Project Introduction:

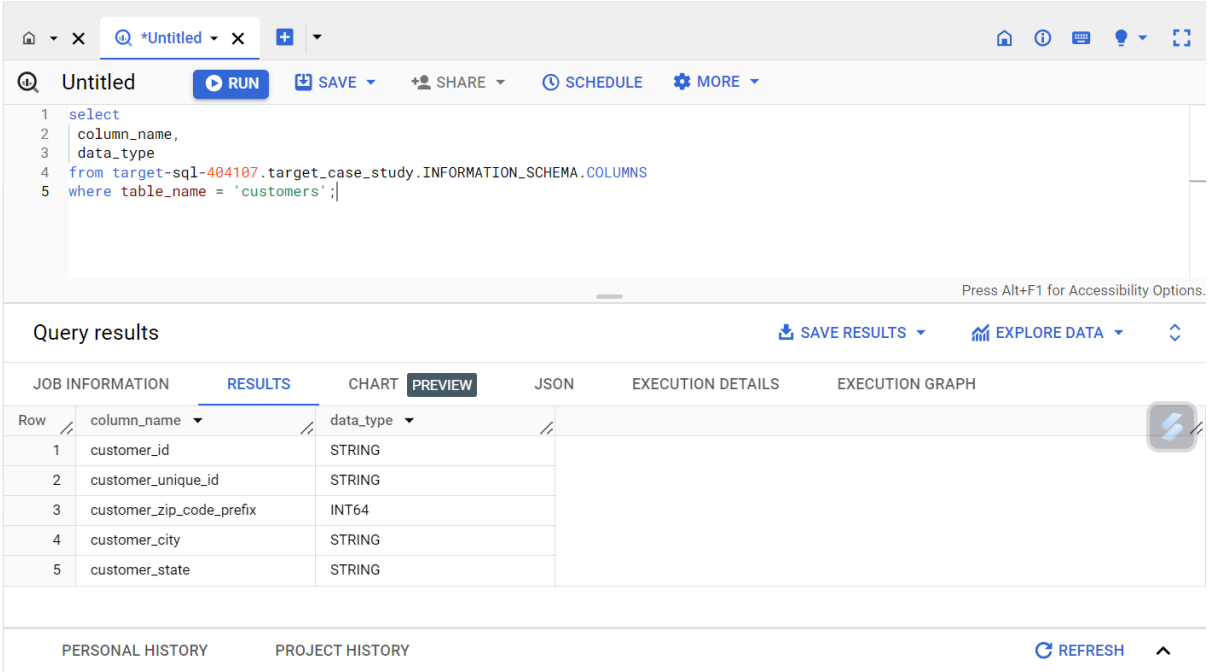
Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This project analyzes Target's operations in Brazil, analyzing 100,000 orders placed between 2016 and 2018. The dataset provides a comprehensive view of various dimensions, including order status, pricing, payment and freight performance, customer locations, product attributes, and customer reviews. Through this analysis, valuable insights can be gained into Target's business operations in Brazil, including order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels. These insights will help understand and improve Target's performance in the Brazilian market.

Q. 1 a)

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

```
select
 column_name,
 data_type
from target-sql-404107.target_case_study.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers';
```



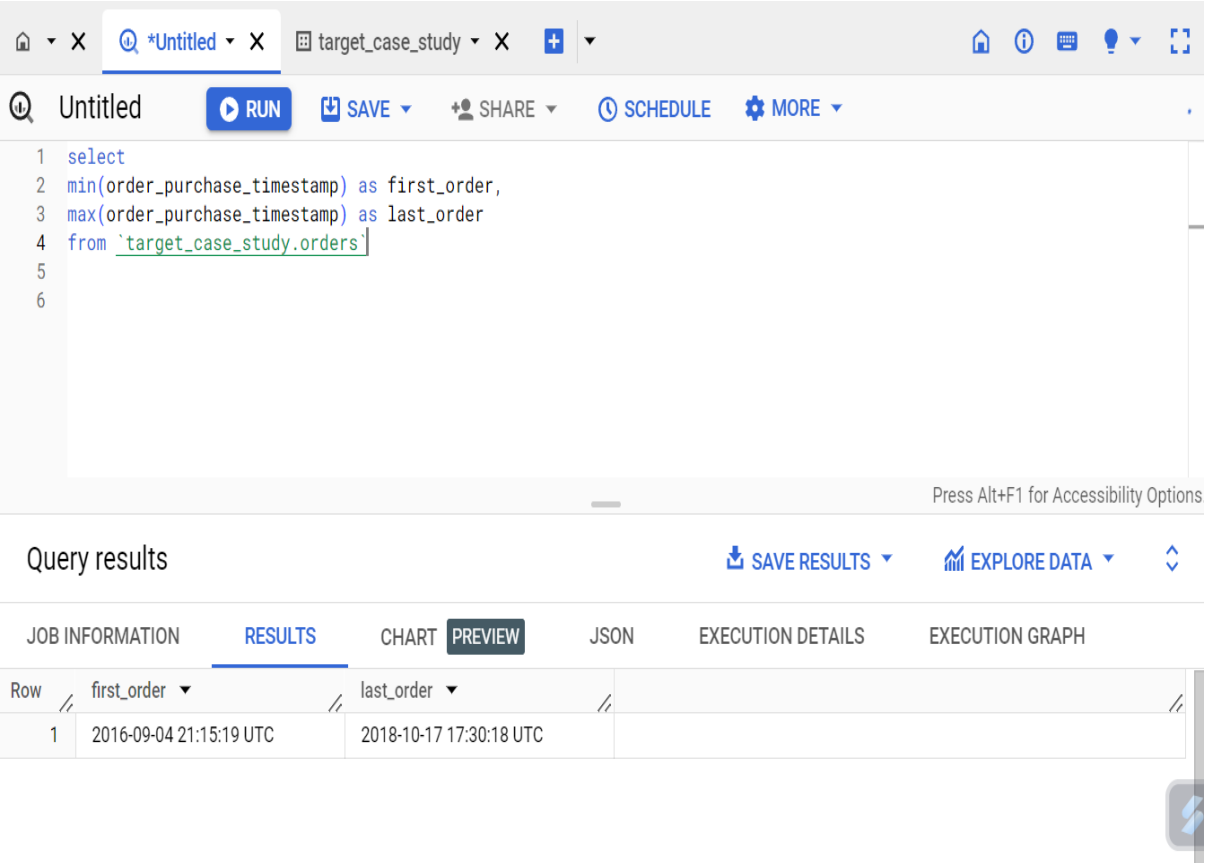
- Insights

  All columns have same data type except customer_zip_code_prefix.

