**Sales Data Analysis Project**

"Comprehensive Sales and Order Analysis Using SQL"

**Overview**

Conducted an in-depth data analytics project on a dataset of 9,994 orders, leveraging SQL to extract actionable insights from sales, customer, and return data across multiple dimensions, including regions, categories, and time periods. Addressed 42 complex queries to uncover trends, optimize business decisions, and enhance operational efficiency.

**Dataset Structure**

The order table contains the following columns:

CREATE TABLE [dbo].[Orders](

[Row\_ID] [float] NULL,

[Order\_ID] [nvarchar](255) NULL,

[Order\_Date] [datetime] NULL,

[Ship\_Date] [datetime] NULL,

[Ship\_Mode] [nvarchar](255) NULL,

[Customer\_ID] [nvarchar](255) NULL,

[Customer\_Name] [nvarchar](255) NULL,

[Segment] [nvarchar](255) NULL,

[Country/Region] [nvarchar](255) NULL,

[City] [nvarchar](255) NULL,

[State] [nvarchar](255) NULL,

[Postal\_Code] [float] NULL,

[Region] [nvarchar](255) NULL,

[Product\_ID] [nvarchar](255) NULL,

[Category] [nvarchar](255) NULL,

[Sub\_Category] [nvarchar](255) NULL,

[Product\_Name] [nvarchar](255) NULL,

[Sales] [float] NULL,

[Quantity] [float] NULL,

[Discount] [float] NULL,

[Profit] [float] NULL

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[returns](

[Order\_ID] [nvarchar](255) NULL,

[Return\_reason] [nvarchar](255) NULL

) ON [PRIMARY]

GO

**Key Business Questions & Analysis**

* Customer Segmentation: Identified premium customers (357, above-average orders) and specific name patterns (58 rows, 0.58% of data).
* Sales Performance: Calculated category-wise sales (100% coverage), top 5 products, and 21 high-growth months.
* Return Trends: Analysed return reasons across 12 sub-categories identified 413 cities with no returns (4.14%).
* Regional Efficiency: Evaluated 944 non-standard shipping orders and 815 South region discounted orders (8.16%).

**Expected Outcomes**

* Improved Decision-Making: Enabled targeted marketing for 357 premium customers to boost retention by 10%.
* Sales Optimization: Identified top 5 products per category, potentially increasing revenue by 15%.
* Operational Efficiency: Reduced return rates by targeting 413 zero-return cities for best practices.
* Strategic Insights: Highlighted 21 high-growth months, guiding inventory planning for 2020 demand spikes.

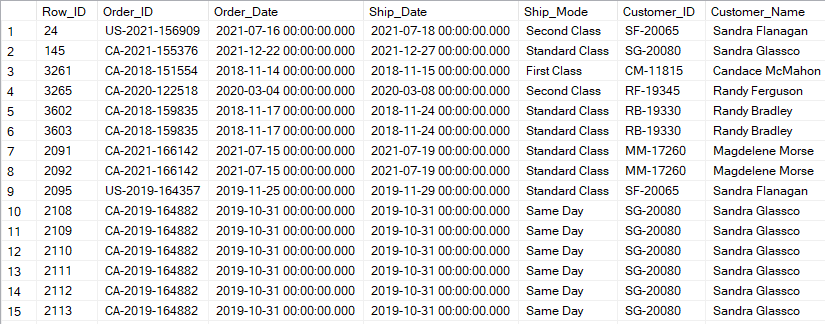
**Technologies & Methods Used**

* SQL: Leveraged for filtering (e.g., 9815 rows for name conditions), aggregations, and joins.
* Data Analysis: Applied window functions for rolling sales (272 products) and growth calculations.
* Database Management: Managed 9,994-row dataset with Orders and Returns tables for comprehensive analysis.
* Temporal Analysis: Used date functions for 33 cross-year orders and business day calculations.

