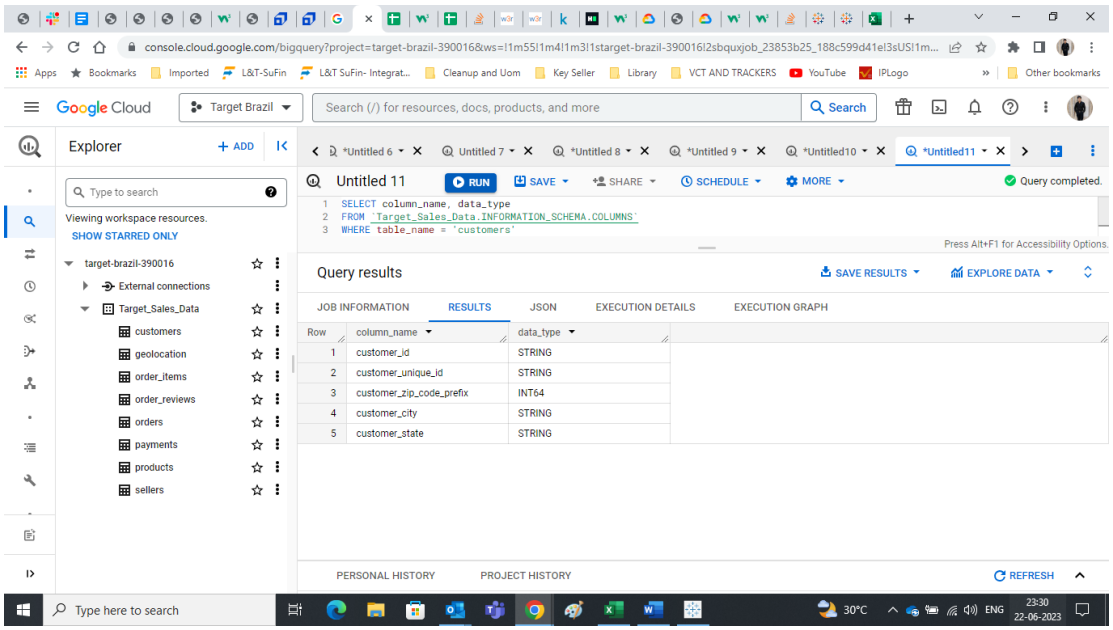
Business Case: Target United States.

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

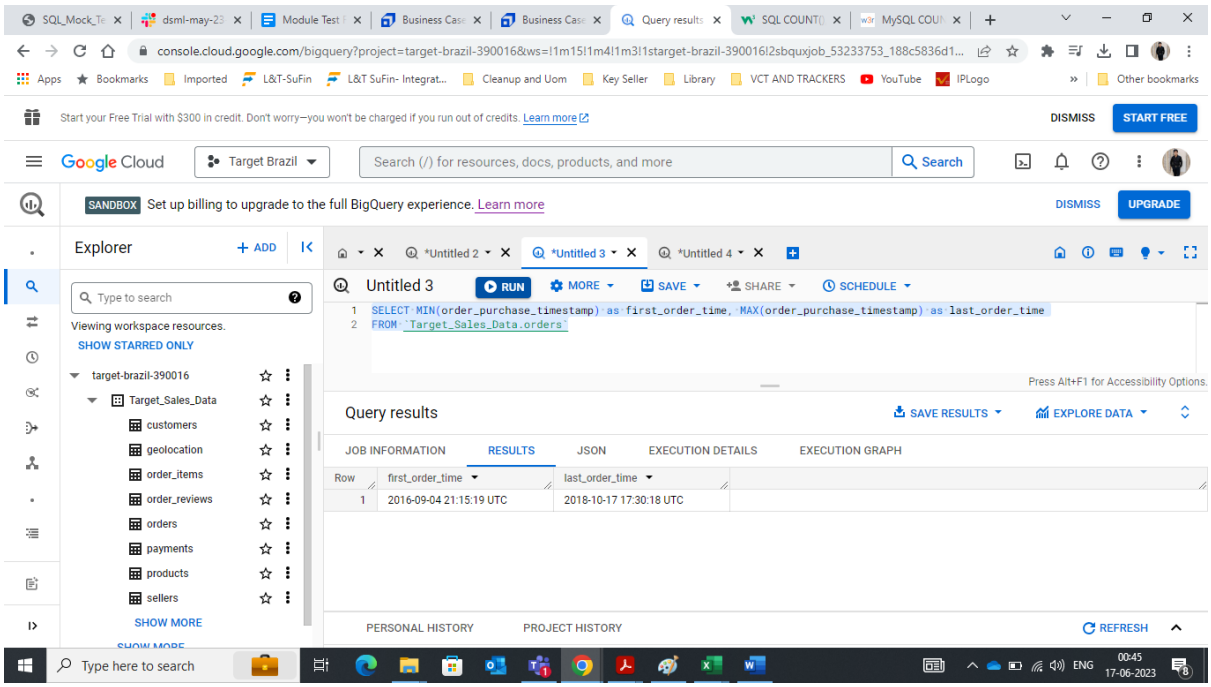
   1.1. **Data type of all columns in the "customers" table.**

```
SELECT column_name, data_type
FROM `Target_Sales_Data.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
```



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

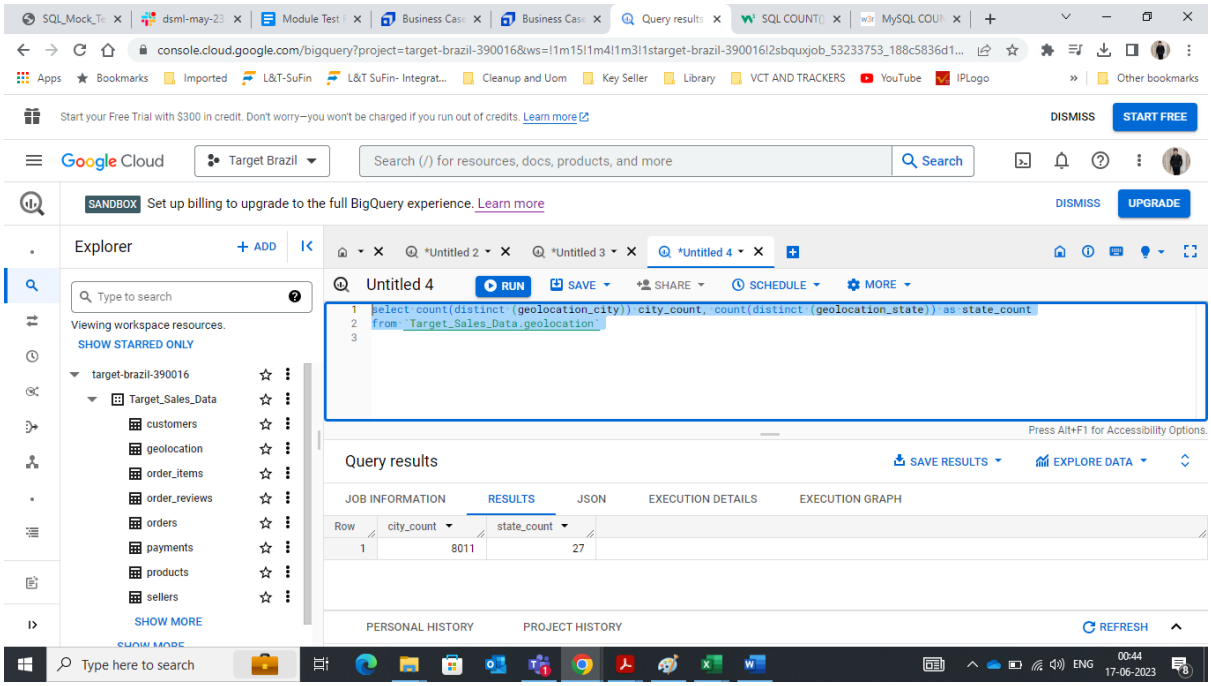
```
SELECT MIN(order_purchase_timestamp) as first_order_time,
MAX(order_purchase_timestamp) as last_order_time
FROM `Target_Sales_Data.orders`
```



