CREATE DATABASE Mysql_project;

use Mysql_project;

/*1)Import the dataset and do usual exploratory analysis steps like checking

the structure & characteristics of the dataset.

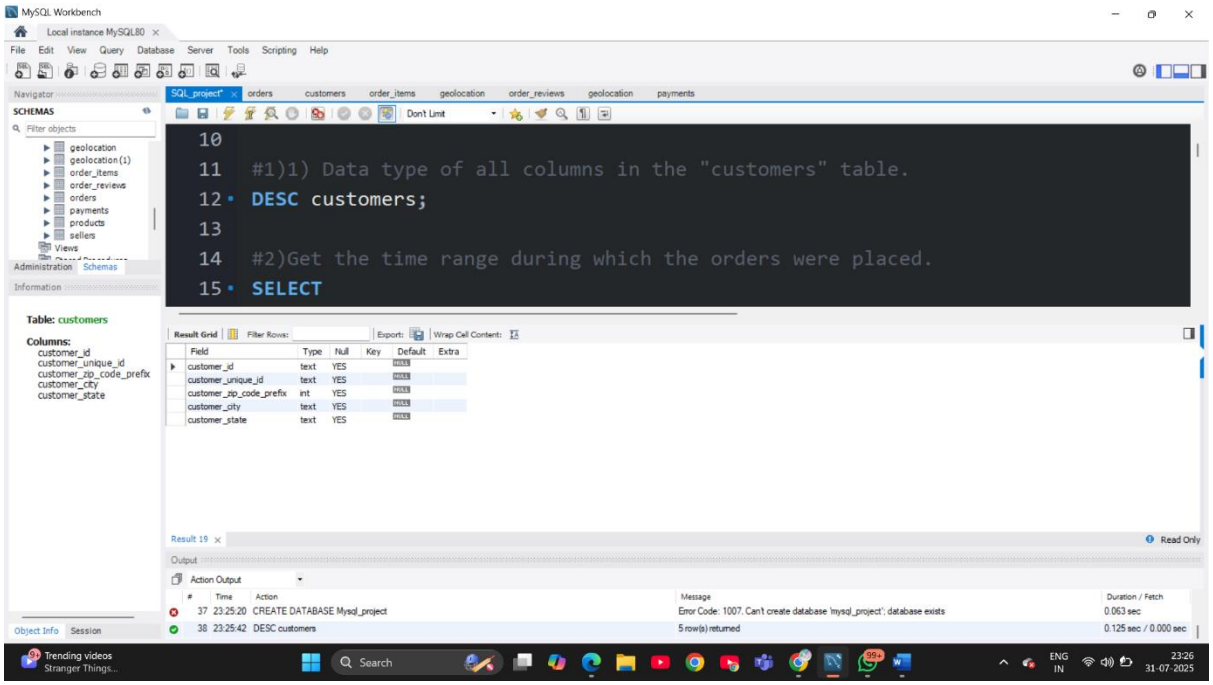
1) Data type of all columns in the "customers" table.

2)Get the time range during which the orders were placed.

3)Count the Cities & States of customers who ordered during the given period.*/

#a) Data type of all columns in the "customers" table.

DESC customers;



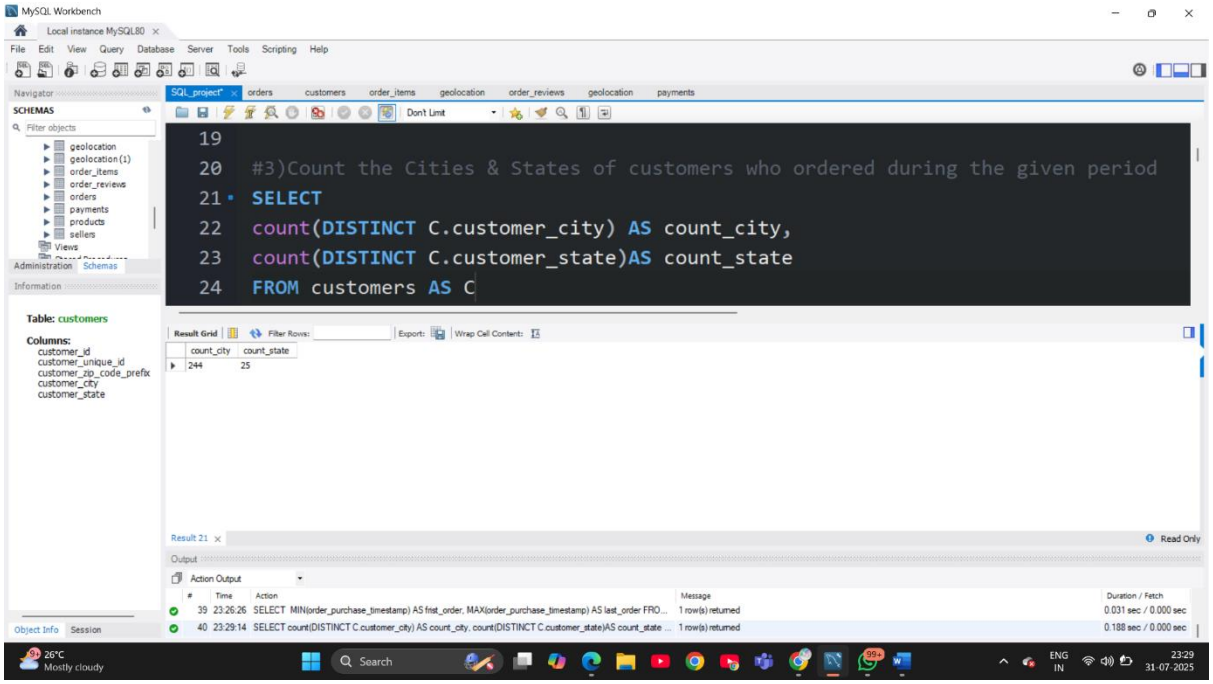
#b)Get the time range during which the orders were placed.

SELECT

MIN(order_purchase_timestamp) AS frist_order,

MAX(order_purchase_timestamp) AS last_order

FROM orders;