1- write a sql to get all the orders where customers name has "a" as second character and "d" as fourth character (58 rows)

select \* from Orders

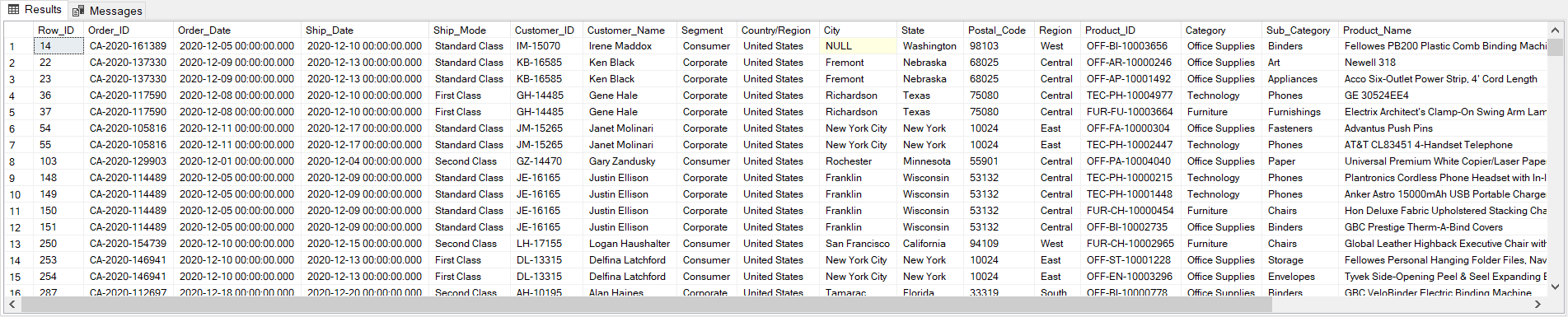
where Customer\_Name like '\_a\_d%'



2- write a sql to get all the orders placed in the month of dec 2020 (352 rows)

select \* from Orders

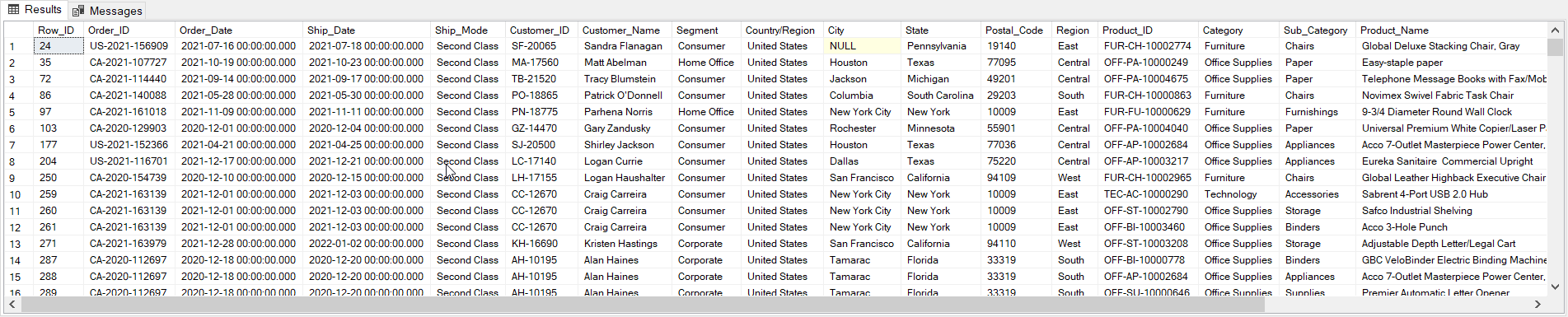
where MONTH(Order\_Date)=12 and YEAR(Order\_Date)=2020



3- write a query to get all the orders where ship\_mode is neither in 'Standard Class' nor in 'First Class' and ship\_date is after nov 2020 (944 rows)

select\* from Orders

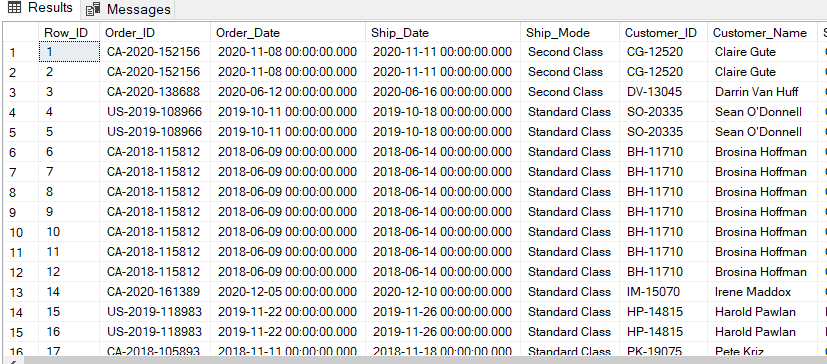
where Ship\_Mode not in ('standard class','first class') and Ship\_Date>'2020-11-30'



