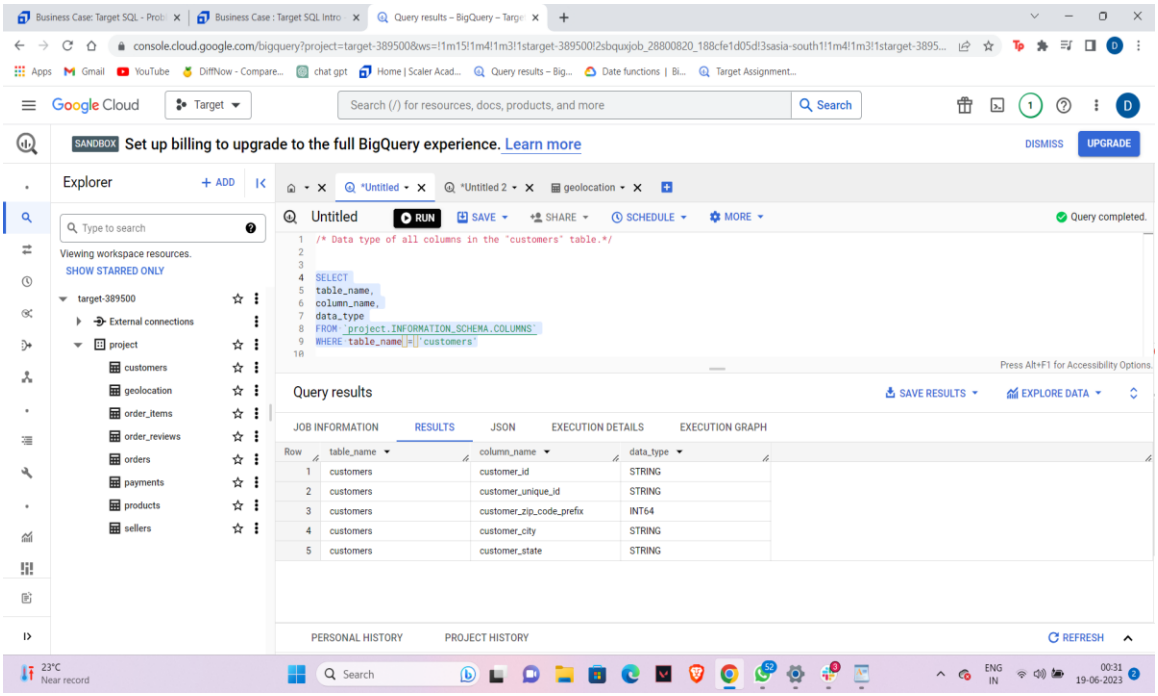
Business Case: Target SQL

Name: D A Santhosh

Q1 /* Data type of all columns in the "customers" table.*/

```
SELECT

table_name,

column_name,

data_type

FROM `project.INFORMATION_SCHEMA.COLUMNS`

WHERE table_name = 'customers'
```



Insights and recommendation: Here we can conclude that there are different types of data in

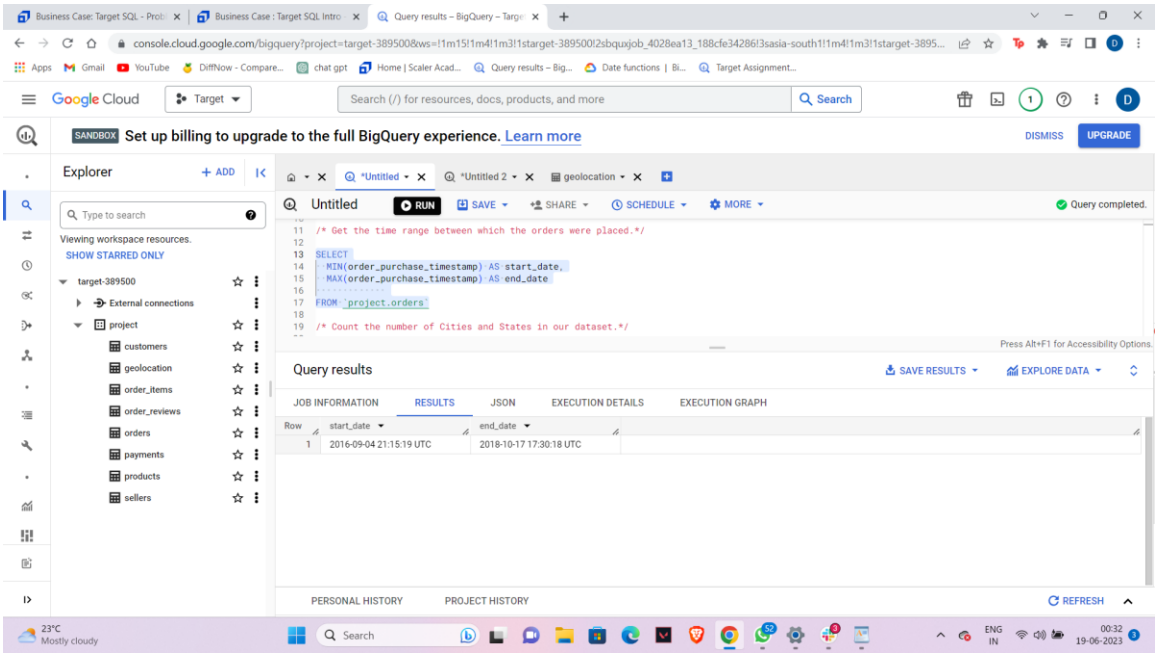
the table

including alphabets which is string data type and numeric data which is int type

hence the table contains both numeric and alphabetic data types.

/* Get the time range between which the orders were placed.*/

```
SELECT
    MIN(order_purchase_timestamp) AS start_date,
    MAX(order_purchase_timestamp) AS end_date

FROM `project.orders`
```



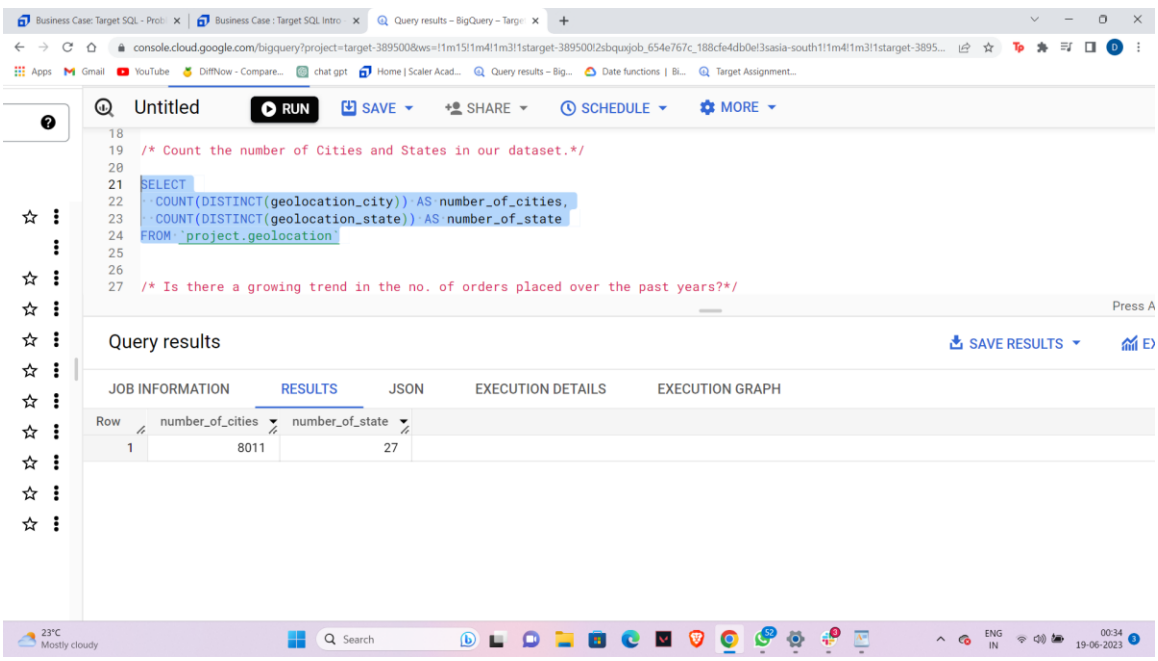
Insights and recommendation: The time range between which orders were mostly placed was between

2016-09-04 at 21:15:19 to

2018-10-17 at 17:30:18 , hence we can conclude that the customers ordered items the most between these two years .

