Target (SQL) Business Case

About the Company:-

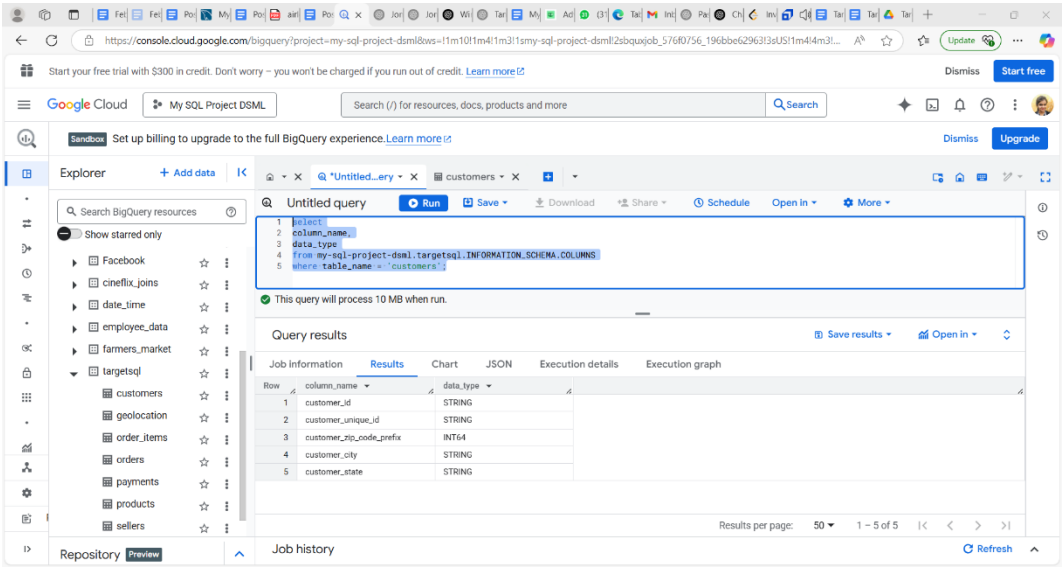
Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

Business Problem:- To gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

I. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

A. Data type of all columns in the “customers” table.

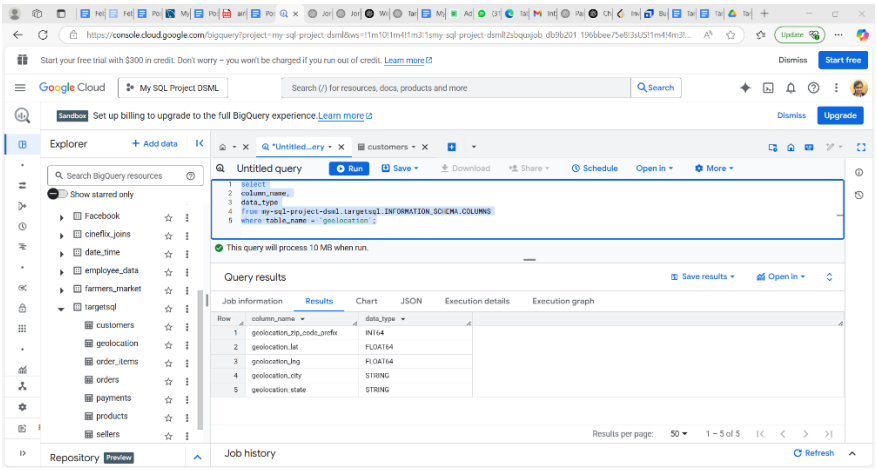
```
select
column_name,
data_type
from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers';
```



Insights & Recommendations:-

As per the above results, customers table have most of the columns of string datatype which we can say is of varchar type.

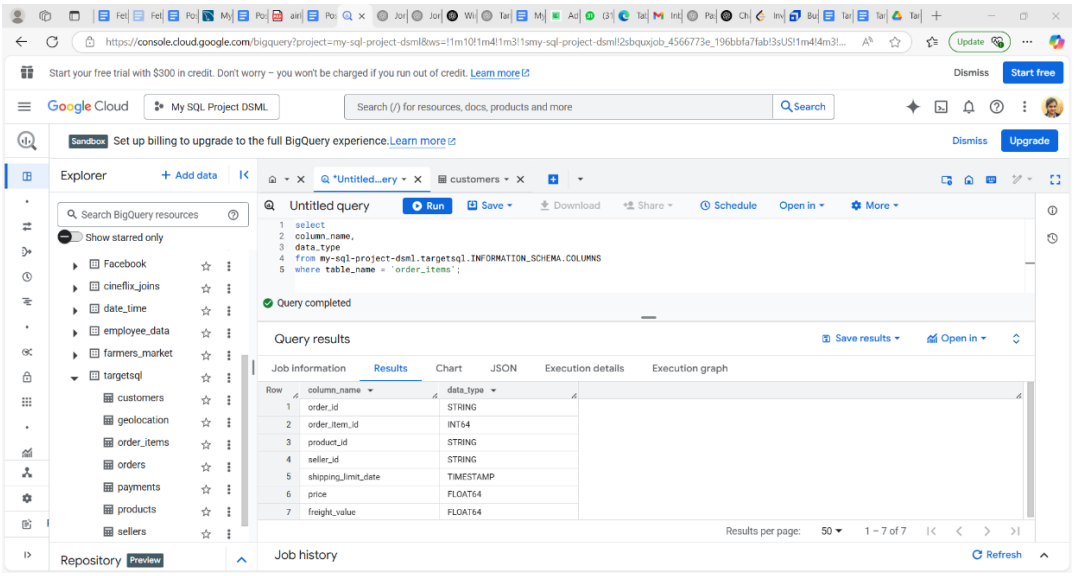
```
select

column_name,

data_type

from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS

where table_name = 'geolocation';
```



Insights & Recommendation:-

We have mixed data types – lat,lon are of float type, zip code is of int type, city and state is of string type hence this table contents complete location details.

```
select

column_name,

data_type

from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS

where table_name = 'order_items';
```



Insights & Recommendations:-

We can see this table contains order_id, product_id, order_id which are string however the date is of timestamp type and the other two price and freight value are float type.

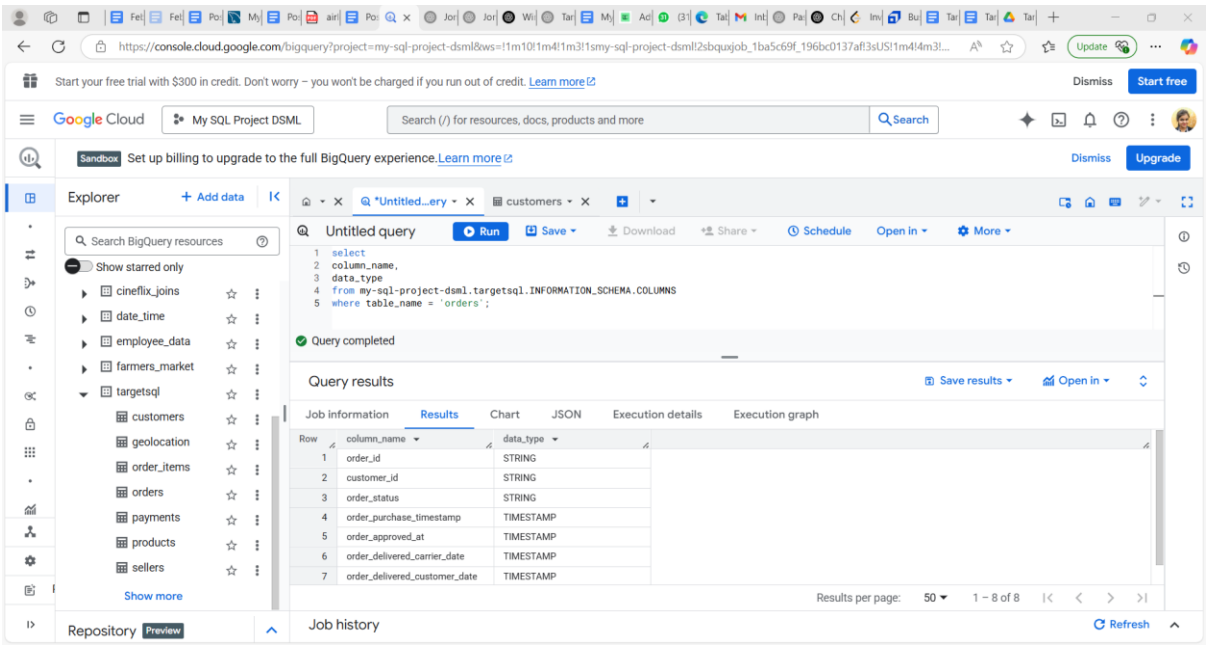
select

column_name,

data_type

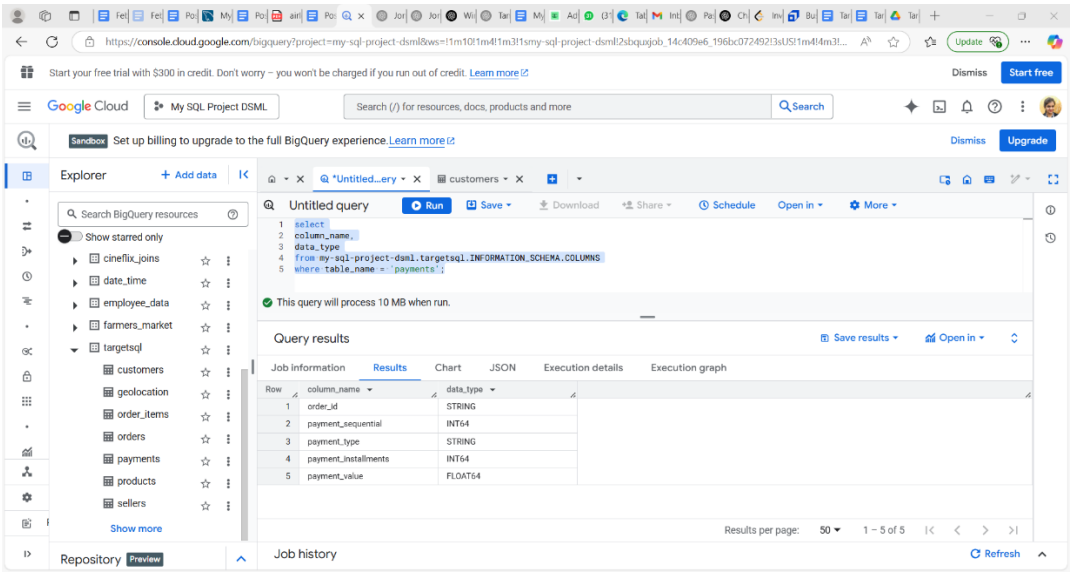
from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS

where table_name = 'orders';