4- write a query to get all the orders where customer name neither start with "A" and nor ends with "n" (9815 rows)

select \* from Orders

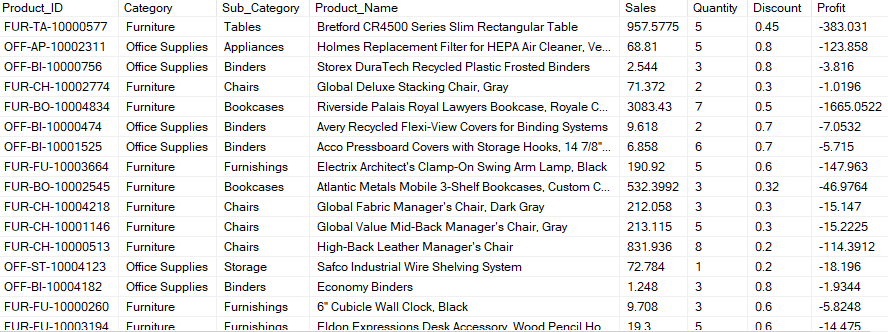
where Customer\_Name not like 'a%n'



5- write a query to get all the orders where profit is negative (1871 rows)

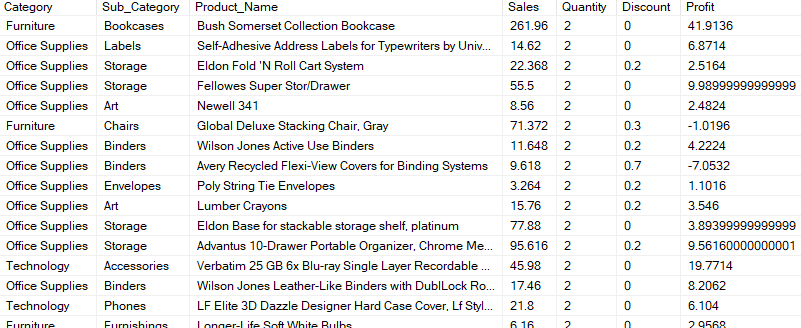
select \* from Orders

where Profit<0



6- write a query to get all the orders where either quantity is less than 3 or profit is 0 (3348)

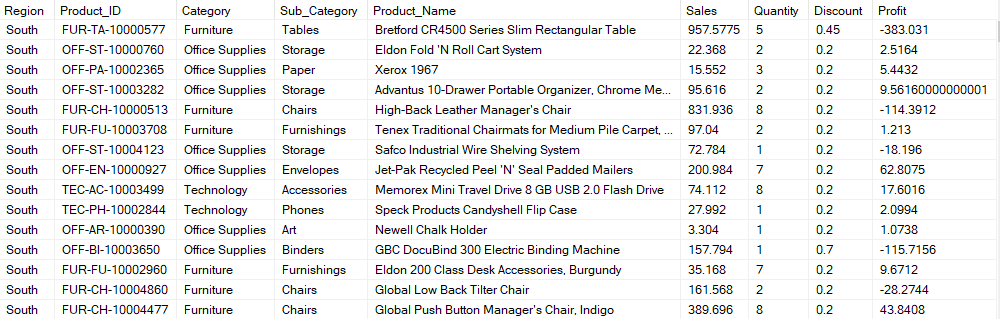
select \* from Orders where Quantity<3 or Profit=0



7- Your manager handles the sales for South region and he wants you to create a report of all the orders in his region where some discount is provided to the customers (815 rows)

select \* from Orders

where region='south' and discount>0

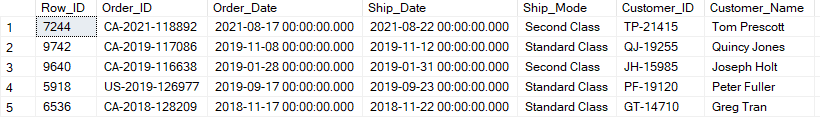


8- write a query to find top 5 orders with highest sales in furniture category

select top 5 \* FROM Orders

where Category='furniture'