/* Count the number of Cities and States in our dataset.*/

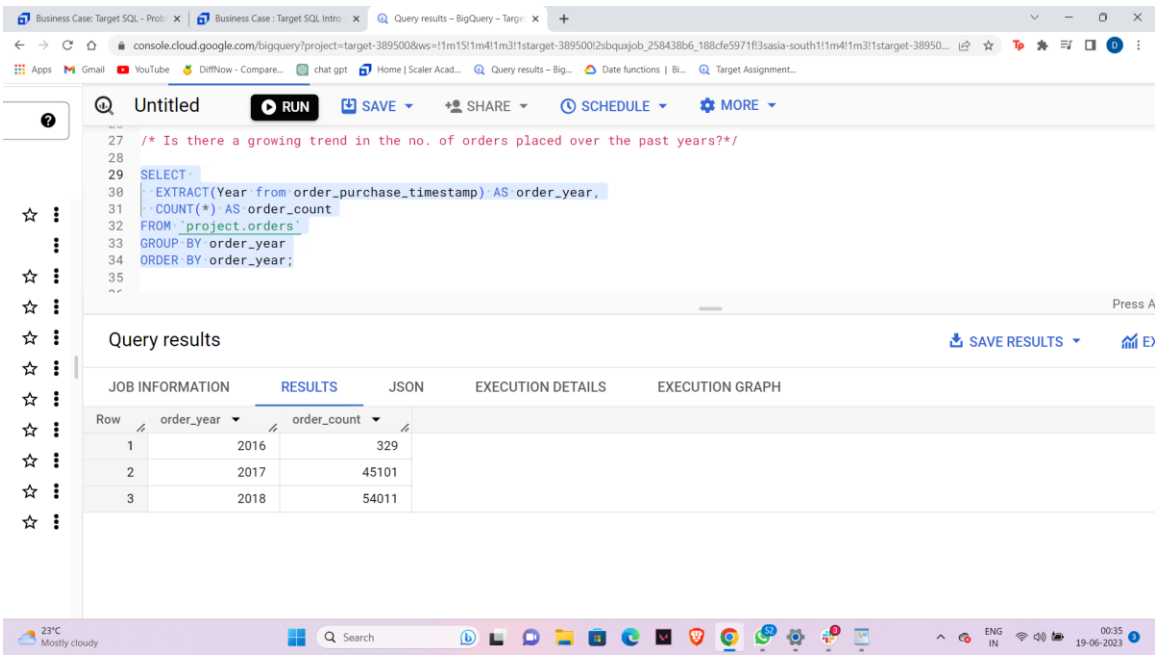
```
SELECT
   COUNT(DISTINCT(geolocation_city)) AS number_of_cities,
   COUNT(DISTINCT(geolocation_state)) AS number_of_state
FROM `project.geolocation`
```



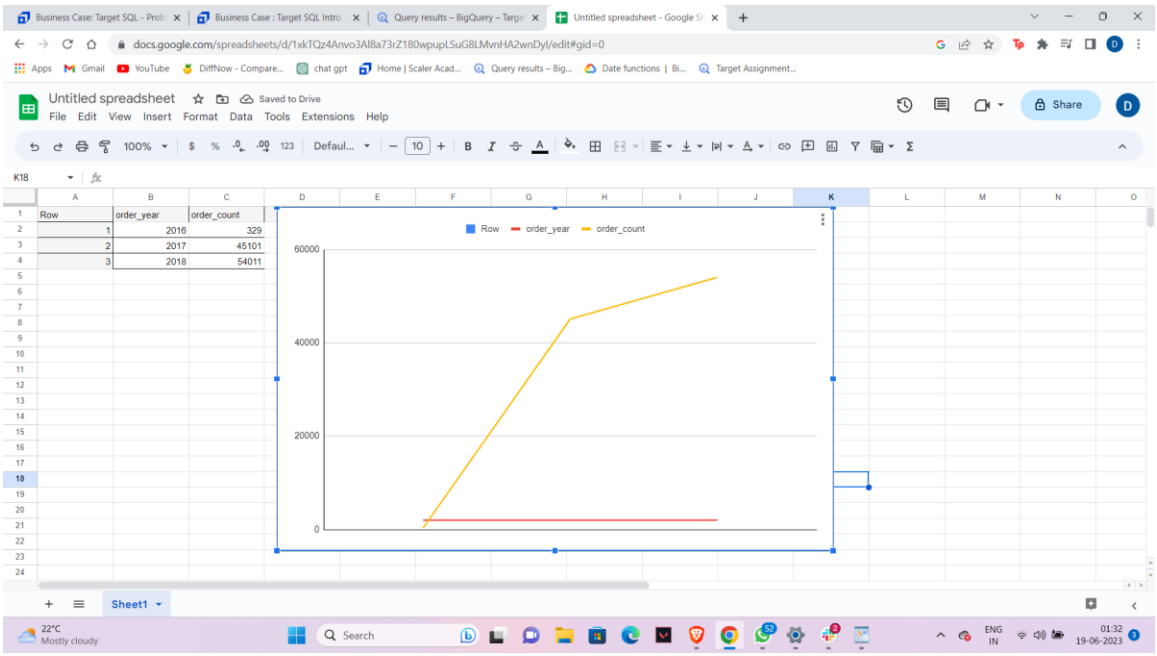
Insights and recommendation: Hence from the obove query we can conclude that there are around 8011 cities and 27 states

Q2 /* Is there a growing trend in the no. of orders placed over the past years?*/

```
SELECT
    EXTRACT(Year from order_purchase_timestamp) AS order_year,
    COUNT(*) AS order_count
FROM `project.orders`
GROUP BY order_year
ORDER BY order_year;
```



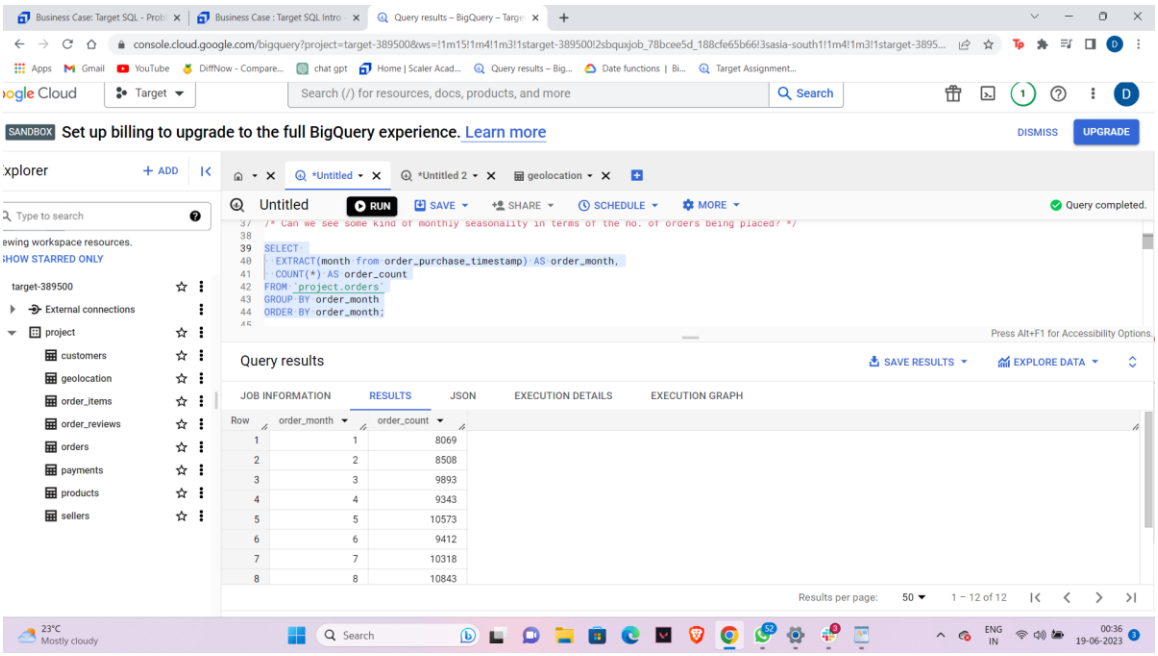
Insights and recommendation: There is Obviously a growing Trend as the number of orders has increased rapidly from 2016 to 2017 from 329 orders to a whooping 45101 orders and a sustainable increase between 2017 and 2018 from 45101 to 54011