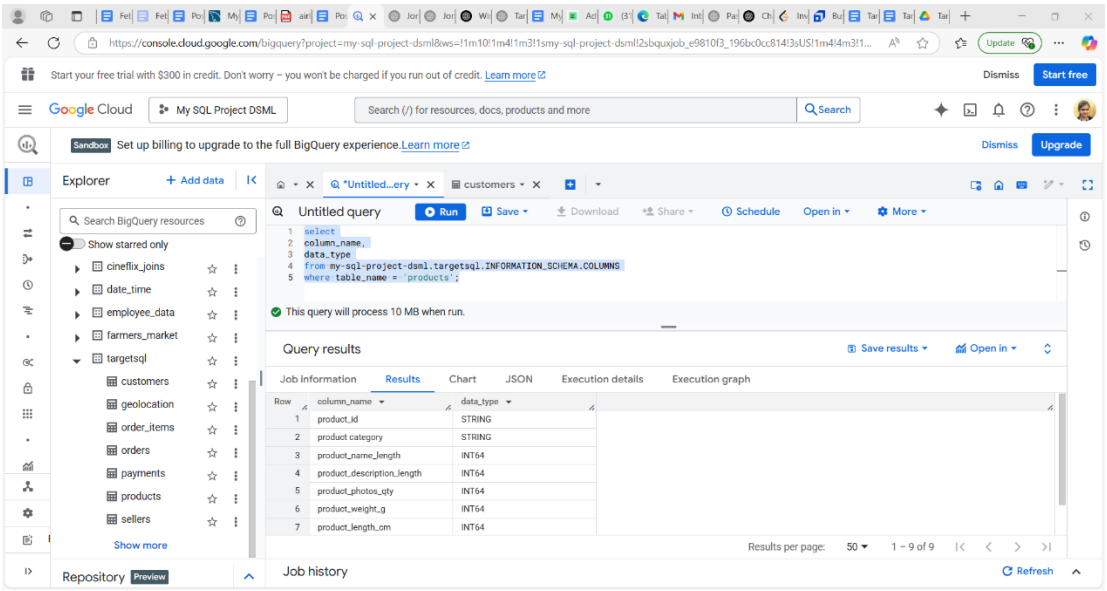
We can see the order_id,customer_id,order_status are of String type; and the rest three are of timestamp type hence the table orders has all the information for an order including the date and time at which the order was placed.

```
select

column_name,

data_type

from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS

where table_name = 'payments';
```



We can see the order_id and payment_type as string , payment_sequential and payment_installments as integer, payment_value as float;2 string , 2 integer and 1 float data type.

```
select

column_name,

data_type

from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS

where table_name = 'products';
```



We can see here two types of data types – String and Integer. Product_id and Product category are string type; Product_name_length, Product_description_length,Product_photos_qty,Product_weight_g,Product_length_cm ,Product_height_cm,Product_weight_cm are Integer type.

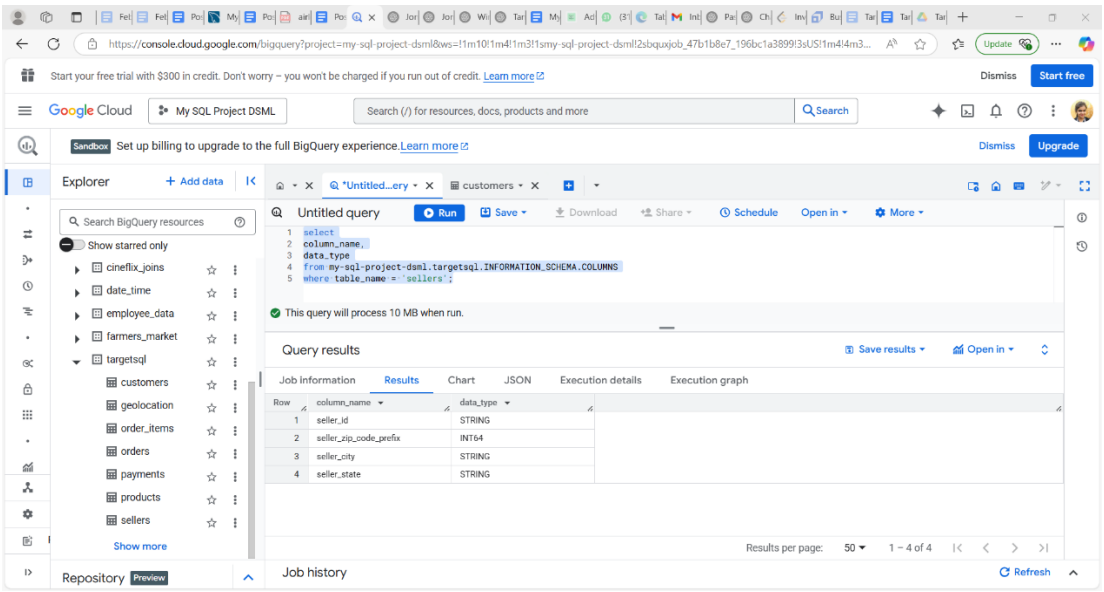
select

column_name,

data_type

from my-sql-project-dsml.targetsql.INFORMATION_SCHEMA.COLUMNS

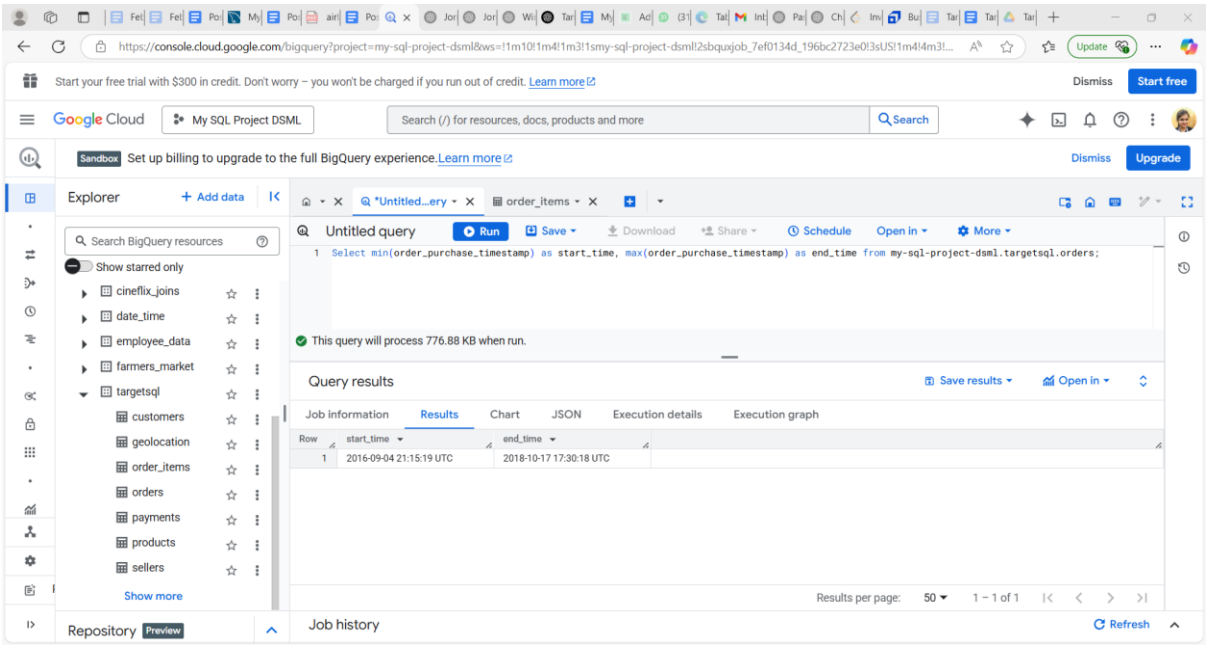
where table_name = 'sellers';



We can see two types of data types- string and integer. Seller city, state and id are string while seller zip code is integer.

B. Get the time range between which the orders were placed.

Select min(order_purchase_timestamp) as start_time, max(order_purchase_timestamp) as end_time from my-sql-project-dsml.targetsql.orders;

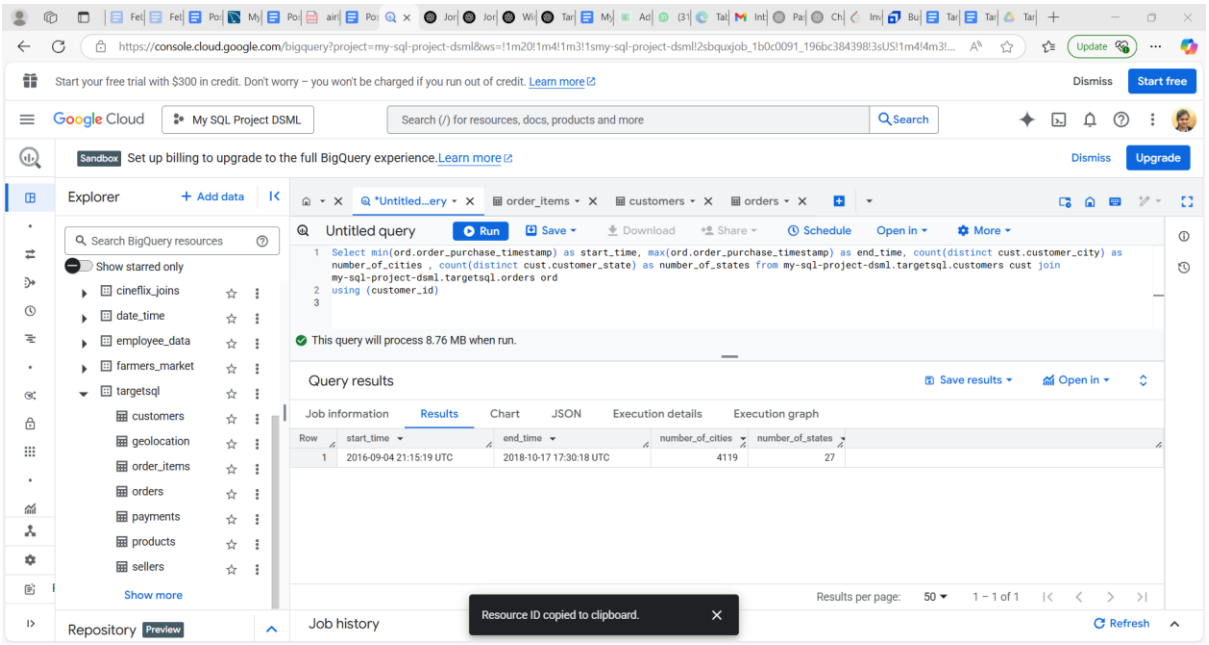


Insights:-

First order was placed at 9:59:19 PM UTC on 4th Sept 2016 and Last order was placed at 5:30:18 PM UTC on 17th Oct 2018.