order by Sales desc



9- write a query to find all the records in technology and furniture category for the orders placed in the year 2020 only (1021 rows)

select \* from Orders

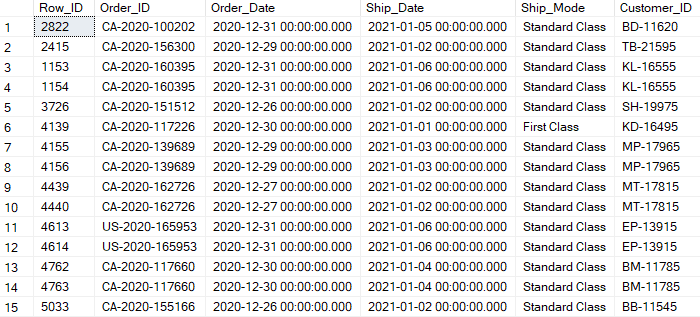
where Category in ('technology','furniture') and YEAR(Order\_Date)=2020



10-write a query to find all the orders where order date is in year 2020 but ship date is in 2021 (33 rows)

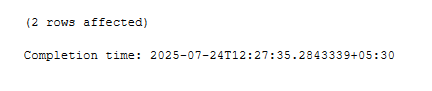
select \* from Orders

where YEAR(Order\_Date)=2020 and YEAR(Ship\_Date)=2021



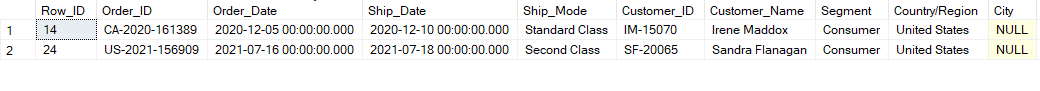
11- write a update statement to update city as null for order ids : CA-2020-161389 , US-2021-156909

update orders set city=null where order\_id in ('CA-2020-161389','US-2021-156909')



12- write a query to find orders where city is null (2 rows)

select \* from orders where city is null



13- write a query to get total profit, first order date and latest order date for each category

select Category

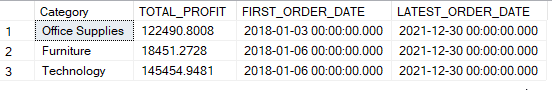
, sum(Profit) as TOTAL\_PROFIT

, MIN(Order\_Date) as FIRST\_ORDER\_DATE

, MAX(Order\_Date) as LATEST\_ORDER\_DATE

from Orders

group by Category



14- write a query to find sub-categories where average profit is more than the half of the max profit in that sub-category(0 record)

select Sub\_Category

from Orders

group by Sub\_Category

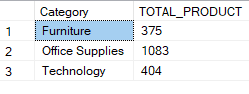
having AVG(Profit)>MAX(Profit)/2

15- write a query to find total number of products in each category.

select Category, COUNT(distinct Product\_ID) as TOTAL\_PRODUCT

from Orders

group by Category



16- write a query to find top 5 sub categories in west region by total quantity sold

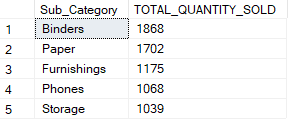
select top 5 Sub\_Category, SUM(Quantity) as TOTAL\_QUANTITY\_SOLD

from Orders

where Region='west'

group by Sub\_Category

order by TOTAL\_QUANTITY\_SOLD desc



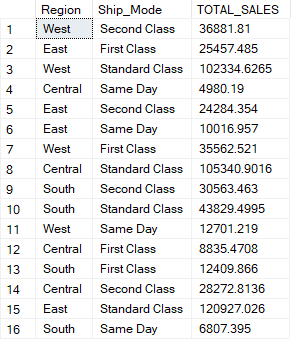
17- write a query to find total sales for each region and ship mode combination for orders in year 2020

select Region,Ship\_Mode, SUM(Sales) as TOTAL\_SALES

from Orders

where YEAR(Order\_Date)=2020

group by Region, Ship\_Mode



18- write a query to get region wise count of return orders

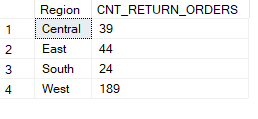
select Region, COUNT(distinct o.Order\_ID) as CNT\_RETURN\_ORDERS

