k;ji m

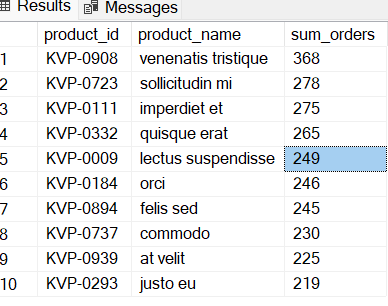
**E-commerce Sales Analytics SQl Queries**

kpi’s

1.List top-selling products.

(select top 10 orders.product\_id,products.product\_name ,sum(quantity) as sum\_orders

* from orders
* left join products
* on orders.product\_id = products.product\_id
* group by orders.product\_id,products.product\_name
* order by sum\_orders desc )



2. Find the total revenue generated.

SELECT SUM(T\_price) as Total\_revenu from (select

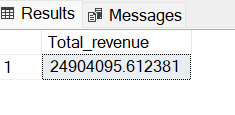
o.product\_id,product\_name,product\_price, quantity ,(quantity\*product\_price) as T\_price

from

orders o

join products p

on o.product\_id=p.product\_id) as sub\_query



3. Identify the most valuable customers.

select top 10 orders.customer\_id,concat(first\_name,' ',last\_name)as Name, round(sum(product\_price\*quantity),2) as price

from orders

left Join products

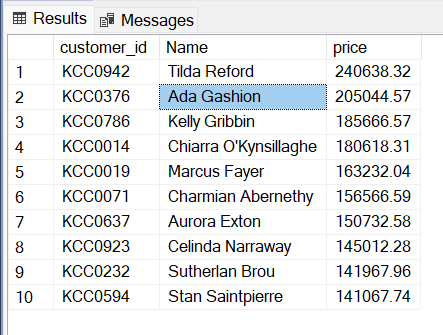
on products.product\_id = orders.product\_id

left join customers

on orders.customer\_id=customers.customer\_id

group by orders.customer\_id,concat(first\_name, ' ',last\_name)

order by price desc;

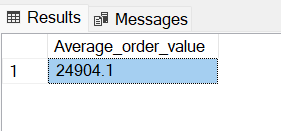


4) Calculate the average order value.

select sum(o.quantity\*p.product\_price) / count(distinct(o.order\_id)) as Average\_order\_value

from orders o

join products p on o.product\_id=p.product\_id



5)Break down the total revenue by month.

select

month(order\_date )as monthz,

DATENAME(MONTH,order\_date) as month ,

count(DATENAME(MONTH,order\_date)) as sales,

round(sum(p.product\_price \*quantity),2) as revenue

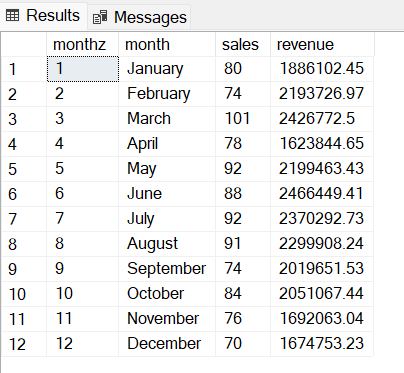
from orders o

left join

products p on o.product\_id = p. product\_id

group by month(order\_date ),DATENAME(MONTH,order\_date)

order by monthz;



7) calculate monthly or quarterly revenue

SELECT

CASE

WHEN MONTH(o.order\_date) IN (1,2,3) THEN '1st quarter'

WHEN MONTH(o.order\_date) IN (4,5,6) THEN '2nd quarter'

WHEN MONTH(o.order\_date) IN (7,8,9) THEN '3rd quarter'

WHEN MONTH(o.order\_date) IN (10,11,12) THEN '4th quarter'

END AS quarters,

round(SUM(o.quantity \* p.product\_price),2) AS revenue

FROM orders o

LEFT JOIN products p ON o.product\_id = p.product\_id

GROUP BY

CASE

WHEN MONTH(o.order\_date) IN (1,2,3) THEN '1st quarter'