#c)Count the Cities & States of customers who ordered during the given period

SELECT

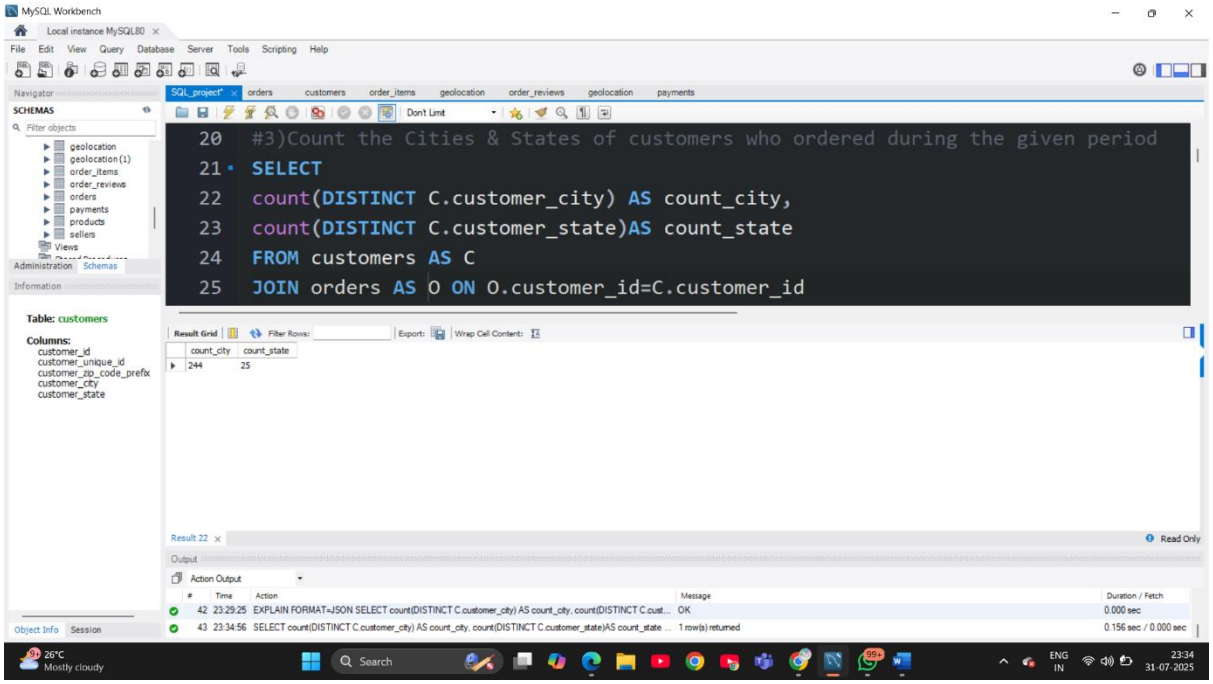
count(DISTINCT C.customer_city) AS count_city,

count(DISTINCT C.customer_state)AS count_state

FROM customers AS C

JOIN orders AS O ON O.customer_id=C.customer_id

WHERE order_id is not null;




/*2)In-depth Exploration:

a) Is there a growing trend in the no. of orders placed over the past years?

b)Can we see some kind of monthly seasonality in terms of the no.

of orders being placed?

c)During what time of the day do Brazilian customers mostly

place their orders? (Dawn, Morning, Afternoon or Night)

0-6 hrs: Dawn

7-12 hrs: Mornings

13-18 hrs: Afternoon

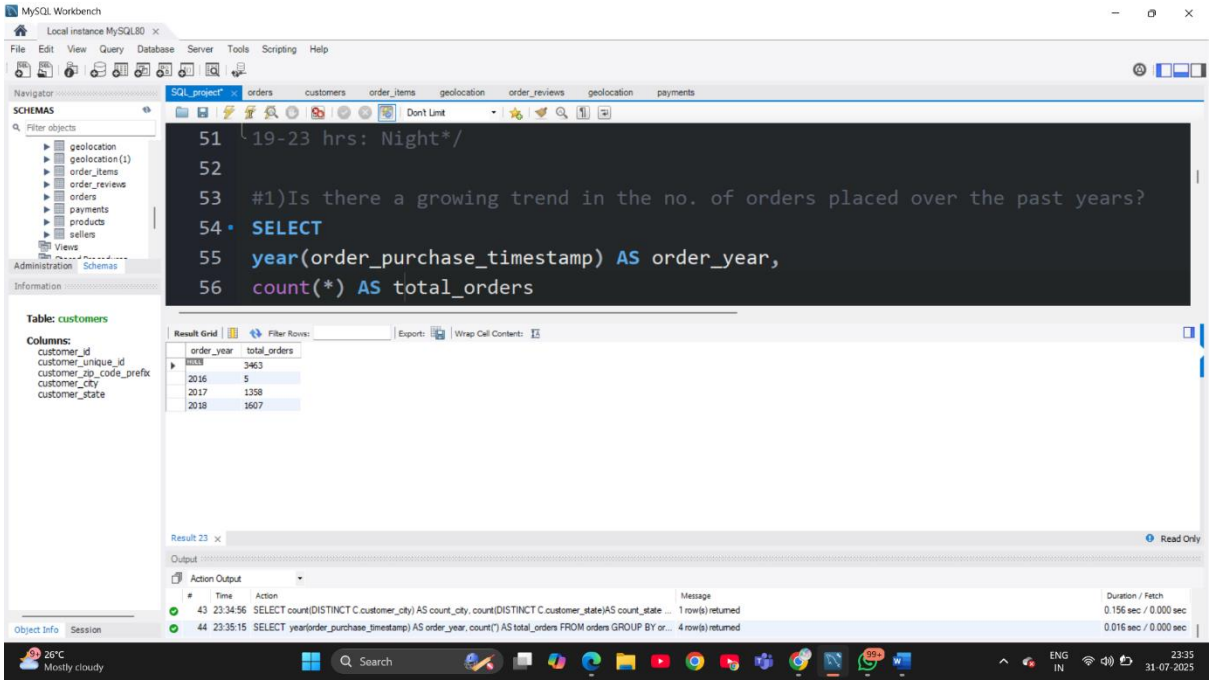
19-23 hrs: Night*/

#a)Is there a growing trend in the no. of orders placed over the past years?

```
SELECT
year(order_purchase_timestamp) AS order_year,
count(*) AS total_orders
FROM orders
```

GROUP BY order_year

ORDER BY order_year ASC;




/*b)Can we see some kind of monthly seasonality in terms of the no.

of orders being placed?*/

SELECT

YEAR(order_purchase_timestamp)AS YEAR,

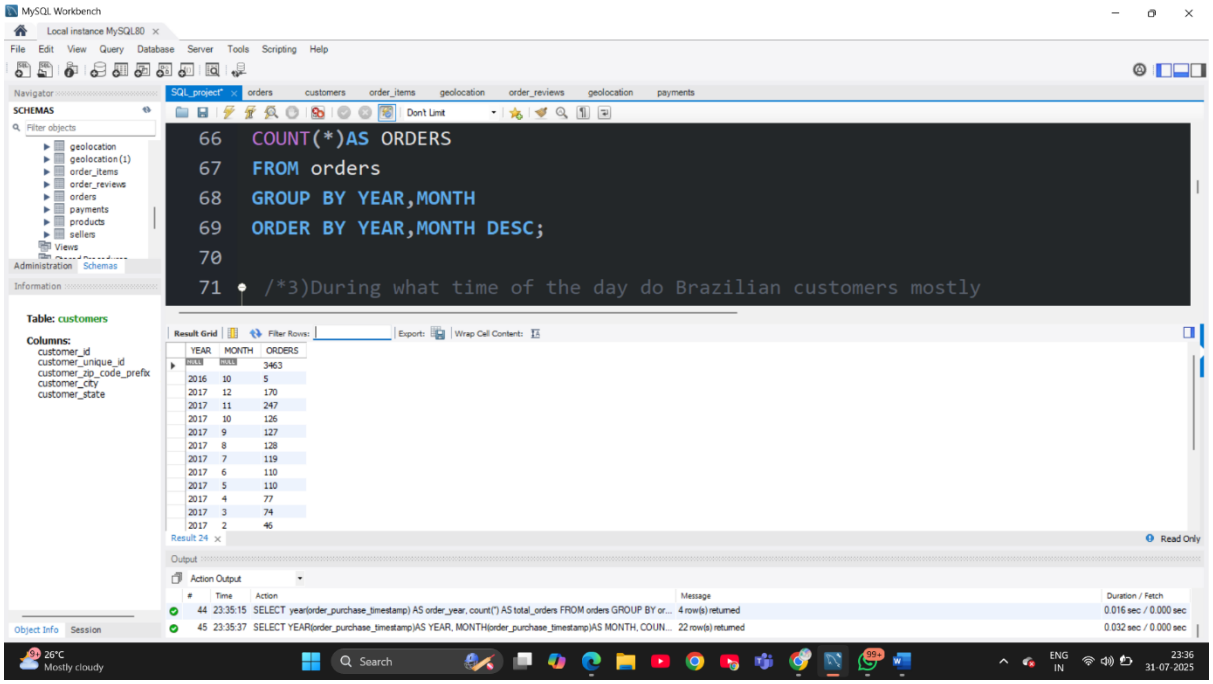
MONTH(order_purchase_timestamp)AS MONTH,