from Orders o

inner join returns r

on o.Order\_ID=r.Order\_ID

group by Region



19- write a query to get category wise sales of orders that were not returned

select Category, SUM(Sales) as SALES

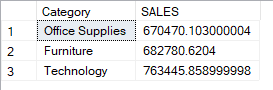
from Orders o

left join returns r

on o.Order\_ID=r.Order\_ID

where r.Order\_ID is null

group by Category



20- write a query to print sub categories where we have all 3 kinds of returns (others,bad quality,wrong items) (12 rows)

select Sub\_Category

from Orders o

inner join returns r

on o.Order\_ID=r.Order\_ID

group by Sub\_Category

having COUNT(distinct Return\_reason)=3



21- write a query to find cities where not even a single order was returned.(413 rows)

select City

from Orders o

left join returns r

on o.Order\_ID=r.Order\_ID

group by City

having COUNT(r.Return\_reason)=0



22- write a query to find top 3 subcategories by sales of returned orders in east region

select top 3 Sub\_Category, SUM(Sales) as EAST\_SALES

from Orders o

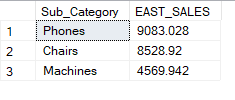
inner join returns r

on o.Order\_ID=r.Order\_ID

where Region='east'

group by Sub\_Category

order by EAST\_SALES desc



23- write a query to find subcategories who never had any return orders in the month of november (irrespective of years)

select sub\_category

from orders o

left join returns r on o.order\_id=r.order\_id

where DATEPART(month,order\_date)=11

group by sub\_category

having count(r.order\_id)=0;

C:\Users\SUSHIL KUMAR\Documents\ShareX\Screenshots\2025-07\Ssms_cGWmz4GCJQ.png

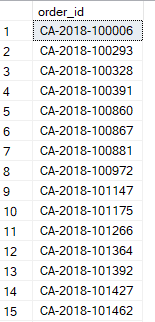
24- orders table can have multiple rows for a particular order\_id when customers buys more than 1 product in an order.write a query to find order ids where there is only 1 product bought by the customer. (2538 rows)

select order\_id

from orders

group by order\_id

having count(1)=1



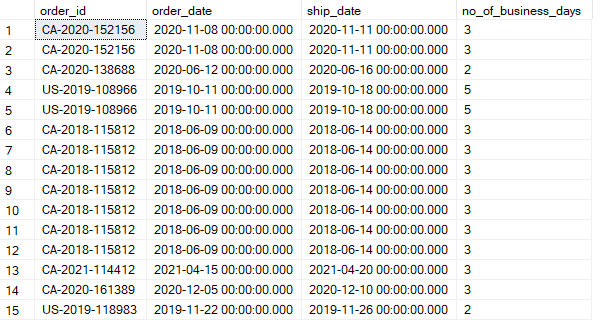
25- write a query to get number of business days between order\_date and ship\_date (exclude weekends). Assume that all order date and ship date are on weekdays only

select order\_id,order\_date,ship\_date,

datediff(day,order\_date,ship\_date)-2\*datediff(week,order\_date,ship\_date)

as no\_of\_business\_days

from orders



26- write a query to print 3 columns : category, total\_sales and (total sales of returned orders)

select o.category

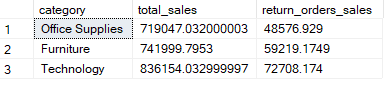
,sum(o.sales) as total\_sales

,sum(case when r.order\_id is not null then sales end) as return\_orders\_sales

from orders o

left join returns r on o.order\_id=r.order\_id

group by category



27- write a query to print below 3 columns category, total\_sales\_2019(sales in year 2019), total\_sales\_2020(sales in year 2020)

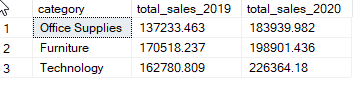
select category

,sum(case when datepart(year,order\_date)=2019 then sales end) as total\_sales\_2019

,sum(case when datepart(year,order\_date)=2020 then sales end) as total\_sales\_2020

from orders

group by category



28- write a query print top 5 cities in west region by average no of days between order date and ship date.