Business Case: Target United States.

1.3. Count the number of Cities and States in our dataset.

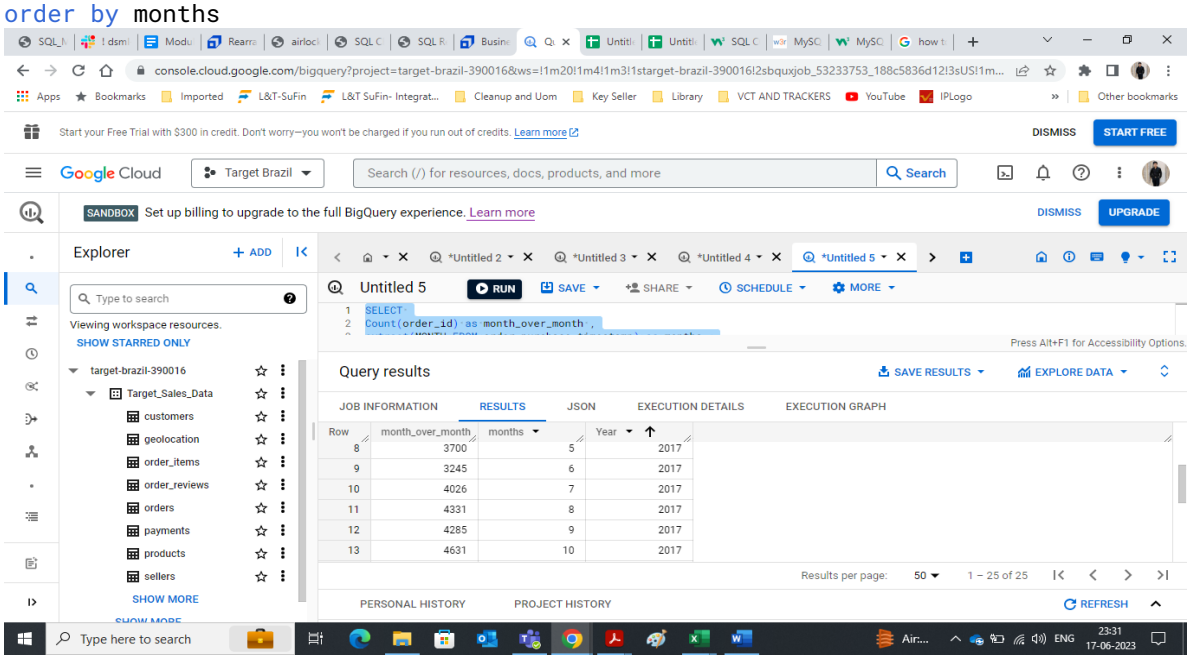
```
select
      count(distinct (geolocation_city)) city_count,
      count(distinct (geolocation_state)) as state_count
from `Target_Sales_Data.geolocation`
```



2. In-depth Exploration:

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

```
SELECT
      Count(order_id) as month_over_month ,
      extract(MONTH FROM order_purchase_timestamp) as months,
      EXTRACT(YEAR FROM order_purchase_timestamp) as Year,
from `Target_Sales_Data.orders`
group by months,Year
order by months
```



Business Case: Target United States.

Observation:

- No of Orders decline every year from march to April.
- There is continues decline in no of orders from march to June in year 2018

2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed? (from above code)