COUNT(*)AS ORDERS

FROM orders

GROUP BY YEAR,MONTH

ORDER BY YEAR,MONTH DESC;

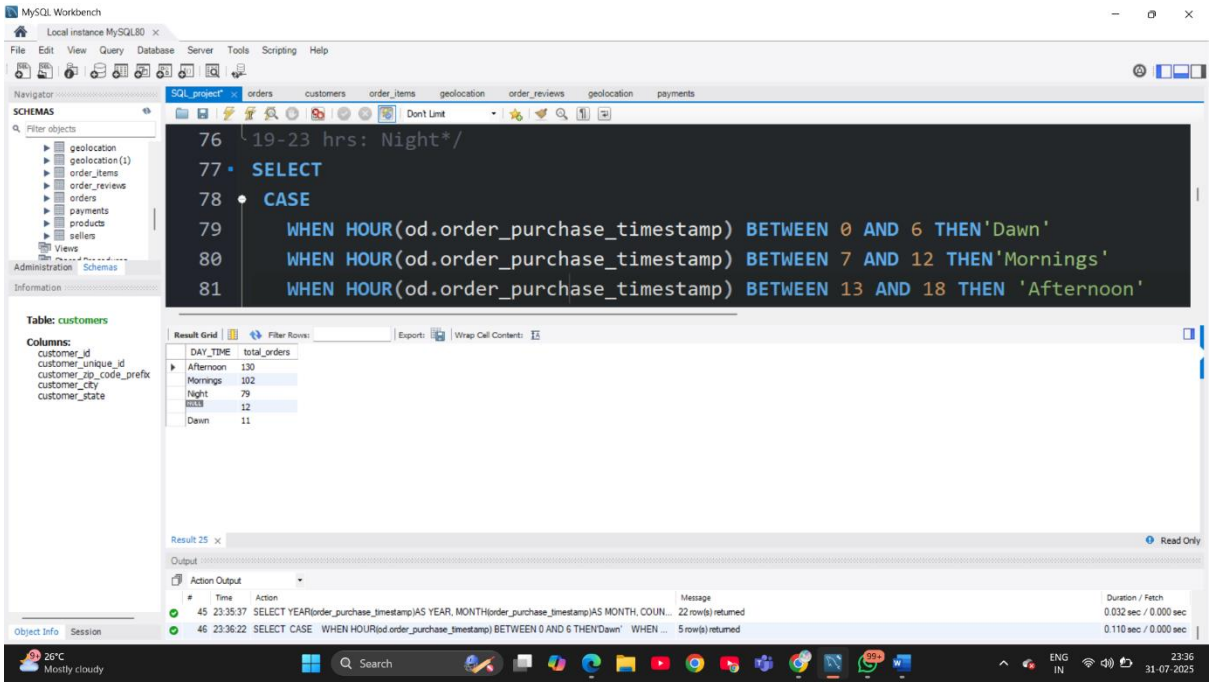



```
/*c)During what time of the day do Brazilian customers mostly
 place their orders? (Dawn, Morning, Afternoon or Night)
 0-6 hrs: Dawn
7-12 hrs: Mornings
13-18 hrs: Afternoon
19-23 hrs: Night*/
SELECT
 CASE
   WHEN HOUR(od.order_purchase_timestamp) BETWEEN 0 AND 6 THEN'Dawn'
   WHEN HOUR(od.order_purchase_timestamp) BETWEEN 7 AND 12 THEN'Mornings'
   WHEN HOUR(od.order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
   WHEN HOUR(od.order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
 END AS DAY_TIME,
 COUNT(*)AS total_orders
FROM orders AS od
JOIN customers AS c ON c.customer_id=od.customer_id
```

WHERE c.customer_state like ('%sp%')

GROUP BY DAY_TIME

ORDER BY total_orders DESC;




/*3)Evolution of E-commerce orders in the Brazil region:

a)Get the month-on-month no. of orders placed in each state.

