Product sub category generated highest profit in each region in the year 2016

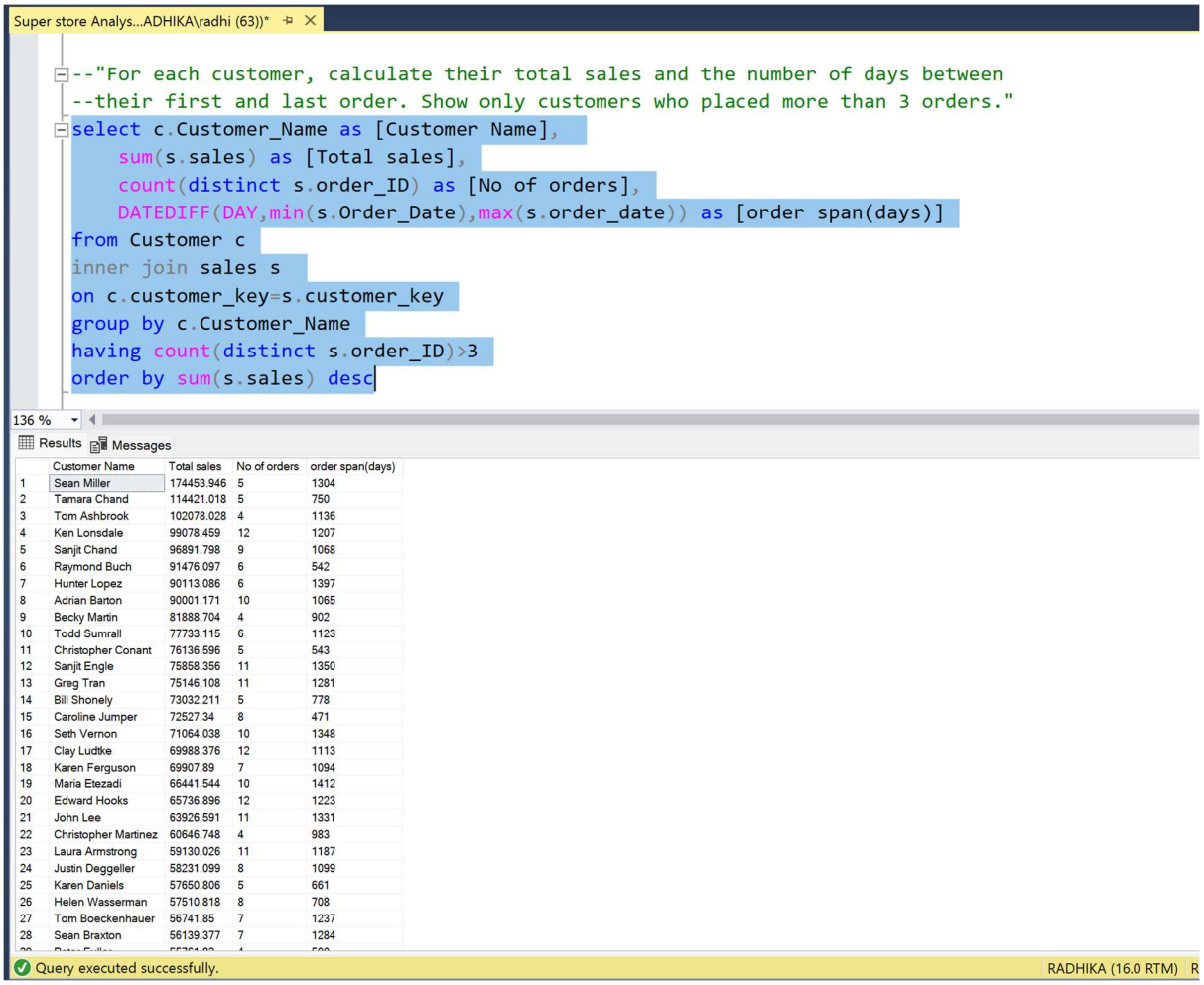


```
--"The regional sales manager wants to know: Which product sub-category generated
--the highest profit in each region in 2016?
with subCat_CTE as(
select c.region,
    p.Sub_Category,sum(s.Profit) as total_profit,
    count(distinct s.Order_ID) as no_of_orders,
    rank () over(partition by c.region order by sum(s.profit) desc ) as Profit_rank
from product p
inner join sales s
on p.product_key=s.product_key
inner join customer c
on s.customer_key=c.customer_key
where YEAR(s.Order_Date)=2016
group by c.Region,p.Sub_Category
)
select * from subCat_CTE
where Profit_rank=1
```

Results Messages

	region	Sub_Category	total_profit	no_of_orders	Profit_rank
1	Central	Copiers	63379.42	5	1
2	East	Binders	43120.34	104	1
3	South	Machines	22569.71	8	1
4	West	Copiers	25244.75	6	1

For each customer, calculate total sales and number of days between their first and last order.