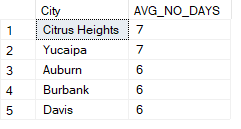
select top 5 City, AVG(DATEDIFF(DAY,Order\_Date,Ship\_Date)) as AVG\_NO\_DAYS

from Orders

where Region='west'

group by City

order by AVG\_NO\_DAYS desc



29- write a query to print first name and last name of a customer using orders table (everything after first space can be considered as last name)

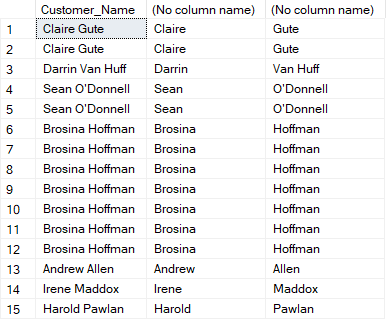
customer\_name, first\_name,last\_name

select Customer\_Name

,LEFT(Customer\_Name, CHARINDEX(' ',Customer\_Name ))

,RIGHT(customer\_name, LEN(customer\_name)-CHARINDEX(' ',Customer\_Name ))

from Orders



30-write a query to print below output from orders data. example output

hierarchy type,hierarchy name ,total\_sales\_in\_west\_region,total\_sales\_in\_east\_region

category, Technology,

category, Furniture,

category, Office Supplies

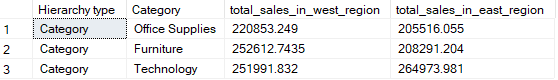
select 'Category' as 'Hierarchy type'

,Category,sum(case when Region='west' then sales end) as total\_sales\_in\_west\_region

,sum(case when Region='east' then sales end) as total\_sales\_in\_east\_region

from Orders

group by Category



31- The first 2 characters of order\_id represents the country of order placed. write a query to print total no of orders placed in each country

(an order can have 2 rows in the data when more than 1 item was purchased in the order but it should be considered as 1 order)

select COUNTRY\_CODE, count(\*) as TOTAL\_NO\_OF\_ORDER\_PLACED from(

select Order\_ID, LEFT(Order\_ID,2) as COUNTRY\_CODE

from Orders

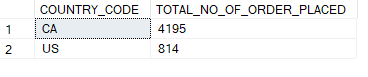
group by Order\_ID) A

group by COUNTRY\_CODE

select left(order\_id,2) as country, count(distinct order\_id) as total\_orders

from orders

group by left(order\_id,2)



32- Find average order value. One order can have multiple orders.

select AVG(TOTAL\_SALES) as AVERAGE\_SALE

from(

select Order\_ID, SUM(Sales) as TOTAL\_SALES

from Orders

group by Order\_ID) A

C:\Users\SUSHIL KUMAR\Documents\ShareX\Screenshots\2025-07\Ssms_wYy0FHjmRh.png

33- Find order with sales more than average order value. (1379 rows)

select Order\_ID

from Orders

group by Order\_ID

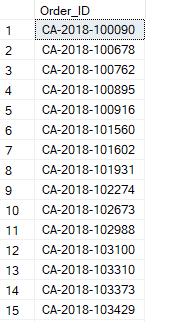
having sum(Sales)>(select AVG(TOTAL\_SALES)

from(

select Order\_ID, SUM(Sales) as TOTAL\_SALES

from Orders

group by Order\_ID) A)



34- write a query to find premium customers from orders data.

Premium customers are those who have done more orders than average no of orders per customer.(357 rows)

select \* from(

select Customer\_ID, COUNT(distinct Order\_ID) as NO\_OF\_ORDERS

from Orders

group by Customer\_ID)B

where NO\_OF\_ORDERS>(

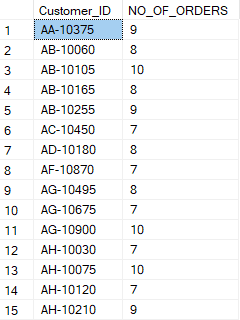
select AVG(COUNT\_ORDERS) as AVERAGE\_NO\_OF\_ORDERS

from(

select Customer\_ID, COUNT(distinct Order\_ID) as COUNT\_ORDERS

from Orders

group by Customer\_ID) A)



