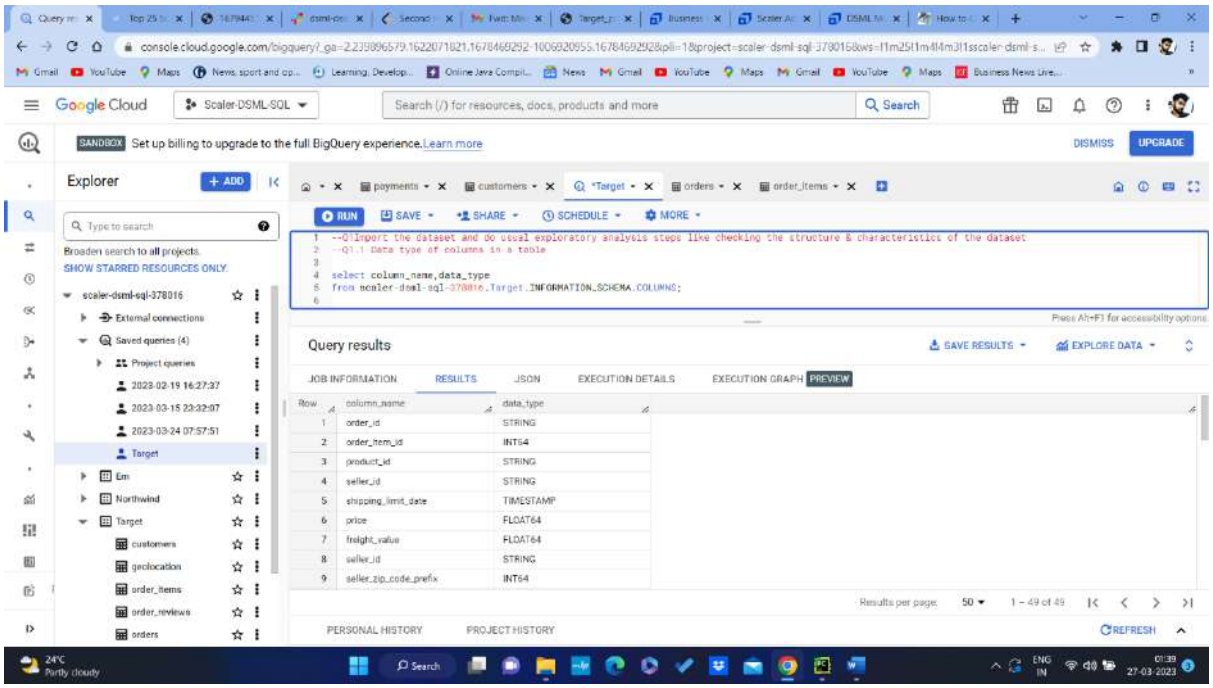
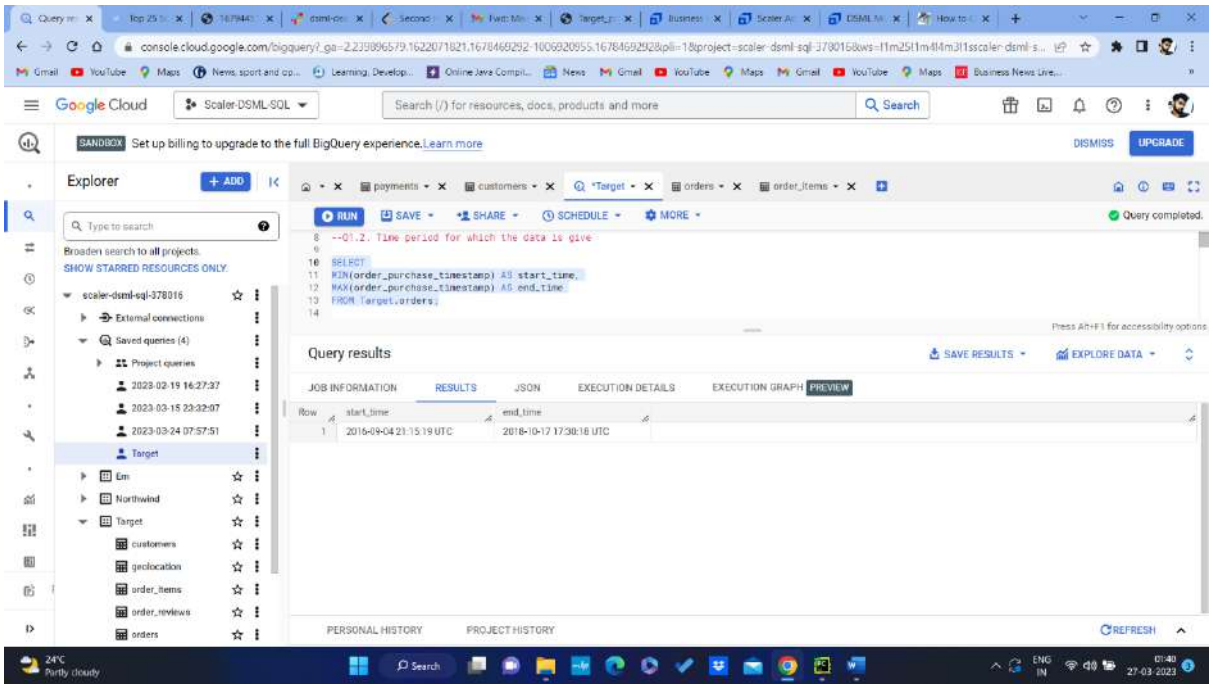
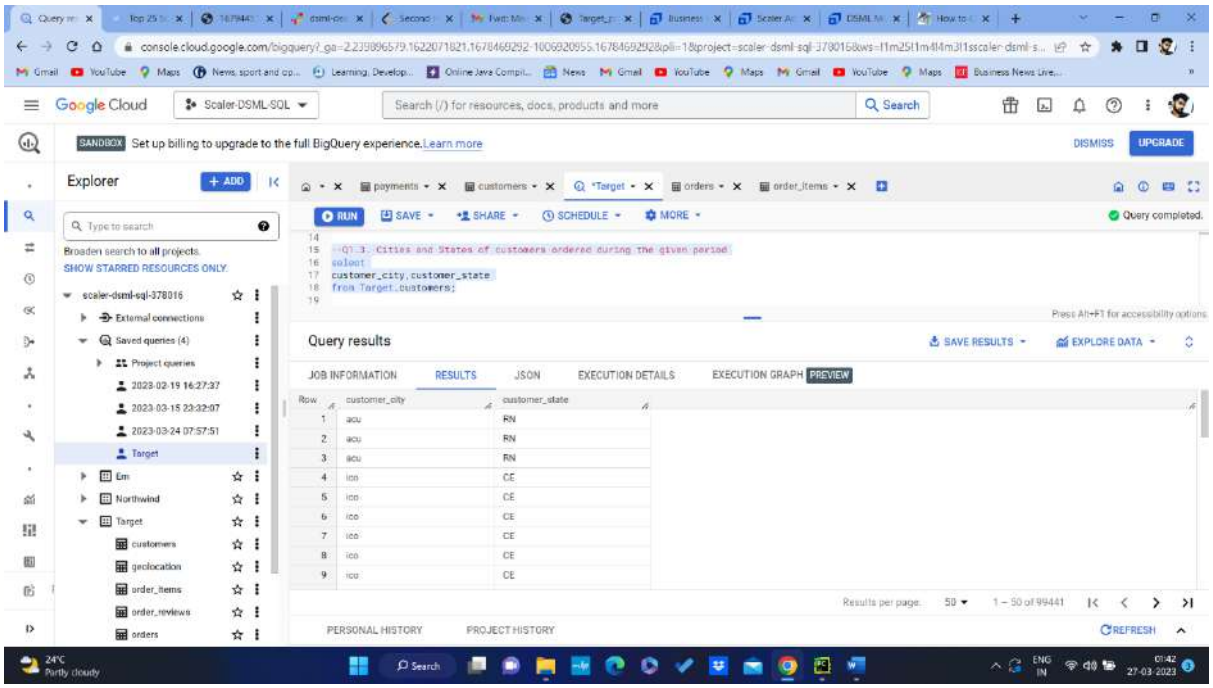
```
--
Q1Import the dataset and do usual exploratory analysis steps like checking the structure &
characteristics of the dataset
--Q1.1 Data type of columns in a table

select column_name,data_type
from scaler-dsml-sql-378016.Target.INFORMATION_SCHEMA.COLUMNS;
```