C. Count the Cities & States of customers who ordered during the given period.

Select min(ord.order_purchase_timestamp) as start_time, max(ord.order_purchase_timestamp) as end_time, count(distinct cust.customer_city) as number_of_cities , count(distinct cust.customer_state) as number_of_states from my-sql-project-dsml.targetsql.customers cust join my-sql-project-dsml.targetsql.orders ord using (customer_id) ;



Insights:-

4119 cities from 27 states placed the order during the given time period from 9:59:19 PM UTC on 4th Sept 2016 to 5:30:18 PM UTC on 17th Oct 2018.

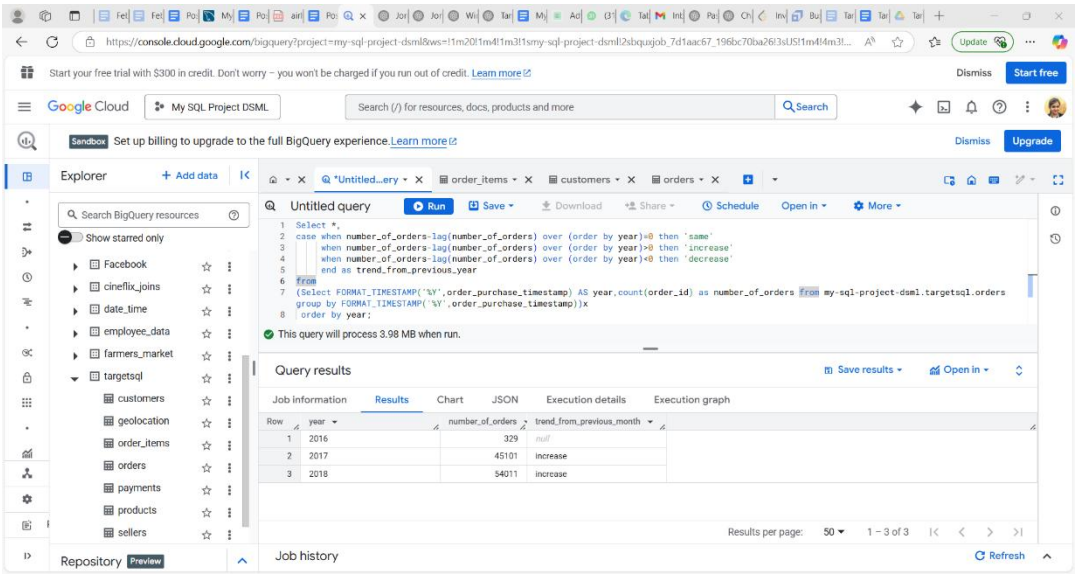
I. In-depth Exploration:

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

Exploration done both year wise and month wise

Year-Wise Analysis

```
Select *,

case when number_of_orders-lag(number_of_orders) over (order by year)=0 then 'same'

    when number_of_orders-lag(number_of_orders) over (order by year)>0 then 'increase'

    when number_of_orders-lag(number_of_orders) over (order by year)<0 then 'decrease'

    end as trend_from_previous_year

from

(Select FORMAT_TIMESTAMP('%Y',order_purchase_timestamp) AS year,count(order_id) as number_of_orders from my-sql-project-dsml.targetsql.orders group by FORMAT_TIMESTAMP('%Y',order_purchase_timestamp))x

 order by year;
```



Month-wise analysis

Select *,

case when number_of_orders-lag(number_of_orders) over (order by month)=0 then 'same'

when number_of_orders-lag(number_of_orders) over (order by month)>0 then 'increase'

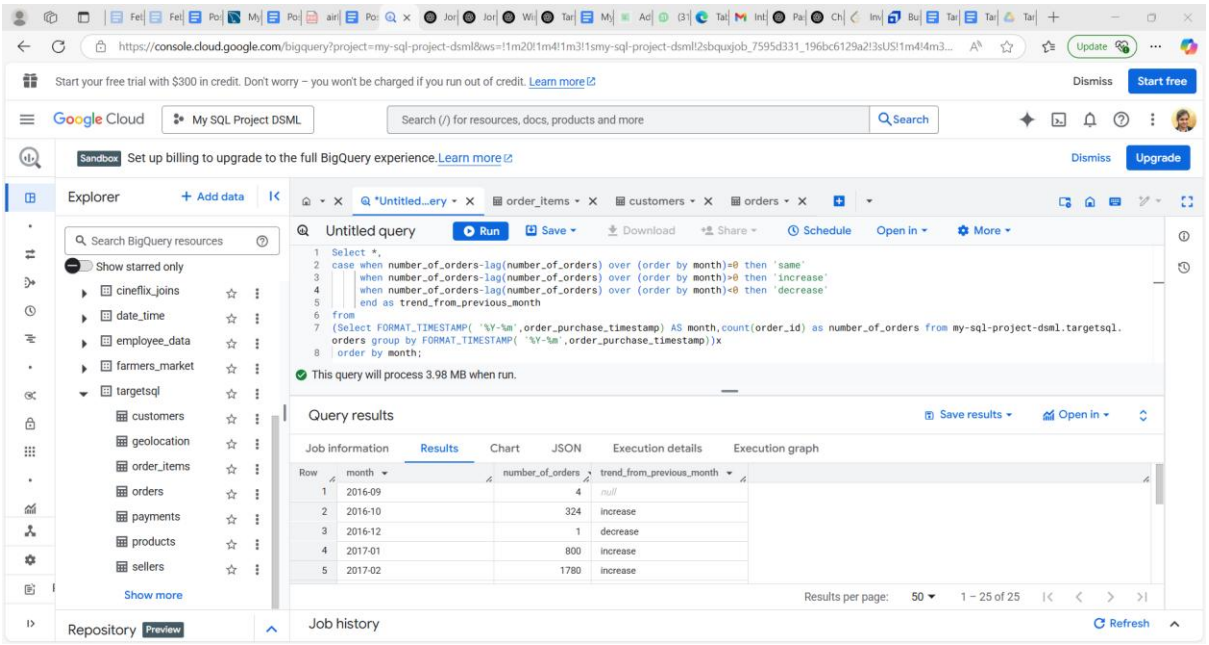
when number_of_orders-lag(number_of_orders) over (order by month)<0 then 'decrease'

end as trend_from_previous_month

from

(Select FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp) AS month,count(order_id) as number_of_orders from my-sql-project-dsml.targetsql.orders group by FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp))x

order by month;



Insights:-

Overall the number of orders have increased each year but if we see month-wise then in

2018 every month the number of orders are mostly decreasing.

In Year 2016, number of order increased from sept to October but then decreased in December.

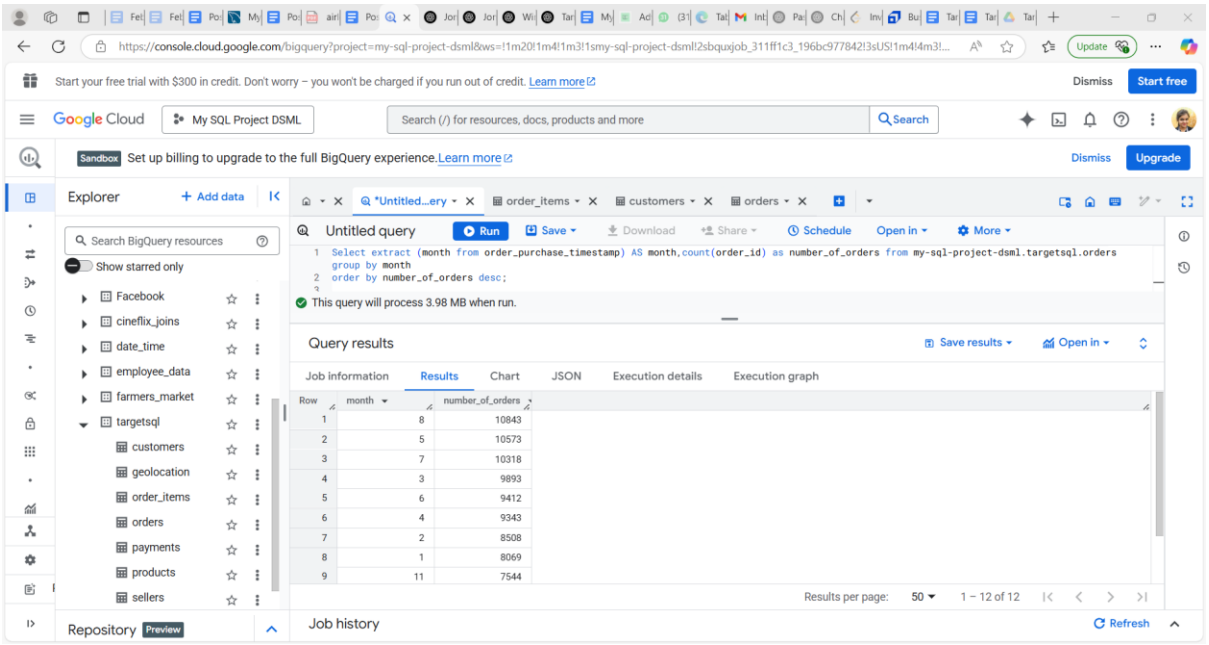
In year 2017, number of orders increased every month except from March to April.May to June, August to Sept and then November to December.