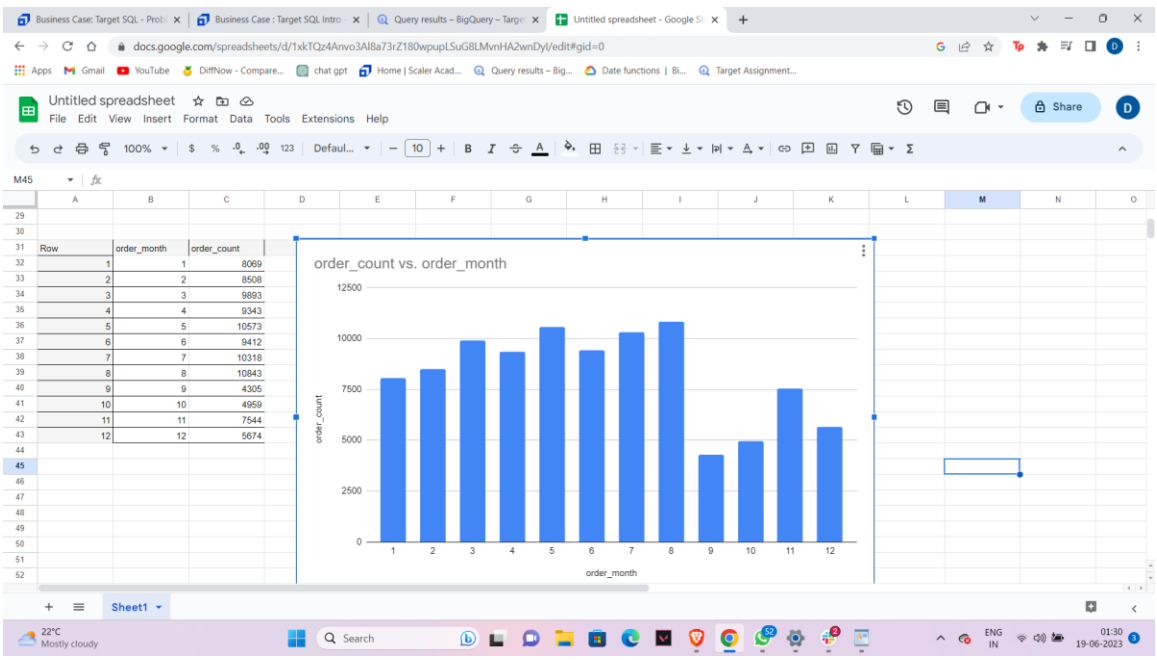


The graph also shows an increase in the trend

/* Can we see some kind of monthly seasonality in terms of the no. of orders being placed? */

```
SELECT
    EXTRACT(month from order_purchase_timestamp) AS order_month,
    COUNT(*) AS order_count
FROM `project.orders`
GROUP BY order_month
ORDER BY order_month;
```





Insights and recommendation: we can clearly observe that the sales started off well during the beginning of the year and hit its peak during the 8th month and showed a sudden fall during the ninth month and gradually picked up during the 11th month but dropped again during the last month.

So giving some discounts coupons or having a sale during the months in the year end would stabilize the sales throughout the year.

/* During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

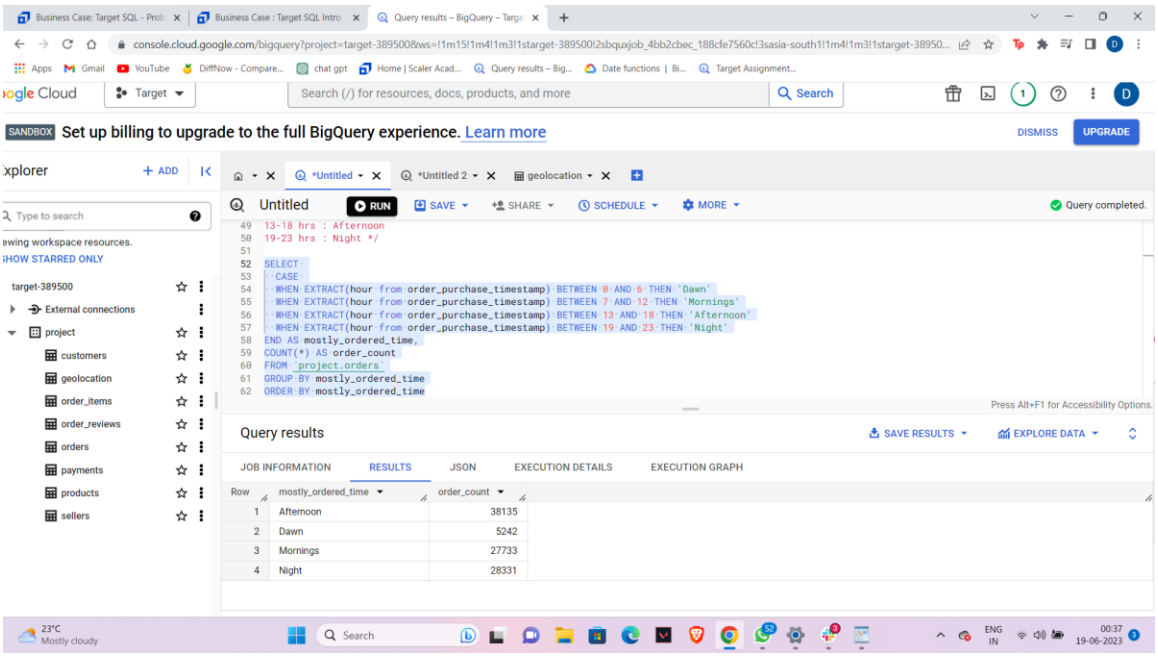
0-6 hrs : Dawn

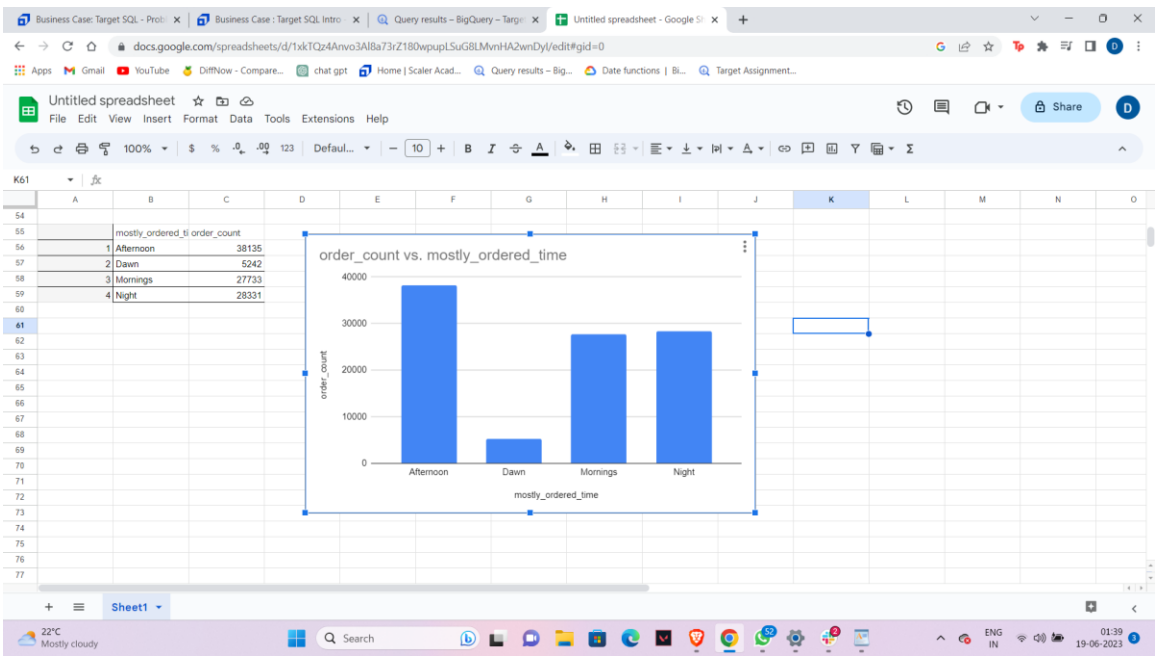
7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night */

```
SELECT

    CASE

    WHEN EXTRACT(hour from order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'

    WHEN EXTRACT(hour from order_purchase_timestamp) BETWEEN 7 AND 12 THEN
'Mornings'

    WHEN EXTRACT(hour from order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'

    WHEN EXTRACT(hour from order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'

END AS mostly_ordered_time,

COUNT(*) AS order_count

FROM `project.orders`

GROUP BY mostly_ordered_time

ORDER BY mostly_ordered_time
```