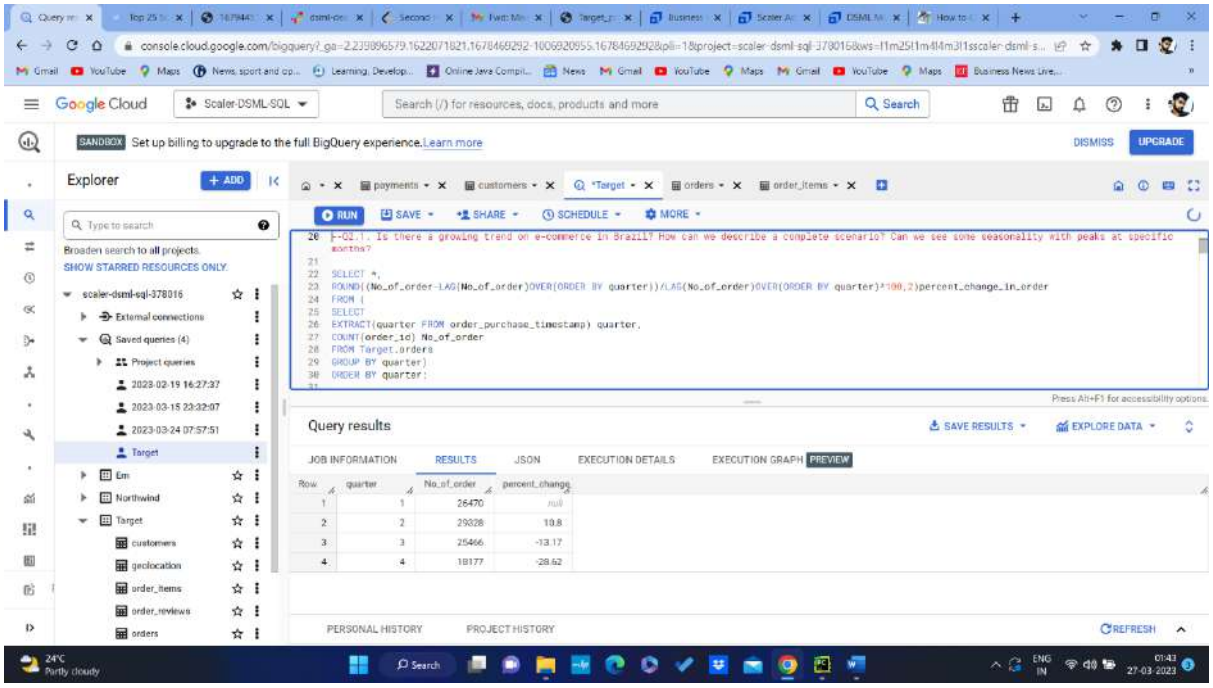
```sql
--Q1.2. Time period for which the data is give

SELECT
MIN(order_purchase_timestamp) AS start_time,
MAX(order_purchase_timestamp) AS end_time
FROM Target.orders;
```



```sql
--Q1.3. Cities and States of customers ordered during the given period
select
customer_city,customer_state
from Target.customers;
```

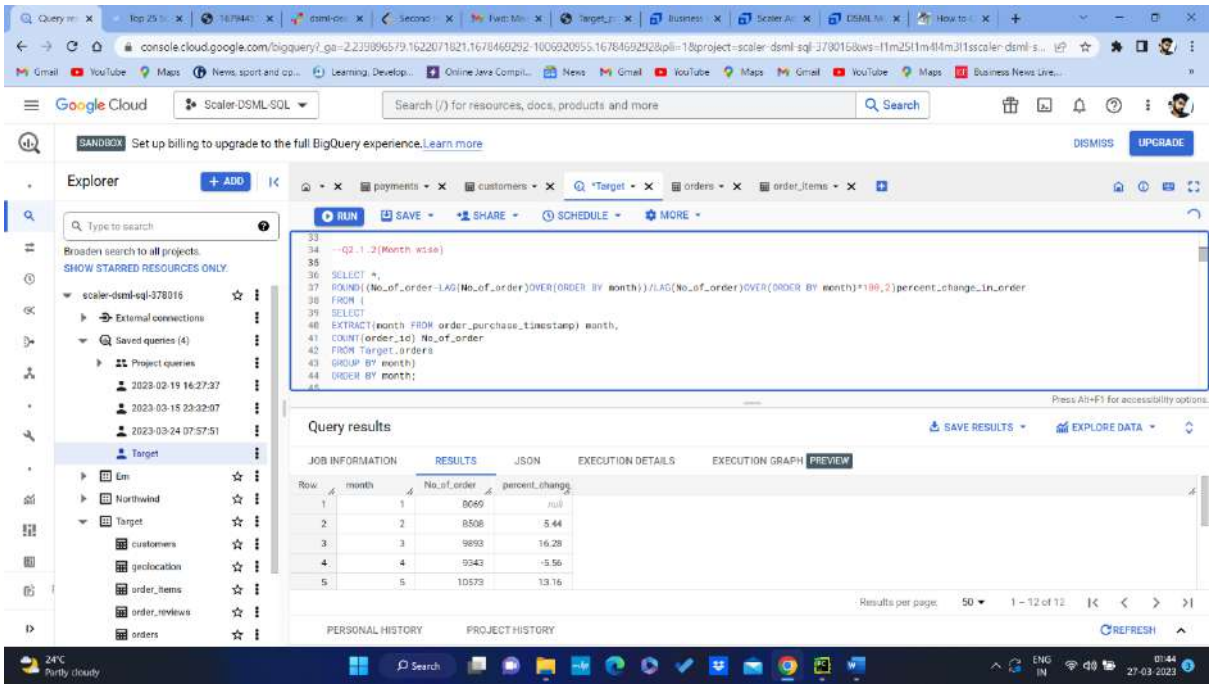


```
--Q2.1. Is there a growing trend on e-
commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality wi
th peaks at specific months?

SELECT *,
ROUND((No_of_order-
LAG(No_of_order)OVER(ORDER BY quarter))/LAG(No_of_order)OVER(ORDER BY quarter)*100,2)percen
t_change_in_order
FROM (
SELECT
EXTRACT(quarter FROM order_purchase_timestamp) quarter,
COUNT(order_id) No_of_order
FROM Target.orders
GROUP BY quarter)
ORDER BY quarter;
```



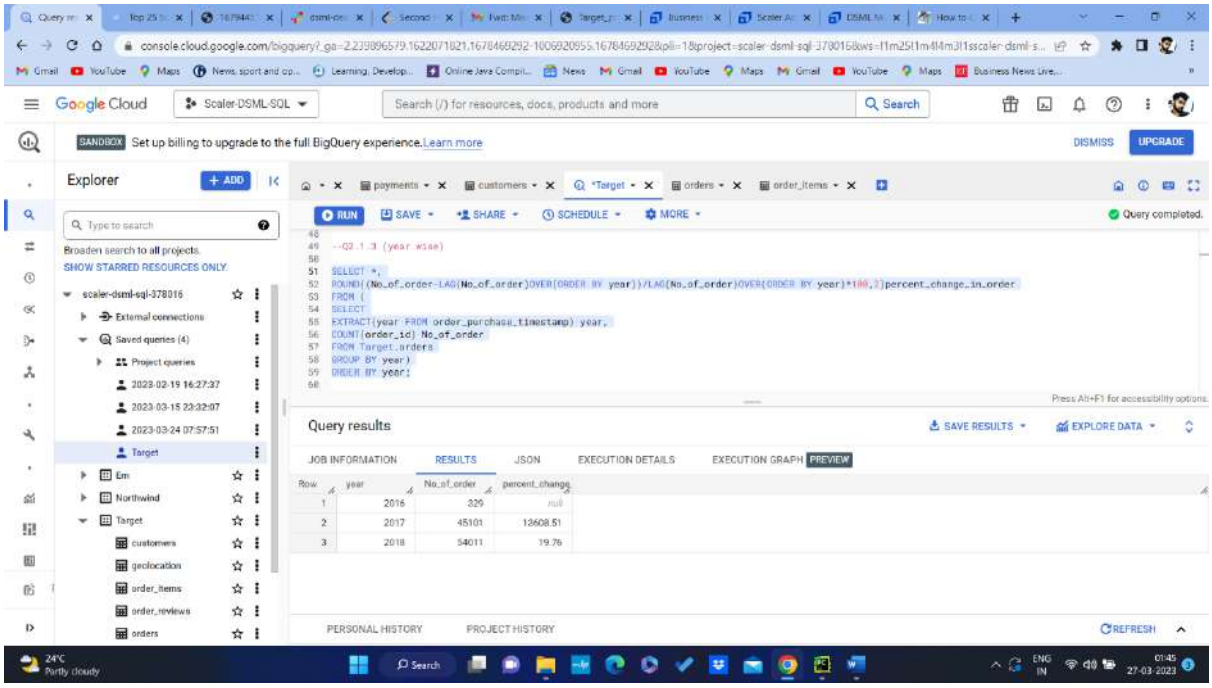

```
--growing trend from Q1 to Q2 . But after Q2 significant drop in no of orders .
```

```
--Q2.1.2(Month wise)


SELECT *,
ROUND((No_of_order-
LAG(No_of_order)OVER(ORDER BY month))/LAG(No_of_order)OVER(ORDER BY month)*100,2)percent_ch
ange_in_order
FROM (
SELECT
EXTRACT(month FROM order_purchase_timestamp) month,
COUNT(order_id) No_of_order
FROM Target.orders
GROUP BY month)
ORDER BY month;
```