Q.1

b) Get the time range between which the orders were placed

```
select
   min(order_purchase_timestamp) as first_order,
   max(order_purchase_timestamp) as last_order
   from `target_case_study.orders`
```



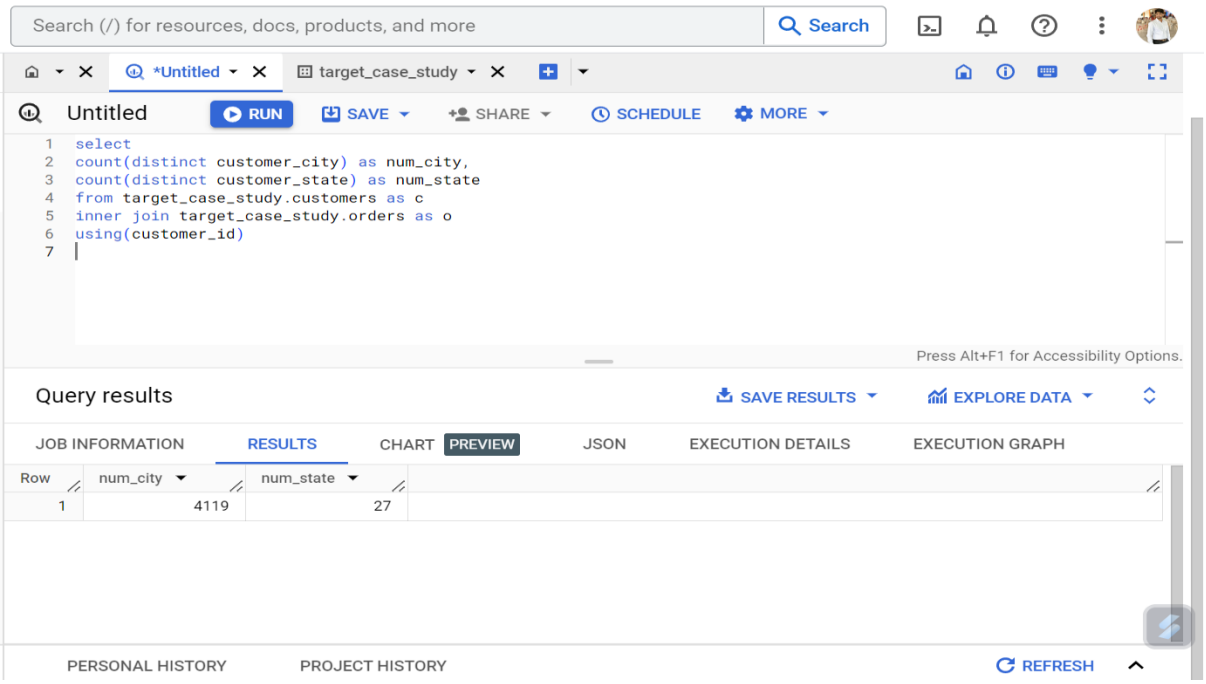
❖ **Insights**

Difference between first_order and last_order is almost 2 Years.

Q.1

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

```
select
count(distinct customer_city) as num_city,
count(distinct customer_state) as num_state
from target_case_study.customers as c
inner join target_case_study.orders as o
using(customer_id);
```



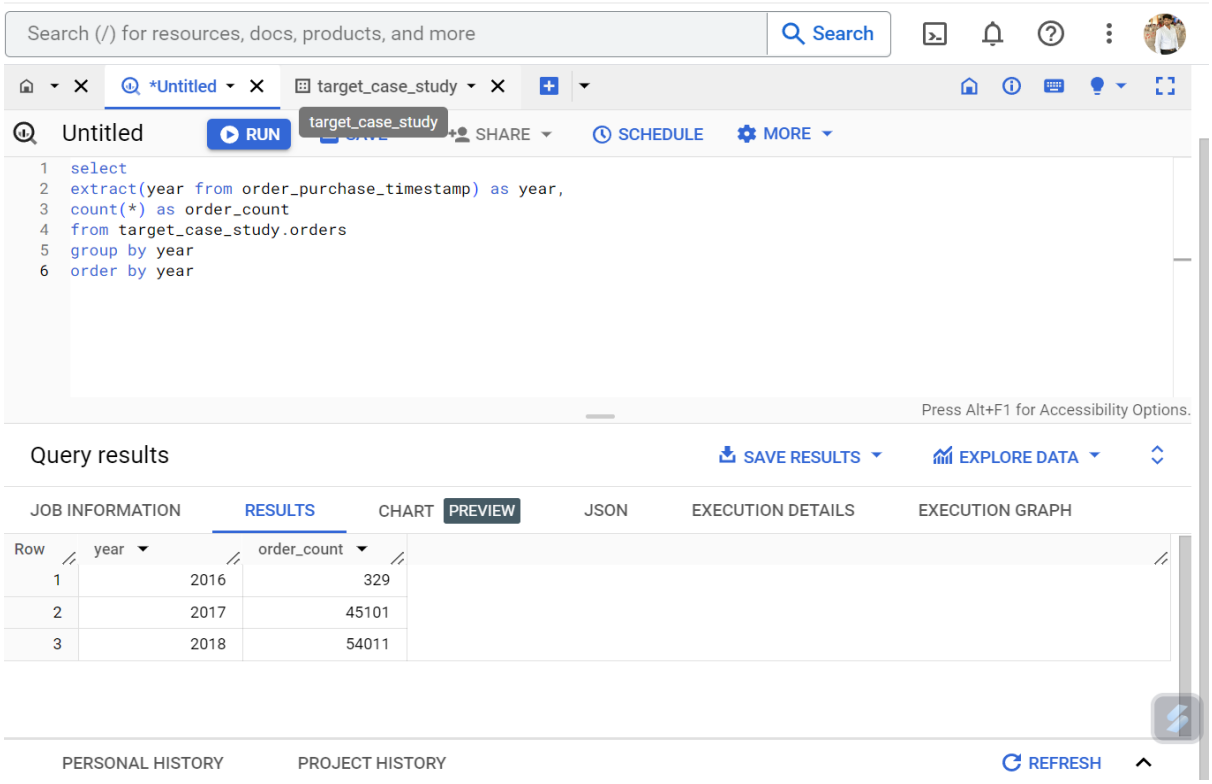
❖ Insights

Orders were received from all states and cities with few exceptions within the given time

Q.2 **In-depth Exploration**

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

```
select
extract(year from order_purchase_timestamp) as year,
count(*) as order_count
from target_case_study.orders
group by year
order by year
```