Observation- No of orders increase in jan in new-year

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

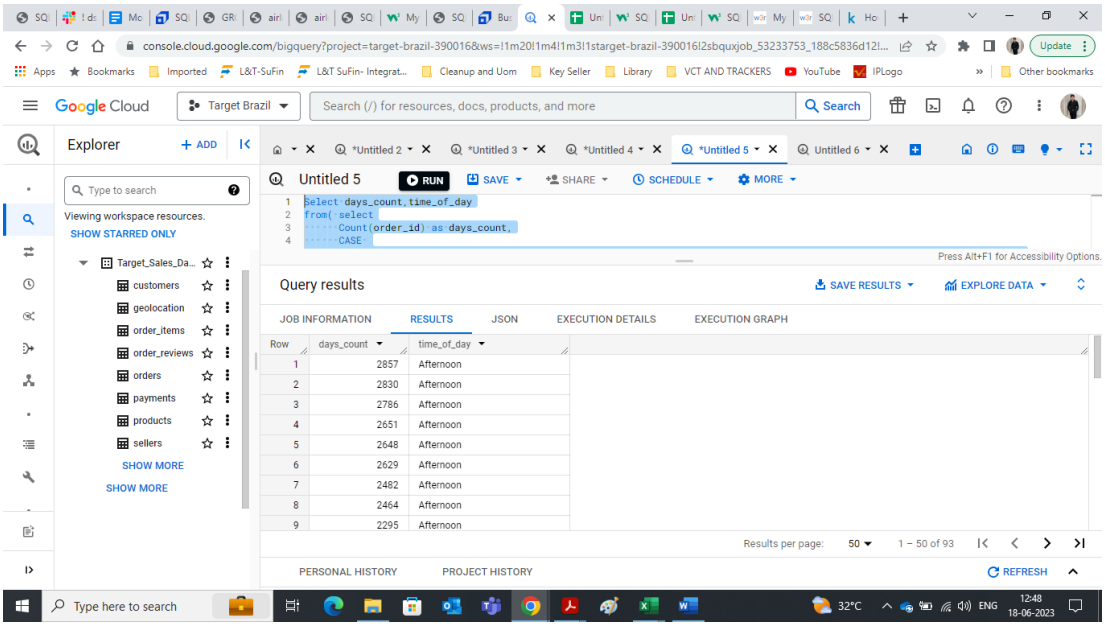
0-6 hrs : Dawn

7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night

```
Select days_count,time_of_day
from( select
      Count(order_id) as days_count,
      CASE
      WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 0 AND EXTRACT(HOUR FROM
order_purchase_timestamp) <= 6 THEN 'Dawn'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 7 AND EXTRACT(HOUR FROM
order_purchase_timestamp) <= 12 THEN 'Morning'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 13 AND EXTRACT(HOUR
FROM order_purchase_timestamp) <= 18 THEN 'Afternoon'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 19 AND EXTRACT(HOUR
FROM order_purchase_timestamp) <= 23 THEN 'Night'
    END AS time_of_day,
      extract(MONTH FROM order_purchase_timestamp) as months,
      EXTRACT(YEAR FROM order_purchase_timestamp) as Year,
      from `Target_Sales_Data.orders`
      GROUP BY time_of_day, months, Year ) AS A
order by A.days_count desc
```



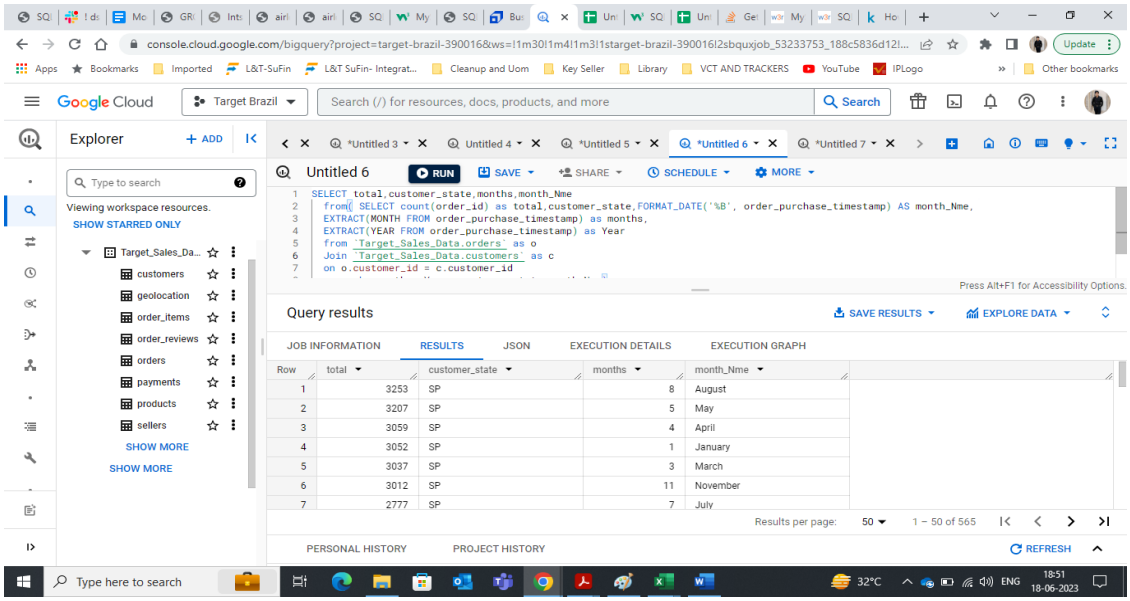
Observation – most orders are placed in after noon

Business Case: Target United States.

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

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

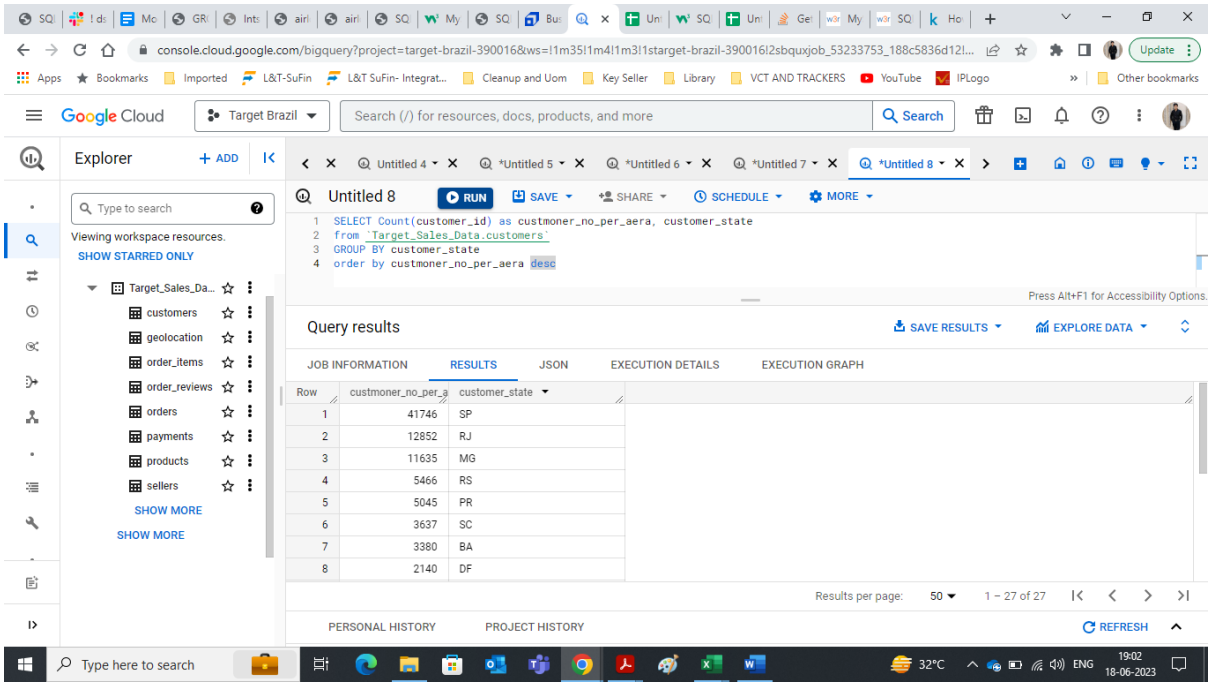
```
SELECT total,customer_state,months,month_Nme
  from( SELECT count(order_id) as total,customer_state,FORMAT_DATE('%B',
  order_purchase_timestamp) AS month_Nme,
  EXTRACT(MONTH FROM order_purchase_timestamp) as months,
  EXTRACT(YEAR FROM order_purchase_timestamp) as Year
  from `Target_Sales_Data.orders` as o
  Join `Target_Sales_Data.customers` as c
  on o.customer_id = c.customer_id
  group by months, Year, customer_state,month_Nme) as a
order by total desc,customer_state asc
```