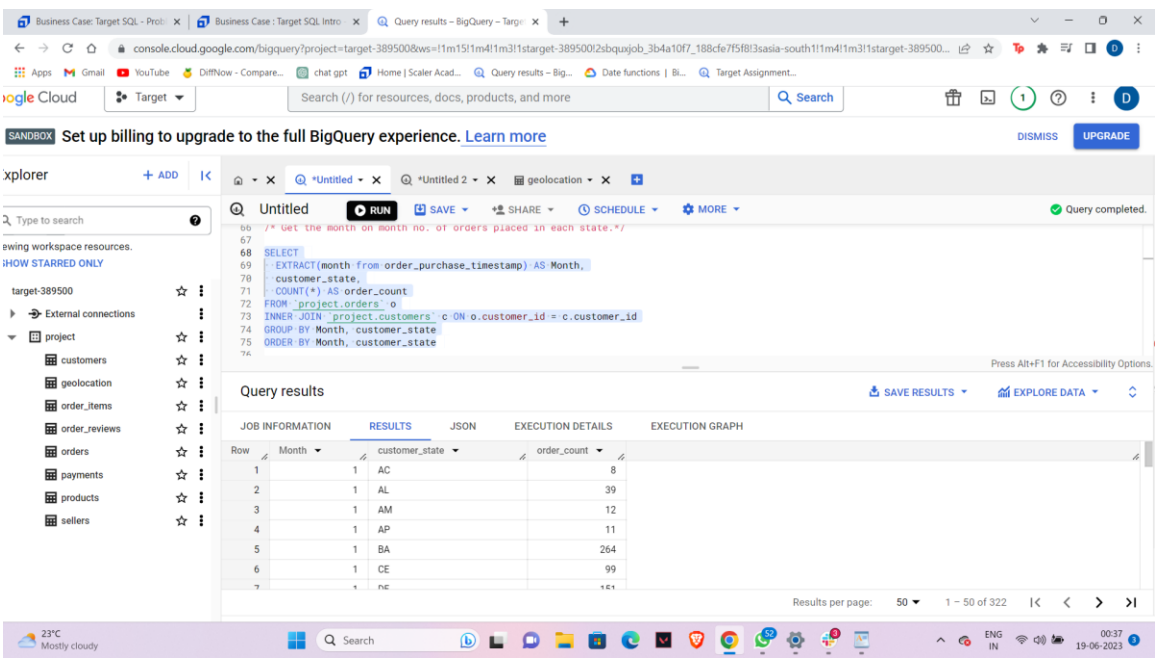


Insights and recommendation: Brazilian customers mostly ordered during the afternoon and good number of orders during the morning and night too, the number of orders is the least during the dawn

so bringing up some promotional events or flash sales during the dawn would increase the number of orders placed during the dawn.

Q3 /* Get the month on month no. of orders placed in each state.*/

```
SELECT
    EXTRACT(month from order_purchase_timestamp) AS Month,
    customer_state,
    COUNT(*) AS order_count
FROM `project.orders` o
INNER JOIN `project.customers` c ON o.customer_id = c.customer_id
GROUP BY Month, customer_state
ORDER BY Month, customer_state
```

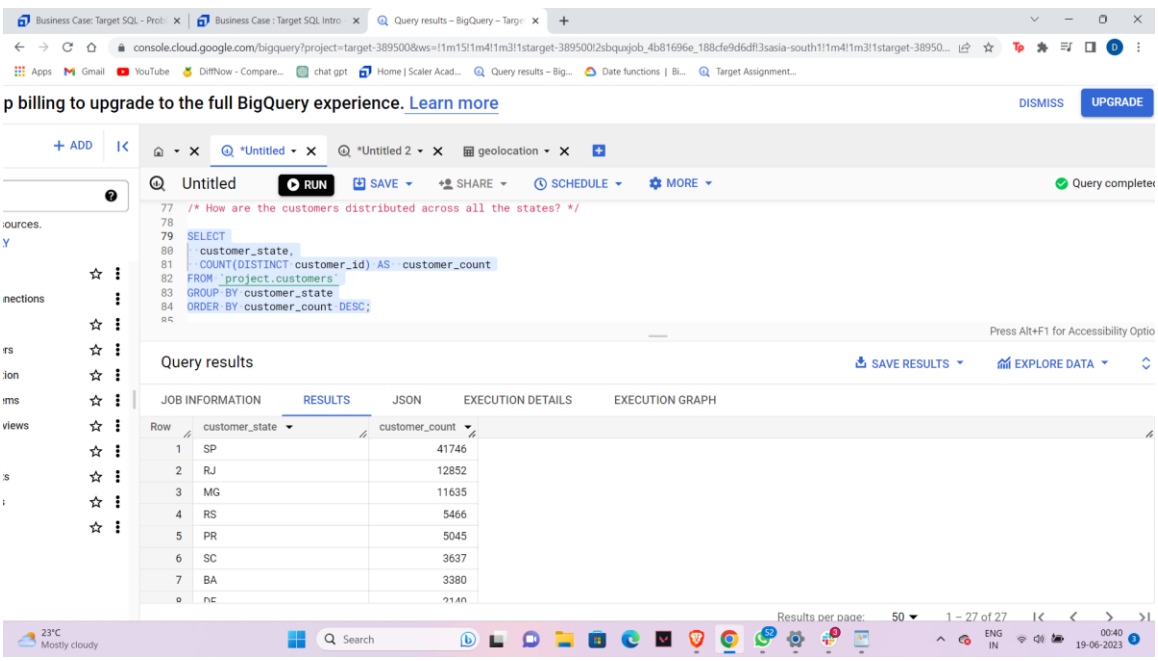


Insights and recommendation: Here by bringing the month on month orders placed by customers in different states we can observe state wise sales sorted in a particular month and

push business accordingly as per requirements and push by advertising more during the times the business is low in a particular state.

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

```
SELECT
    customer_state,
    COUNT(DISTINCT customer_id) AS    customer_count
FROM `project.customers`
GROUP BY customer_state
ORDER BY customer_count DESC;
```

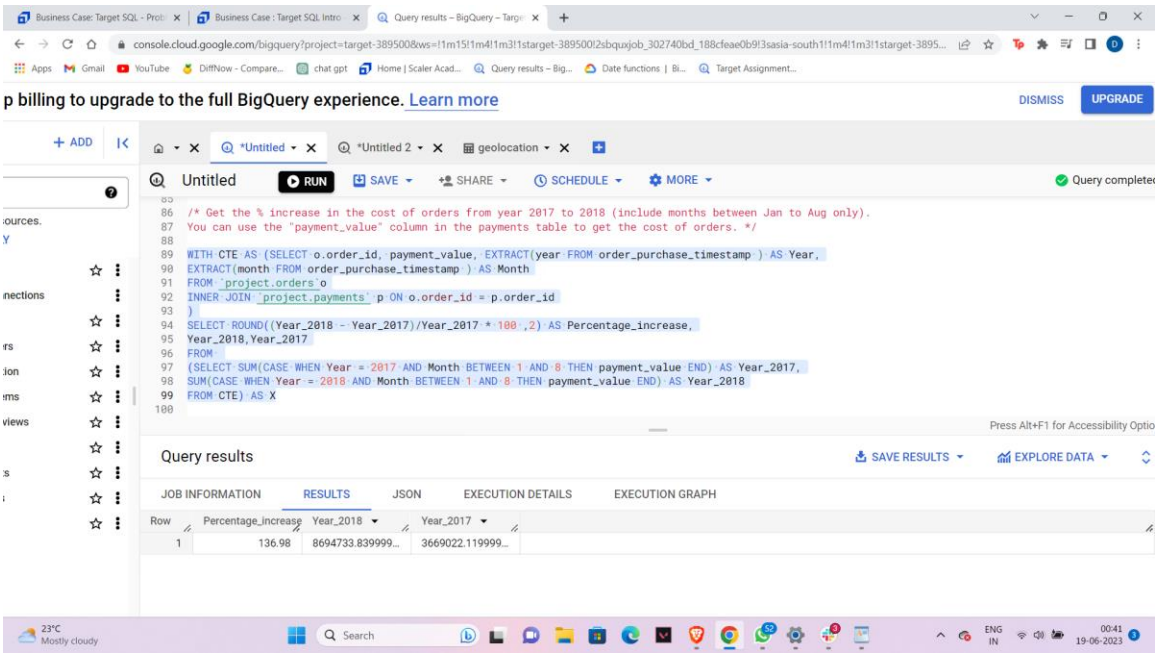


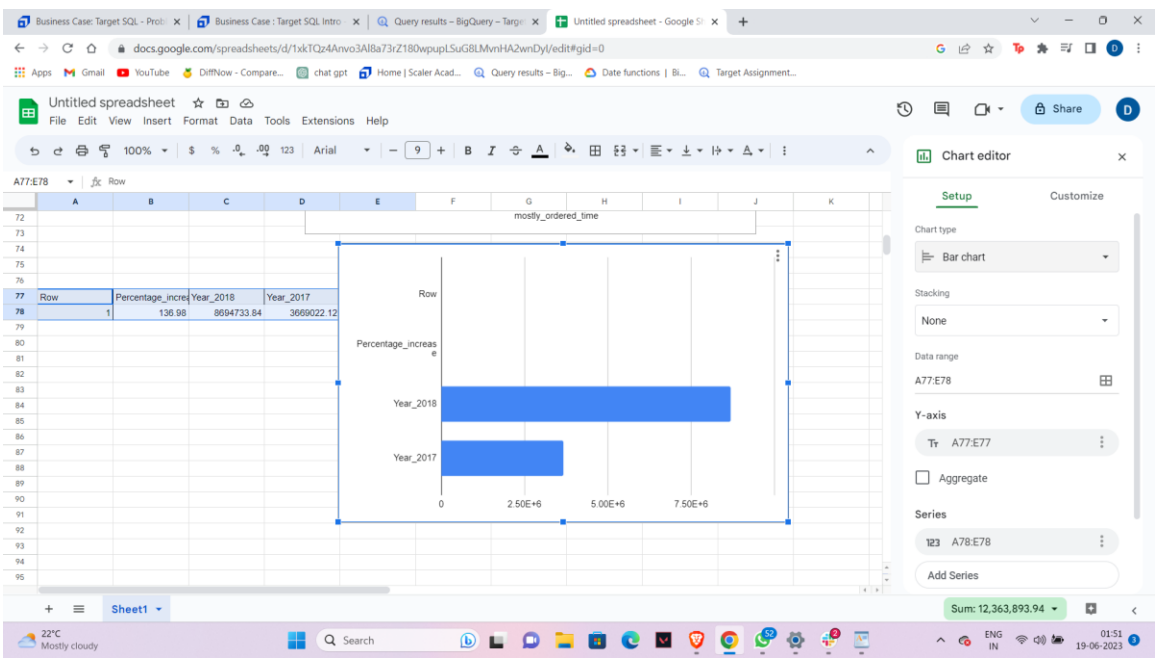
Insights and recommendation: Here we get to see a data of the number of customers according to the states , which will in turn help us stabilize the positives done to get maximum sales in the states where there is more business and work on where we may be going wrong and cater the needs according to that particular region when the sales is less.