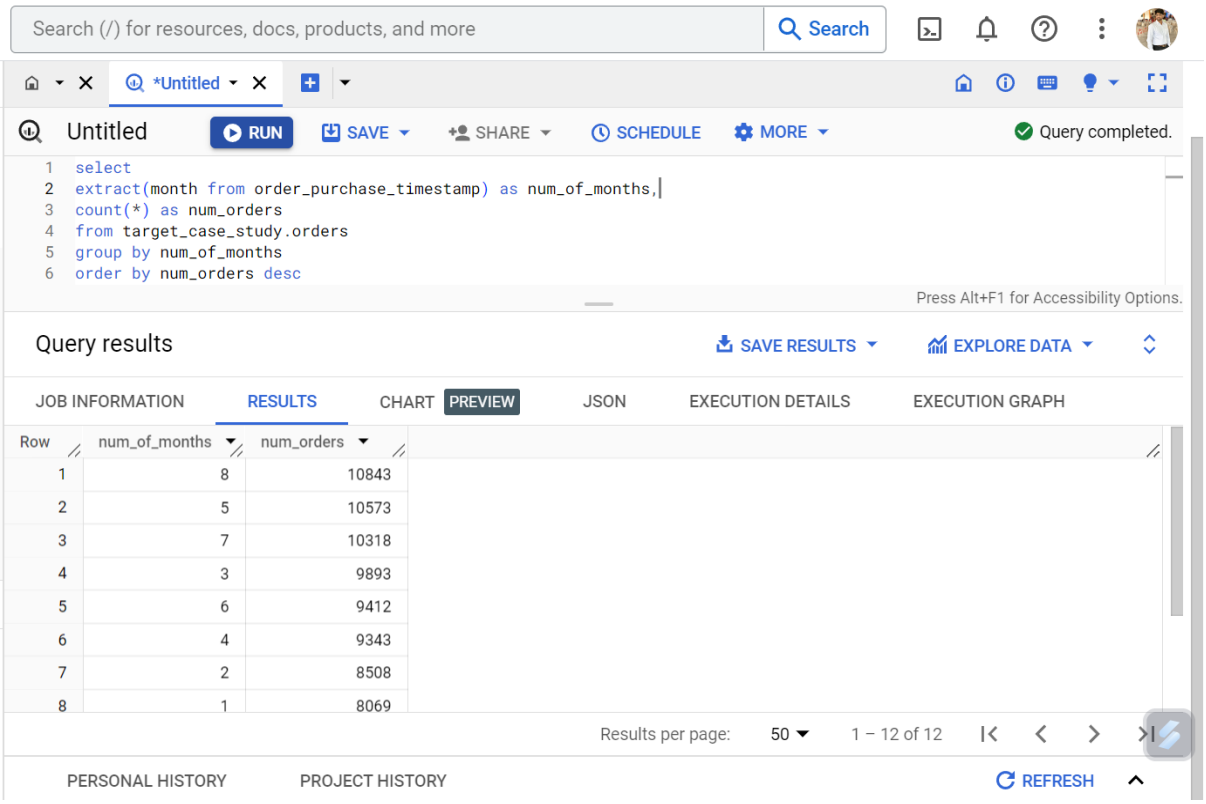
❖ Insights

YES, We can see that orders have been increased over the years

Q. 2

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 num_of_months,
count(*) as num_orders
from target_case_study.orders
group by num_of_months
order by num_orders desc
```



❖ Insights

Yes,we can see that the num_of_months decreases the num_of_orders also decreases.

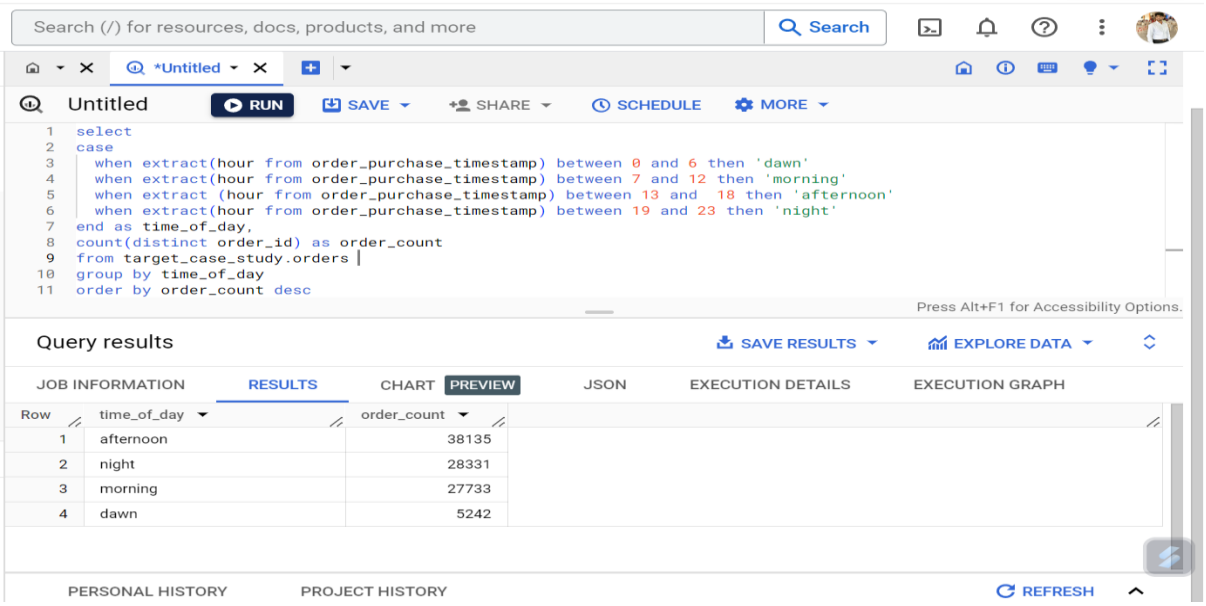
Q.2

c) During what time of the day, do the Brazilian customers mostly place the 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 'morning'
  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 time_of_day,
count(distinct order_id) as order_count
from target_case_study.orders
group by time_of_day
order by order_count desc
```