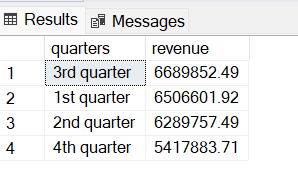
WHEN MONTH(o.order\_date) IN (4,5,6) THEN '2nd quarter'

WHEN MONTH(o.order\_date) IN (7,8,9) THEN '3rd quarter'

WHEN MONTH(o.order\_date) IN (10,11,12) THEN '4th quarter'

END

order by revenue desc ;



8) Identify the top product categories by total revenue.

select product\_category,round(sum(product\_price\*quantity),2) as Revenue

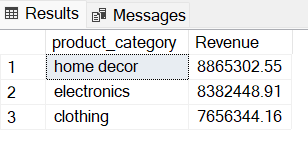
from orders

left join products

on orders.product\_id=products.product\_id

group by product\_category

order by Revenue desc;



9)Count the number of repeat customers.

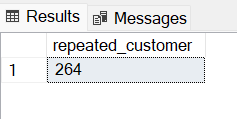
select count(count\_repeated\_customer) as repeated\_customer from

(select count(customer\_id) as count\_repeated\_customer

from orders

group by customer\_id) as customer\_count

where count\_repeated\_customer >1



10) Calculate the customer churn rate.

select count(distinct customer\_id) as total\_customer from customers

select count(distinct customer\_id) as order\_customer from orders

SELECT

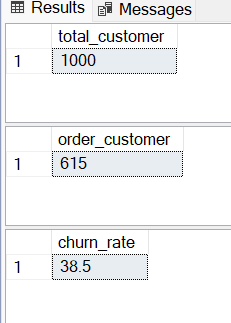
Cast (count (c.customer\_id) - count (DISTINCT o.customer\_id) as float )

/ COUNT(c.customer\_id)\* 100 AS churn\_rate

from Customers c

JOIN Orders o

on c.customer\_id = o.customer\_id;



11) Find the most popular products in a specific category

select count(orders.product\_id) as count,orders.product\_id,product\_name

from orders

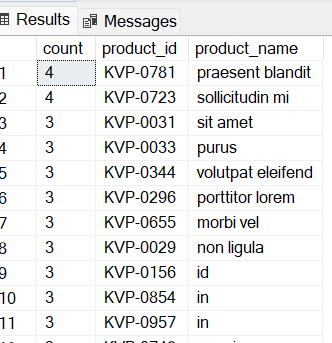
left join products

on products.product\_id=orders.product\_id

where products.product\_category = 'electronics'

group by orders.product\_id,product\_name

order by count desc;



12)Identify products that haven't been sold in the last three months.

WITH Last3\_MONTH AS (SELECT

p.product\_id,

p.product\_name,

p.product\_category,

order\_date,

DATEDIFF(MONTH, o.order\_date, (SELECT MAX(order\_date) FROM orders)) AS Month\_difference

FROM

products p

join

orders o ON p.product\_id = o.product\_id

AND DATEDIFF(MONTH, o.order\_date, (SELECT MAX(order\_date) FROM orders)) <= 3)