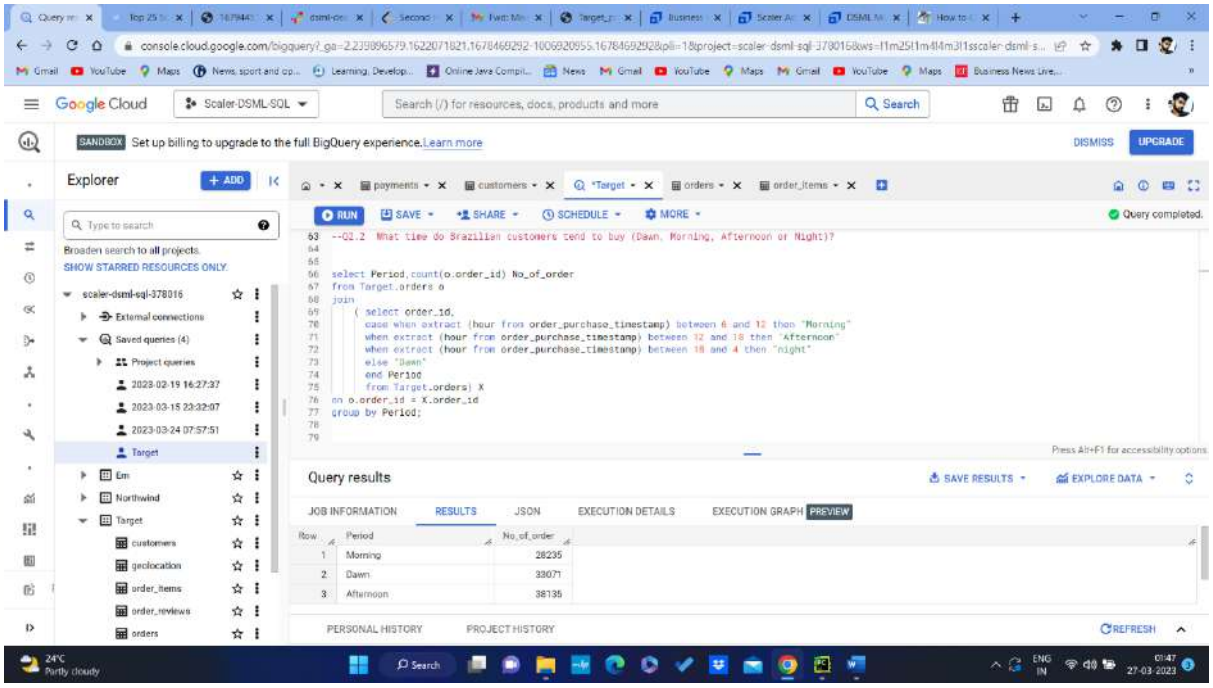



```
--AUGUST has highest no of orders and by july.
```

```sql
--Q2.1.3 (year wise)

SELECT *,
ROUND((No_of_order-
LAG(No_of_order)OVER(ORDER BY year))/LAG(No_of_order)OVER(ORDER BY year)*100,2)percent_chan
ge_in_order
FROM (
SELECT
EXTRACT(year FROM order_purchase_timestamp) year,
COUNT(order_id) No_of_order
FROM Target.orders
GROUP BY year)
ORDER BY year;
```

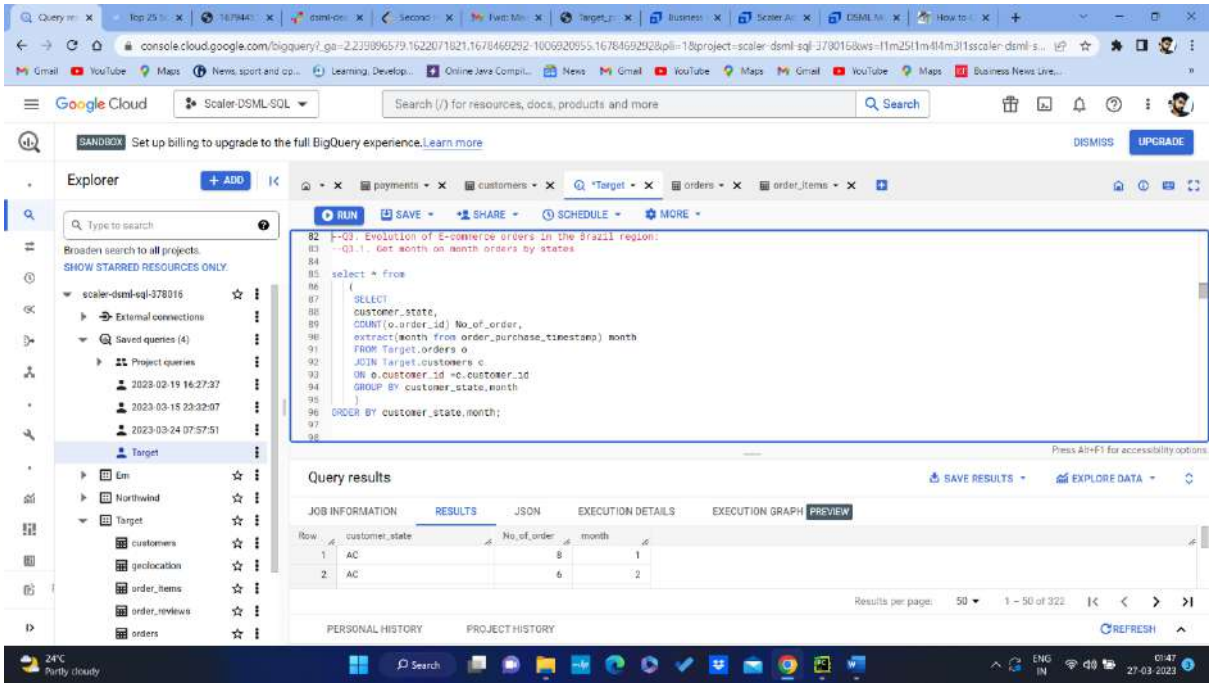


```
--Q2.2  What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?


select Period,count(o.order_id) No_of_order
from Target.orders o
join
    ( select order_id,
      case when extract (hour from order_purchase_timestamp) between 6 and 12 then "Morning
"
      when extract (hour from order_purchase_timestamp) between 12 and 18 then "Afternoon"
      when extract (hour from order_purchase_timestamp) between 18 and 4 then "night"
      else "Dawn"
      end Period
      from Target.orders) X
on o.order_id = X.order_id
group by Period;
```

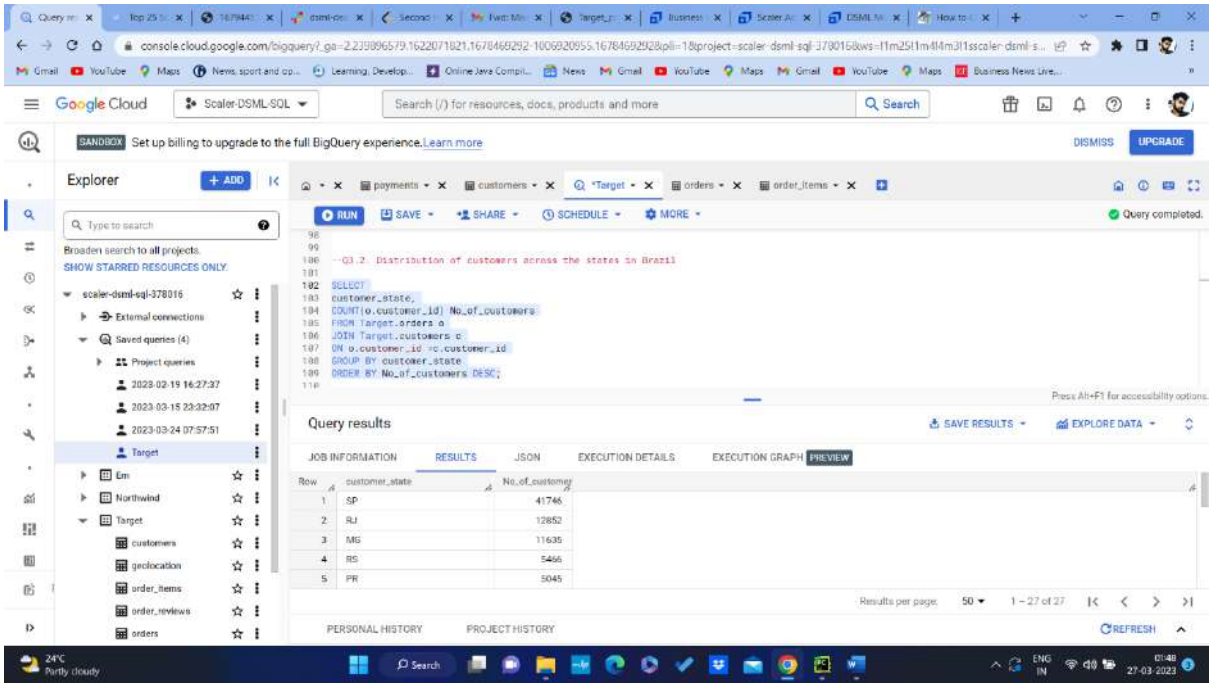



```
--Brazilian most preferablly like to buy in afternoon.
```