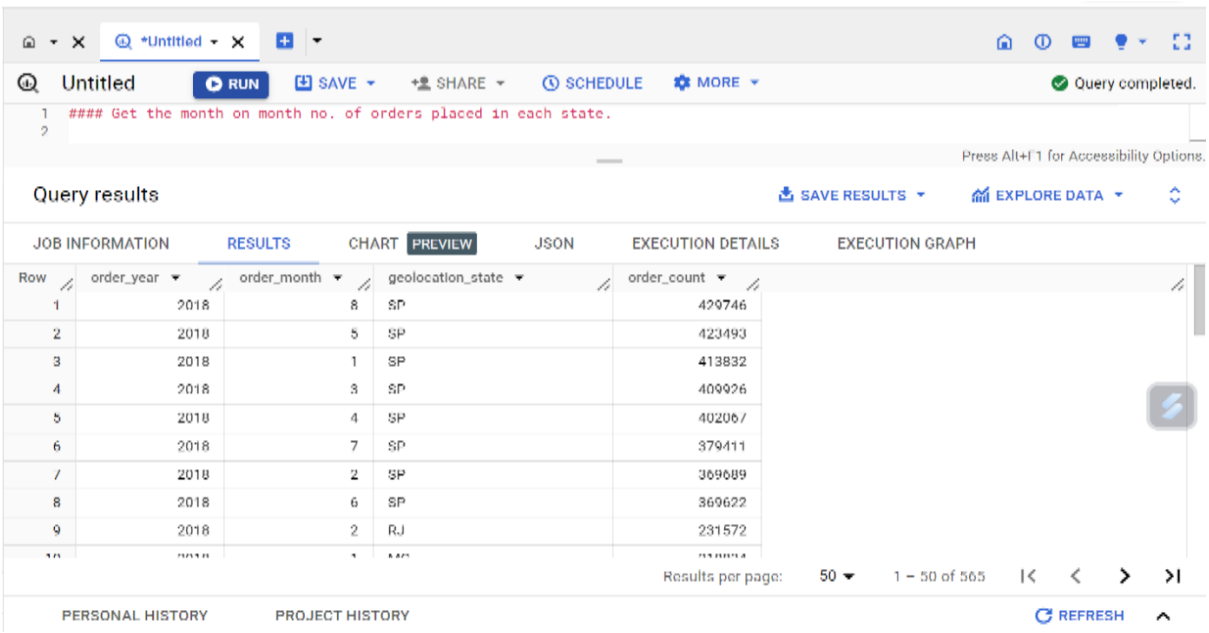
❖ Insights

At afternoon Brazilian peoples places highest amout orders followed by night and morning very few orders places at dawn.

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

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

```
SELECT
EXTRACT(YEAR FROM  o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
g.geolocation_state,
COUNT(o.order_id) AS order_count
FROM target_case_study.orders as o
inner join target_case_study.customers as c  on o.customer_id =
c.customer_id
inner join target_case_study.geolocation as g on
c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
GROUP BY
    order_year,
    order_month,
    geolocation_state
order by
    order_count desc,
    order_year desc
```



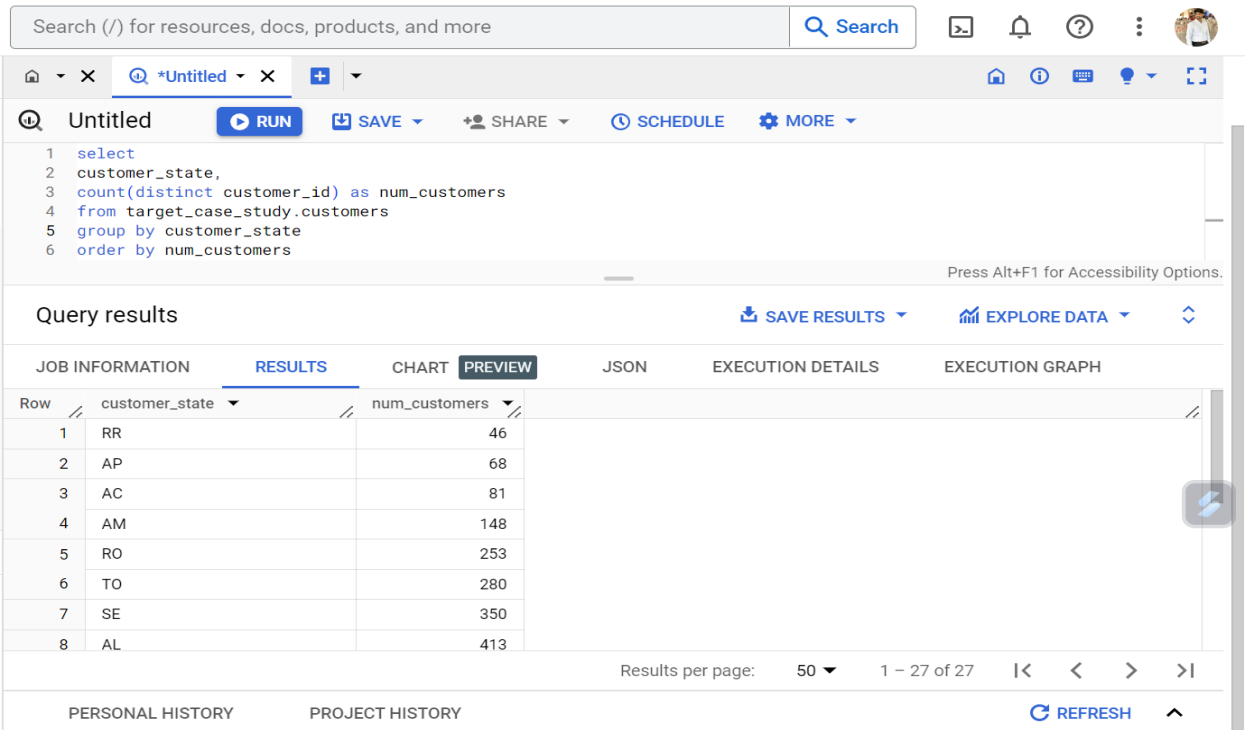
Insights

In 2018 SP is 8 times out of top 10 by order_count .

Q.3

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

```
select
customer_state,
count(distinct customer_id) as num_customers
from target_case_study.customers
group by customer_state
order by num_customers
```



❖ Insights

The distribution of the customers is not uniform, there is major variation present between state wise data.

Q.4 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 cte_2017 as

(select extract(year FROM o.order_purchase_timestamp) as year,

sum(payment_value) as pur_1

 from target_case_study.orders o INNER JOIN target_case_study.payments p
USING(order_id)
 where extract(year FROM o.order_purchase_timestamp) = 2017
 AND extract(month FROM o.order_purchase_timestamp) BETWEEN 1 and 8
 group by year),
cte_2018 as
(select
 extract(year FROM o.order_purchase_timestamp) as year,
 sum(payment_value) as pur_2
 from target_case_study.orders o INNER JOIN target_case_study.payments p
USING(order_id)
 where extract(year FROM o.order_purchase_timestamp) = 2018
 AND extract(month FROM o.order_purchase_timestamp) BETWEEN 1 and 8
 group by year)
select
 ROUND((pur_2 - pur_1) / pur_1 * 100,2) as percent_increase
from cte_2017,cte_2018
```