b)How are the customers distributed across all the states?*/

#a)Get the month-on-month no. of orders placed in each state

SELECT

year(o.order_purchase_timestamp) AS year,

MONTH(O.order_purchase_timestamp)AS month,

c.customer_state AS STATE,

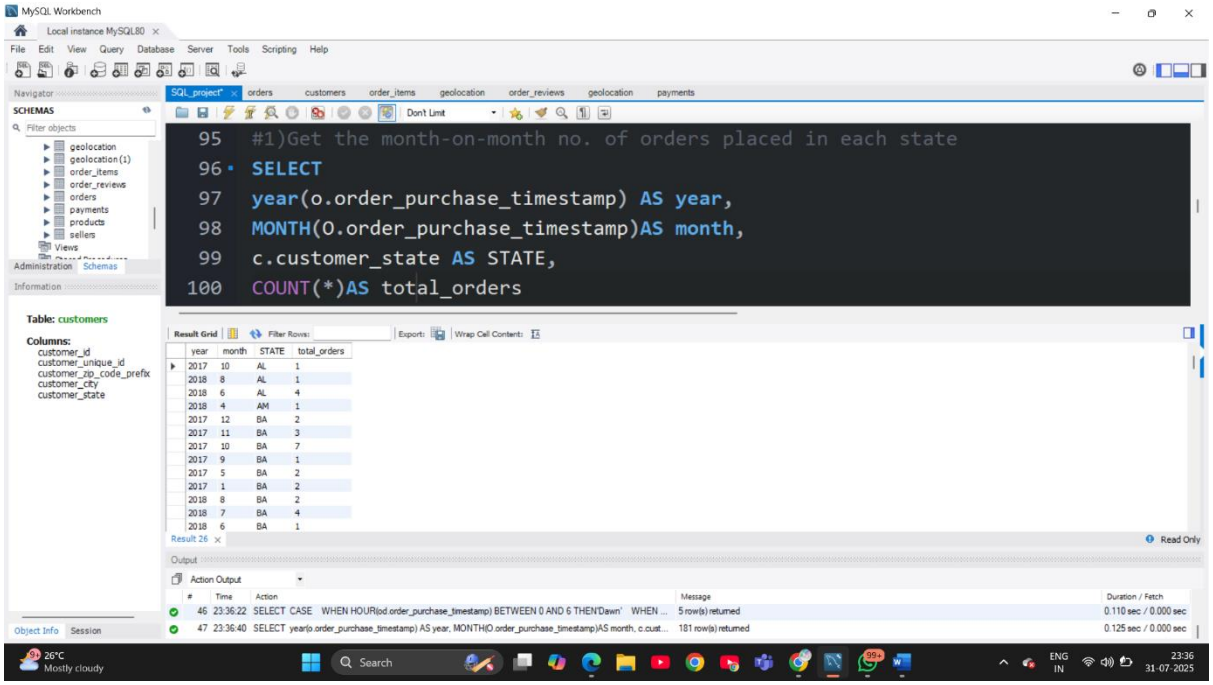
COUNT(*)AS total_orders

FROM orders AS o

JOIN customers AS C ON o.customer_id=c.customer_id

GROUP BY
year(o.order_purchase_timestamp),MONTH(O.order_purchase_timestamp),c.customer_state

ORDER BY STATE,year,month DESC;




#b)How are the customers distributed across all the states?

SELECT

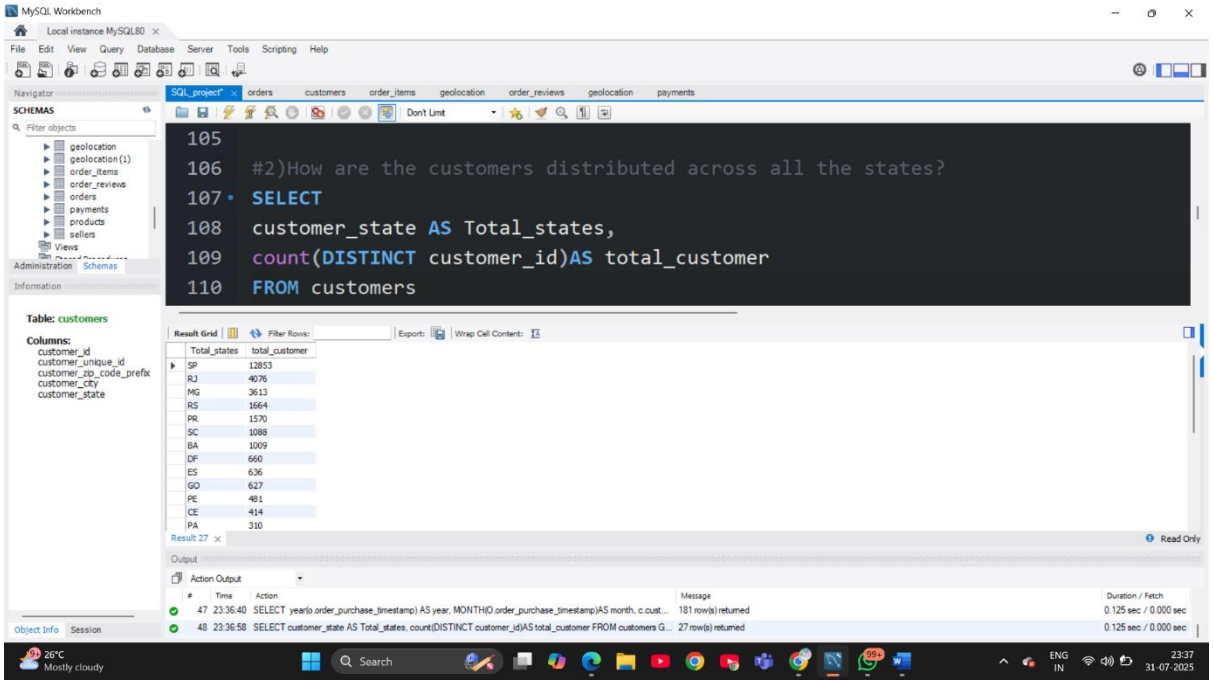
customer_state AS Total_states,

count(DISTINCT customer_id)AS total_customer

FROM customers

GROUP BY customer_state

ORDER BY total_customer DESC;




/*4)Impact on Economy: Analyse the money movement by e-commerce

by looking at order prices, freight and others.

a)Get the % increase in the cost of orders from year 2017 to 2018

(include months between January to Aug only).

You can use the "payment_value" column in the payments table

to get the cost of orders.

b)Calculate the Total & Average value of order price for each state.

c)Calculate the Total & Average value of order freight for each state.*/

/*a)Get the % increase in the cost of orders from year 2017 to 2018

(include months between January to Aug only).

You can use the "payment_value" column in the payments table

to get the cost of orders.*/

SELECT

year(O.order_purchase_timestamp)AS order_year,

sum(P.payment_value)AS total_payment

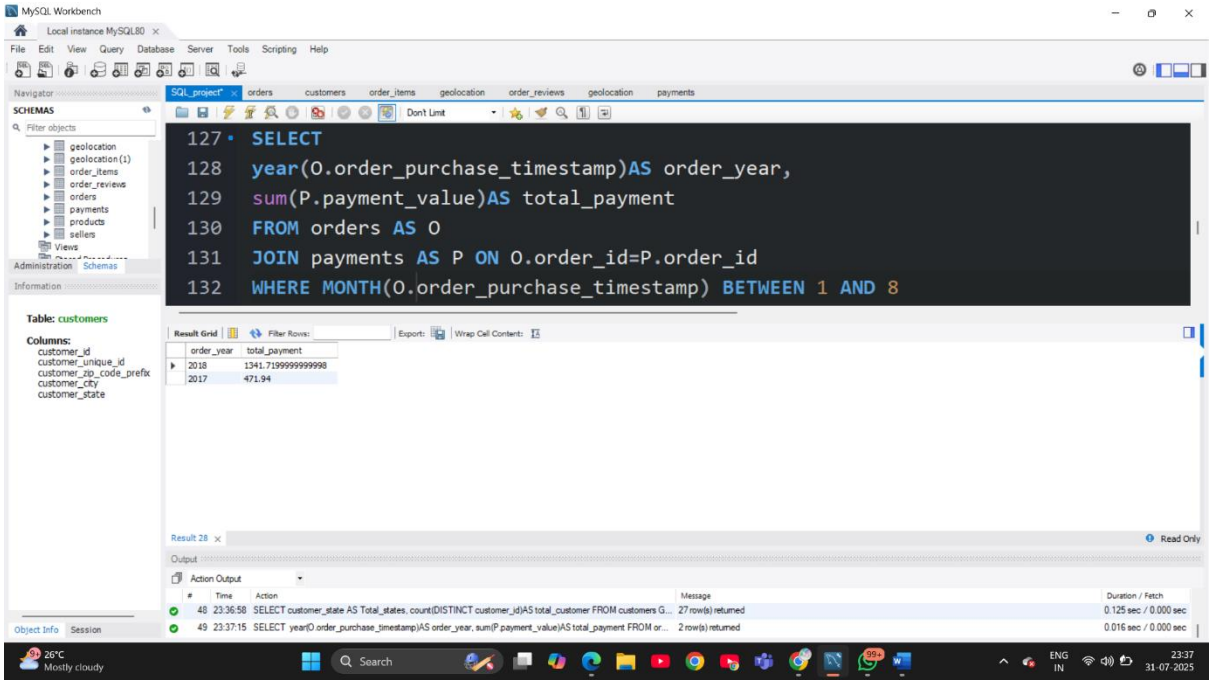
FROM orders AS O

JOIN payments AS P ON O.order_id=P.order_id

WHERE MONTH(O.order_purchase_timestamp) BETWEEN 1 AND 8

GROUP BY ORDER_YEAR

ORDER BY ORDER_YEAR DESC;