In year 2018, number of orders increased in the Month of January but then it started decreasing it only increased from Feb to March and then from June to July.

B. 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 month,count(order_id) as number_of_orders from my-sql-project-dsml.targetsql.orders group by month

order by number_of_orders desc;



Insights:-

Overall, highest number of orders have been placed during August,May and July Month, means, during mid-winter in Brazil(July,August) and also during cultural holidays (like Mother's Day widely celebrated in Brazil in May).

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

0-6 hrs : Dawn

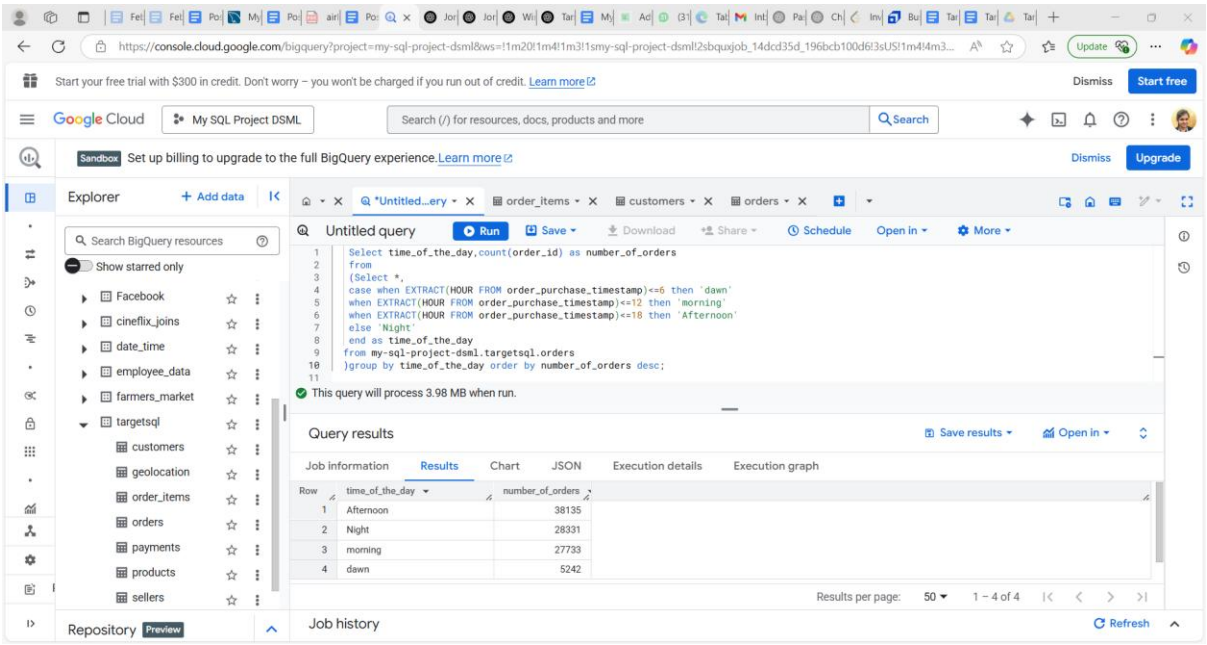
7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night

```
Select time_of_the_day,count(order_id) as number_of_orders
  from
  (Select *,
  case when EXTRACT(HOUR FROM order_purchase_timestamp)<=6 then 'dawn'
  when EXTRACT(HOUR FROM order_purchase_timestamp)<=12 then 'morning'
  when EXTRACT(HOUR FROM order_purchase_timestamp)<=18 then 'Afternoon'
```

```
else 'Night'
end as time_of_the_day
from my-sql-project-dsml.targetsql.orders
)
group by time_of_the_day
order by number_of_orders desc;
```



Insights:-

Highest number of orders have been placed in the afternoon (possibly because people Take lunch breaks during work) and after 7 PM they are free from work when it's their Family time going out during dinner so they are free to do online or offline shopping And making decisions with their families.

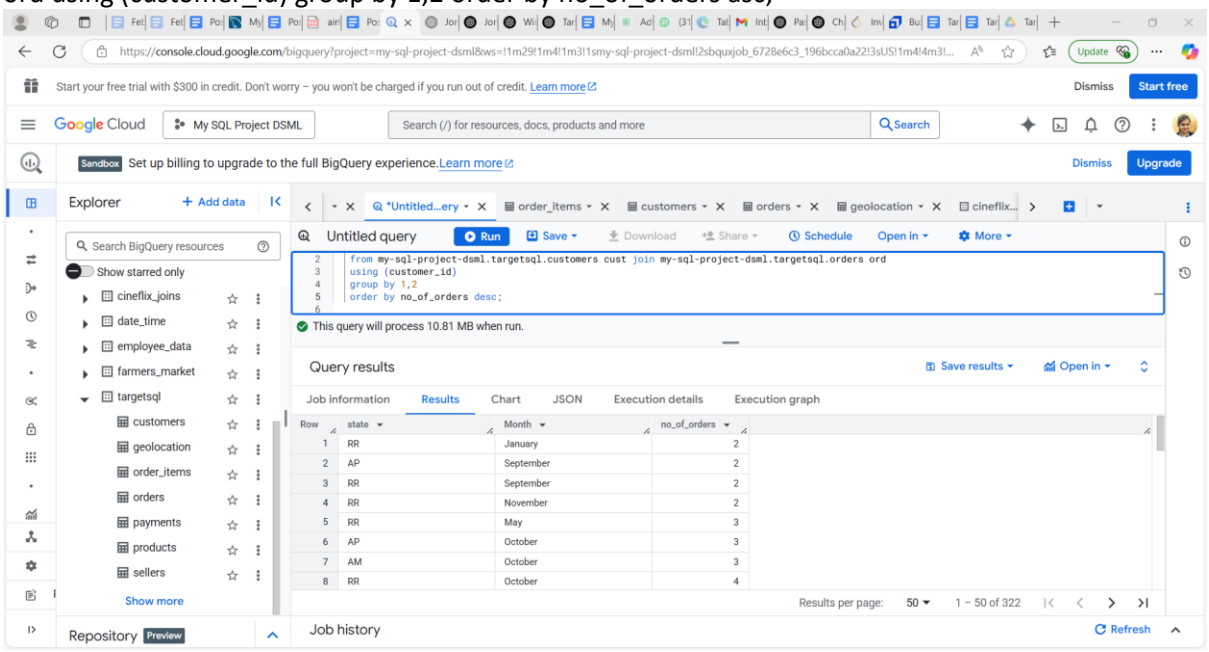
I. Evolution of E-commerce orders in the Brazil region:

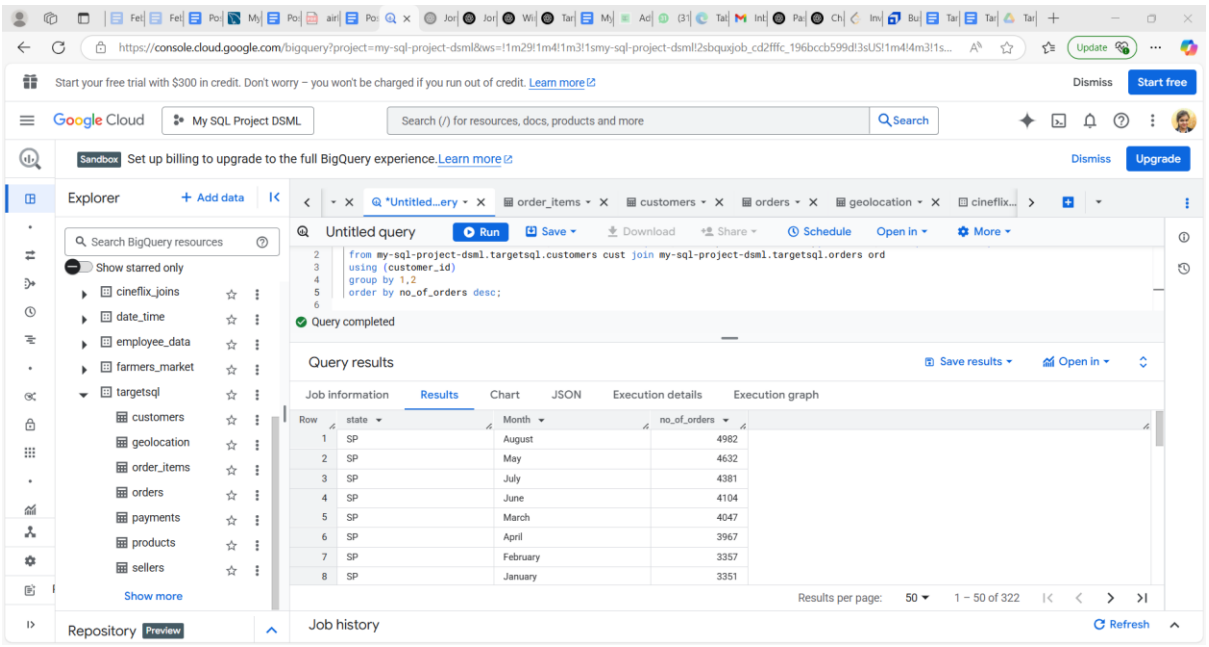
A. Get the month on month no. of orders placed in each state.

```
Select distinct (cust.customer_state) as state,FORMAT_TIMESTAMP('%B',
order_purchase_timestamp) as Month, count(ord.order_id) as no_of_orders
```

from my-sql-project-dsml.targetsql.customers cust join my-sql-project-dsml.targetsql.orders ord using (customer_id) group by 1,2 order by no_of_orders desc;