```
extract(year FROM o.order_purchase_timestamp) as year,
sum(payment_value) as pur_2
from target_case_study.orders o INNER JOIN target_case_study.payments p USING(order_id)
```

Press Alt+F1 for Accessibility Opti

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	percent_increase
1	136.98

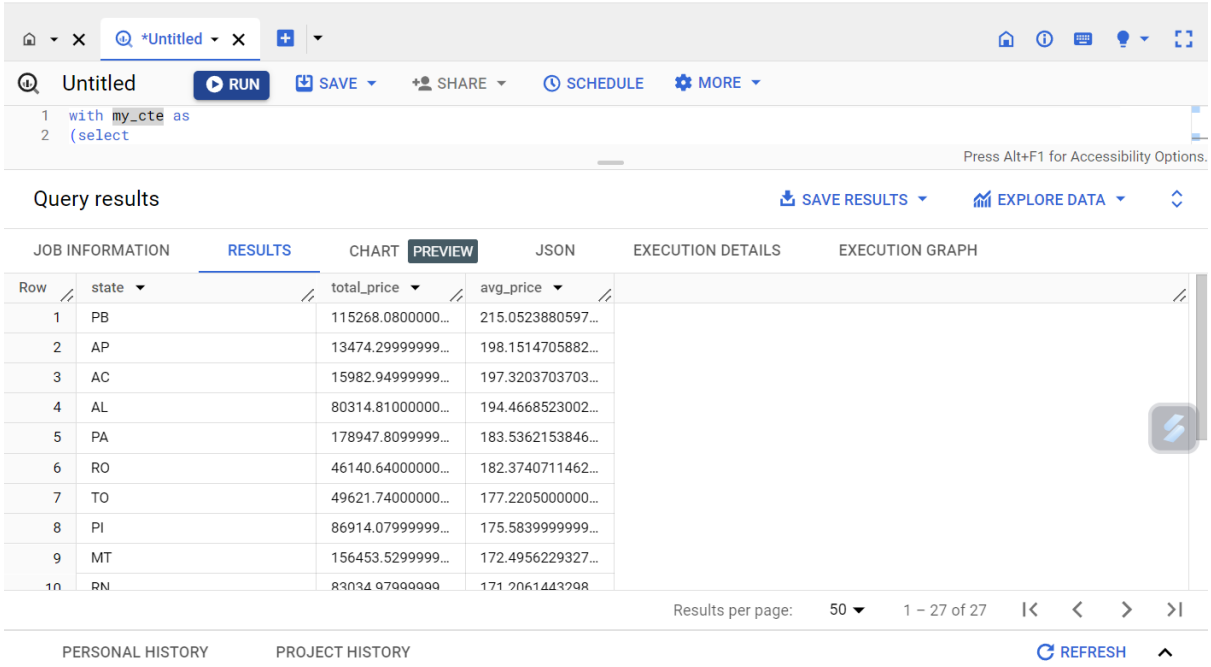
❖ Insights

The % increase in the the cost of orders from year 2017 to 2018 include Months between Jan to Aug) is **136.98%**

Q.4

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

```
with my_cte as
(select
 c.customer_state as state,
 SUM(price) as total_price,
 COUNT(distinct o.order_id) as num_orders
 from target_case_study.customers c LEFT JOIN target_case_study.orders o
USING(customer_id)
 LEFT JOIN target_case_study.order_items oi USING(order_id)
 group by state)
select
 state, my_cte.total_price,
 (my_cte.total_price/num_orders) as avg_price
from my_cte
order by avg_price desc
```



Row	state	total_price	avg_price
1	PB	115268.0800000...	215.0523880597...
2	AP	13474.29999999...	198.1514705882...
3	AC	15982.94999999...	197.3203703703...
4	AL	80314.81000000...	194.4668523002...
5	PA	178947.8099999...	183.5362153846...
6	RO	46140.64000000...	182.3740711462...
7	TO	49621.74000000...	177.2205000000...
8	PI	86914.07999999...	175.5839999999...
9	MT	156453.5299999...	172.4956229327...
10	RN	83034.97999999...	171.2061443298...

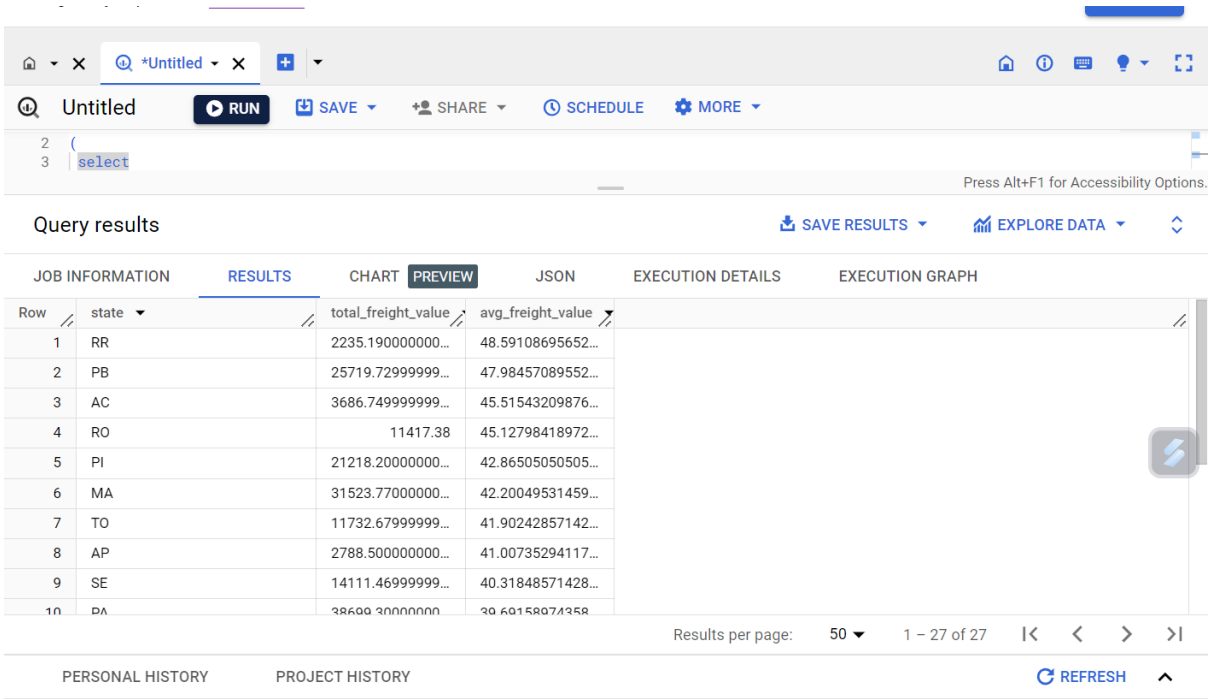
❖ Insights

The state with the highest average cost is the twice the state with the lowest cost. Max avg_price = 215.05 and Min avg_price = 124.63

Q. 4

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