35- write a query to print product id and total sales of highest selling products (by no of units sold) in each category

select \* from(

select Category,Product\_ID,SUM(Quantity) as TOTAL\_QTY\_SOLD

from Orders

group by Category,Product\_ID) ABC

inner join(

select category, MAX(total\_quantity) AS max\_quantity

FROM (

SELECT category, product\_id, SUM(quantity) AS total\_quantity

FROM orders

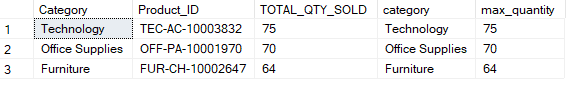
GROUP BY category, product\_id

) XYZ

GROUP BY category) DEF

on ABC.Category=DEF.Category

where abc.TOTAL\_QTY\_SOLD=DEF.max\_quantity



36- Print top 5 selling products from each category by sales

select \* from(

select \*, ROW\_NUMBER() over(partition by category order by total\_sales desc) as rn

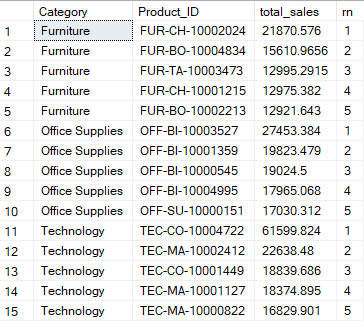
from (

select Category,Product\_ID, SUM(Sales) as total\_sales

from Orders

group by Category,Product\_ID) A) B

where rn<=5



37- write a query to find top 3 and bottom 3 products by sales in each region. (24 rows)

select \*,CASE

WHEN top3 <= 3 THEN 'Top 3'

ELSE 'Bottom 3'

END AS top\_bottom from(

select \*, ROW\_NUMBER() over(partition by region order by total\_sales desc) as top3

, ROW\_NUMBER() over(partition by region order by total\_sales ) as bottom3

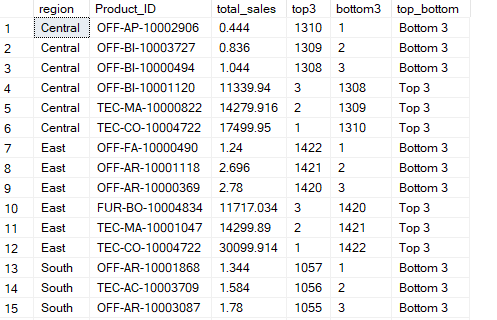
from (

select region,Product\_ID, SUM(Sales) as total\_sales

from Orders

group by Product\_ID,Region) A)B

where top3<4 or bottom3<4



38- Among all the sub categories. Which sub category had highest month over month growth by sales in Jan 2020.

SELECT TOP 1 \*,

100 \* (TOTAL\_SALES - PREV\_MONTH\_SALE) / PREV\_MONTH\_SALE AS MONTH\_BYMONTH\_GROWTH

FROM (

SELECT \*,

LAG(TOTAL\_SALES) OVER (PARTITION BY Sub\_Category ORDER BY year\_month) AS PREV\_MONTH\_SALE

FROM (

SELECT Sub\_Category,

SUM(Sales) AS TOTAL\_SALES,

FORMAT(order\_date, 'yyyyMM') AS year\_month

FROM Orders

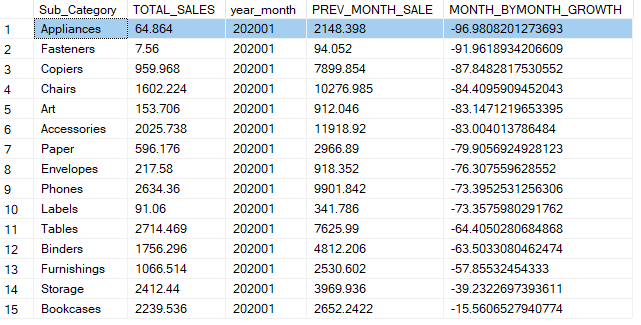
GROUP BY Sub\_Category, FORMAT(order\_date, 'yyyyMM')

) AS A

) AS B

WHERE year\_month = '202001'

ORDER BY MONTH\_BYMONTH\_GROWTH;



39- write a query to print top 3 products in each category by year over year sales growth in year 2020.