#b)Calculate the Total & Average value of order price for each state

SELECT

C.customer_state AS state,

SUM(oi.price) AS total_price,

AVG(oi.price)AS total_avg

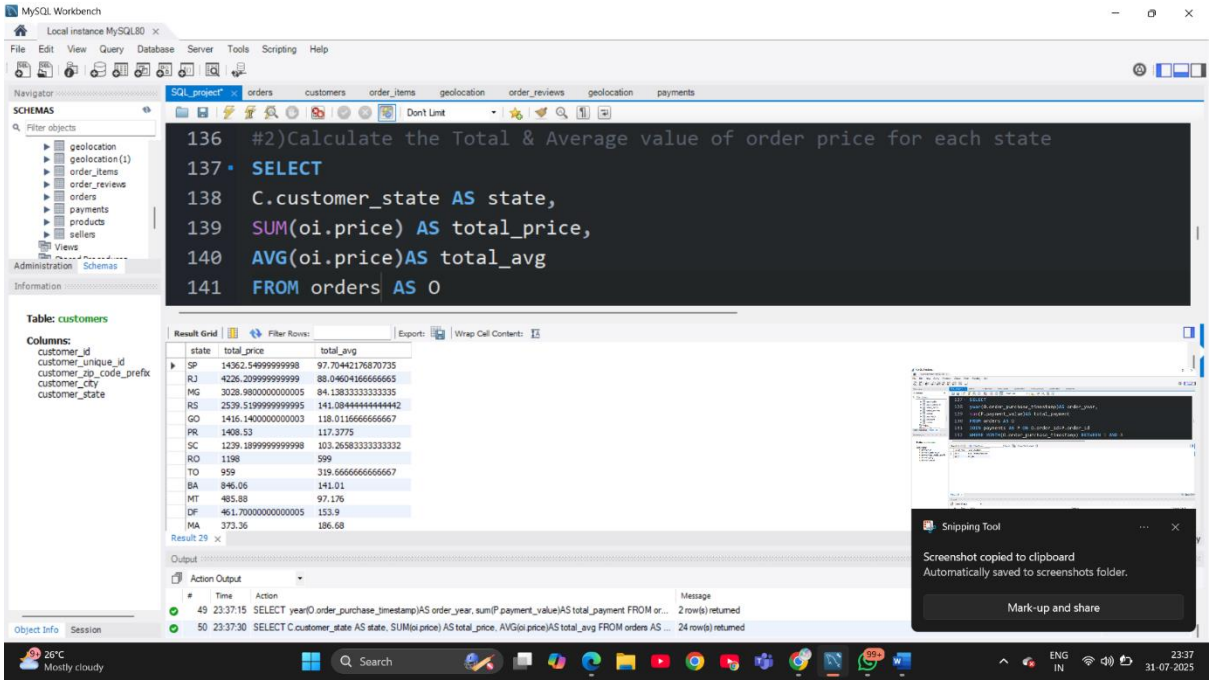
FROM orders AS O

JOIN customers AS C ON O.customer_id=C.customer_id

JOIN order_items AS oi ON O.order_id=oi.order_id

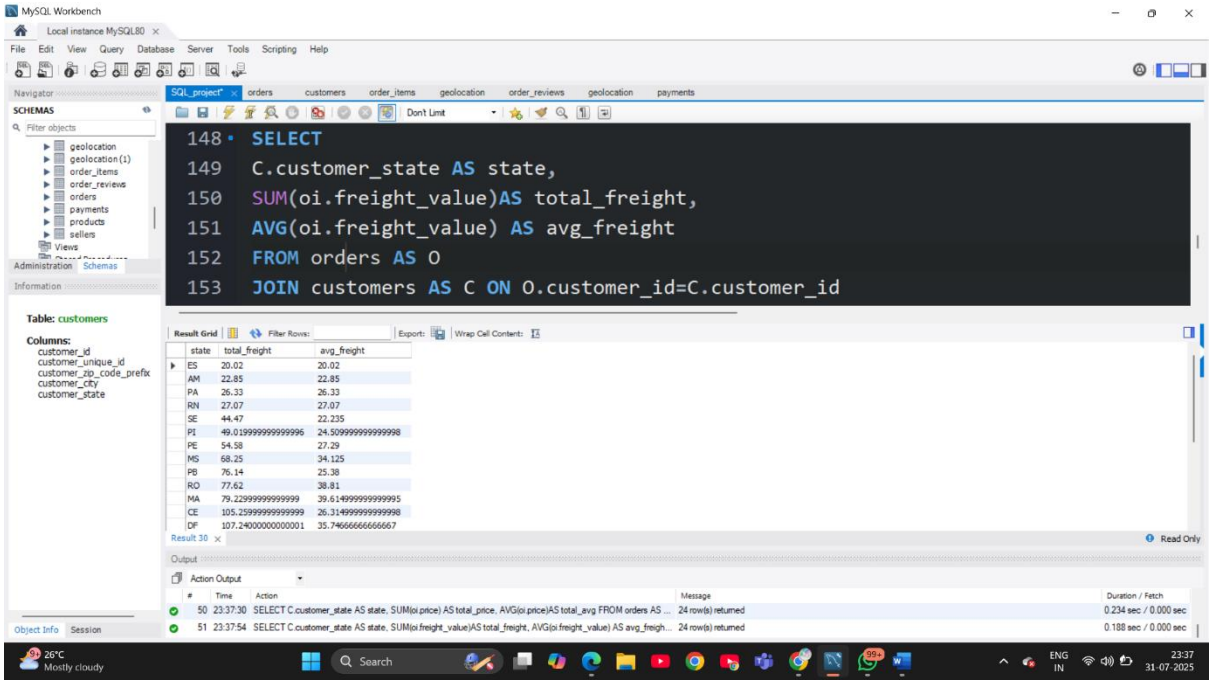
GROUP BY state

ORDER BY total_price DESC;




#c)Calculate the Total & Average value of order freight for each state

```
SELECT
C.customer_state AS state,
SUM(oi.freight_value)AS total_freight,
AVG(oi.freight_value) AS avg_freight
FROM orders AS O
JOIN customers AS C ON O.customer_id=C.customer_id
JOIN order_items AS oi ON O.order_id=oi.order_id
GROUP BY state
ORDER BY total_freight;
```




/*5)Analysis based on sales, freight and delivery time.

a)Find the no. of days taken to deliver each order from the

order's purchase date as delivery time.