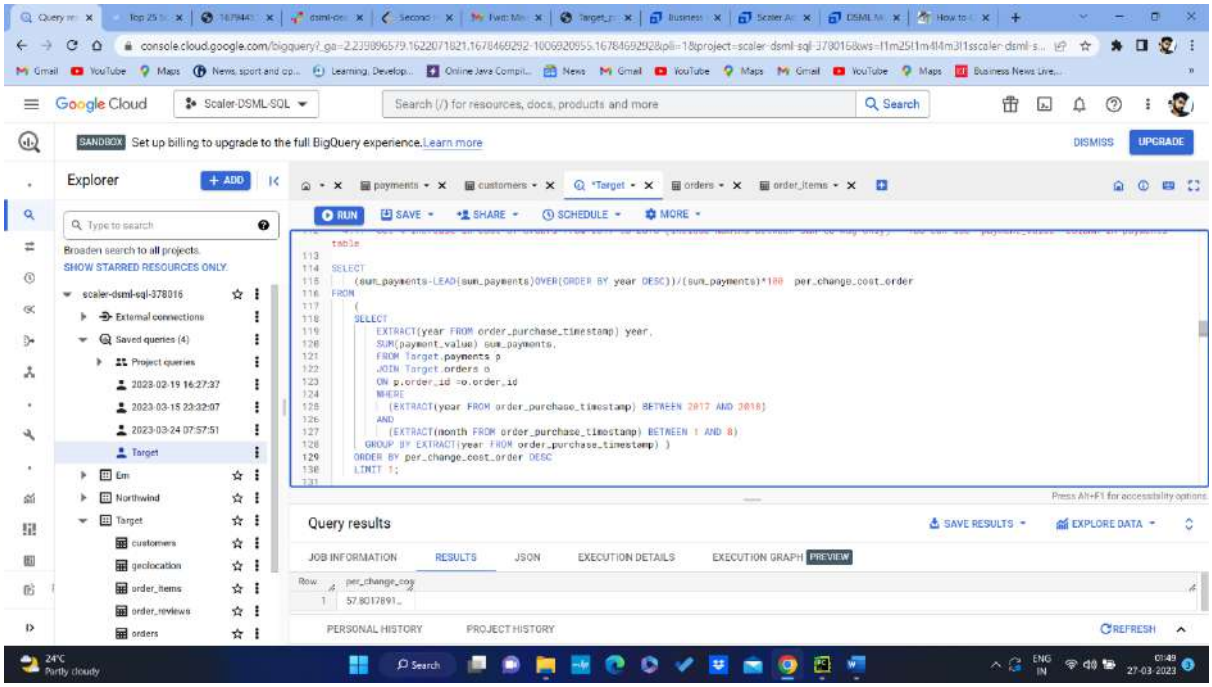
```sql
--Q3. Evolution of E-commerce orders in the Brazil region:
--Q3.1. Get month on month orders by states

select * from
   (
    SELECT
    customer_state,
    COUNT(o.order_id) No_of_order,
    extract(month from order_purchase_timestamp) month
    FROM Target.orders o
    JOIN Target.customers c
    ON o.customer_id =c.customer_id
    GROUP BY customer_state,month
    )
ORDER BY customer_state,month;
```



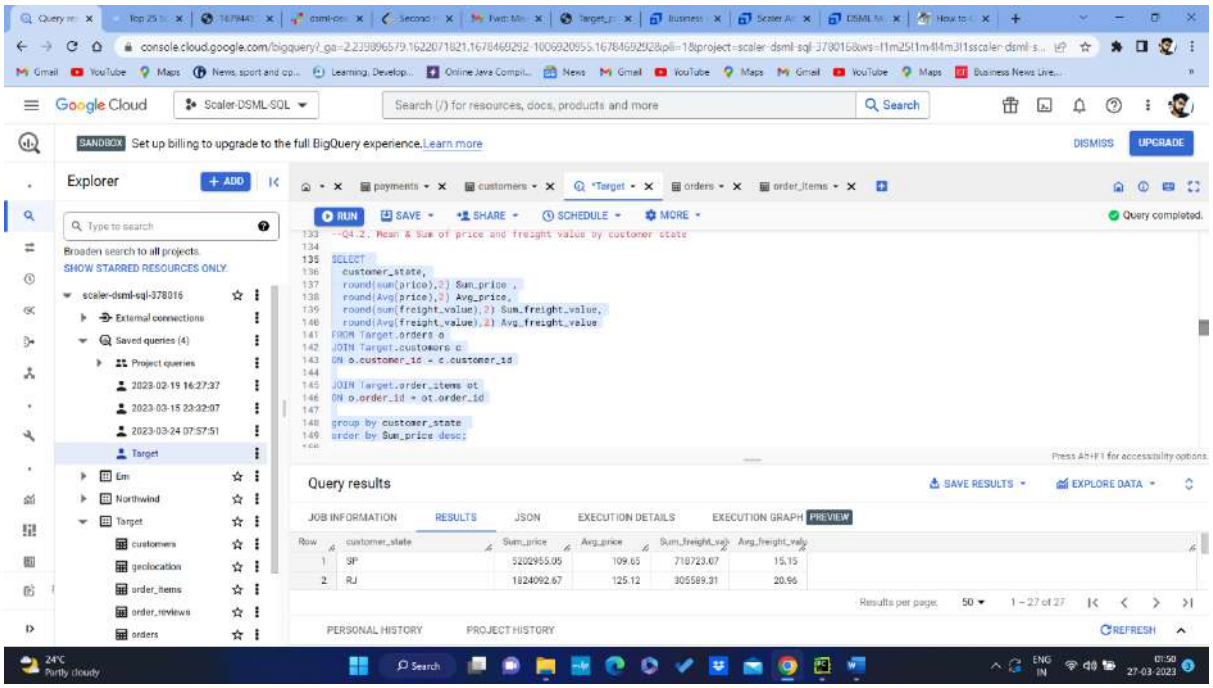
```sql
--Q3.2. Distribution of customers across the states in Brazil

SELECT
customer_state,
COUNT(o.customer_id) No_of_customers
FROM Target.orders o
JOIN Target.customers c
ON o.customer_id =c.customer_id
GROUP BY customer_state
ORDER BY No_of_customers DESC;
```