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

```
SELECT Count(customer_id) as custmoner_no_per_aera, customer_state
from `Target_Sales_Data.customers`
GROUP BY customer_state
order by custmoner_no_per_aera desc
```

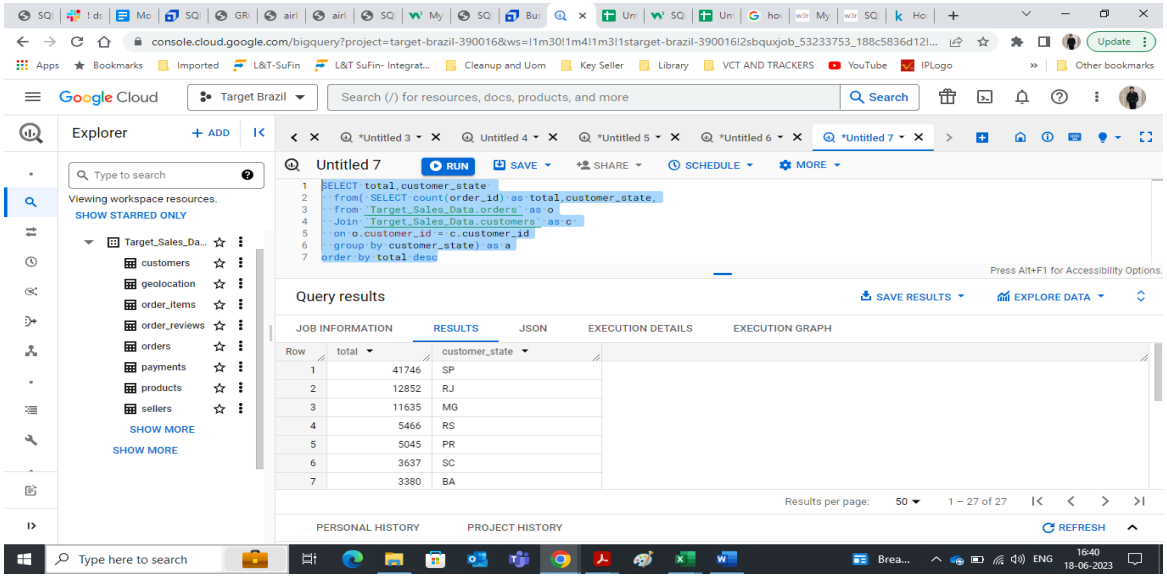
Business Case: Target United States.



Additional - If we want Distribution of Order across states –

```
SELECT total,customer_state
  from( SELECT count(order_id) as total,customer_state,
  from `Target_Sales_Data.orders` as o
  Join `Target_Sales_Data.customers` as c
  on o.customer_id = c.customer_id
  group by customer_state) as a
order by total desc
```

Observation - Most Customers are from SP

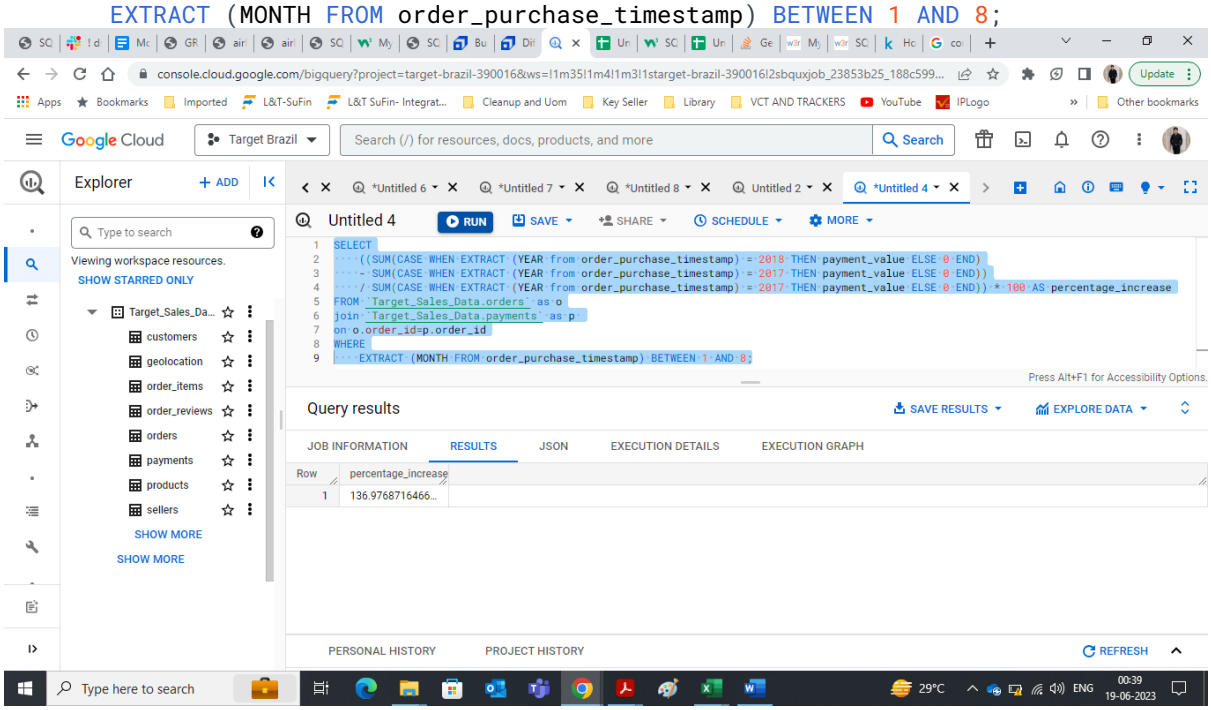


4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others

Business Case: Target United States.