Also, calculate the difference (in days) between

the estimated & actual delivery date of an order.

Do this in a single query.

You can calculate the delivery time and the difference

between the estimated & actual delivery date using the given formula:

a)time_to_deliver = order_delivered_customer_date - order_purchase_timestamp

b)diff_estimated_delivery = order_delivered_customer_date

- order_estimated_delivery_date

b) Find out the top 5 states with the highest & lowest average freight value.

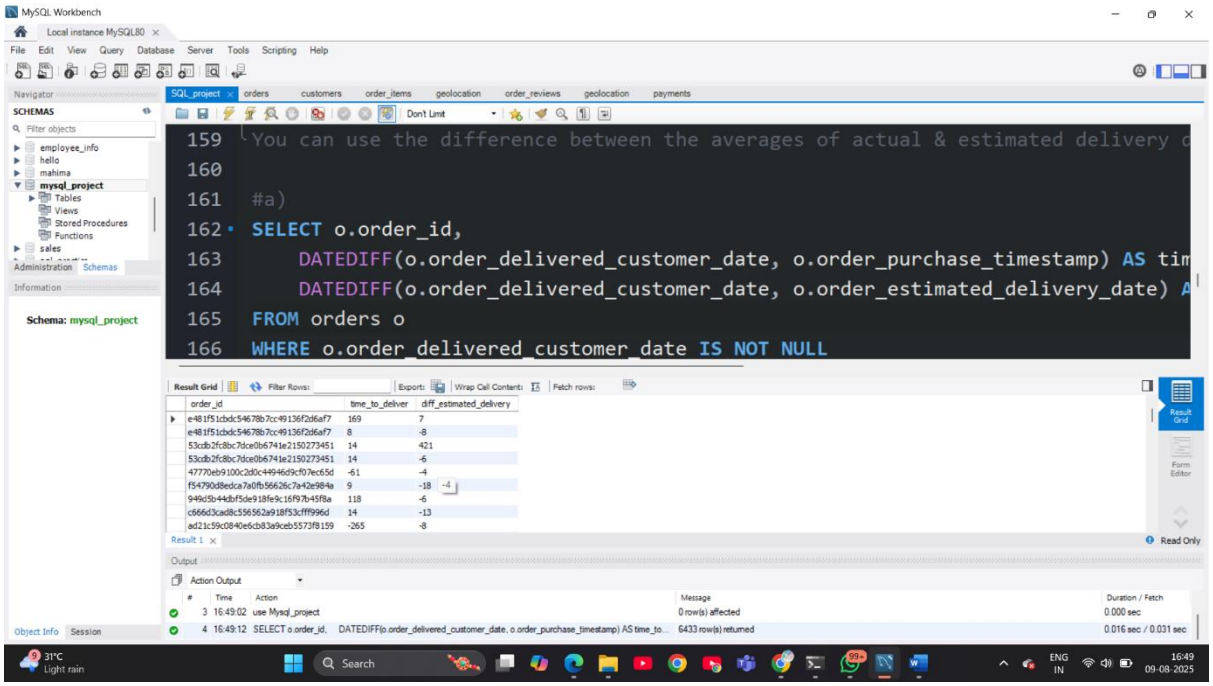
c)Find out the top 5 states with the highest & lowest average delivery time.

d)Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

You can use the difference between the averages of actual & estimated delivery dates to figure out how fast the delivery was for each state*/

#a)

```
SELECT o.order_id,
   DATEDIFF(o.order_delivered_customer_date, o.order_purchase_timestamp) AS time_to_deliver,
   DATEDIFF(o.order_delivered_customer_date, o.order_estimated_delivery_date) AS diff_estimated_delivery
FROM orders o
WHERE o.order_delivered_customer_date IS NOT NULL
 AND o.order_estimated_delivery_date IS NOT NULL
 AND o.order_purchase_timestamp IS NOT NULL;
```




#b)Find out the top 5 states with the highest & lowest average freight value

```
SELECT
c.customer_state AS state,
```

ROUND(AVG(oi.freight_value),2) AS avg_freight

FROM order_items oi

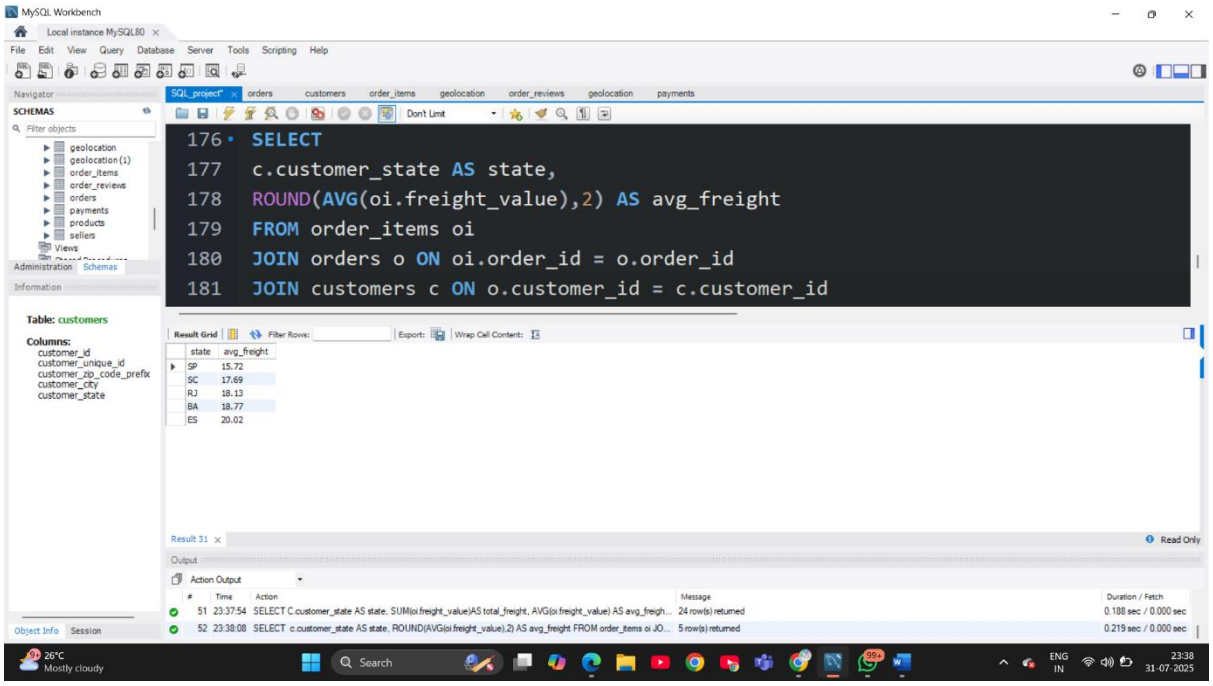
JOIN orders o ON oi.order_id = o.order_id

JOIN customers c ON o.customer_id = c.customer_id

GROUP BY c.customer_state

ORDER BY avg_freight ASC

LIMIT 5;




#c)Find out the top 5 states with the highest & lowest average delivery time.