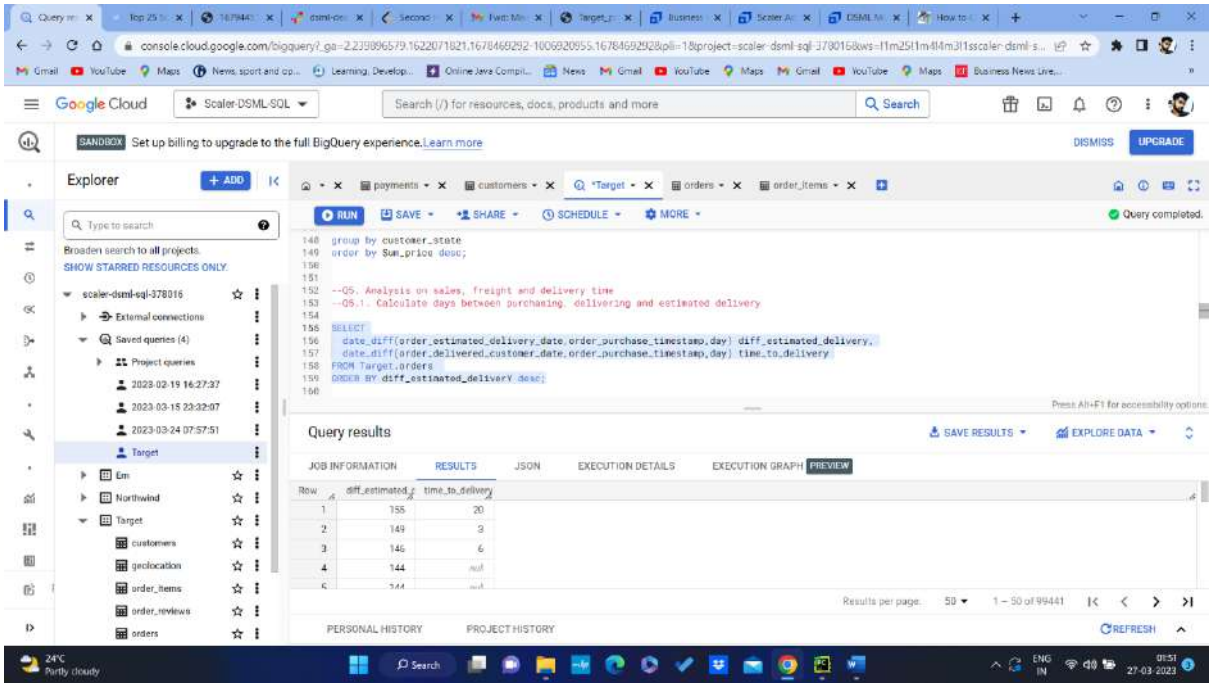
```
--Q4. Impact on Economy: Analyze the money movement by e-
commerce by looking at order prices, freight and others.
--
4.1.  Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug
 only) - You can use "payment_value" column in payments table

SELECT
    (sum_payments-
LEAD(sum_payments)OVER(ORDER BY year DESC))/(sum_payments)*100  per_change_cost_order
FROM
    (
    SELECT
        EXTRACT(year FROM order_purchase_timestamp) year,
        SUM(payment_value) sum_payments,
        FROM Target.payments p
        JOIN Target.orders o
        ON p.order_id =o.order_id
        WHERE
          (EXTRACT(year FROM order_purchase_timestamp) BETWEEN 2017 AND 2018)
        AND
          (EXTRACT(month FROM order_purchase_timestamp) BETWEEN 1 AND 8)
      GROUP BY EXTRACT(year FROM order_purchase_timestamp) )
    ORDER BY per_change_cost_order DESC
    LIMIT 1;
```



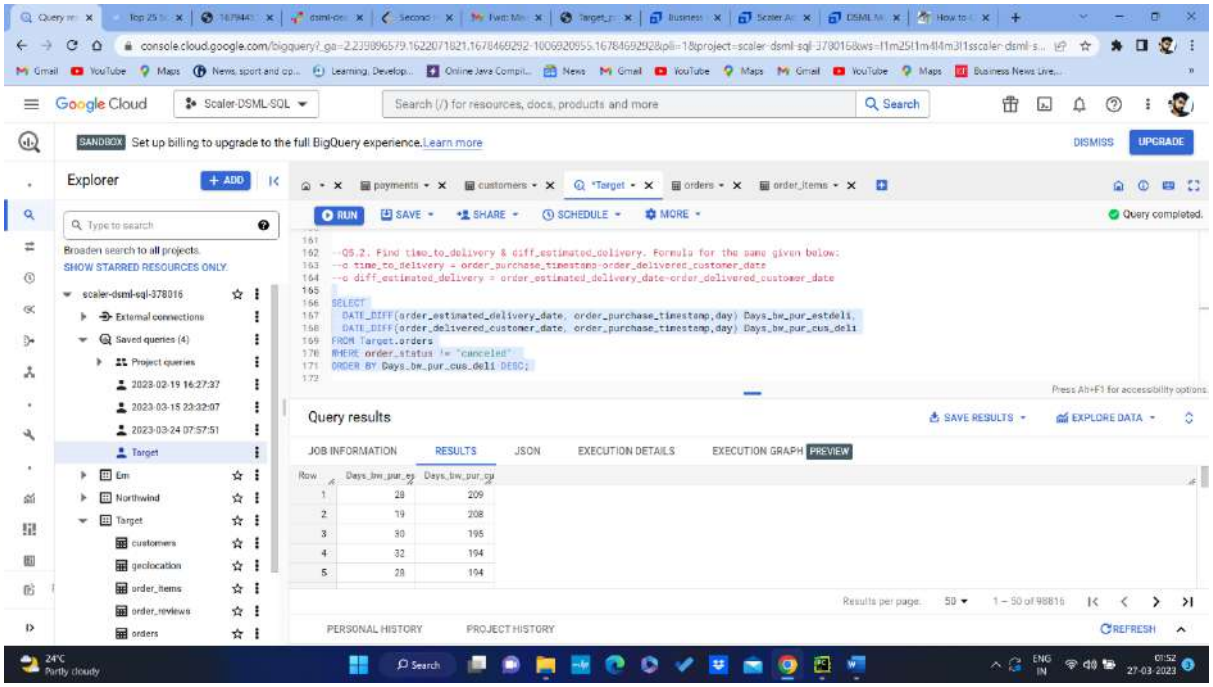
```sql
--Q4.2. Mean & Sum of price and freight value by customer state

SELECT
  customer_state,
  round(sum(price),2) Sum_price ,
  round(Avg(price),2) Avg_price,
  round(sum(freight_value),2) Sum_freight_value,
  round(Avg(freight_value),2) Avg_freight_value
FROM Target.orders o
JOIN Target.customers c
ON o.customer_id = c.customer_id

JOIN Target.order_items ot
ON o.order_id = ot.order_id

group by customer_state
order by Sum_price desc;
```



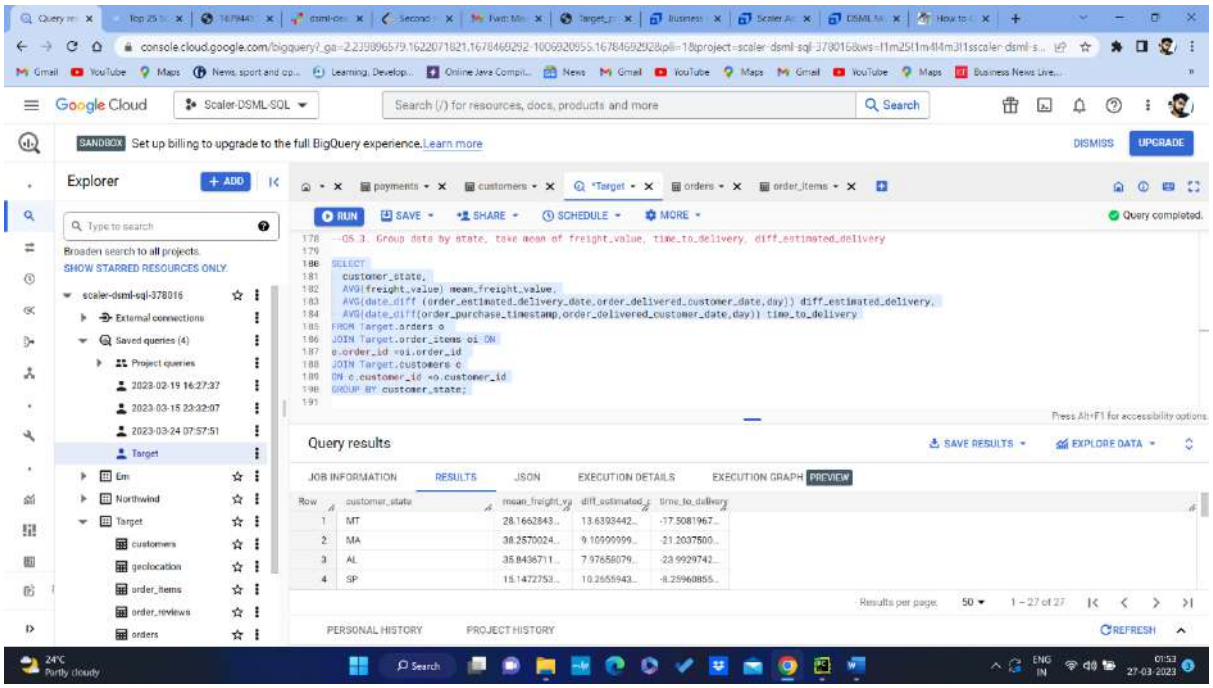
```sql
--Q5. Analysis on sales, freight and delivery time
--Q5.1. Calculate days between purchasing, delivering and estimated delivery

SELECT
  date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) diff_estimated_deli
very,
  date_diff(order_delivered_customer_date,order_purchase_timestamp,day) time_to_delivery
FROM Target.orders
ORDER BY diff_estimated_deliverY desc;
```