Q4 /* Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

You can use the "payment_value" column in the payments table to get the cost of orders. */

```
WITH CTE AS (SELECT o.order_id, payment_value, EXTRACT(year FROM
order_purchase_timestamp ) AS Year,

EXTRACT(month FROM order_purchase_timestamp ) AS Month

FROM `project.orders`o

INNER JOIN `project.payments` p ON o.order_id = p.order_id

)

SELECT ROUND((Year_2018 - Year_2017)/Year_2017 * 100 ,2) AS Percentage_increase,

Year_2018,Year_2017

FROM

(SELECT SUM(CASE WHEN Year = 2017 AND Month BETWEEN 1 AND 8 THEN payment_value
END) AS Year_2017,

SUM(CASE WHEN Year = 2018 AND Month BETWEEN 1 AND 8 THEN payment_value END) AS
Year_2018

FROM CTE) AS X
```

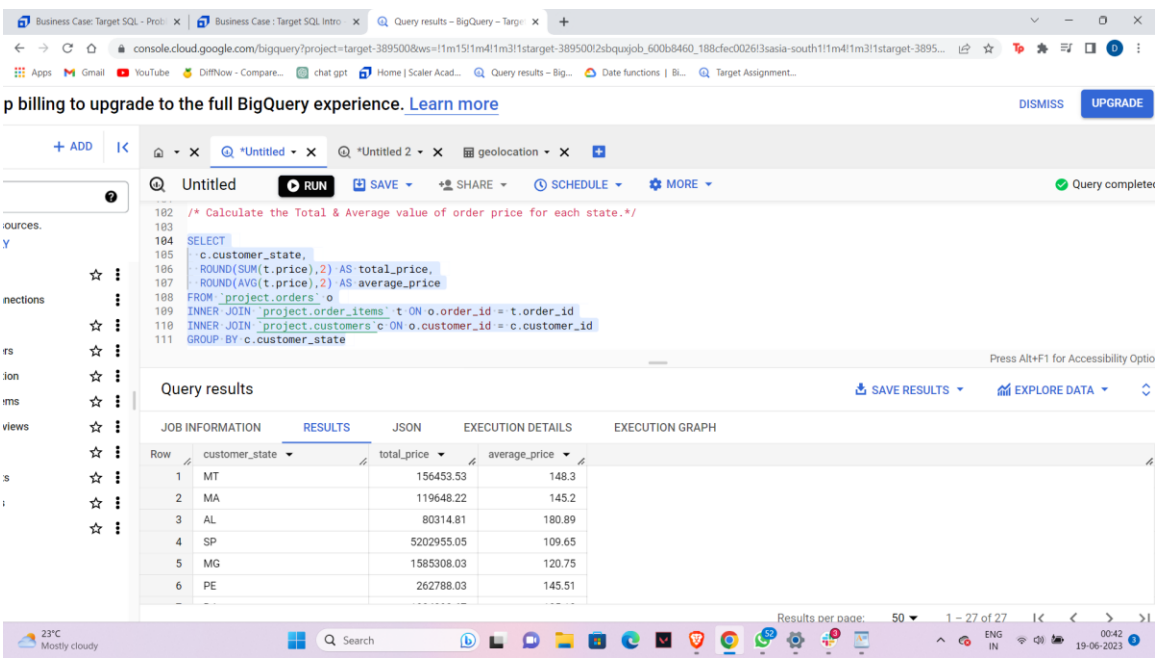




Insights and recommendation: the % increase in the cost of orders from year 2017 to 2018 seems to be about 2 times more , we can compare the same using the graphical visualization of the data of cost between the two years.

/* Calculate the Total & Average value of order price for each state.*/

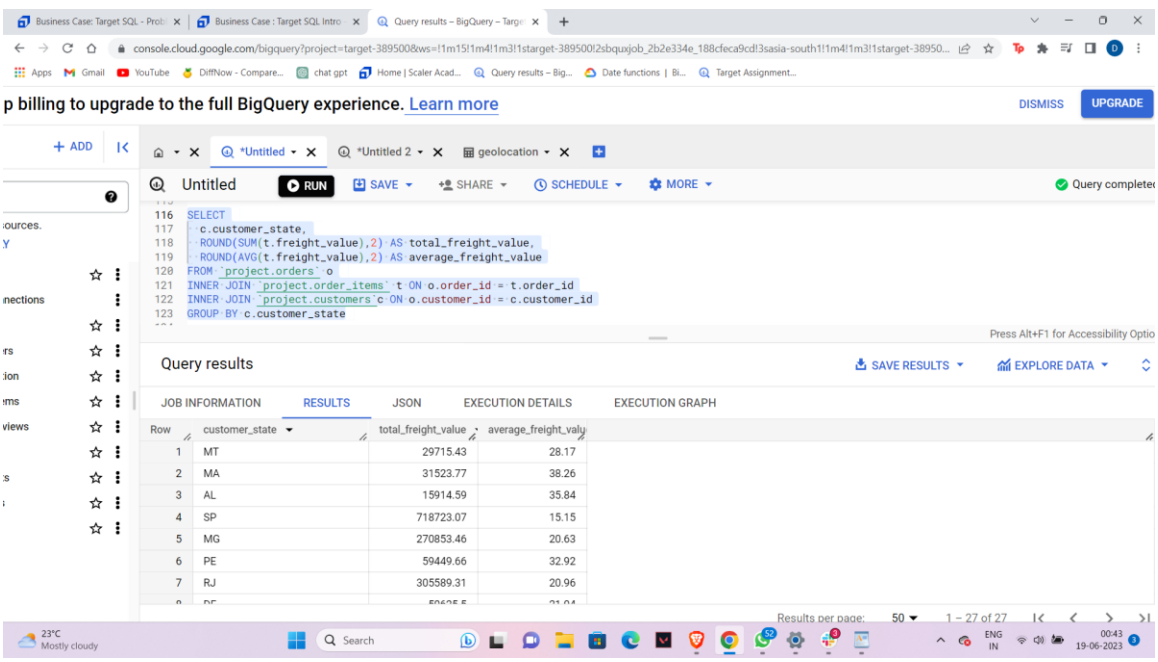
```
SELECT
    c.customer_state,
    ROUND(SUM(t.price),2) AS total_price,
    ROUND(AVG(t.price),2) AS average_price
FROM `project.orders` o
INNER JOIN `project.order_items` t ON o.order_id = t.order_id
INNER JOIN `project.customers`c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
```



Insights and recommendation: Here we are calculating the total price and average price according to the state by joining multiple tables to sort data accordingly.

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

```
SELECT
    c.customer_state,
    ROUND(SUM(t.freight_value),2) AS total_freight_value,
    ROUND(AVG(t.freight_value),2) AS average_freight_value
FROM `project.orders` o
INNER JOIN `project.order_items` t ON o.order_id = t.order_id
INNER JOIN `project.customers`c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
```



Insights and recommendation: Here we are calculating the total freight and average freight according to the state by joining multiple tables to sort data accordingly.

Q5 /* 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.