```
with my_cte as (
 select
 c.customer_state as state,
 SUM(freight_value) as total_freight_value,
 COUNT(distinct o.order_id) as num_orders
 from target_case_study.customers c LEFT JOIN
target_case_study.orders o USING(customer_id)
 LEFT JOIN target_case_study.order_items oi USING(order_id)
 group by state)
select state,
 my_cte.total_freight_value,
 (my_cte.total_freight_value/num_orders) as avg_freight_value
from my_cte
order by avg_freight_value desc
```



Row	state	total_freight_value	avg_freight_value
1	RR	2235.190000000...	48.59108695652...
2	PB	25719.72999999...	47.98457089552...
3	AC	3686.749999999...	45.51543209876...
4	RO	11417.38	45.12798418972...
5	PI	21218.20000000...	42.86505050505...
6	MA	31523.77000000...	42.20049531459...
7	TO	11732.67999999...	41.90242857142...
8	AP	2788.500000000...	41.00735294117...
9	SE	14111.46999999...	40.31848571428...
10	PA	38699.30000000...	39.69158974358...

❖ **Insights**

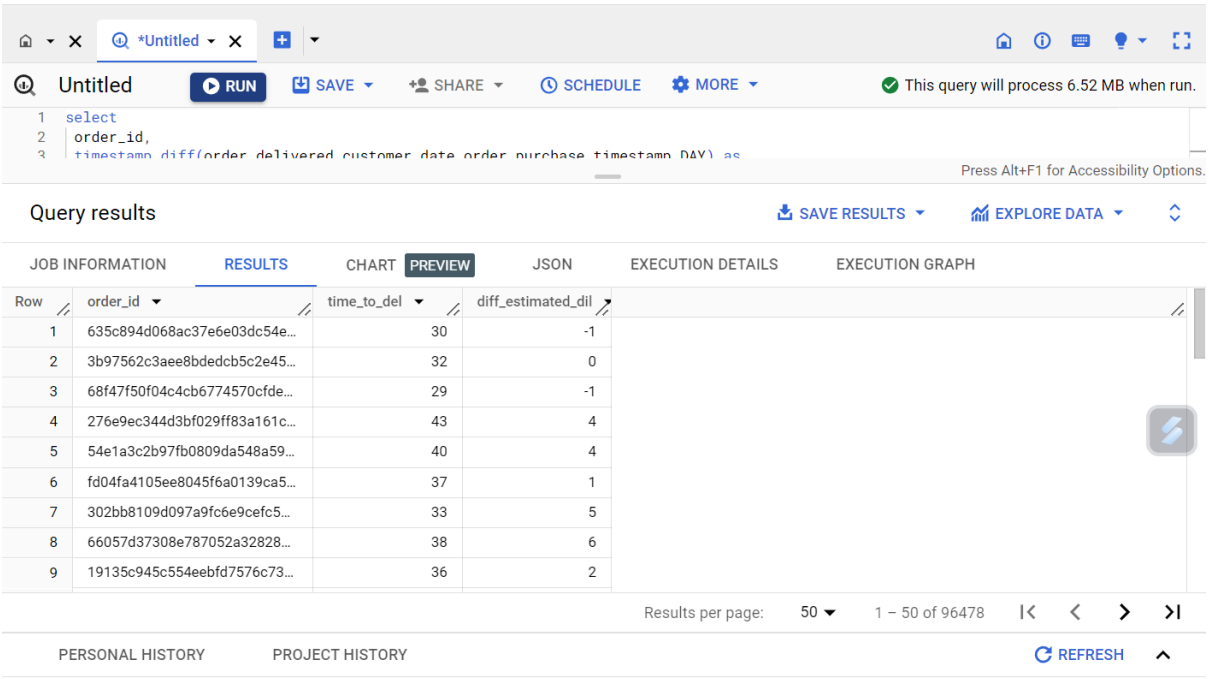
The PR state has heights average fright value.

Q.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:

- **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date
-

```
select order_id,
 timestamp_diff(order_delivered_customer_date,order_purchase_t
imestamp,DAY) as time_to_del,
 timestamp_diff(order_delivered_customer_date,order_estimated_
delivery_date,DAY)
as diff_estimated_dil
from target_case_study.orders where order_status = 'delivered'
```



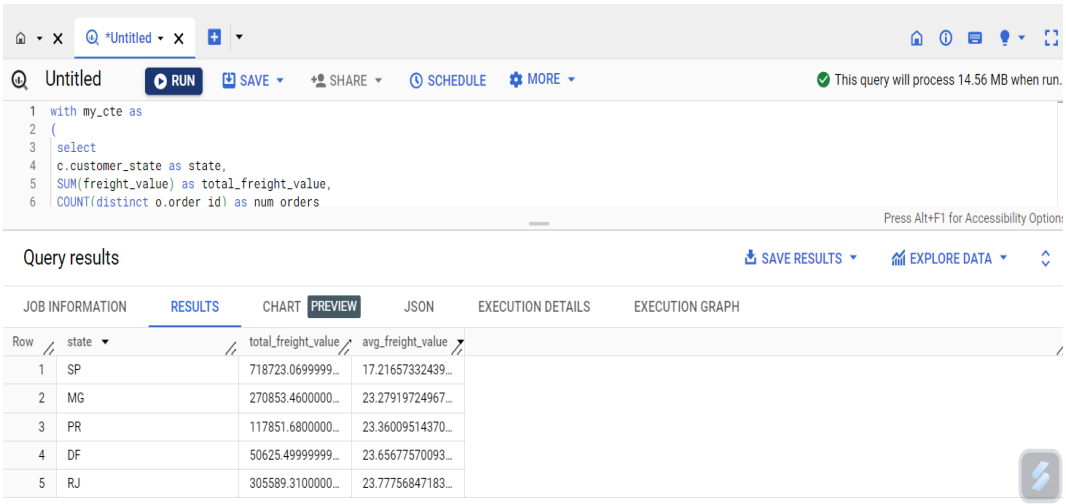
❖ **Insights :**

Delivery is on time except few orders.

Q.5

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

```
with my_cte as (select
 c.customer_state as state,
 SUM(freight_value) as total_freight_value,
 COUNT(distinct o.order_id) as num_orders
 from target_case_study.customers c LEFT JOIN
target_case_study.orders o USING(customer_id)
 LEFT JOIN target_case_study.order_items oi USING(order_id)
 group by state)
select
state,
  my_cte.total_freight_value,
 (my_cte.total_freight_value/num_orders) as avg_freight_value
from my_cte
order by avg_freight_value asc
limit 5;
```



Row	state	total_freight_value	avg_freight_value
1	SP	718723.0699999...	17.21657332439...
2	MG	270853.4600000...	23.27919724967...
3	PR	117851.6800000...	23.36009514370...
4	DF	50625.49999999...	23.65677570093...
5	RJ	305589.3100000...	23.77756847183...

❖ **Insights**

Top 5 state with the highest and lowest average freight value is SP, MG, PR, DF, RJ.

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