select

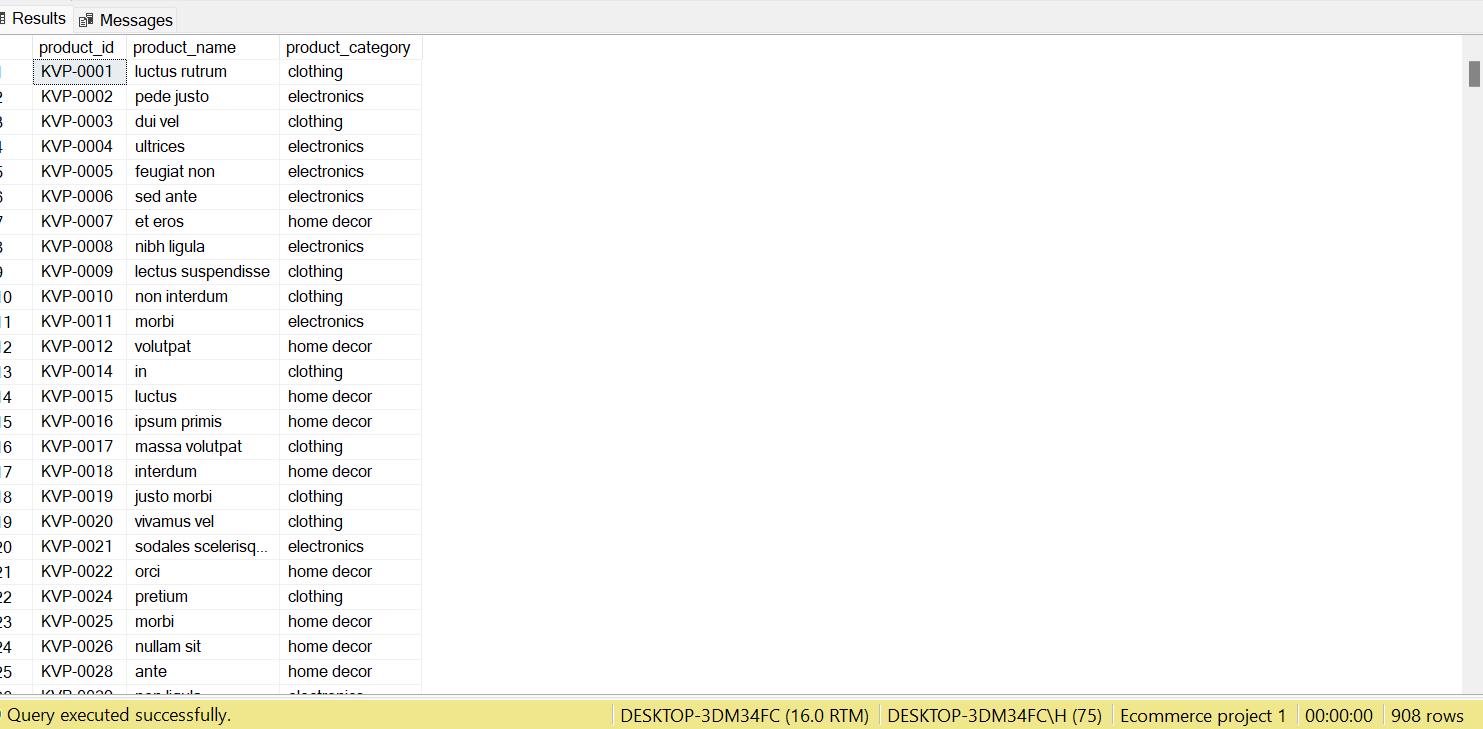
product\_id,

product\_name,

product\_category

from products

where product\_id not in (SELECT product\_id FROM Last3\_MONTH)



13)Calculate the average transaction value for each customer

select

customer\_id,sum(quantity \* price) as transac,

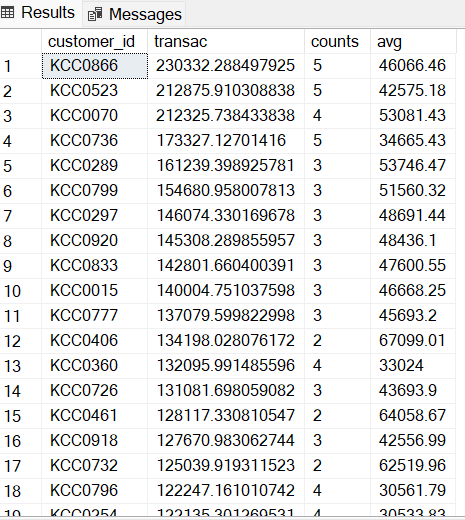
count(customer\_id) as counts ,

round(sum(quantity \* price)/count(customer\_id),2) as avg

from [transaction]

group by customer\_id

order by transac desc;



14)Determine the percentage of total revenue contributed by the top 10 customers.