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

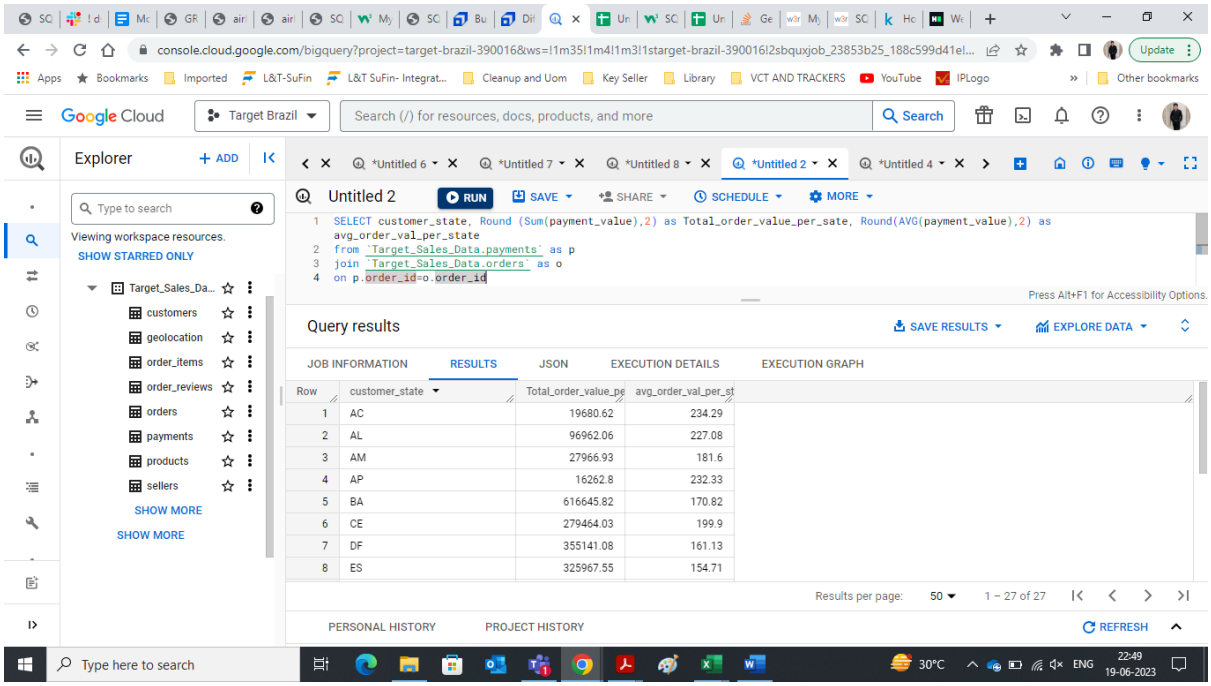
```
SELECT
    ((SUM(CASE WHEN EXTRACT (YEAR from order_purchase_timestamp) = 2018 THEN
payment_value ELSE 0 END)
    - SUM(CASE WHEN EXTRACT (YEAR from order_purchase_timestamp) = 2017 THEN
payment_value ELSE 0 END))
    / SUM(CASE WHEN EXTRACT (YEAR from order_purchase_timestamp) = 2017 THEN
payment_value ELSE 0 END)) * 100 AS percentage_increase
FROM `Target_Sales_Data.orders` as o
join `Target_Sales_Data.payments` as p
on o.order_id=p.order_id
WHERE
    EXTRACT (MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8;
```



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

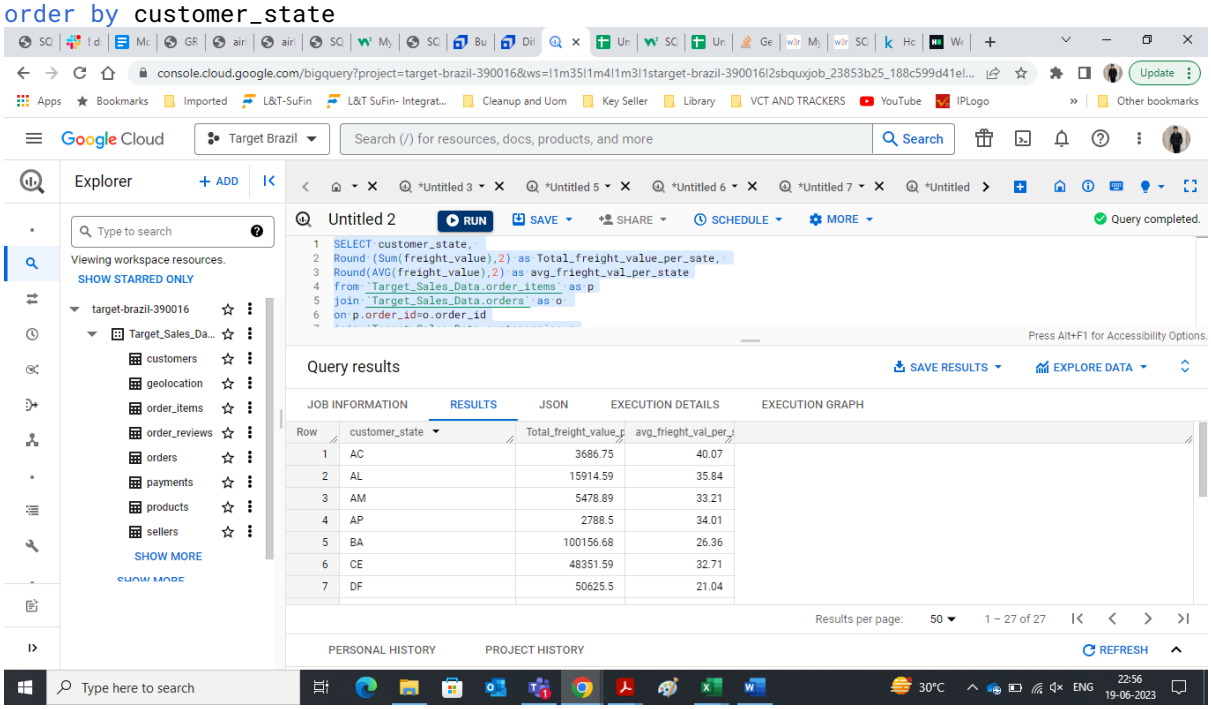
```
SELECT customer_state, Round (Sum(payment_value),2) as Total_order_value_per_sate,
Round(AVG(payment_value),2) as avg_order_val_per_state
from `Target_Sales_Data.payments` as p
join `Target_Sales_Data.orders` as o
on p.order_id=o.order_id
join `Target_Sales_Data.customers`as c
on o.customer_id=c.customer_id
group by customer_state
order by customer_state
```

Business Case: Target United States.



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

```
SELECT customer_state,
Round (Sum(freight_value),2) as Total_freight_value_per_sate,
Round(AVG(freight_value),2) as avg_frieght_val_per_state
from `Target_Sales_Data.order_items` as p
join `Target_Sales_Data.orders` as o
on p.order_id=o.order_id
join `Target_Sales_Data.customers`as c
on o.customer_id=c.customer_id
group by customer_state
order by customer_state
```