	Customer Name	Total sales	No of orders	order span(days)
1	Sean Miller	174453.946	5	1304
2	Tamara Chand	114421.018	5	750
3	Tom Ashbrook	102078.028	4	1136
4	Ken Lonsdale	99078.459	12	1207
5	Sanjit Chand	96891.798	9	1068
6	Raymond Buch	91476.097	6	542
7	Hunter Lopez	90113.086	6	1397
8	Adrian Barton	90001.171	10	1065
9	Becky Martin	81888.704	4	902
10	Todd Sumrall	77733.115	6	1123
11	Christopher Conant	76136.596	5	543
12	Sanjit Engle	75858.356	11	1350
13	Greg Tran	75146.108	11	1281
14	Bill Shonely	73032.211	5	778
15	Caroline Jumper	72527.34	8	471
16	Seth Vernon	71064.038	10	1348
17	Clay Ludtke	69988.376	12	1113
18	Karen Ferguson	69907.89	7	1094
19	Maria Etezadi	66441.544	10	1412
20	Edward Hooks	65736.896	12	1223
21	John Lee	63926.591	11	1331
22	Christopher Martinez	60646.748	4	983
23	Laura Armstrong	59130.026	11	1187
24	Justin Deggeller	58231.099	8	1099
25	Karen Daniels	57650.806	5	661
26	Helen Wasserman	57510.818	8	708
27	Tom Boeckenhauer	56741.85	7	1237
28	Sean Braxton	56139.377	7	1284

Region wise top performing categories in 2017

```sql
--"Find the top 2 product categories in each region based
--on total sales in the year 2017. Show category name, region, total sales, and number of orders."
with region_CTE as(
select p.Category,c.Region,
    sum(s.sales) as [Total sales],
    count(distinct s.Order_ID) as [No of orders],
    rank() over(partition by c.region order by sum(s.sales) desc) as [Sales Rank]
from Product p
inner join sales s
on p.product_key=s.Product_key
inner join Customer c
on s.Customer_key=c.customer_key
where  year(s.Order_Date)=2017
group by p.Category,c.Region
)
select * from region_CTE
where [Sales Rank]<=2
```

Results Messages

	Category	Region	Total sales	No of orders	Sales Rank
1	Technology	Central	576103.586	138	1
2	Office Supplies	Central	473563.816	352	2
3	Technology	East	458944.758	142	1
4	Office Supplies	East	452434.596	365	2
5	Technology	South	380227.855	111	1
6	Furniture	South	277249.8054	117	2
7	Office Supplies	West	445697.664	360	1
8	Technology	West	370017.396	150	2