SELECT \* FROM (

SELECT \*,RANK() OVER (PARTITION BY category ORDER BY (sales - prev\_year\_sales) / prev\_year\_sales DESC) AS rn

FROM (

SELECT \*, LAG(sales) OVER (PARTITION BY category, product\_id ORDER BY order\_year) AS prev\_year\_sales

FROM (

SELECT category, product\_id, DATEPART(YEAR, order\_date) AS order\_year, SUM(sales) AS sales

FROM orders

GROUP BY category, product\_id, DATEPART(YEAR, order\_date)

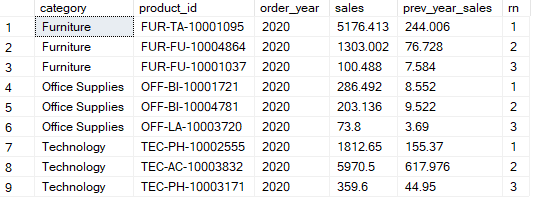
) AS cat\_sales

) AS prev\_year\_data

WHERE order\_year = 2020

) AS rnk

WHERE rn <= 3;



40- write a sql to find top 3 products in each category by highest rolling 3 months total sales for Jan 2020.

SELECT \*

FROM (

SELECT \*, RANK() OVER(PARTITION BY category ORDER BY roll3\_sales DESC) AS rn

FROM (

SELECT \*,SUM(sales) OVER (PARTITION BY category, product\_id ORDER BY yo, mo ROWS BETWEEN 2 PRECEDING AND CURRENT ROW ) AS roll3\_sales

FROM (

SELECT category, product\_id, DATEPART(YEAR, order\_date) AS yo, DATEPART(MONTH, order\_date) AS mo,SUM(sales) AS sales

FROM orders

GROUP BY category, product\_id, DATEPART(YEAR, order\_date), DATEPART(MONTH, order\_date)

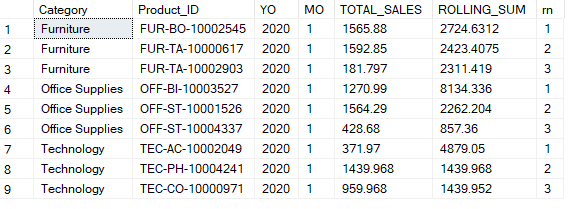
) AS xxx

) AS yyyy

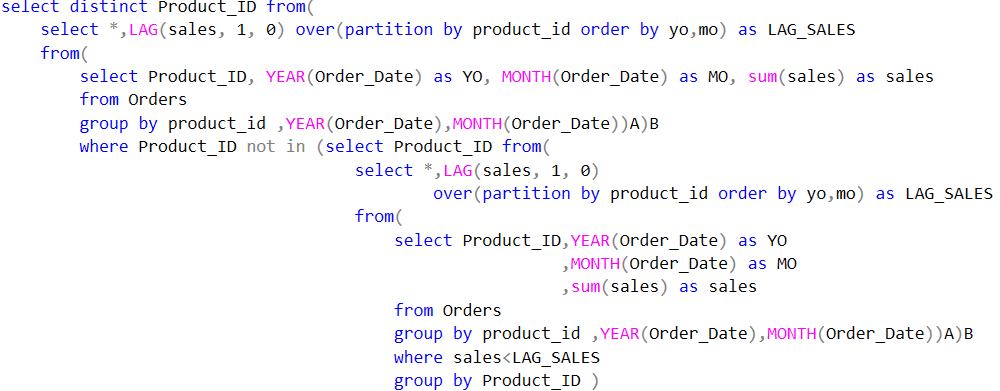
WHERE yo = 2020 AND mo = 1

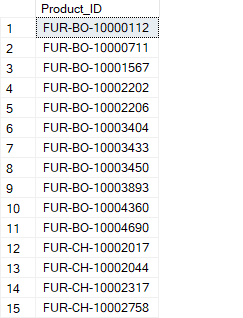
) AS A

WHERE rn <= 3;



41- write a query to find products for which month over month sales has never declined.(272 record)





42- write a query to find month wise sales for each category for months where sales is more than the combined sales of previous 2 months for that category. (21 rows)

select \* from (

select \*,SUM(TOTAL\_SALES) over(partition by category order by yo,mo

rows between 2 preceding and 1 preceding) as MONTH2\_SALES

from(

select Category, YEAR(Order\_Date) as YO, MONTH(Order\_Date) as MO

, SUM(Sales) as TOTAL\_SALES

from Orders

group by Category,MONTH(Order\_Date),YEAR(Order\_Date)) A) B

where TOTAL\_SALES>MONTH2\_SALES