Select distinct (cust.customer_state) as state,FORMAT_TIMESTAMP('%B', order_purchase_timestamp) as Month, count(ord.order_id) as no_of_orders
from my-sql-project-dsml.targetsql.customers cust join my-sql-project-dsml.targetsql.orders ord using (customer_id) group by 1,2 order by no_of_orders asc;



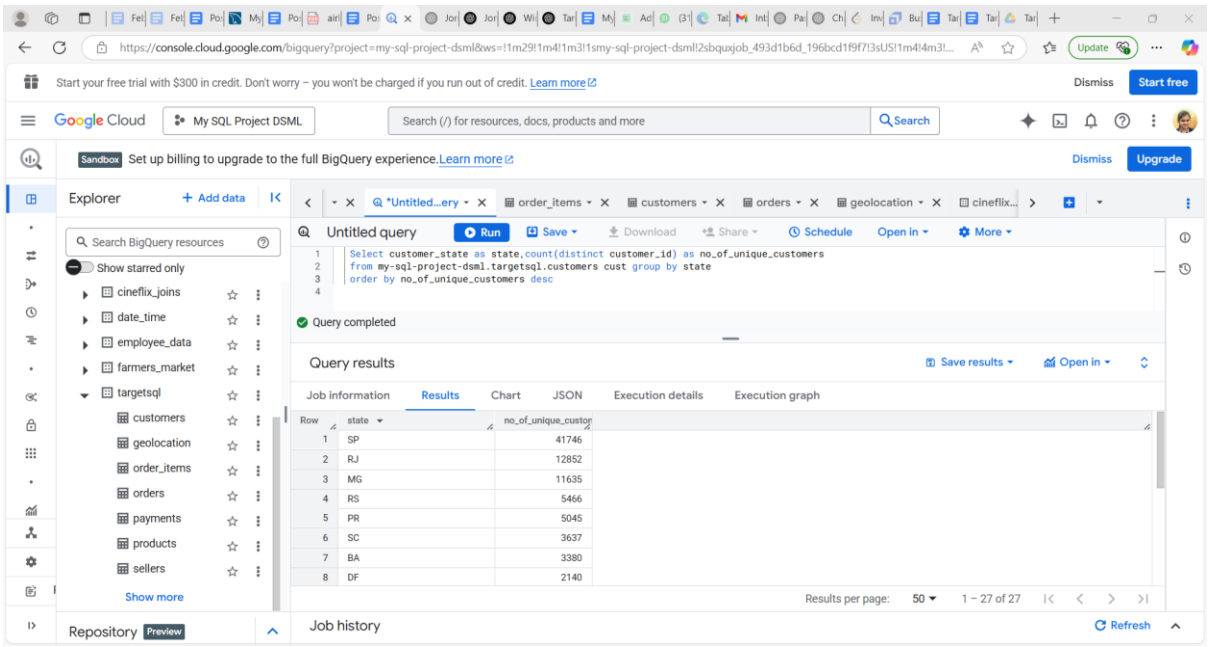


Insights:-

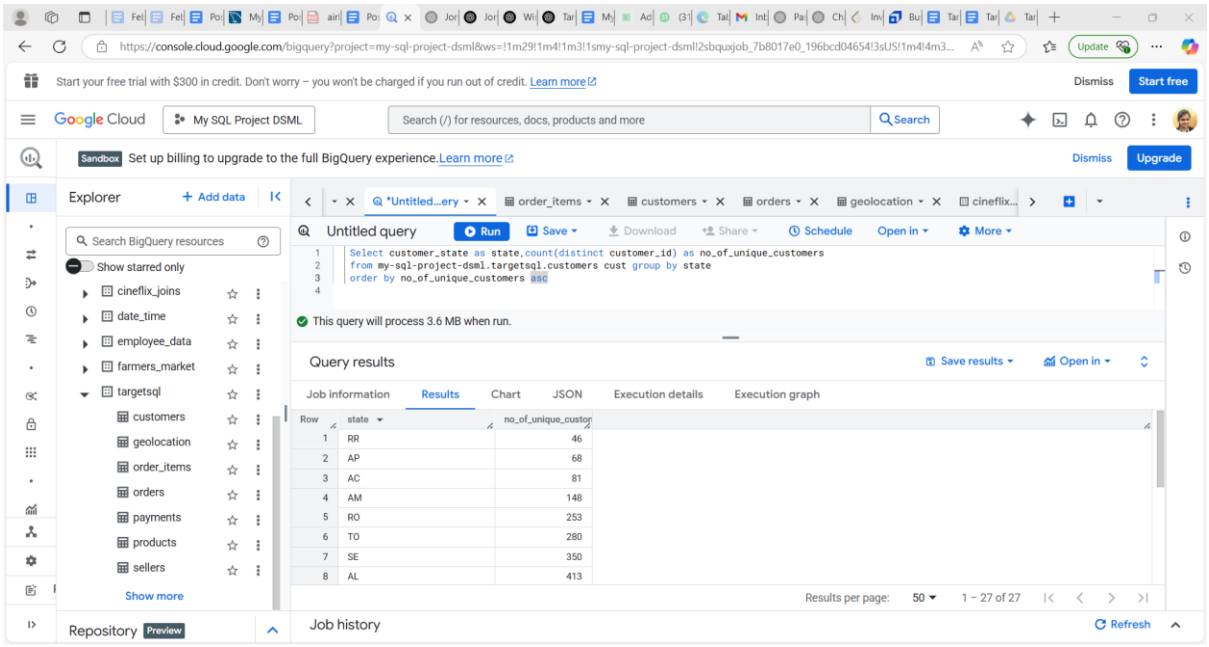
Highest number of orders have been placed in the Month of August in state SP , and lowest number of orders have been placed in the month of January in RR

B. How are the customers distributed across all the states?

Select customer_state as state,count(distinct customer_id) as no_of_unique_customers from my-sql-project-dsml.targetsql.customers cust group by state order by no_of_unique_customers desc



Select customer_state as state,count(distinct customer_id) as no_of_unique_customers from my-sql-project-dsml.targetsql.customers cust group by state order by no_of_unique_customers asc



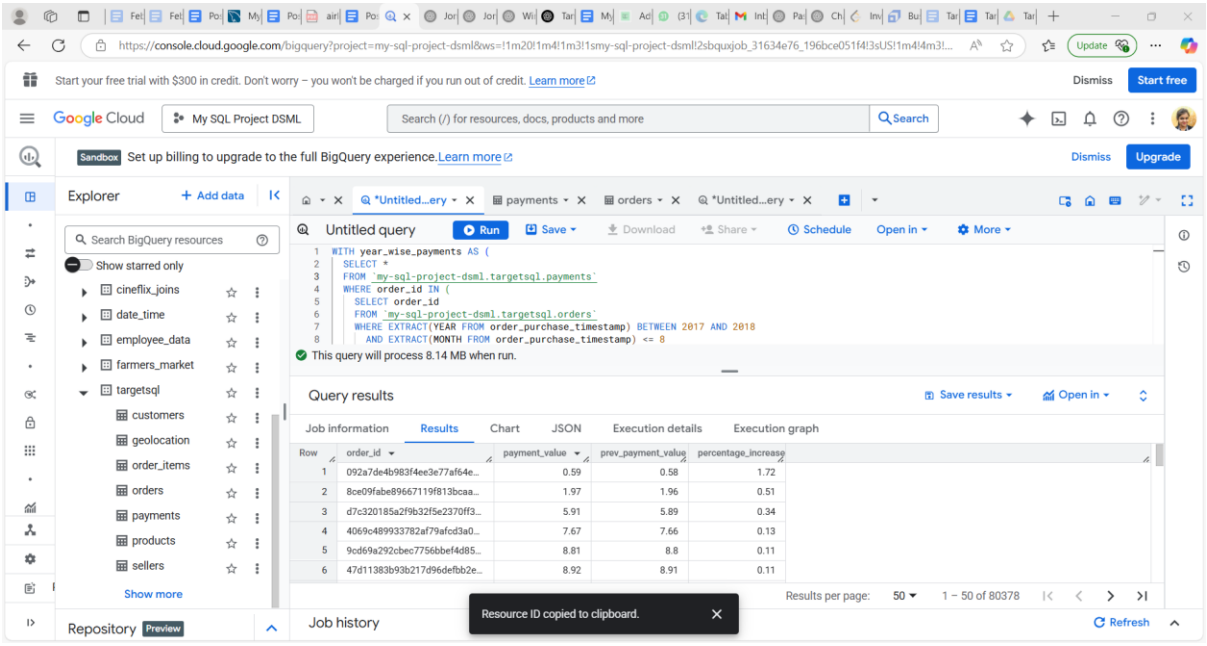
Insights:- Highest number of customers from state SP while lowest number of customers from state RR

I. Impact on Economy: Analyze 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 Jan to Aug only).*

```
 WITH year_wise_payments AS (
SELECT *
FROM `my-sql-project-dsml.targetsql.payments`
WHERE order_id IN (
  SELECT order_id
  FROM `my-sql-project-dsml.targetsql.orders`
  WHERE EXTRACT(YEAR FROM order_purchase_timestamp) BETWEEN 2017 AND 2018
    AND EXTRACT(MONTH FROM order_purchase_timestamp) <= 8
 )
),
lagged_payments AS (
 SELECT
  order_id,
  payment_value,
  LAG(payment_value) OVER (ORDER BY payment_value) AS prev_payment_value
 FROM year_wise_payments
)

SELECT
```