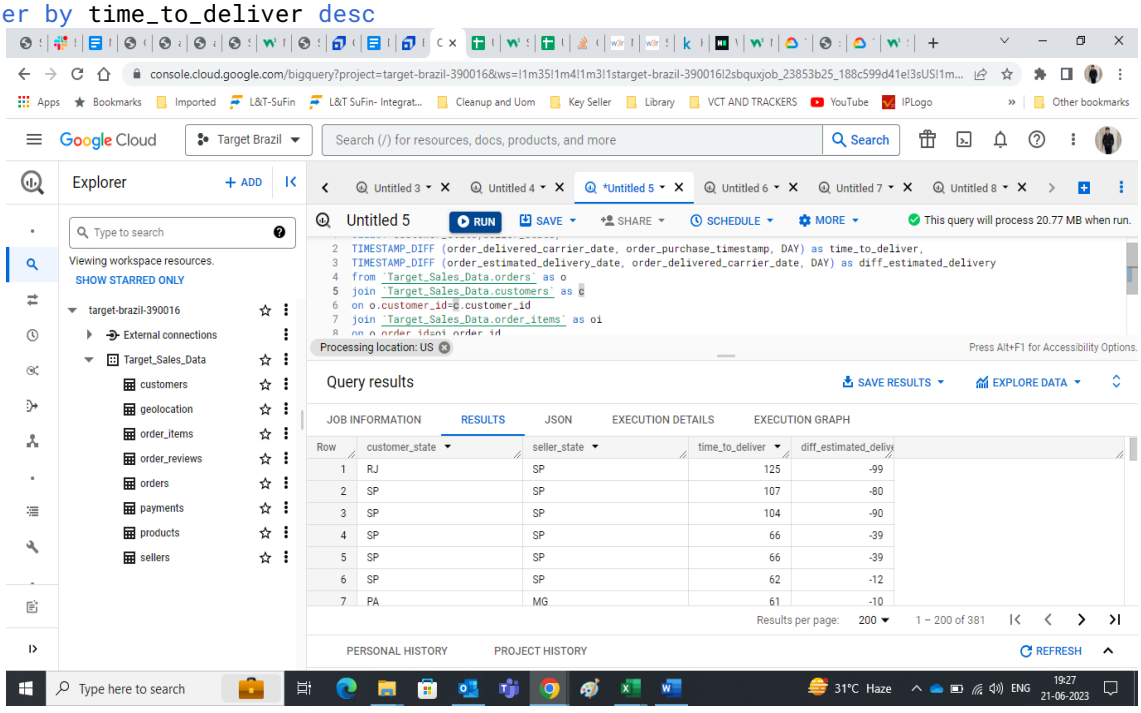
Business Case: Target United States.

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

5.1. 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.

Below is gives actual delivery time and difference in estimated & actual delivery date only where order were delayed with sate of customer and seller.

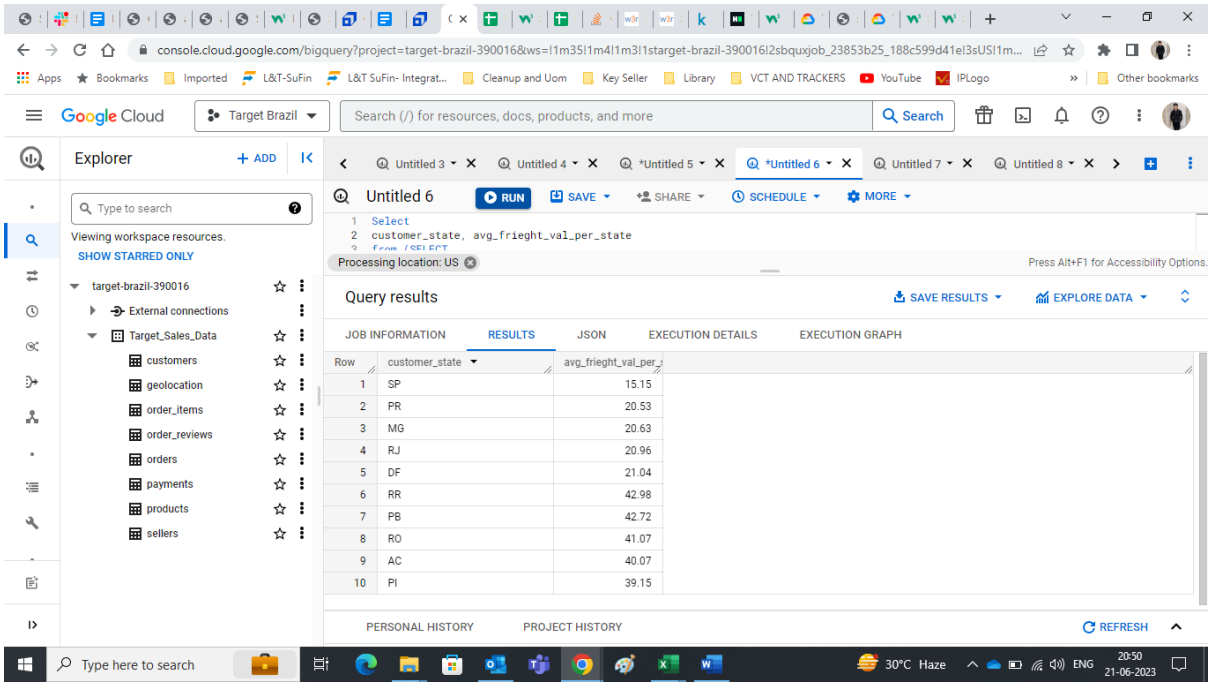
```
SELECT customer_state,seller_state,
TIMESTAMP_DIFF (order_delivered_carrier_date, order_purchase_timestamp, DAY) as
time_to_deliver,
TIMESTAMP_DIFF (order_estimated_delivery_date, order_delivered_carrier_date,
DAY) as diff_estimated_delivery
from `Target_Sales_Data.orders` as o
join `Target_Sales_Data.customers` as c
on o.customer_id=c.customer_id
join `Target_Sales_Data.order_items` as oi
on o.order_id=oi.order_id
join `Target_Sales_Data.sellers`as s
on oi.seller_id=s.seller_id
where LOWER(order_status) = "delivered" and
TIMESTAMP_DIFF (order_estimated_delivery_date, order_delivered_carrier_date,
DAY) < 0
Order by time_to_deliver desc
```



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

```
Select
customer_state, avg_frieght_val_per_state
from (SELECT
  customer_state,
  Round(AVG(freight_value),2) as avg_frieght_val_per_state
  from `Target_Sales_Data.order_items` as p
  join `Target_Sales_Data.orders` as o
  on p.order_id=o.order_id
```