SELECT TOP 10

o.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) AS Name,

ROUND(SUM(o.quantity \* p.product\_price), 2) AS customer\_revenue,

ROUND((SUM(o.quantity \* p.product\_price) /

(SELECT SUM(o.quantity \* p.product\_price) FROM orders o

LEFT JOIN products p ON o.product\_id = p.product\_id)) \* 100, 2) AS revenue\_percentage

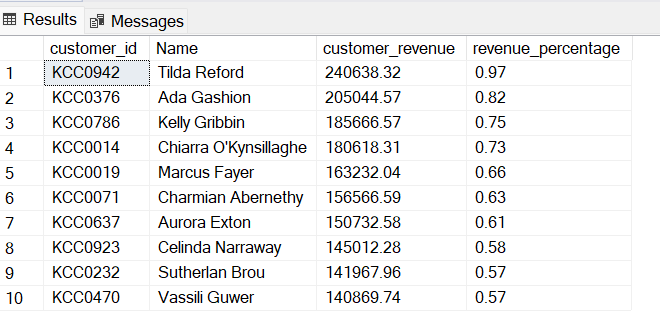
FROM orders o

LEFT JOIN customers c ON o.customer\_id = c.customer\_id

LEFT JOIN products p ON p.product\_id = o.product\_id

GROUP BY o.customer\_id, c.first\_name, c.last\_name

ORDER BY revenue\_percentage DESC;



15) Identify the days with the highest number of sales.

YEAR(order\_date) AS yearz,

DATENAME(month,order\_date) as monthzname,

day(order\_date) as dayz,

round(sum(p.product\_price \*quantity),2) as revenue

from orders o

left join

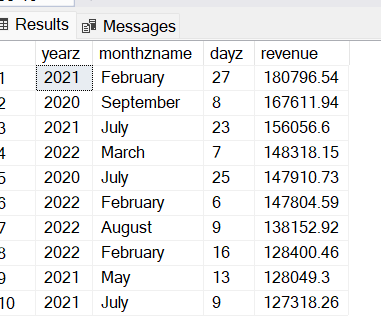
products p on o.product\_id = p. product\_id

group by YEAR ( order\_date),

DATENAME(month,order\_date),

day (order\_date)

order by revenue desc;



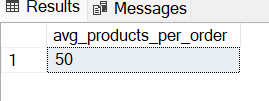
16) Calculate the average number of products included in each order.

SELECT

ROUND(SUM(quantity) / COUNT(DISTINCT order\_id), 2) AS avg\_products\_per\_order

FROM

orders;



17) Customers segment

select orders.customer\_id, concat(first\_name,' ',last\_name),count(orders.customer\_id) as counts,

case

when count(orders.customer\_id) = 1 then 'New Customer'

when count(orders.customer\_id) <= 4 then 'Repeat Customer'

when count(orders.customer\_id) > 4 then 'High-Value Customer'

END AS customer\_category

from orders

left join customers

on customers.customer\_id=orders.customer\_id

group by orders.customer\_id, concat(first\_name,' ',last\_name)

order by counts desc

;



18) Growth rate(YOY)

with cte as

(SELECT

YEAR(o.order\_date) AS year,

round(SUM(p.product\_price \* o.quantity),2) AS revenue,

LAG(SUM(p.product\_price \* o.quantity)) OVER (ORDER BY YEAR(o.order\_date)) AS previous\_year

FROM orders o

JOIN products p

ON o.product\_id = p.product\_id

GROUP BY YEAR(o.order\_date)

)

select year,revenue,

ROUND( isnull(((revenue - previous\_year)/previous\_year)\*100,0),2) as growth\_rate

from cte