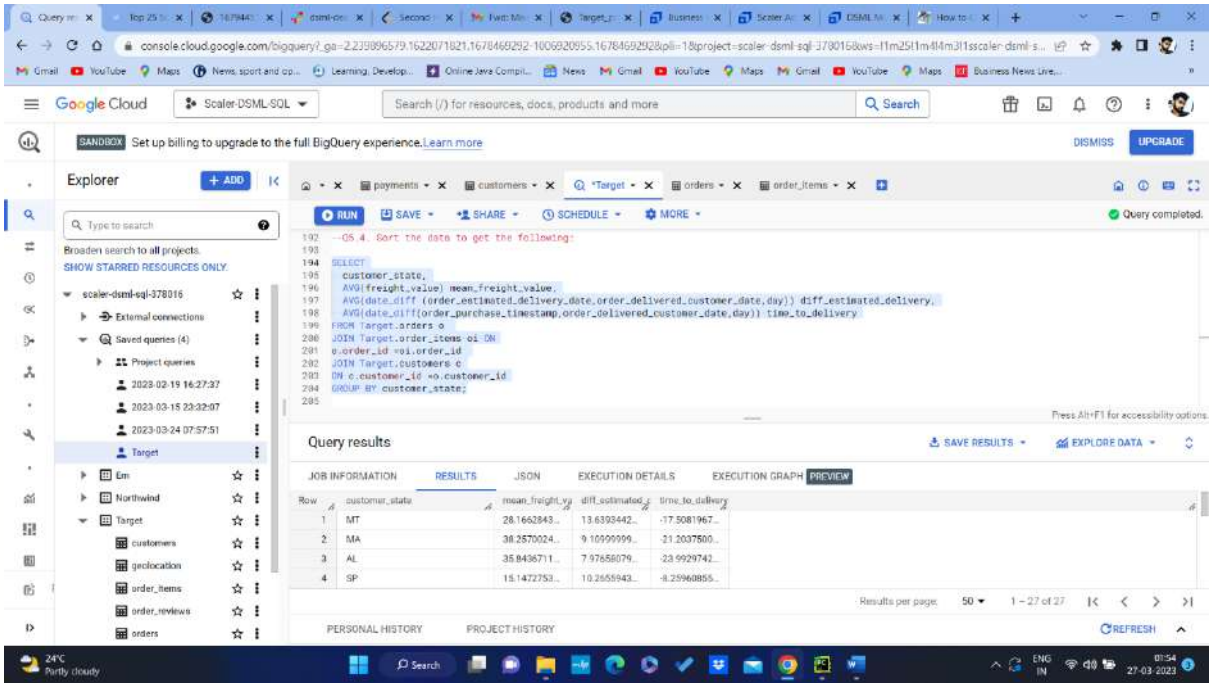
```sql
--Q5.2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
--o time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
--o diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

SELECT
  DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp,day) Days_bw_pur_estdel
i,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day) Days_bw_pur_cus_de
li
FROM Target.orders
WHERE order_status != "canceled"
ORDER BY Days_bw_pur_cus_deli DESC;
```



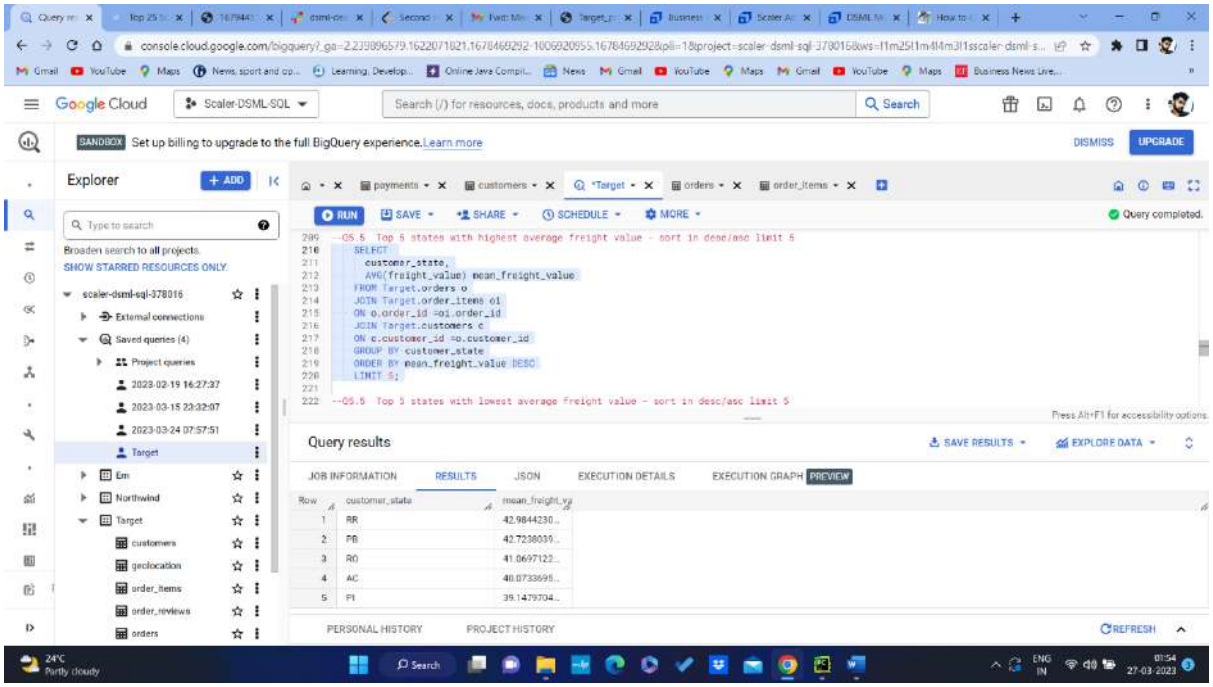
```
--
Q5.3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_del
ivery

SELECT
  customer_state,
  AVG(freight_value) mean_freight_value,
  AVG(date_diff (order_estimated_delivery_date,order_delivered_customer_date,day)) diff_est
imated_delivery,
  AVG(date_diff(order_purchase_timestamp,order_delivered_customer_date,day)) time_to_delive
ry
FROM Target.orders o
JOIN Target.order_items oi ON
o.order_id =oi.order_id
JOIN Target.customers c
ON c.customer_id =o.customer_id
GROUP BY customer_state;
```



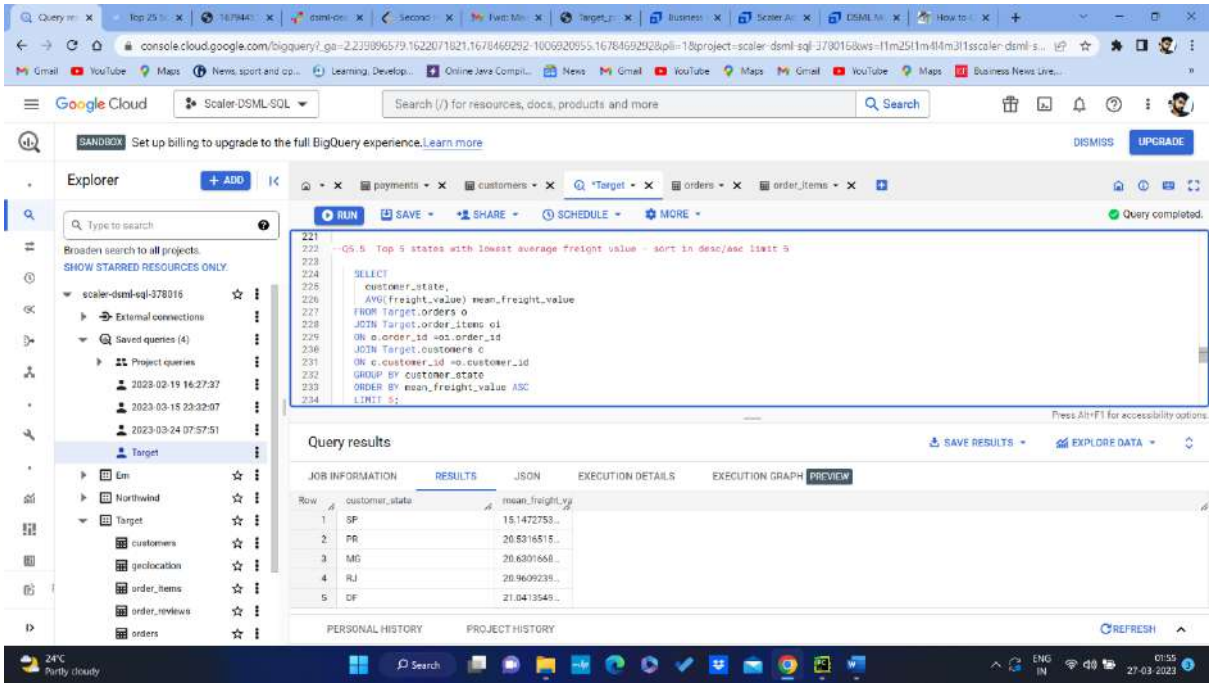
```
--Q5.4. Sort the data to get the following:

SELECT
  customer_state,
  AVG(freight_value) mean_freight_value,
  AVG(date_diff (order_estimated_delivery_date,order_delivered_customer_date,day)) diff_est
imated_delivery,
  AVG(date_diff(order_purchase_timestamp,order_delivered_customer_date,day)) time_to_delive
ry
FROM Target.orders o
JOIN Target.order_items oi ON
o.order_id =oi.order_id
JOIN Target.customers c
ON c.customer_id =o.customer_id
GROUP BY customer_state;
```