```
SELECT
customer_state AS state,
ROUND(SUM(TIMESTAMP_DIFF(order_delivered_customer_date,
order_purchase_timestamp,
DAY))/COUNT(ORDER_ID), 2) AS average_time_for_del,
ROUND(SUM(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_purchase_timestamp,
DAY))/COUNT(ORDER_ID), 2) AS average_est_dil_time,
FROM `target_case_study.orders` o
INNER JOIN `target_case_study.customers` c
ON o.customer_id=c.customer_id
WHERE order_status='delivered'
GROUP BY customer_state
ORDER BY average_time_for_del asc/des      #### asc for lowest and desc for
highest
limit 5
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	state	average_time_for_de	average_est_dil_time
1	RR	28.98	45.63
2	AP	26.73	45.87
3	AM	25.99	44.92
4	AL	24.04	32.21
5	PA	23.32	36.79

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	state	average_time_for_de	average_est_dil_time
1	SP	8.3	18.78
2	PR	11.53	24.25
3	MG	11.54	24.19
4	DF	12.51	23.95
5	SC	14.48	25.41

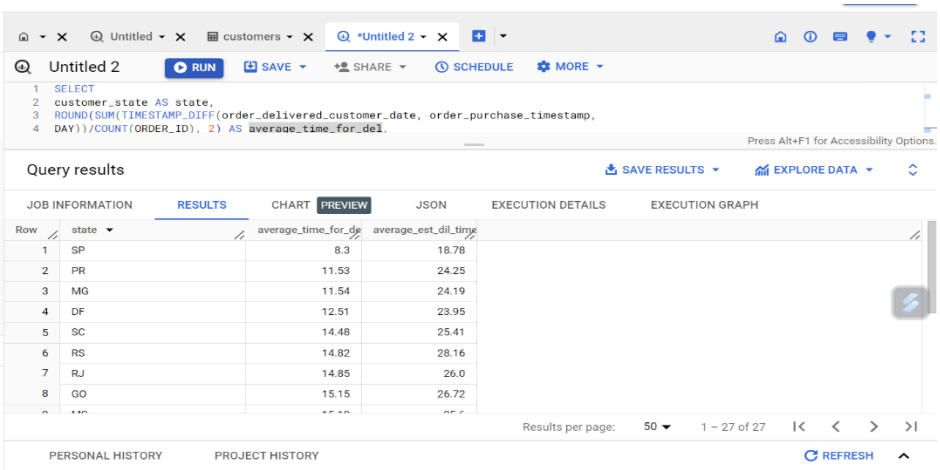
❖ **Insights**

Average time for delivery is half of average estimate delivery time.

Q.5

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

```
SELECT
customer_state AS state,
ROUND(SUM(TIMESTAMP_DIFF(order_delivered_customer_date,
order_purchase_timestamp,
DAY))/COUNT(ORDER_ID), 2) AS average_time_for_del,
ROUND(SUM(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_purchase_timestamp,
DAY))/COUNT(ORDER_ID), 2) AS average_est_dil_time,
FROM `target_case_study.orders` o
INNER JOIN `target_case_study.customers` c
ON o.customer_id=c.customer_id
WHERE order_status='delivered'
GROUP BY customer_state
ORDER BY average_time_for_del
```



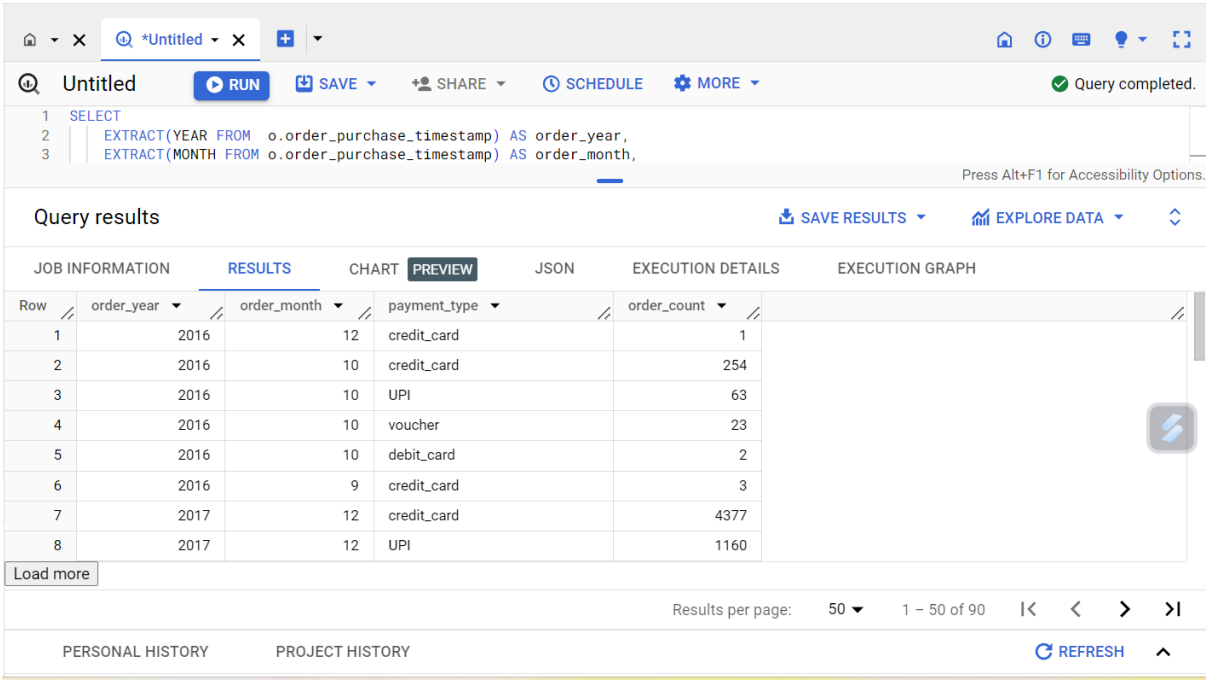
❖ **Insights**

The average time for delivery and average estimate delivery time in SP state is less as compare to other state.

Q.6

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

```
SELECT
EXTRACT(YEAR FROM  o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
p.payment_type,
COUNT(o.order_id) AS order_count
FROM target_case_study.orders as o
inner join target_case_study.payments as p on o.order_id = p.order_id
GROUP BY
    order_year,
    order_month,
    payment_type
ORDER BY
    order_year ,
    order_month desc,
    order_count desc;
```



Row	order_year	order_month	payment_type	order_count
1	2016	12	credit_card	1
2	2016	10	credit_card	254
3	2016	10	UPI	63
4	2016	10	voucher	23
5	2016	10	debit_card	2
6	2016	9	credit_card	3
7	2017	12	credit_card	4377
8	2017	12	UPI	1160