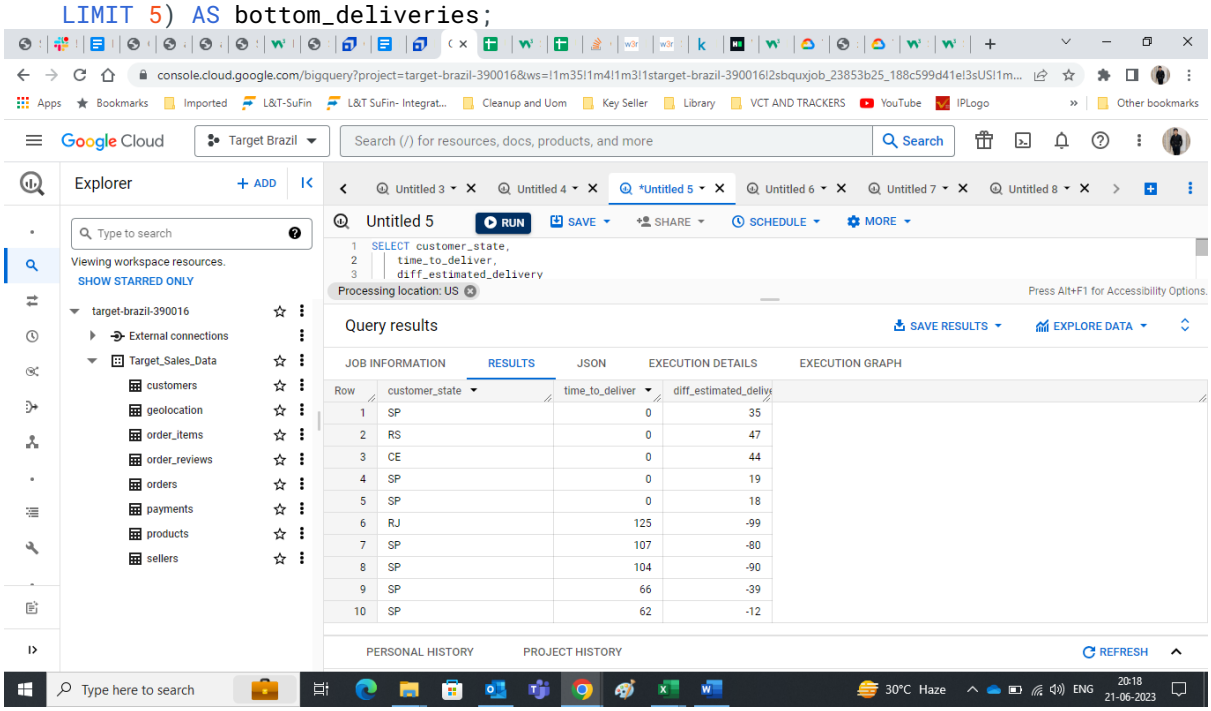
```
  join `Target_Sales_Data.customers`as c
  on o.customer_id=c.customer_id
  group by customer_state
  order by avg_frieght_val_per_state
  LIMIT 5) as lower_freight
  UNION ALL
  Select
customer_state, avg_frieght_val_per_state
from (SELECT
  customer_state,
  Round(AVG(freight_value),2) as avg_frieght_val_per_state
  from `Target_Sales_Data.order_items` as p
  join `Target_Sales_Data.orders` as o
  on p.order_id=o.order_id
  join `Target_Sales_Data.customers`as c
  on o.customer_id=c.customer_id
  group by customer_state
  order by avg_frieght_val_per_state DESC
  LIMIT 5) as higest_freight
```



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

```
SELECT customer_state,
    time_to_deliver,
    diff_estimated_delivery
FROM
    (SELECT customer_state,
        TIMESTAMP_DIFF(order_delivered_carrier_date, order_purchase_timestamp,
DAY) AS time_to_deliver,
        TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_carrier_date, DAY) AS diff_estimated_delivery
    FROM `Target_Sales_Data.orders` AS o
    JOIN `Target_Sales_Data.customers` AS c
    ON o.customer_id = c.customer_id
```

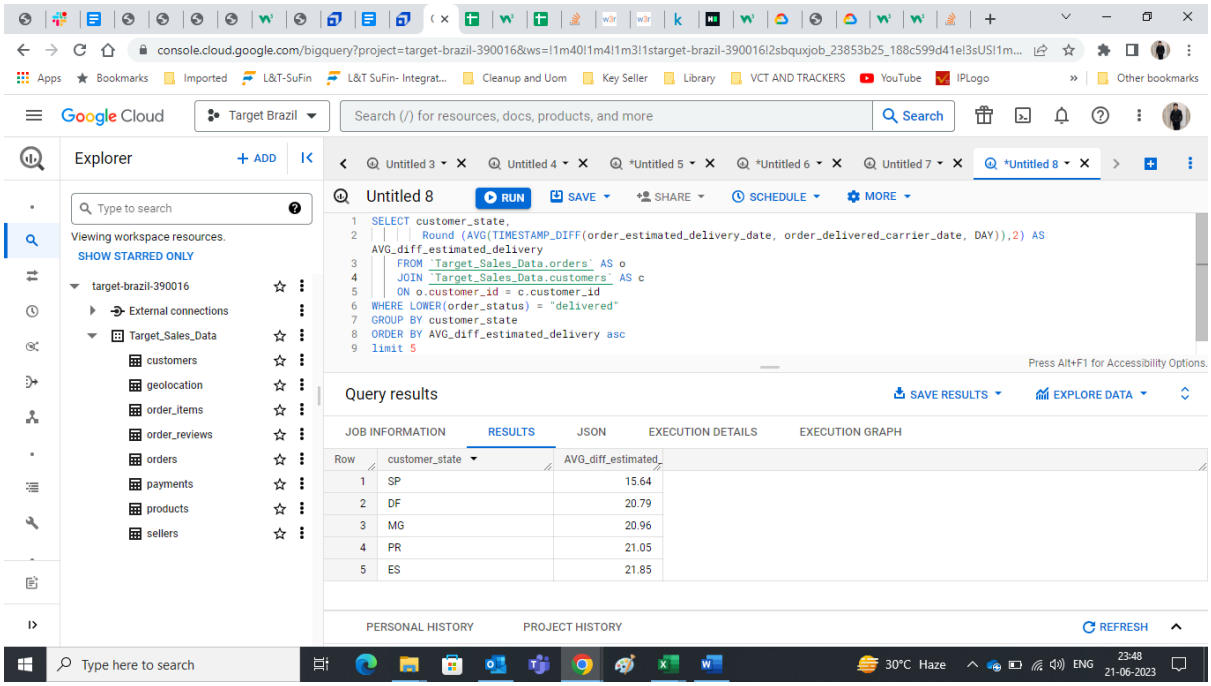
```
    WHERE LOWER(order_status) = "delivered" AND
TIMESTAMP_DIFF(order_delivered_carrier_date, order_purchase_timestamp, DAY) = 0
    ORDER BY time_to_deliver
    LIMIT 5) AS top_deliveries
UNION ALL
SELECT customer_state,
    time_to_deliver,
    diff_estimated_delivery
FROM
    (SELECT customer_state,
        TIMESTAMP_DIFF(order_delivered_carrier_date, order_purchase_timestamp,
DAY) AS time_to_deliver,
        TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_carrier_date, DAY) AS diff_estimated_delivery
    FROM `Target_Sales_Data.orders` AS o
    JOIN `Target_Sales_Data.customers` AS c
    ON o.customer_id = c.customer_id
    WHERE LOWER(order_status) = "delivered"
    ORDER BY time_to_deliver DESC
    LIMIT 5) AS bottom_deliveries;
```



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

```
SELECT customer_state,
        Round (AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_carrier_date, DAY)),2) AS AVG_diff_estimated_delivery
    FROM `Target_Sales_Data.orders` AS o
    JOIN `Target_Sales_Data.customers` AS c
    ON o.customer_id = c.customer_id
WHERE LOWER(order_status) = "delivered"
GROUP BY customer_state
ORDER BY AVG_diff_estimated_delivery asc
limit 5
```

Business Case: Target United States.