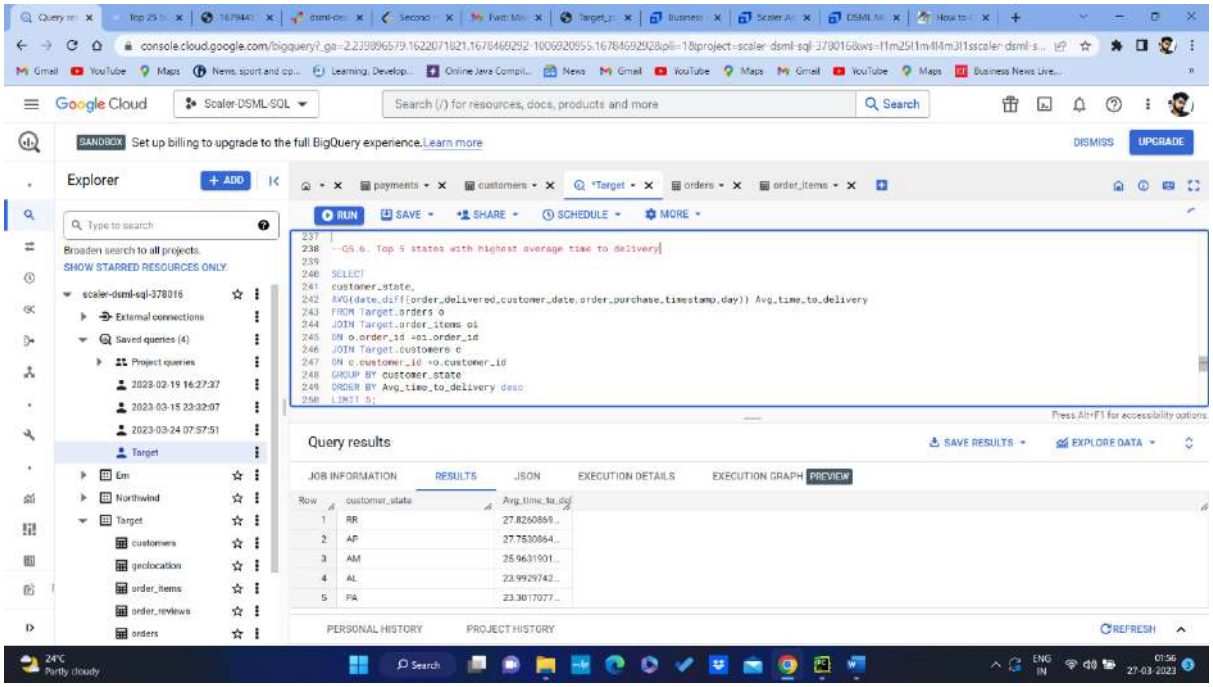
```sql
--Q5.5  Top 5 states with highest average freight value - sort in desc/asc limit 5
    SELECT
      customer_state,
      AVG(freight_value) mean_freight_value
    FROM Target.orders o
    JOIN Target.order_items oi
    ON o.order_id =oi.order_id
    JOIN Target.customers c
    ON c.customer_id =o.customer_id
    GROUP BY customer_state
    ORDER BY mean_freight_value DESC
    LIMIT 5;
```



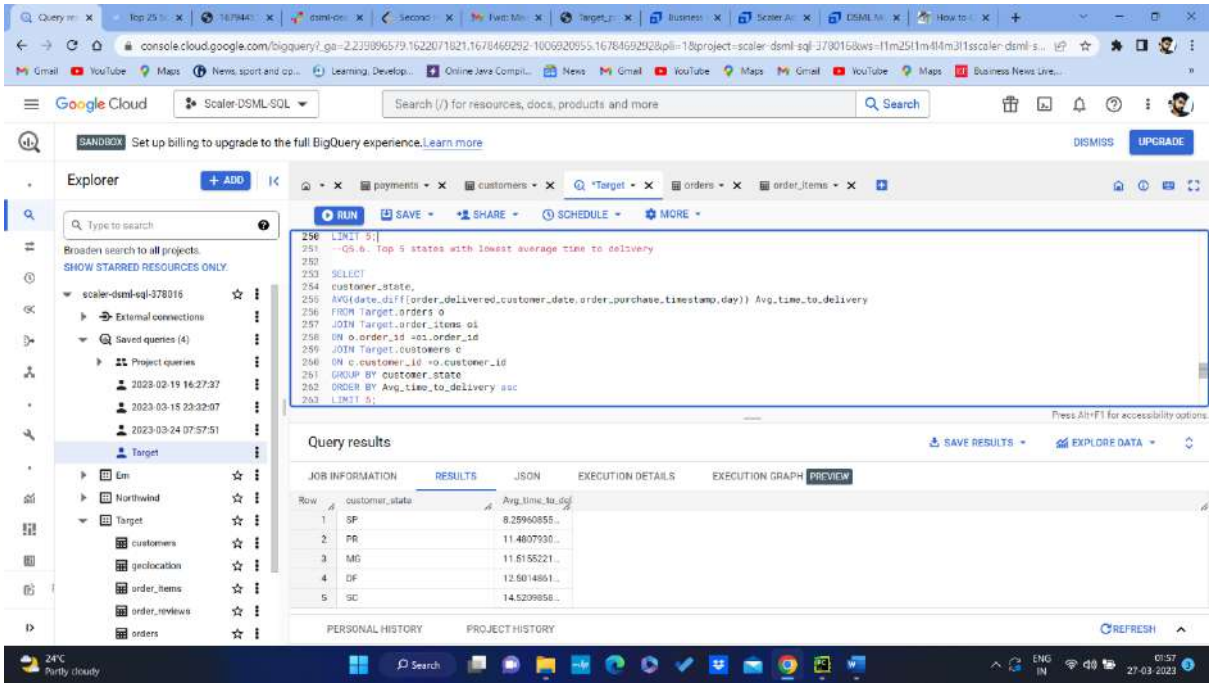
```sql
--Q5.5  Top 5 states with lowest average freight value - sort in desc/asc limit 5

    SELECT
      customer_state,
      AVG(freight_value) mean_freight_value
    FROM Target.orders o
    JOIN Target.order_items oi
    ON o.order_id =oi.order_id
    JOIN Target.customers c
    ON c.customer_id =o.customer_id
    GROUP BY customer_state
    ORDER BY mean_freight_value ASC
    LIMIT 5;
```



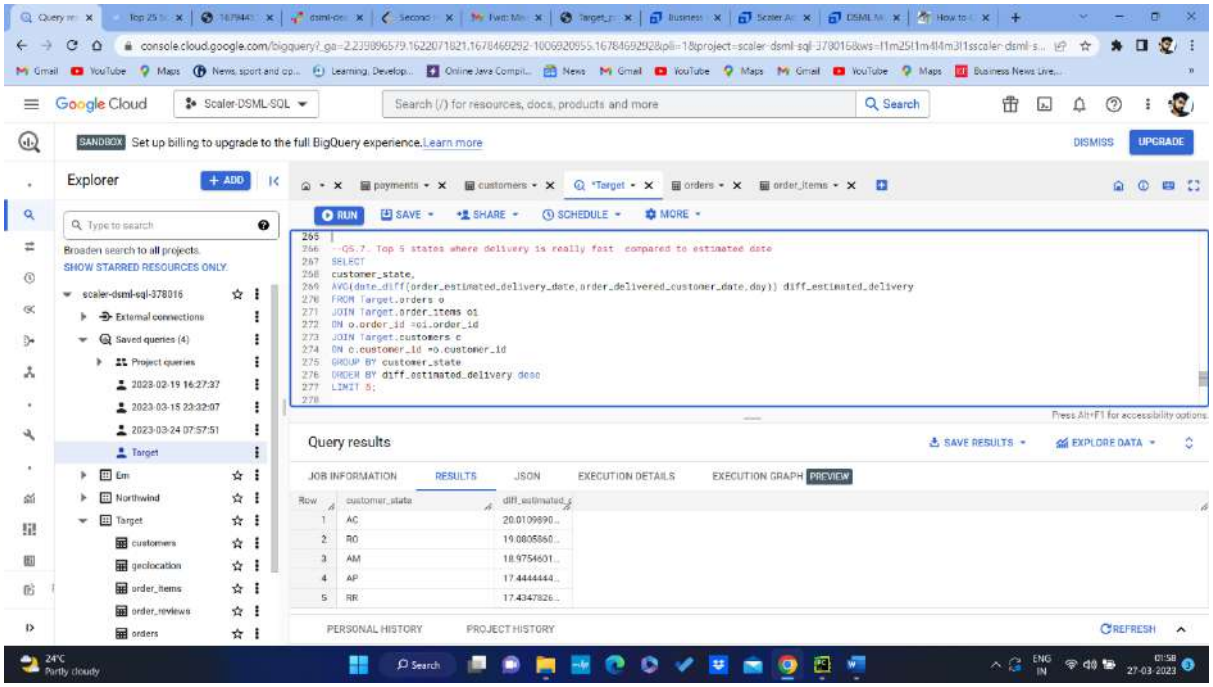
```
--Q5.6. Top 5 states with highest average time to delivery


SELECT
customer_state,
AVG(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)) Avg_time_to_deli
very
FROM Target.orders o
JOIN Target.order_items oi
ON o.order_id =oi.order_id
JOIN Target.customers c
ON c.customer_id =o.customer_id
GROUP BY customer_state
ORDER BY Avg_time_to_delivery desc
LIMIT 5;
```