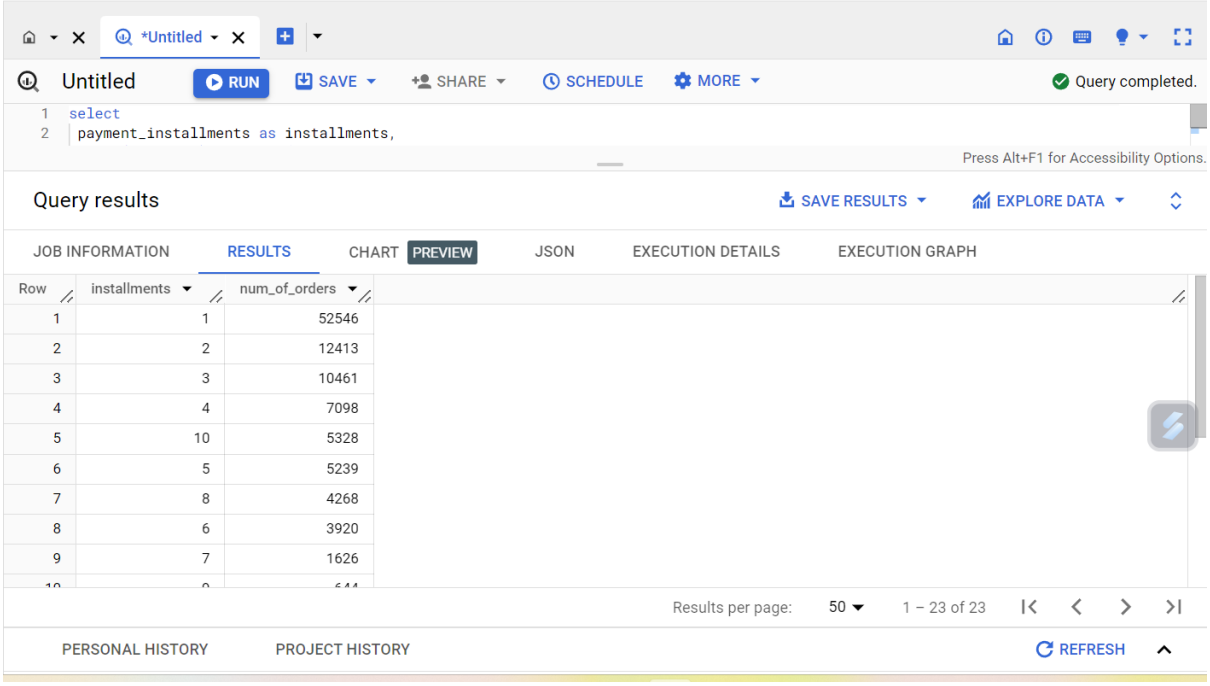
❖ Insights

The most favorite payment type of customers is credit_card .

Q.6

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

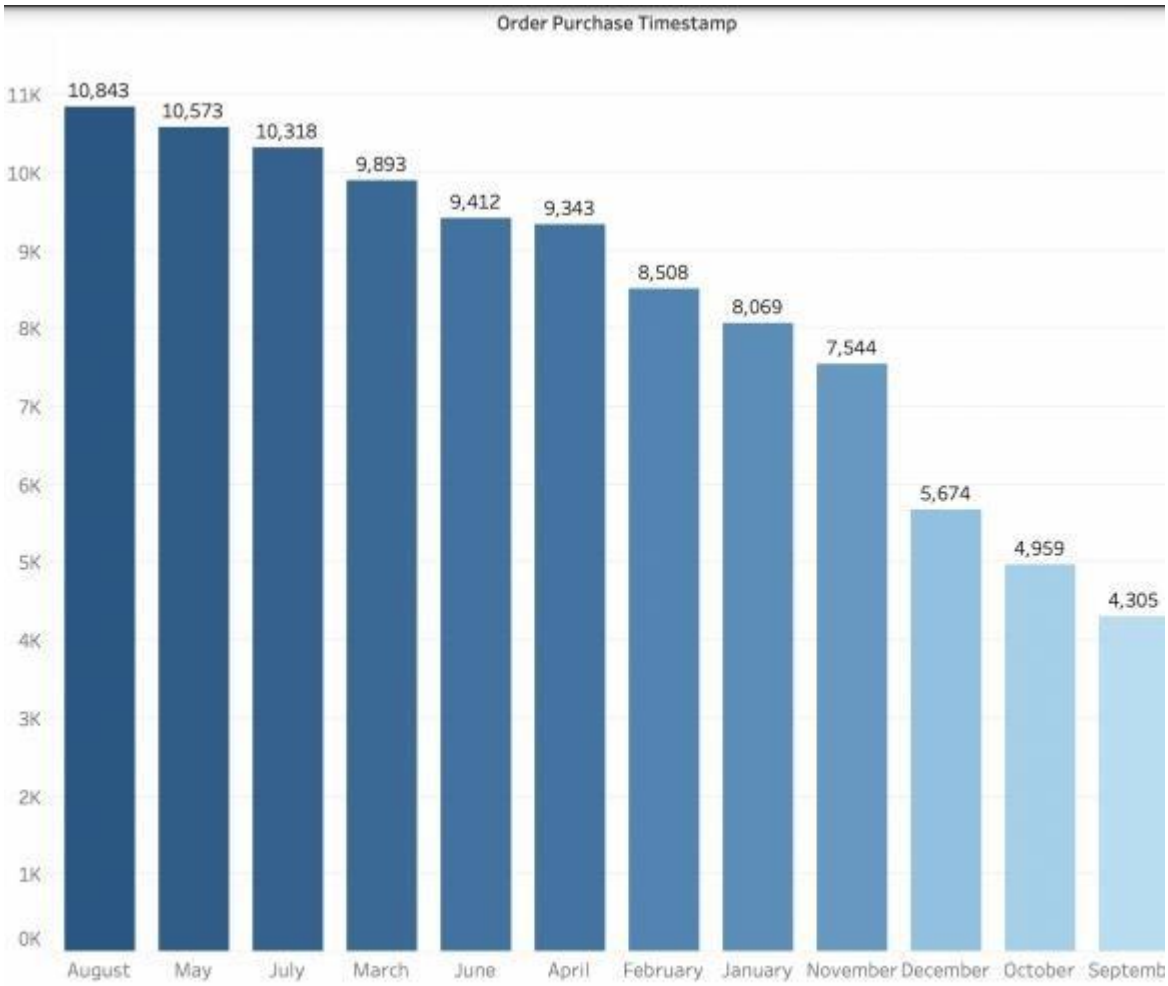
```
select
payment_installments as installments,
count(order_id) as num_of_orders
from target_case_study.payments
where payment_installments >= 1
group by payment_installments
order by num_of_orders desc
```



Row	installments	num_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626

❖ Insights

With few exceptions, installments decrease while num_of_order increase.



Recommendation: The month of August shows the highest sale in the entire year, that shows we can plan more ideas to increase the sales bar. We can provide them some good offers or we can try mid sale and discounts to increase the sale. We can provide some vouchers or gift cards to the customers who come under medium range buyers category. We can Send additional small gifts to boost the customer buying interest like some freebies.