SELECT

c.customer_state,

ROUND(AVG(DATEDIFF(o.order_delivered_customer_date, o.order_purchase_timestamp)),2) AS avg_delivery_days

FROM orders o

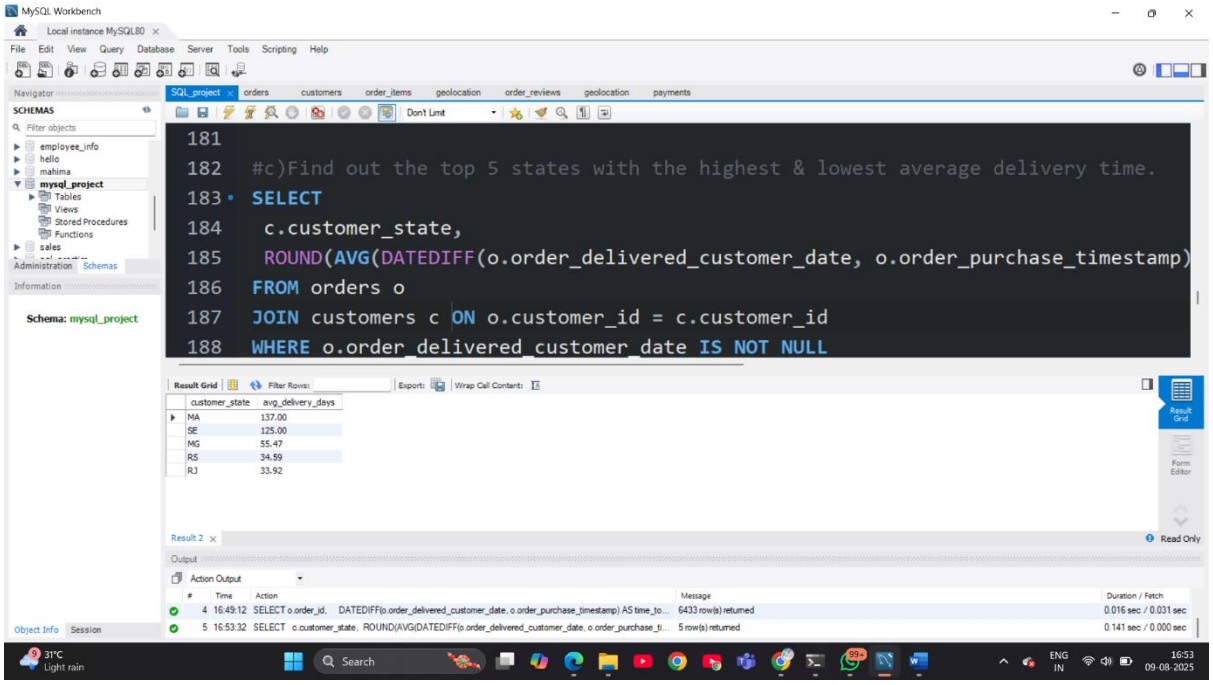
JOIN customers c ON o.customer_id = c.customer_id

WHERE o.order_delivered_customer_date IS NOT NULL

GROUP BY c.customer_state

ORDER BY avg_delivery_days DESC

LIMIT 5;




#d)Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

SELECT

c.customer_state,

ROUND(AVG(DATEDIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date)), 2) AS avg_days_early

FROM orders o

JOIN customers c ON o.customer_id = c.customer_id

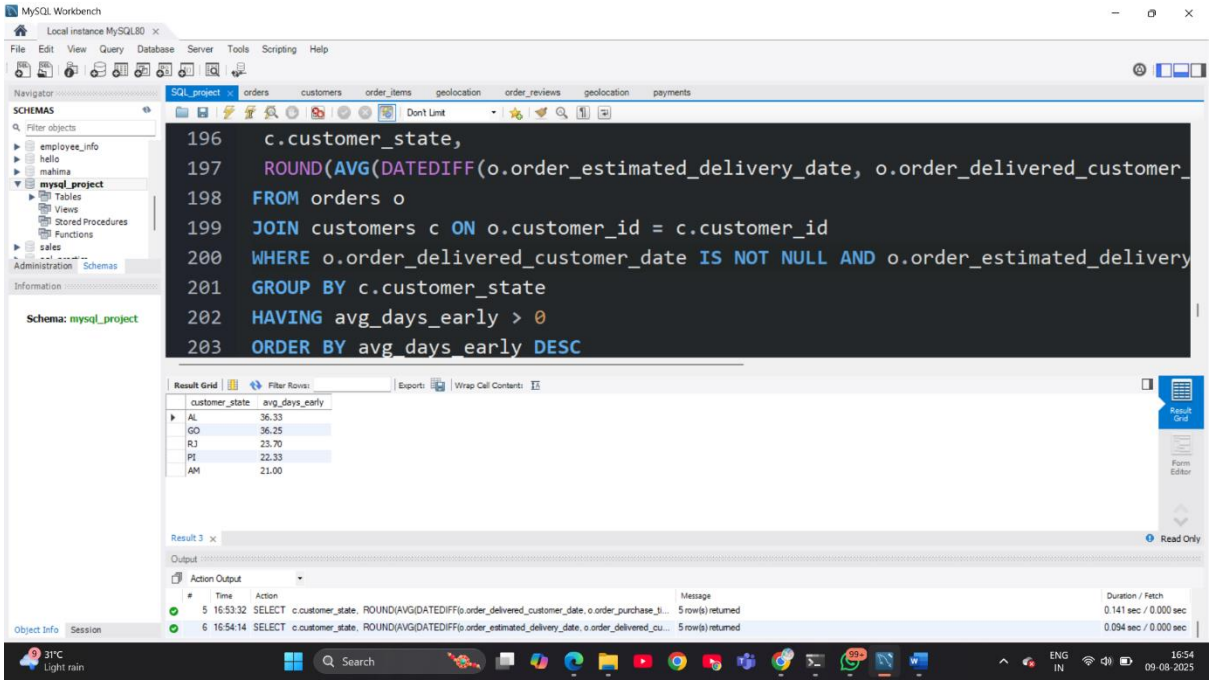
WHERE o.order_delivered_customer_date IS NOT NULL AND o.order_estimated_delivery_date IS NOT NULL

GROUP BY c.customer_state

HAVING avg_days_early > 0

ORDER BY avg_days_early DESC

LIMIT 5;




/*6)Analysis based on the payments:

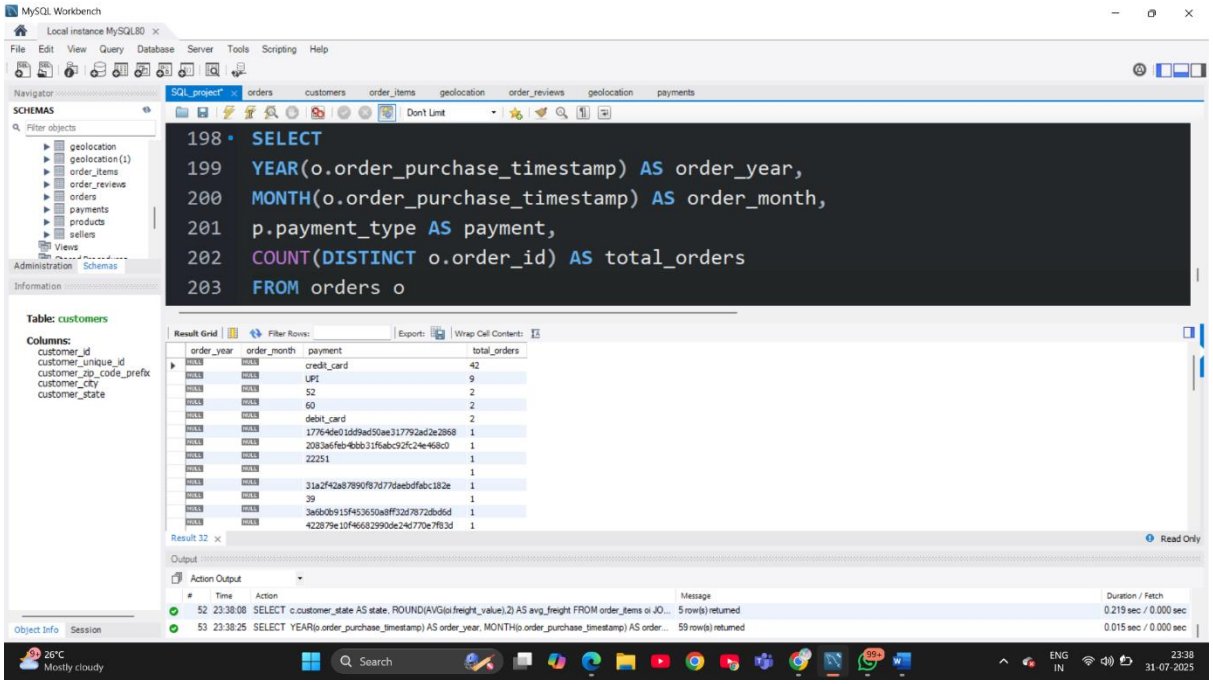
a)Find the month-on-month no. of orders placed using different payment types.

b)Find the no. of orders placed based on the payment instalments

that have been paid.*/

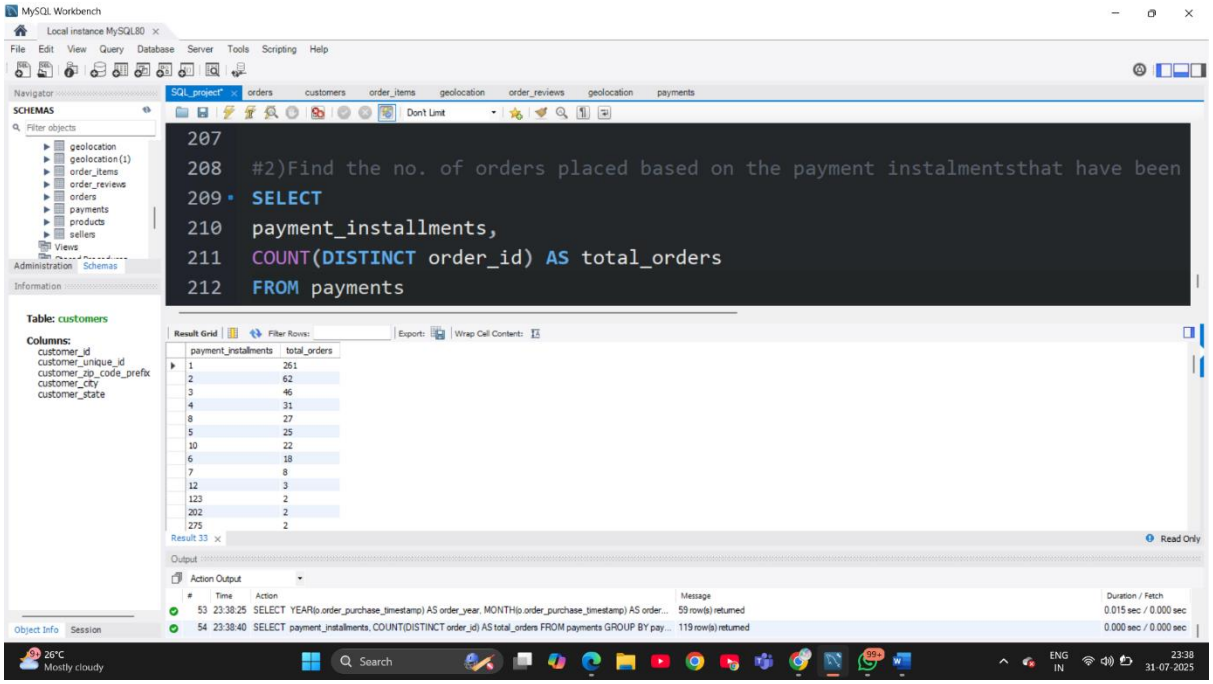
#a)Find the month-on-month no. of orders placed using different payment types.

```
SELECT
YEAR(o.order_purchase_timestamp) AS order_year,
MONTH(o.order_purchase_timestamp) AS order_month,
p.payment_type AS payment,
COUNT(DISTINCT o.order_id) AS total_orders
FROM orders o
JOIN payments p ON o.order_id = p.order_id
GROUP BY order_year, order_month, p.payment_type
ORDER BY order_year, order_month, total_orders DESC;
```




#b)Find the no. of orders placed based on the payment instalmentsthat have been paid

```
SELECT
payment_installments,
COUNT(DISTINCT order_id) AS total_orders
FROM payments
GROUP BY payment_installments
ORDER BY total_orders DESC;
```


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL_project* orders customers order_items geolocation order_reviews geolocation payments
Don't Limit
#2)Find the no. of orders placed based on the payment instalmentsthat have been
SELECT
payment_installments,
COUNT(DISTINCT order_id) AS total_orders
FROM payments
Result Grid
payment_installments total_orders
1 261
2 62
3 46
4 31
8 27
5 25
10 22
6 18
7 8
12 3
123 2
202 2
275 2
Result 33
Read Only
Output
Action Output
53 23:38:25 SELECT YEAR(o.order_purchase_timestamp) AS order_year, MONTH(o.order_purchase_timestamp) AS order... 59 row(s) returned 0.015 sec / 0.000 sec
54 23:38:40 SELECT payment_installments, COUNT(DISTINCT order_id) AS total_orders FROM payments GROUP BY pay... 119 row(s) returned 0.000 sec / 0.000 sec
Table: customers
Columns:
customer_id
customer_unique_id
customer_zip_code_prefix
customer_city
customer_state
26°C
Mostly cloudy
Search
ENG
IN
23:38
31-07-2025