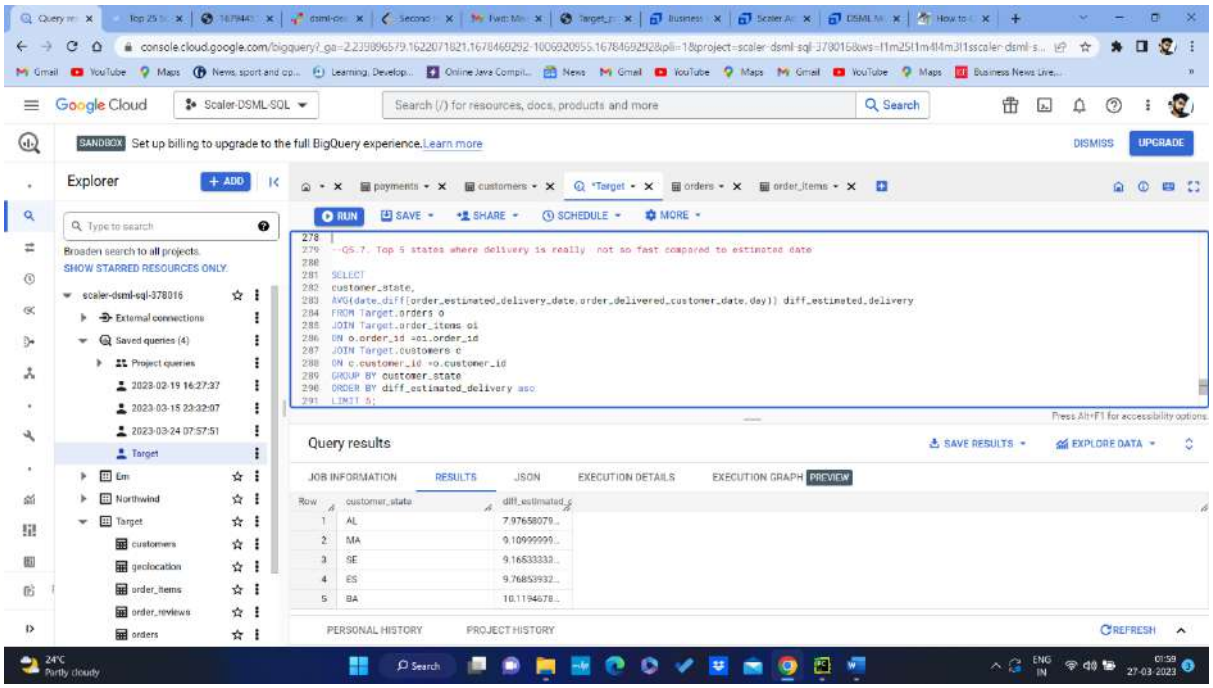
```sql
--Q5.6. Top 5 states with lowest average time to delivery

SELECT
customer_state,
AVG(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)) Avg_time_to_deli
very
FROM Target.orders o
JOIN Target.order_items oi
ON o.order_id =oi.order_id
JOIN Target.customers c
ON c.customer_id =o.customer_id
GROUP BY customer_state
ORDER BY Avg_time_to_delivery asc
LIMIT 5;
```



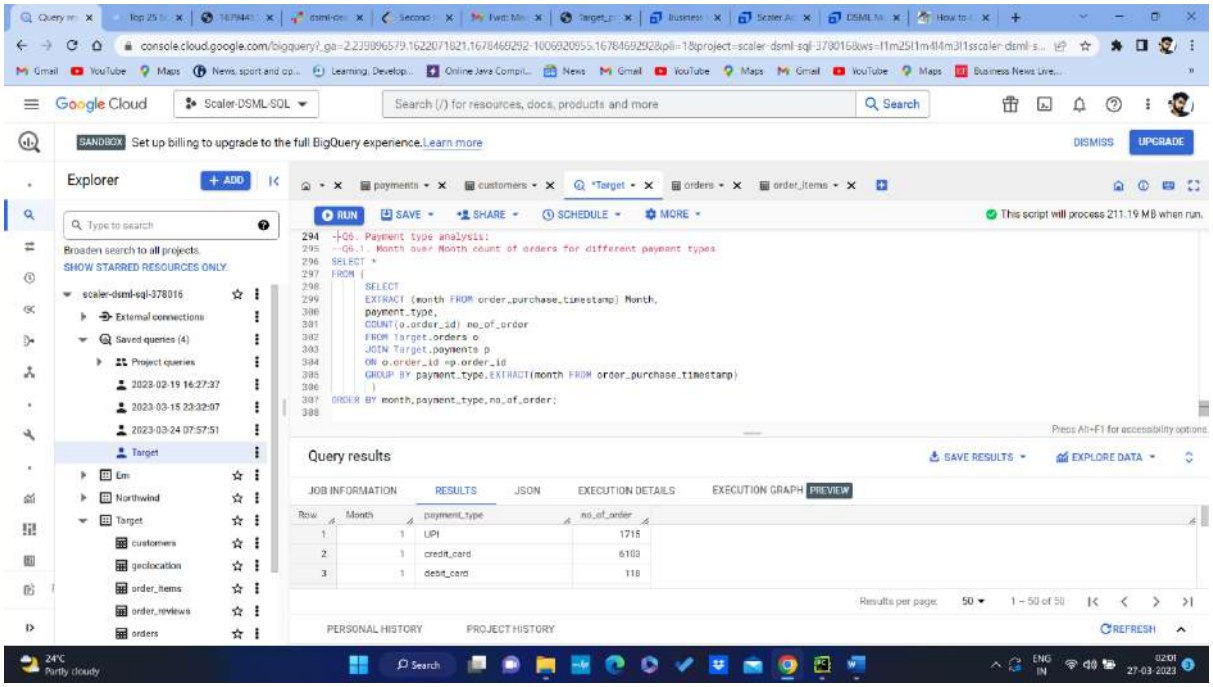
```sql
--Q5.7. Top 5 states where delivery is really fast  compared to estimated date
SELECT
customer_state,
AVG(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)) diff_estimated_delivery
FROM Target.orders o
JOIN Target.order_items oi
ON o.order_id =oi.order_id
JOIN Target.customers c
ON c.customer_id =o.customer_id
GROUP BY customer_state
ORDER BY diff_estimated_delivery desc
LIMIT 5;
```



```sql
--Q5.7. Top 5 states where delivery is really  not so fast compared to estimated date

SELECT
customer_state,
AVG(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)) diff_estimated_delivery
FROM Target.orders o
JOIN Target.order_items oi
ON o.order_id =oi.order_id
JOIN Target.customers c
ON c.customer_id =o.customer_id
GROUP BY customer_state
ORDER BY diff_estimated_delivery asc
LIMIT 5;
```



```sql
--Q6. Payment type analysis:
--Q6.1. Month over Month count of orders for different payment types
SELECT *
FROM (
      SELECT
      EXTRACT (month FROM order_purchase_timestamp) Month,
      payment_type,
      COUNT(o.order_id) no_of_order
      FROM Target.orders o
      JOIN Target.payments p
      ON o.order_id =p.order_id
      GROUP BY payment_type,EXTRACT(month FROM order_purchase_timestamp)
       )
ORDER BY month,payment_type,no_of_order;
```



```sql
--Q6.2. Count of orders based on the no. of payment installments

SELECT payment_installments, COUNT(DISTINCT order_id) as num_orders
FROM Target.payments
GROUP BY payment_installments
order by num_orders;
```