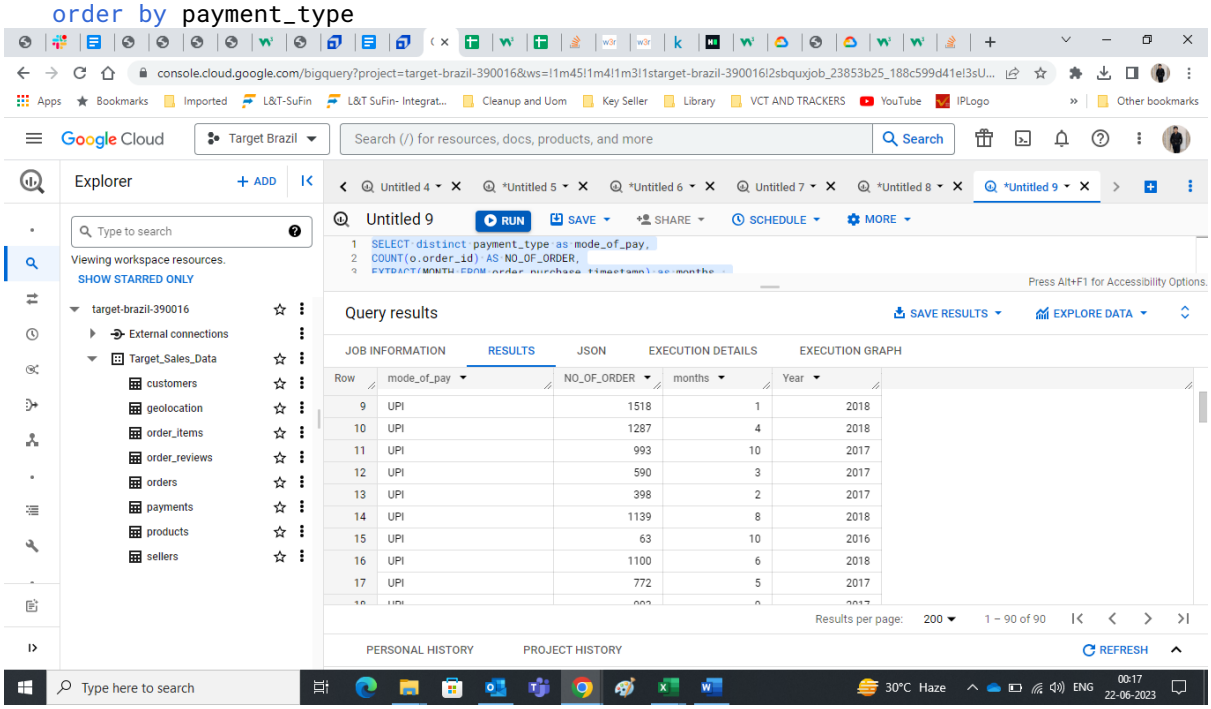


Observation: SP has fastest delivery

6. Analysis based on the payments:

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

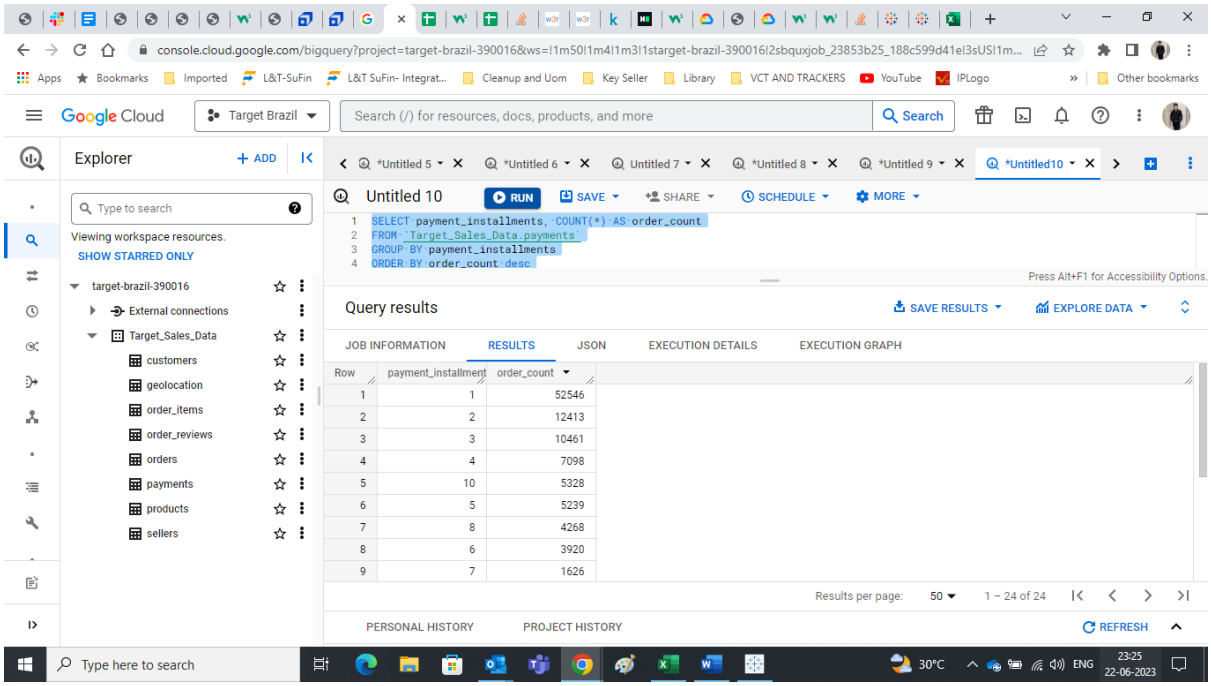
```
SELECT distinct payment_type as mode_of_pay,
    COUNT(o.order_id) AS NO_OF_ORDER,
    EXTRACT(MONTH FROM order_purchase_timestamp) as months,
    EXTRACT(YEAR FROM order_purchase_timestamp) as Year
from `Target_Sales_Data.orders` as o
join `Target_Sales_Data.payments` as p
on o.order_id=p.order_id
GROUP BY months,Year,payment_type
order by payment_type
```



Business Case: Target United States.

6.2. Find the no. of orders placed on the basis of the payment instalments that have been paid

```
SELECT payment_installments, COUNT(*) AS order_count
FROM `Target_Sales_Data.payments`
GROUP BY payment_installments
ORDER BY order_count desc
```



Observation – More than 9 instalment is least preferred.