Category wise running total of sales for 2017

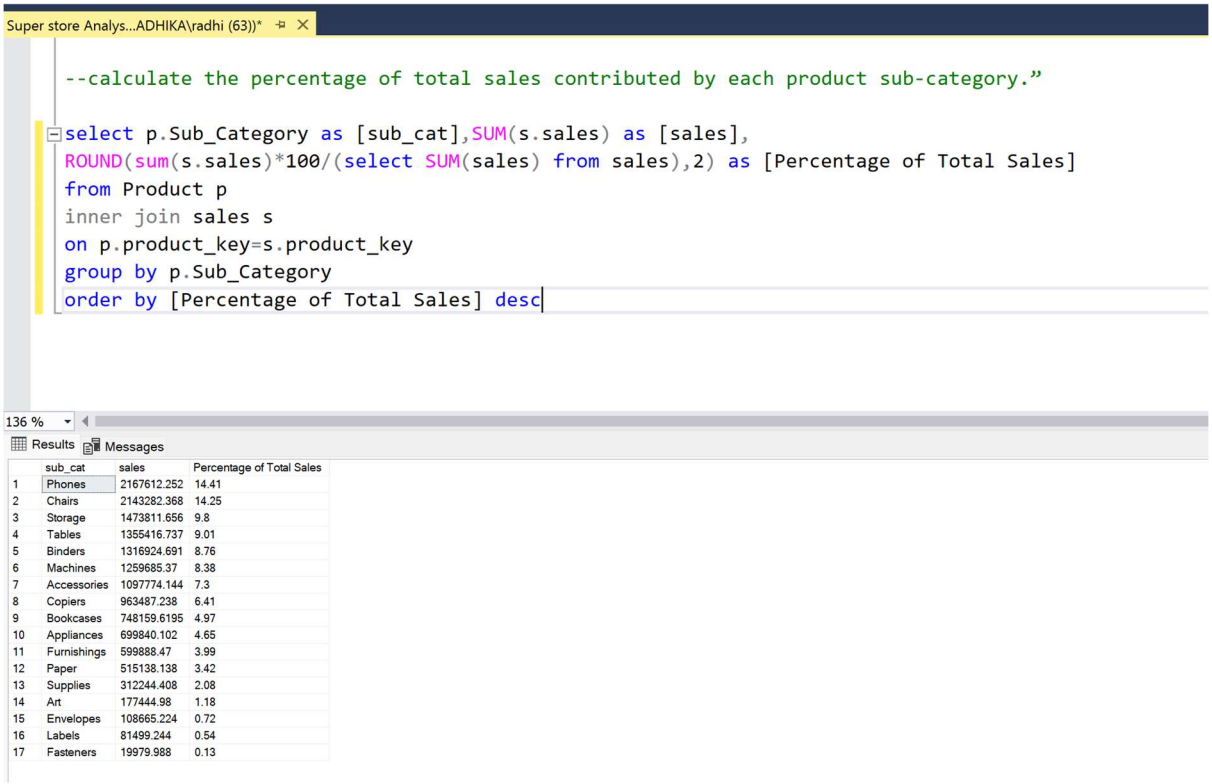
```sql
--"For each product category, calculate the running total of sales for each month in 2017."

with cat_CTE as(
select p.category as [Category],
    d.nmonth [month num],
    d.month_name as [Month],
    sum(s.sales) as [Total sales],
    Rank() over (partition by p.category order by d.nmonth) as [Rank]
    from DimDate d
inner join sales s
on d.Order_Date=s.Order_Date
inner join Product p
on p.product_key=s.product_key
where YEAR(s.Order_Date)=2017
group by p.category,d.month_name,d.nmonth
--order by d.nmonth
)
select [Category],[Month],[Total sales],
sum([Total sales]) over (partition by [category] order by [Month num]) as [Runnning total]
from cat_CTE -
```

Results Messages

	Category	Month	Total sales	Runnning total
1	Furniture	January	38781.816	38781.816
2	Furniture	February	44362.9714	83144.7874
3	Furniture	March	71140.4076	154285.195
4	Furniture	April	59947.9277	214233.1227
5	Furniture	May	109436.014	323669.1367
6	Furniture	June	123319.5757	446988.7124
7	Furniture	July	76089.449	523078.1614
8	Furniture	August	99210.15	622288.3114
9	Furniture	September	189197.433	811485.7444
10	Furniture	October	142947.1684	954432.9128
11	Furniture	November	245941.6968	1200374.6096
12	Furniture	December	205791.9496	1406166.5592
13	Office Supplies	January	132890.754	132890.754
14	Office Supplies	February	47357.189	180247.943
15	Office Supplies	March	95813.603	276061.546
16	Office Supplies	April	101563.24	377624.786

Query executed successfully. RADHIKA (16.0 RTM) RADHIKA\radhi (63)