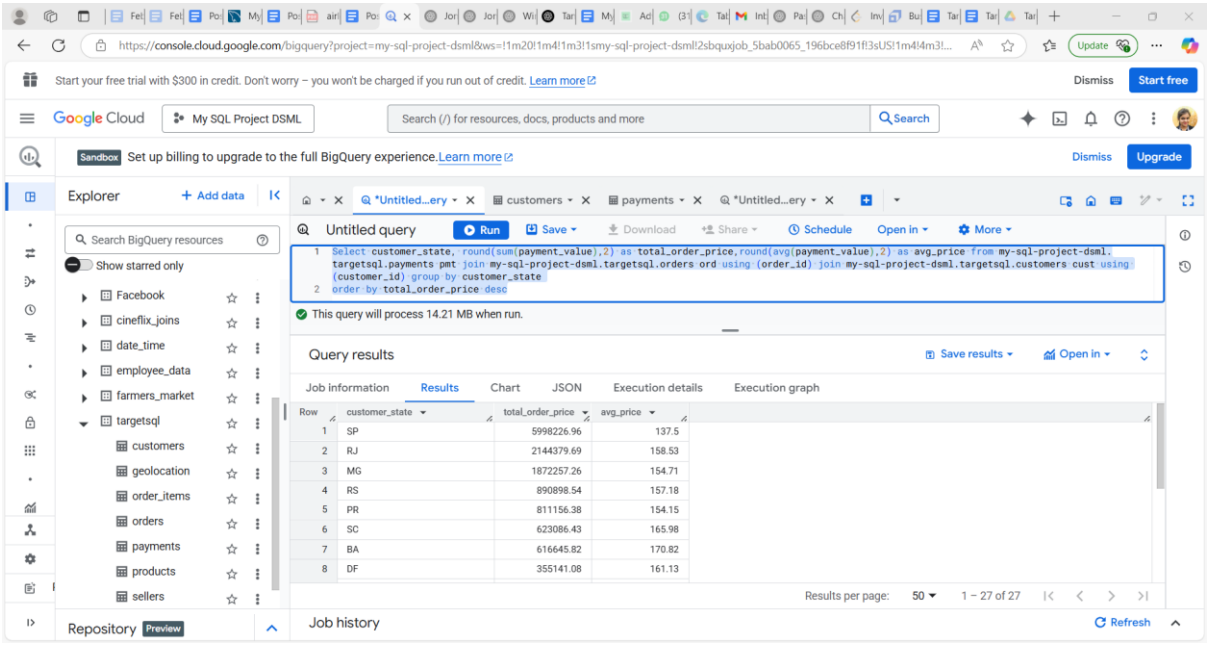
```
 order_id,
 payment_value,
 prev_payment_value,
 ROUND(
  100 * (payment_value - prev_payment_value) / prev_payment_value,
  2
 ) AS percentage_increase
FROM lagged_payments
WHERE prev_payment_value <> 0;
```



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

Select customer_state, round(sum(payment_value),2) as total_order_price,round(avg(payment_value),2) as avg_price from my-sql-project-dsml.targetsql.payments pmt join my-sql-project-dsml.targetsql.orders ord using (order_id) join my-sql-project-dsml.targetsql.customers cust using (customer_id) group by customer_state

order by total_order_price desc



Insights:- Highest order value from State SP while lowest order value from State RR

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

Select customer_state, round(sum(freight_value),2) as total_freight_value,round(avg(freight_value),2) as avg_freight_value from my-sql-project-dsml.targetsql.order_items oi join my-sql-project-dsml.targetsql.orders ord using (order_id) join my-sql-project-dsml.targetsql.customers cust using (customer_id) group by customer_state

order by total_freight_value desc

Insights:- Highest freight value from State SP while lowest freight order value from State RR

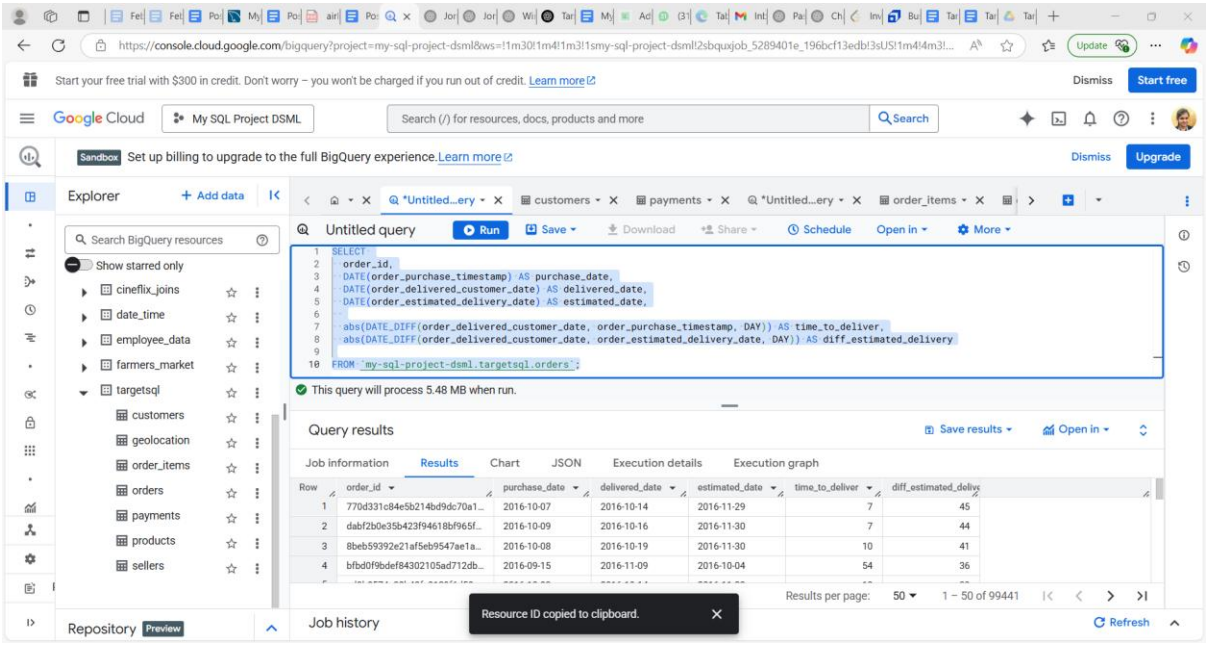
I.Analysis based on sales, freight and delivery time.

A. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

```
SELECT
  order_id,
  DATE(order_purchase_timestamp) AS purchase_date,
  DATE(order_delivered_customer_date) AS delivered_date,
  DATE(order_estimated_delivery_date) AS estimated_date,

  abs(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
time_to_deliver,
```

abs(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY)) AS diff_estimated_delivery

FROM `my-sql-project-dsml.targetsql.orders`;



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

select cust.customer_state,avg(oi.freight_value) as avg_freight_value from my-sql-project-dsml.targetsql.customers cust join my-sql-project-dsml.targetsql.orders ord using(customer_id) join my-sql-project-dsml.targetsql.order_items oi

using (order_id)

group by customer_state

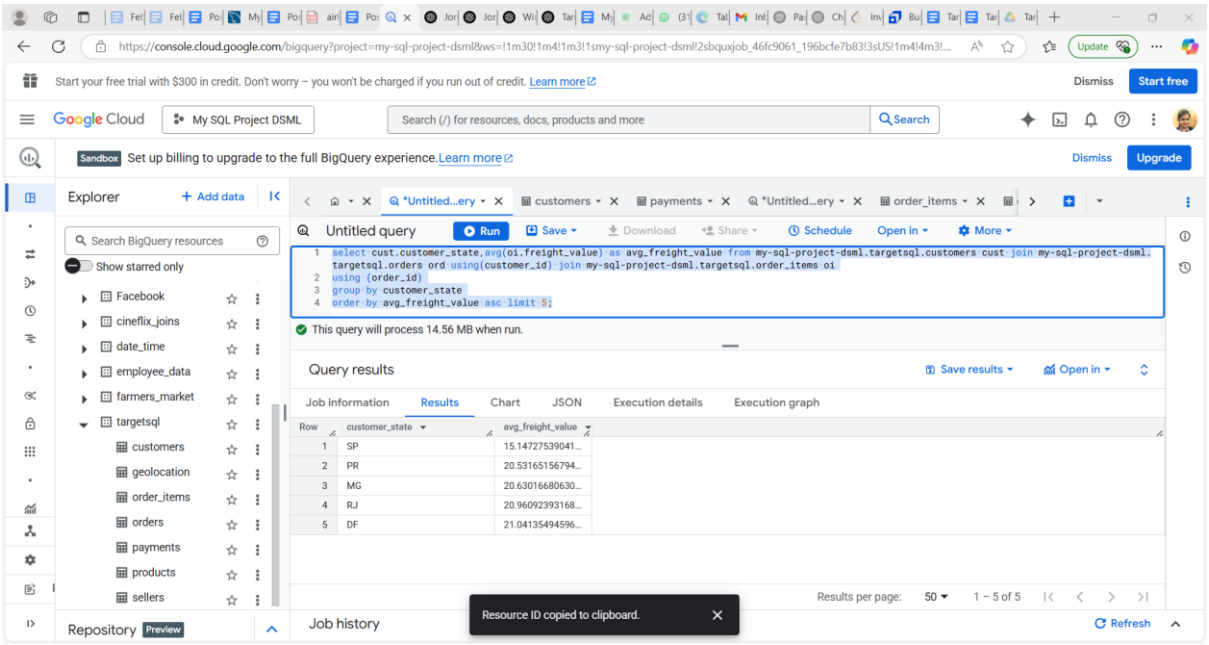
order by avg_freight_value asc limit 5;