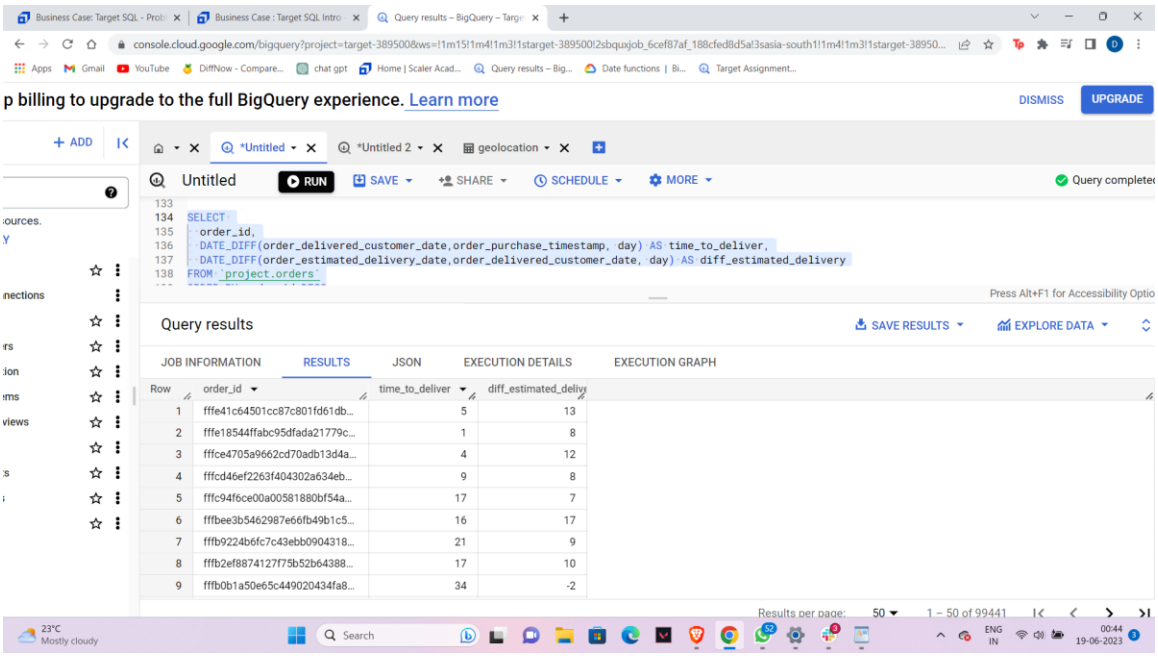
Do this in a single query.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

time_to_deliver = order_delivered_customer_date - order_purchase_timestamp

diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date */

```
SELECT

    order_id,

    DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, day) AS
time_to_deliver,

    DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date, day) AS
diff_estimated_delivery

FROM `project.orders`

ORDER BY order_id DESC
```



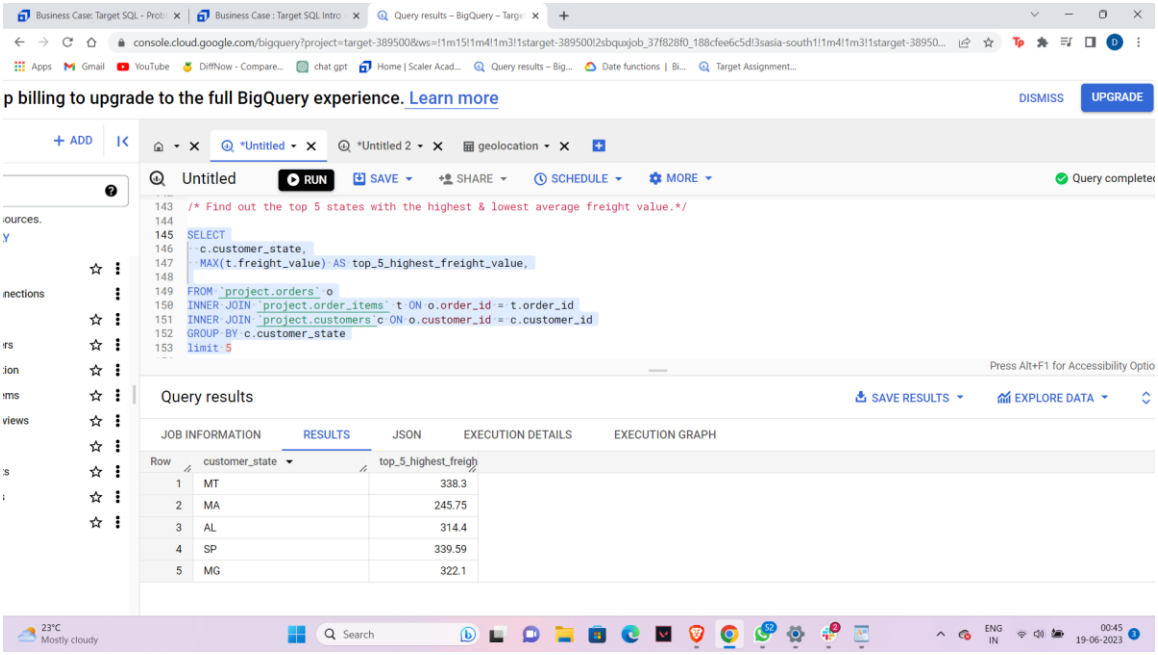
Insights and recommendation: This calculation of delivery time is taken to capture an average of time taken to deliver each product and draw insights to approximate the delivery time better for customer to have an idea of how quick the items would reach them.

/* Find out the top 5 states with the highest & lowest average freight value.*/

```
SELECT

    c.customer_state,

    MAX(t.freight_value) AS top_5_highest_freight_value,


FROM `project.orders` o

INNER JOIN `project.order_items` t ON o.order_id = t.order_id

INNER JOIN `project.customers`c ON o.customer_id = c.customer_id
```

GROUP BY c.customer_state

limit 5



Insights and recommendation: Here we are checking the top 5 record to see why it is on the top most and implement the similar approach for the other states