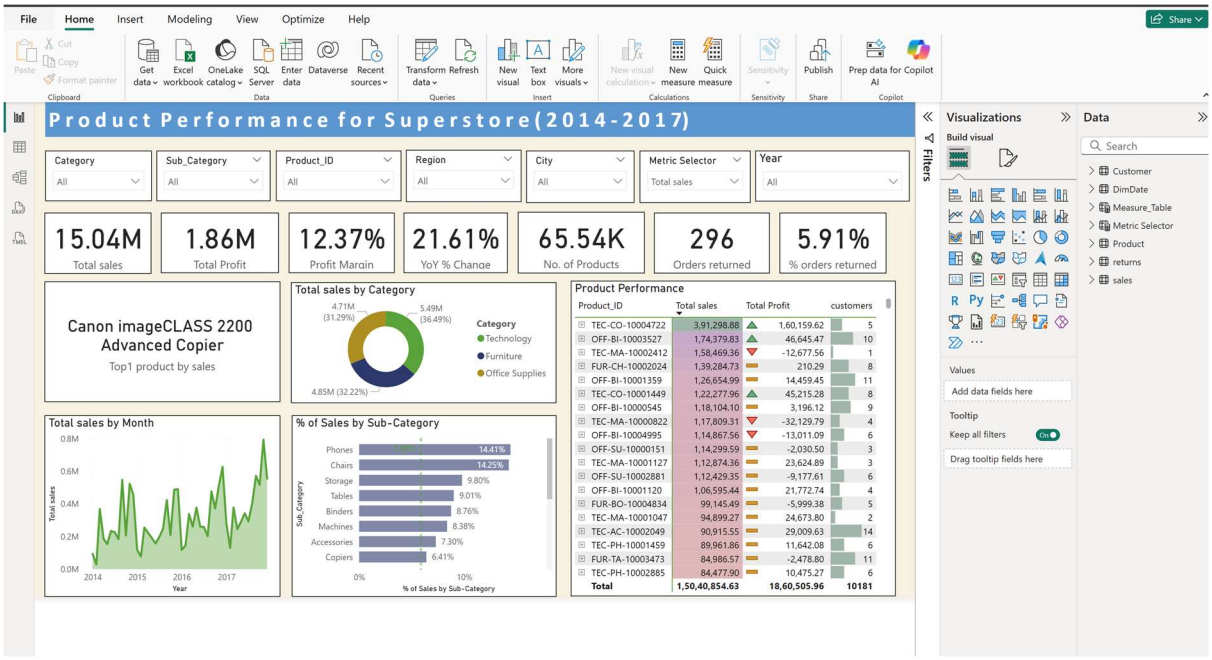
Product sub category wise Percentage of Total sales contributed



	sub_cat	sales	Percentage of Total Sales
1	Phones	2167612.252	14.41
2	Chairs	2143282.368	14.25
3	Storage	1473811.656	9.8
4	Tables	1355416.737	9.01
5	Binders	1316924.691	8.76
6	Machines	1259685.37	8.38
7	Accessories	1097774.144	7.3
8	Copiers	963487.238	6.41
9	Bookcases	748159.6195	4.97
10	Appliances	699840.102	4.65
11	Furnishings	599888.47	3.99
12	Paper	515138.138	3.42
13	Supplies	312244.408	2.08
14	Art	177444.98	1.18
15	Envelopes	108665.224	0.72
16	Labels	81499.244	0.54
17	Fasteners	19979.988	0.13

Power BI Dashboard Summary

- Created KPIs: Total Sales, Profit, Profit Margin, YoY % Growth, No. of Products
- Used DAX for various calculations.
- Added slicers for Year, Region, Category
- Highlighted top-performing product: *Canon imageCLASS 2200(changes according to slicers)*
- Included % Returned Orders (5.91%)
- Applied conditional formatting for Product performance matrix.



Dax:

Current year sales =

var curr_year=MAX(DimDate[nyear])

RETURN CALCULATE([Total sales],DimDate[nyear]=curr_year)

Last year sales =

var curr_year=MAX(DimDate[nmonth])

RETURN CALCULATE([Total sales],DimDate[nyear]=curr_year-1)

Profit Margin = ([Total Profit]/[Total sales])

Total Profit = SUM(sales[Profit])

Total sales = SUM(sales[Sales])

```
YoY % Change =
VAR _CurrentYear = MAX('DimDate'[nyear])
VAR _CurrentSales =
   CALCULATE(
      [Total Sales],
      FILTER(
         ALL('DimDate'),
         'DimDate'[nyear] = _CurrentYear
      )
   )
VAR _PreviousSales =
   CALCULATE(
      [Total Sales],
      FILTER(
         ALL('DimDate'),
         'DimDate'[nyear] = _CurrentYear - 1
      )
   )
RETURN
DIVIDE(_CurrentSales - _PreviousSales, _PreviousSales, 0)
```

Why YOY% again calculating Current year and previous year sales ?

As this has to be changed dynamically with slicers I am calculating dynamically current year sales and previous year sales.

Conclusion:

“This project demonstrates the ability to use SQL for business logic and Power BI for impactful dashboards that drive insights.”