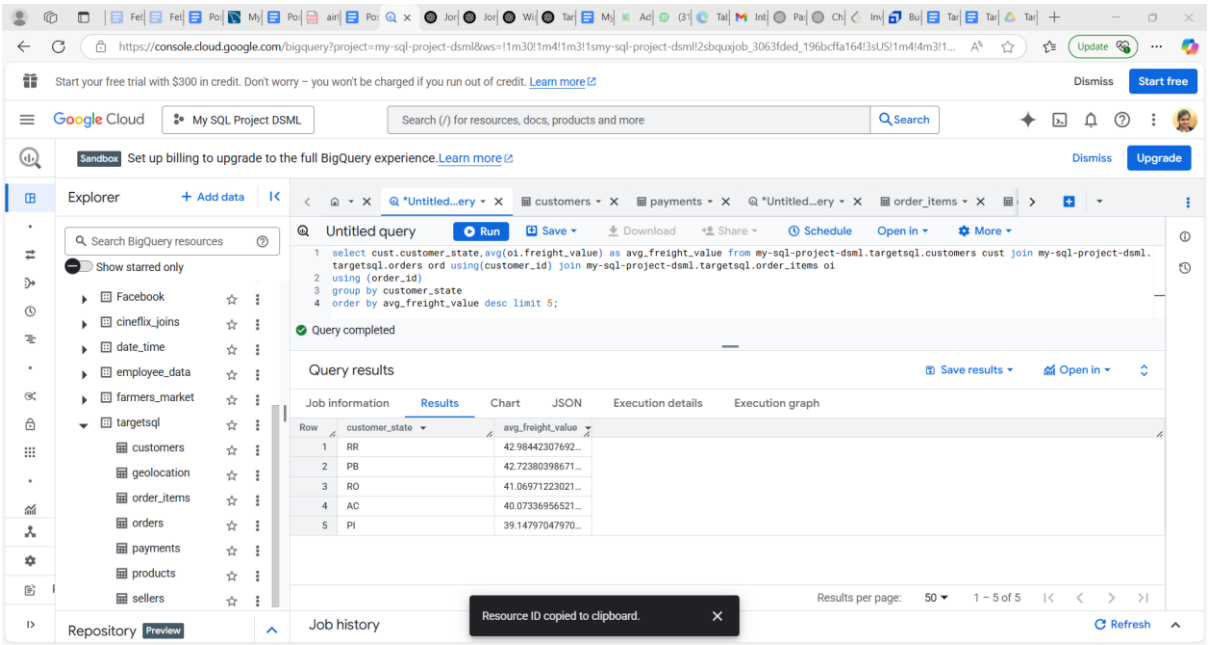
select cust.customer_state,avg(oi.freight_value) as avg_freight_value from my-sql-project-dsml.targetsql.customers cust join my-sql-project-dsml.targetsql.orders ord using(customer_id) join my-sql-project-dsml.targetsql.order_items oi

using (order_id)

group by customer_state

order by avg_freight_value desc limit 5;





Insights:- SP is the state with the lowest average freight value and RR is the state with the highest freight value

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

select cust.customer_state,avg(ord.time_to_deliver) as avg_time_to_deliver from my-sql-project-dsml.targetsql.customers cust join

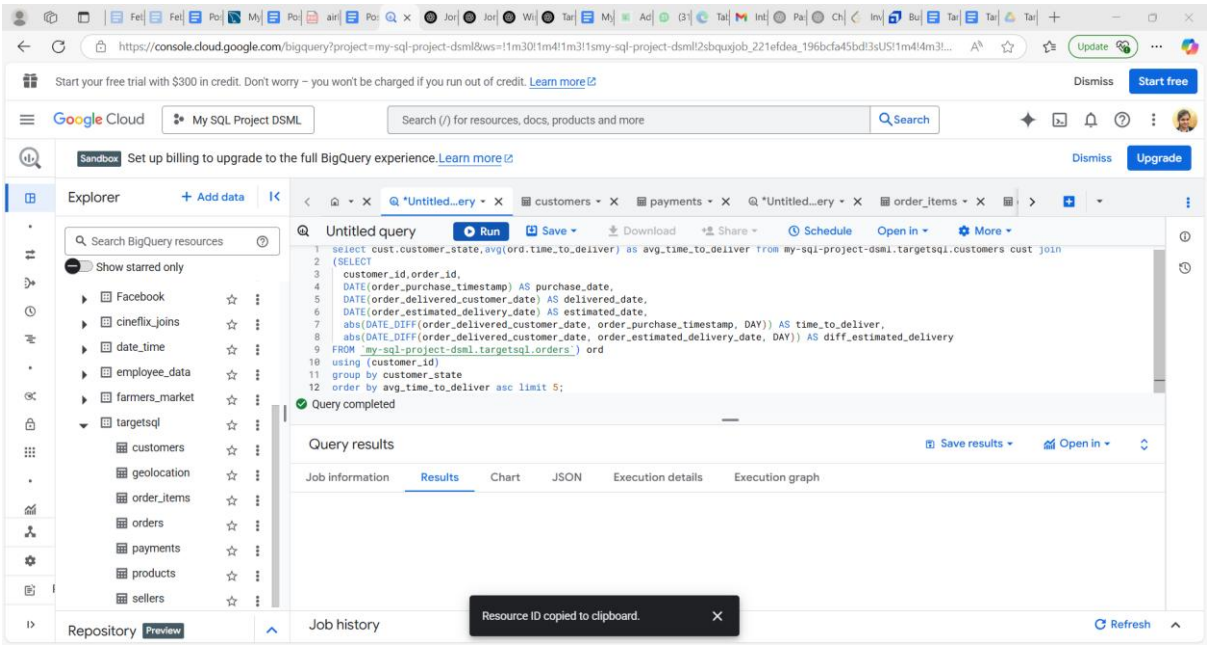
(SELECT

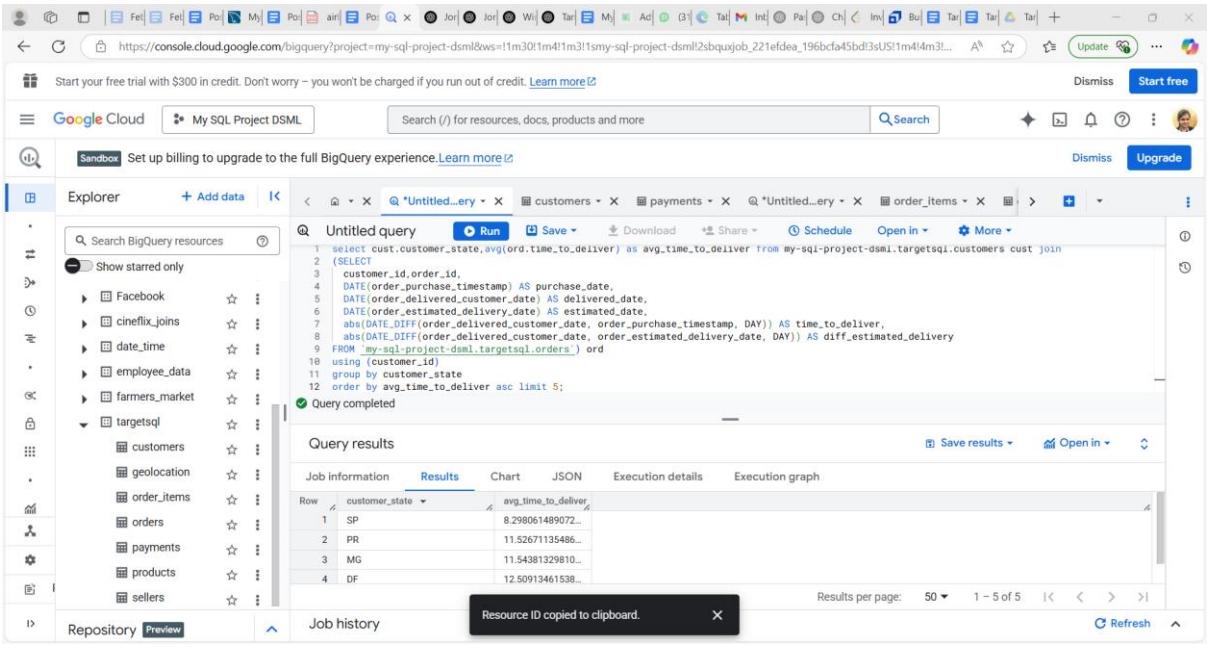
customer_id,order_id,

```
  DATE(order_purchase_timestamp) AS purchase_date,

  DATE(order_delivered_customer_date) AS delivered_date,

  DATE(order_estimated_delivery_date) AS estimated_date,



  abs(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
time_to_deliver,

  abs(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY)) AS
diff_estimated_delivery



FROM `my-sql-project-dsml.targetsql.orders`) ord

using (customer_id)

group by customer_state

order by avg_time_to_deliver desc limit 5;

 select cust.customer_state,avg(ord.time_to_deliver) as avg_time_to_deliver from my-sql-
project-dsml.targetsql.customers cust join

(SELECT

  customer_id,order_id,

  DATE(order_purchase_timestamp) AS purchase_date,

  DATE(order_delivered_customer_date) AS delivered_date,

  DATE(order_estimated_delivery_date) AS estimated_date,



  abs(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)) AS
time_to_deliver,

  abs(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY)) AS
diff_estimated_delivery



FROM `my-sql-project-dsml.targetsql.orders`) ord

using (customer_id)
```

group by customer_state

order by avg_time_to_deliver asc limit 5;

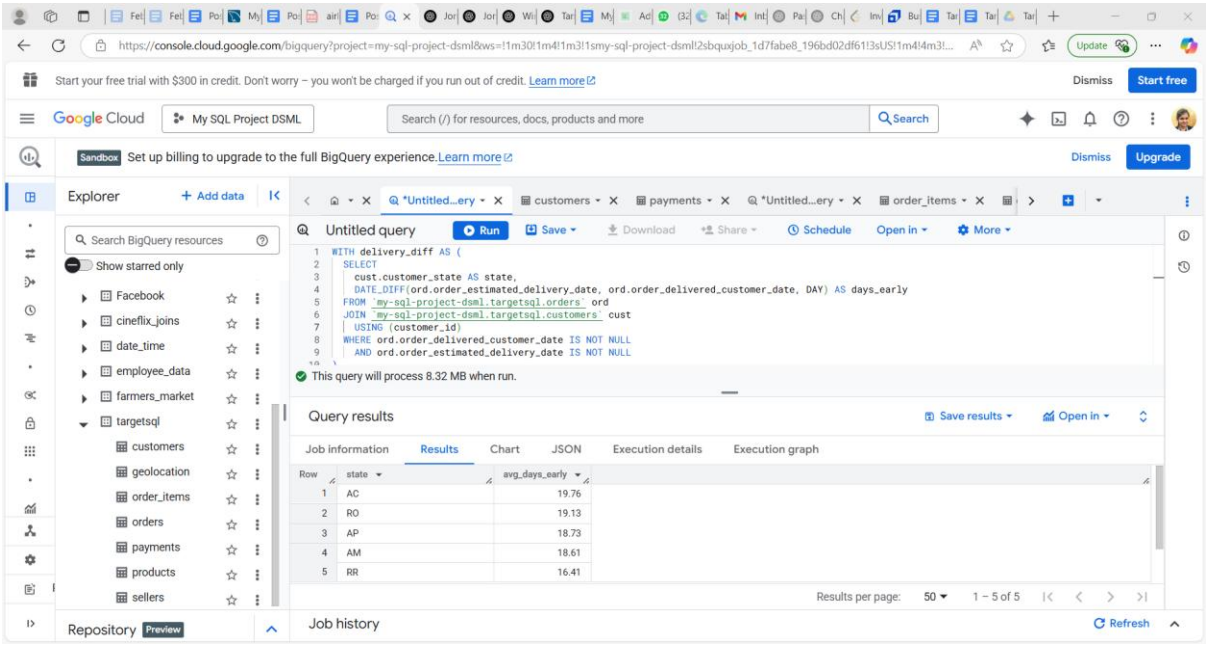




Insights:- SP is the state with the lowest average time delivery and RR is the state with the highest average time delivery