SELECT

c.customer_state,

MIN(t.freight_value) AS top_5_lowest_freight_value,

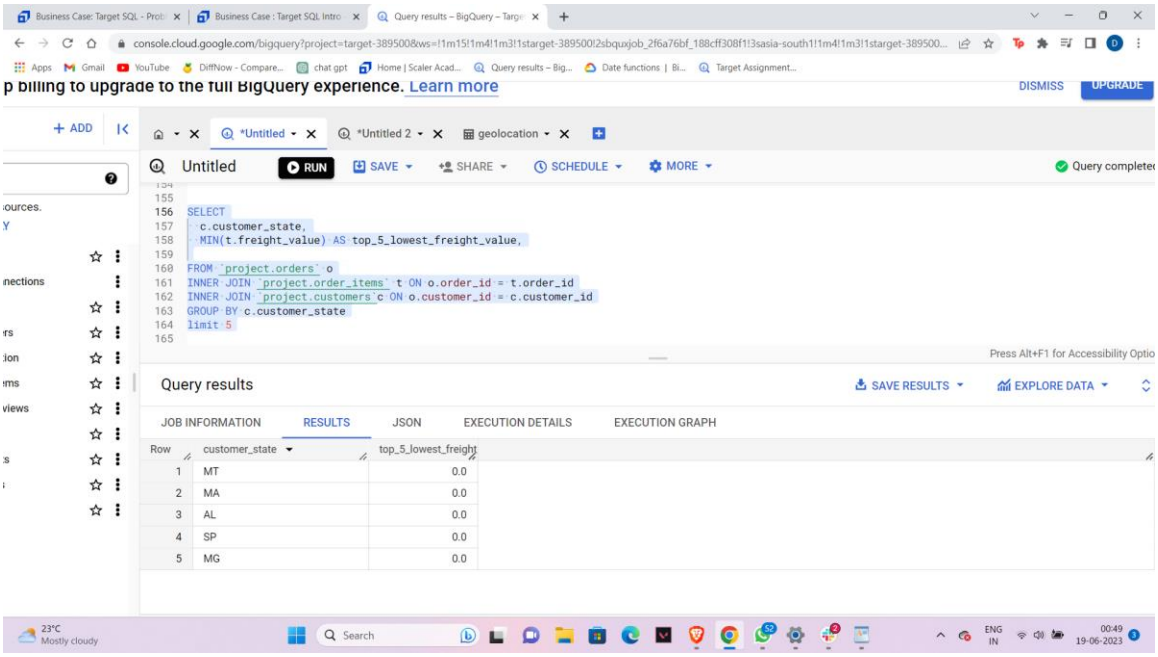
FROM `project.orders` o

INNER JOIN `project.order_items` t ON o.order_id = t.order_id

INNER JOIN `project.customers`c ON o.customer_id = c.customer_id

GROUP BY c.customer_state

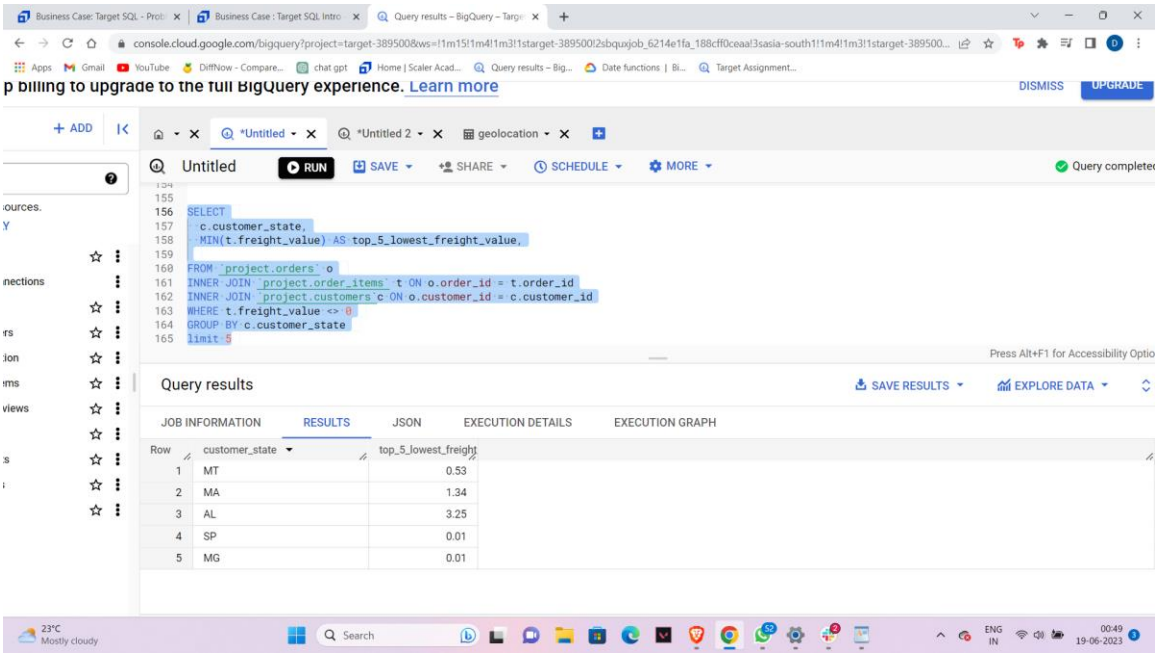
limit 5



Since all the values of freight in the lowest shows value as 0 , checked for values other than 0's

```
SELECT
    c.customer_state,
    MIN(t.freight_value) AS top_5_lowest_freight_value,

FROM `project.orders` o
INNER JOIN `project.order_items` t ON o.order_id = t.order_id
INNER JOIN `project.customers`c ON o.customer_id = c.customer_id
WHERE t.freight_value <> 0
GROUP BY c.customer_state
```

limit 5

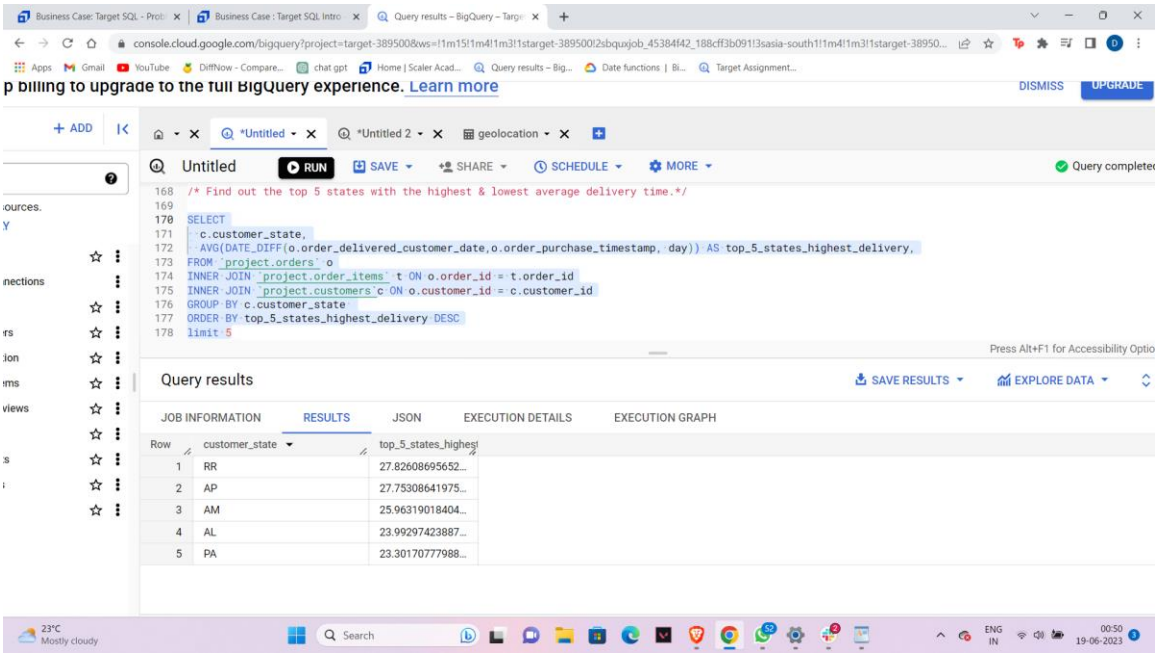


Insights and recommendation: Here we are checking the bottom 5 record to see why it is on the bottom most and improve the similar approach for the states

/* Find out the top 5 states with the highest & lowest average delivery time.*/

```
SELECT

    c.customer_state,

    AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, day)) AS
top_5_states_highest_delivery,

FROM `project.orders` o

INNER JOIN `project.order_items` t ON o.order_id = t.order_id

INNER JOIN `project.customers`c ON o.customer_id = c.customer_id

GROUP BY c.customer_state

ORDER BY top_5_states_highest_delivery DESC
```

limit 5



Insights and recommendation: Here we are checking the top 5 record to see the most quick deliveries done to the states and check why it is so , may be the fact being there are many warehouses accessible and near the city limits making it faster.

Try improving by implementing similar approach in other states where delivery time is late.

SELECT

c.customer_state,

AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, day)) AS top_5_states_lowest_delivery,

FROM `project.orders` o

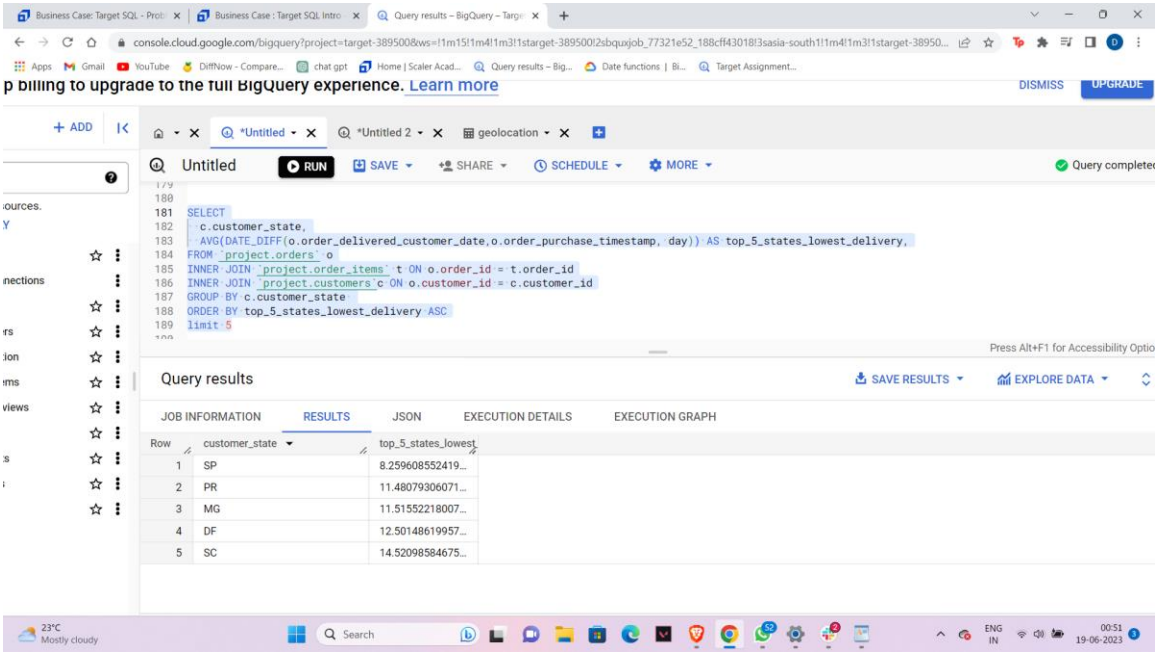
INNER JOIN `project.order_items` t ON o.order_id = t.order_id

INNER JOIN `project.customers`c ON o.customer_id = c.customer_id

GROUP BY c.customer_state

ORDER BY top_5_states_lowest_delivery ASC

limit 5



Insights and recommendation: Here we are checking the lowest 5 record to see the least speed of delivery or a delay in the delivery time and bring improvement on the same.