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

```
WITH delivery_diff AS (
  SELECT
    cust.customer_state AS state,
    DATE_DIFF(ord.order_estimated_delivery_date, ord.order_delivered_customer_date, DAY)
AS days_early
  FROM `my-sql-project-dsml.targetsql.orders` ord
  JOIN `my-sql-project-dsml.targetsql.customers` cust
    USING (customer_id)
  WHERE ord.order_delivered_customer_date IS NOT NULL
    AND ord.order_estimated_delivery_date IS NOT NULL
)
, avg_early_delivery AS (
  SELECT
    state,
    ROUND(AVG(days_early), 2) AS avg_days_early
  FROM delivery_diff
  GROUP BY state
)
SELECT *
FROM avg_early_delivery
WHERE avg_days_early > 0
ORDER BY avg_days_early DESC
LIMIT 5;
```



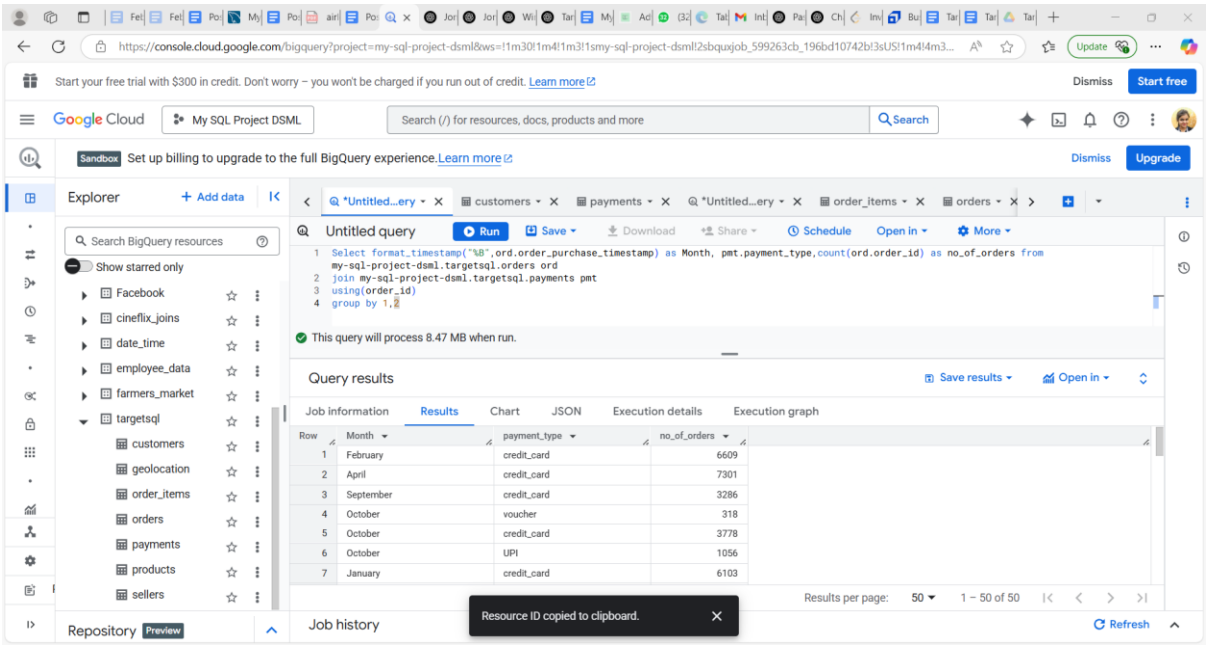
Insights:- Order delivery is really fast as compared to estimated delivery date in state AC.
Top 5 States:-

	state	avg_days_early	
1	AC	19.76	
2	RO	19.13	
3	AP	18.73	
4	AM	18.61	
5	RR	16.41	

I.Analysis based on the payments:

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

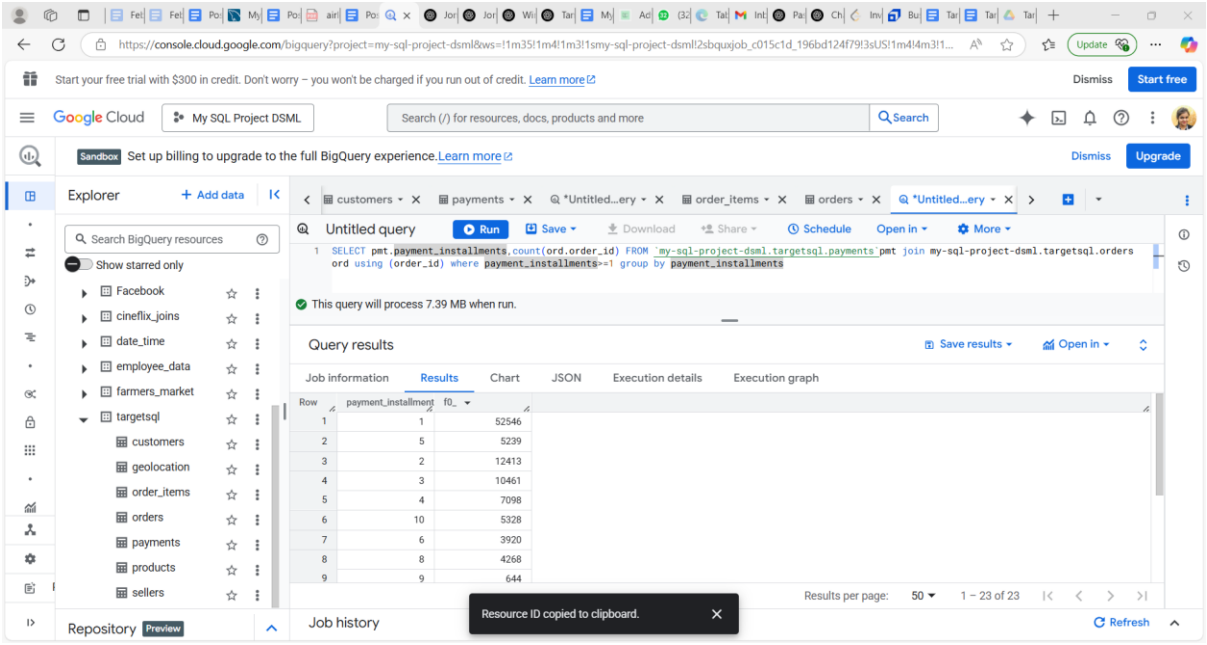
```
Select format_timestamp("%B",ord.order_purchase_timestamp) as Month,
pmt.payment_type,count(ord.order_id) as no_of_orders from my-sql-project-
dsml.targetsql.orders ord
join my-sql-project-dsml.targetsql.payments pmt
using(order_id)
group by 1,2
```



Insights:- Highest number of orders have been placed using credit card in the month of May.

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

SELECT pmt.payment_installments,count(ord.order_id) FROM `my-sql-project-dsml.targetsql.payments`pmt join my-sql-project-dsml.targetsql.orders ord using (order_id) where payment_installments>=1 group by payment_installments



Insights:- 52,546 orders have been placed where atleast 1 payment installment have been made.

Overall Insights:-

1. The number of orders have increased each year
2. In the month of 2018, the order have mostly decreased over the months.
3. Highest number of orders have been placed during August,May and July Month, means, during mid-winter in Brazil(July,August) and also during cultural holidays (like Mother's Day widely celebrated in Brazil in May).
4. Highest number of orders have been placed in the afternoon (possibly because people take lunch breaks during work) and after 7 PM they are free from work when it's their Family time going out during dinner so they are free to do online or offline shopping and making decisions with their families. Lowest orders in September.
5. Highest number of orders have been placed in the Month of August in state SP , and lowest number of orders have been placed in the month of January in RR
6. Highest Order and Freight Value from State SP while the lowest from state RR.
7. SP is the state with the lowest average time delivery and RR is the state with the highest average time delivery
8. Order delivery is really fast as compared to estimated delivery date in state AC.
9. Highest number of orders have been placed using credit card in the month of May.
10. For 52,546 orders , atleast 1 payment installment has been made.

Recommendations:-

1. Target should launch special discount and offers in the month of May especially on mother's day in Brazil in May.
2. Target should launch special discount and offers in the month of July and August as well when there is winter season especially on winter clothes and accessories .
3. Target should give good discounts on credit cards because it is the mostly used payment method especially in the months of May,July and August.
4. Target should work on reducing the delivery time for state RR because that is possibly the reason why this state has lowest number of orders, should consider launching more stores or warehouses , partner with more local delivery companies.
5. For state SP, Target can launch some rewards for it's most loyal customers as it is the state with highest order .
6. Target should give special discounts specifically between 7 AM-12 PM on online purchases especially on products for housewives or elderly people who don't commute to work to boost sales in the morning time as well.
7. Run digital ads and push notifications between **1–3 PM** (lunch) and **7–10 PM** (family & shopping time).
8. AC is the state with really fast delivery as compared to estimated delivery date, target should promote this as "early delivery guarantee" maximum in this state to increase number of orders.
9. For orders with higher value, credit card with EMI options should be more visible on the application and websites. Also no-cost EMI options should be given during mid year.
10. As there are lowest orders in September Month (when there is spring season in Brazil), more promotions and offers on clothing and accessories (like spring season essentials) or may be pre-holiday kits because from November holiday season starts can boost sales.
11. In January , more promotions and offers on products like new year planners can boost sales. Special Instagram giveaways or flash sales can work.