/* 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 date to figure out how fast the delivery was for each state. */

SELECT customer_state,ROUND(AVG(diff_estimated_delivery),2) AS delivery_speed

FROM (

SELECT

o.order_id,

c.customer_state,

DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) AS

diff_estimated_delivery

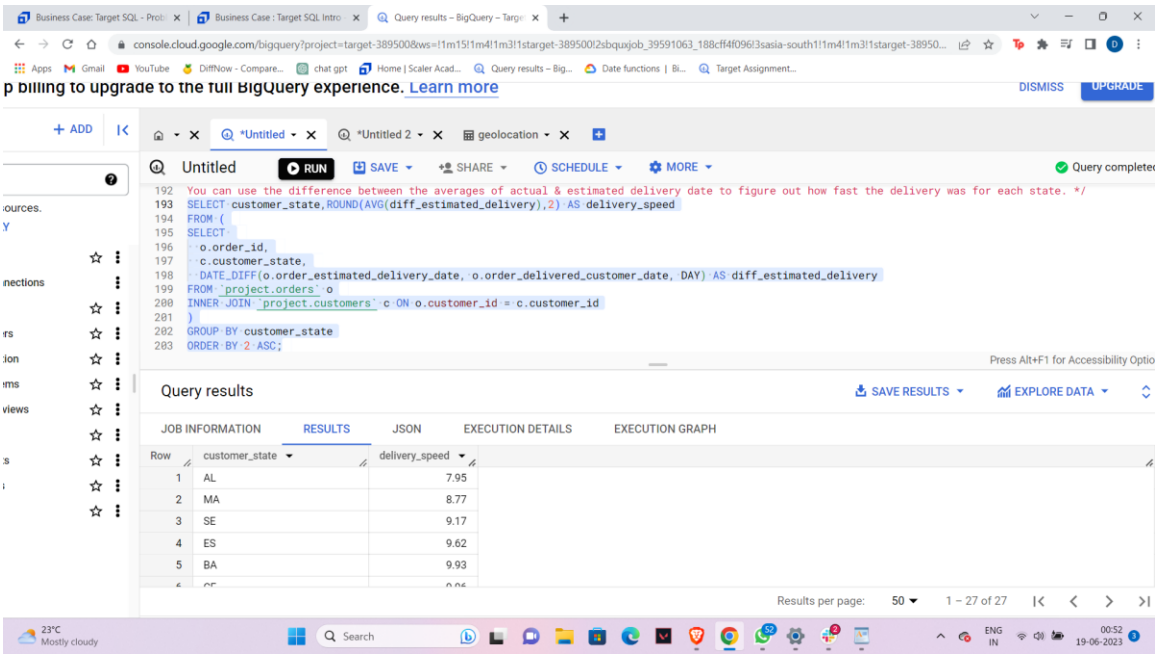
FROM `project.orders` o

INNER JOIN `project.customers` c ON o.customer_id = c.customer_id

)

GROUP BY customer_state

ORDER BY 2 ASC;

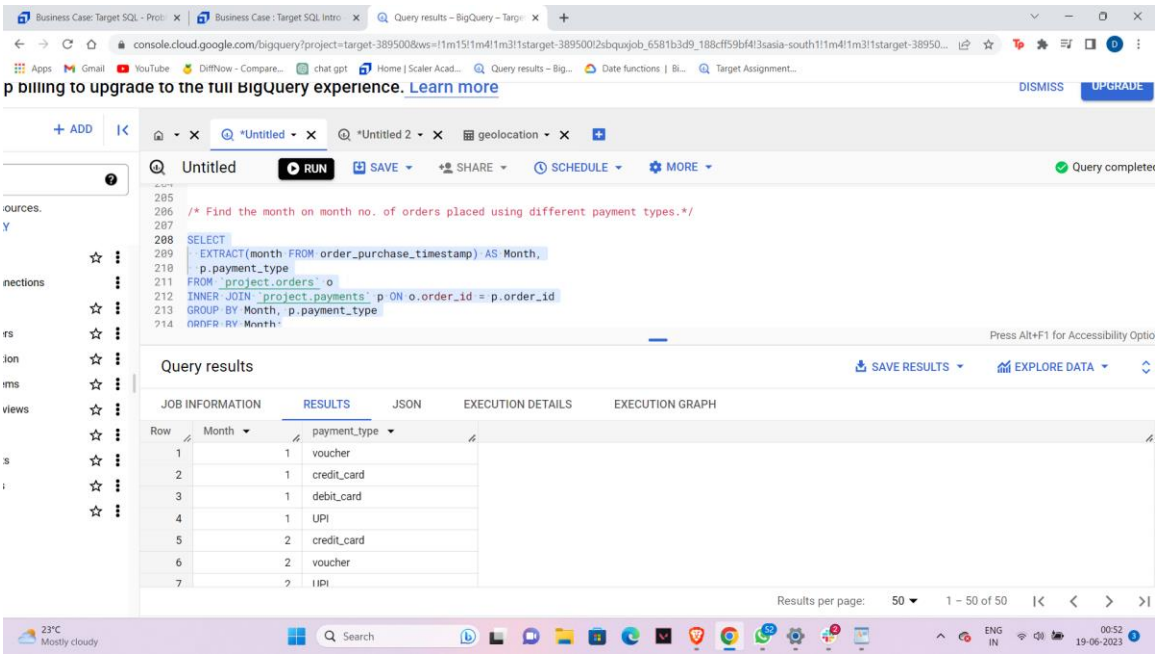


Insights and recommendation: Here we are checking how the delivery is faster than the estimated time and how this can improve customer satisfaction and draw insights from customer reviews about the same.

Q6/* Find the month on month no. of orders placed using different payment types.*/

SELECT

```
    EXTRACT(month FROM order_purchase_timestamp) AS Month,
    p.payment_type
FROM `project.orders` o
INNER JOIN `project.payments` p ON o.order_id = p.order_id
GROUP BY Month, p.payment_type
ORDER BY Month;
```

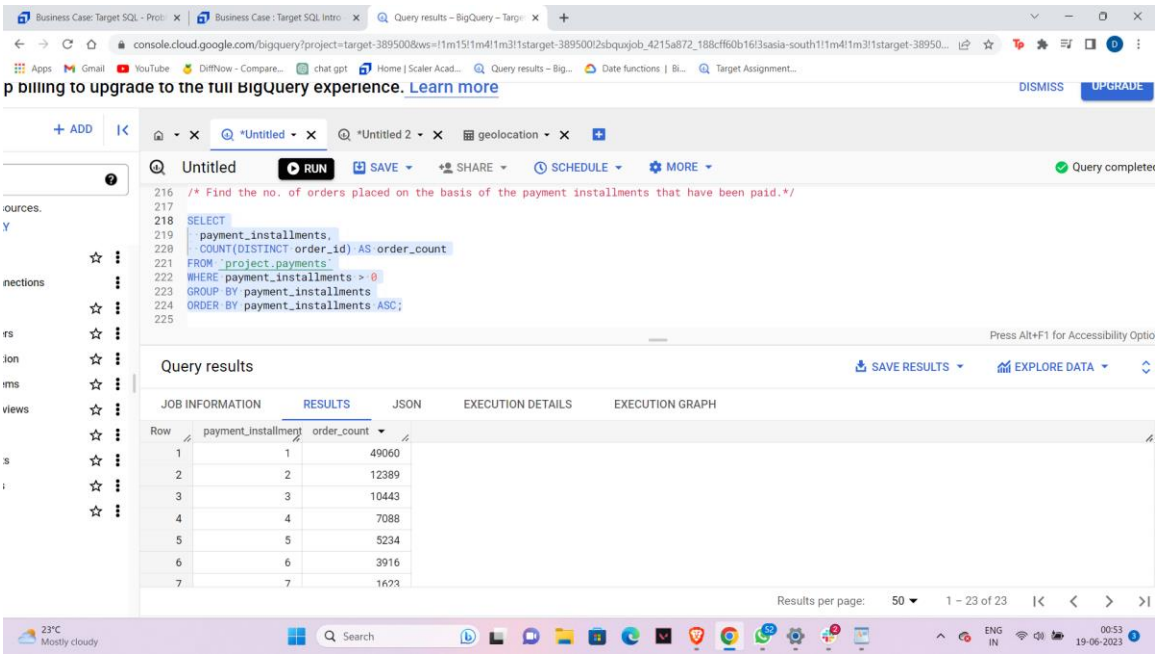


Insights and recommendation: Here we can conclude the different payment methods used by customers and understand the ease of payment and offer an analysis to check which of the methods have high success rates of making the payment without causing an error.

/* Find the no. of orders placed on the basis of the payment installments that have been paid.*/

```
SELECT
    payment_installments,
```

```
    COUNT(DISTINCT order_id) AS order_count
FROM `project.payments`
WHERE payment_installments > 0
GROUP BY payment_installments
ORDER BY payment_installments ASC;
```



Insights and recommendation: Here we can see how much percentage of people bought items because of the installment options and we how we could push more sales if given a no cost month installment option.

Conclusion:

1) We can improve sales by giving discounts during the year end where the sales is low

2) Improve the delivery speed in some regions by setting up warehouses within the city limits

3) Launch excited products during the dawn when the sales is really low and advertise more about it to improve the sales.

4) We must analyze what customers need according to the state they live in , check the culture , weather conditions and the need of the hour products and launch them exclusively.

5) Reduce the price of products that are being least purchased.

6) Offer discount coupons or vouchers to retain the customer to make another purchase.

And finally take insight from the top products sold in a region and mimic the approach according to the region/ state for better marketing.