TARGET – BUSINESS CASE STUDY

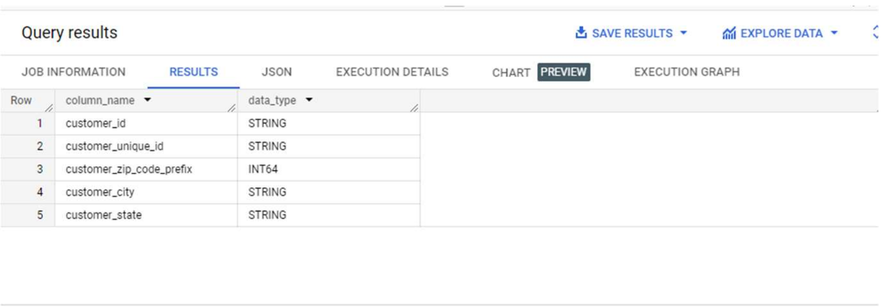
1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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

QUERY:

```
select column_name, data_type
from `target.information_schema.columns`
where table_name = 'customers'
```

OUTPUT:



Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

INSIGHTS:

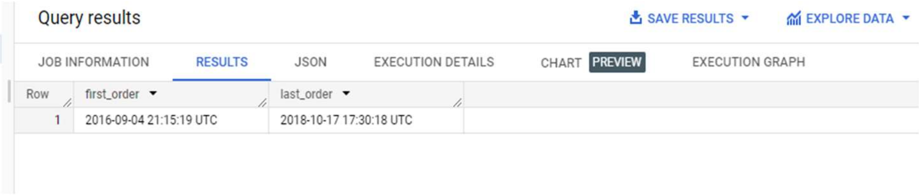
This will help us to understand the table and its data types.

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

QUERY:

```
Select
min(order_purchase_timestamp) as  first_order ,
max(order_purchase_timestamp) as  last_order
from `target.orders`
```

OUTPUT:



Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

INSIGHTS:

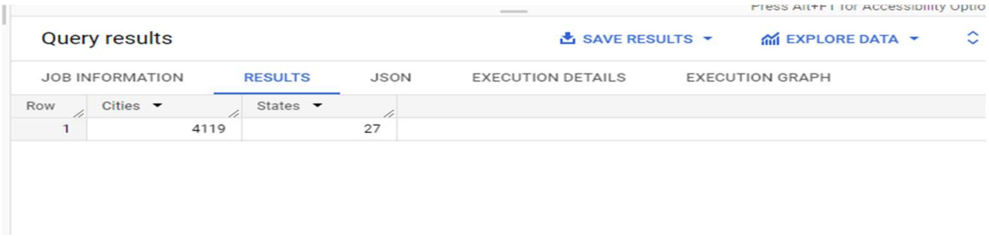
Here we get to know the first and last order made by the customers in the period between 2016 and 2018.

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

QUERY:

```
select count(distinct c.customer_city) as Cities,
 count(distinct c.customer_state) as States
from `target.customers` as c join `target.orders` as o
on c.customer_id = o.customer_id
```

OUTPUT:



Row	Cities	States
1	4119	27

INSIGHTS:

We can see that customers ordered placed from 27 states and 4119 cities.

RECOMMENDATION:

Target can expand their business to multiple cities.

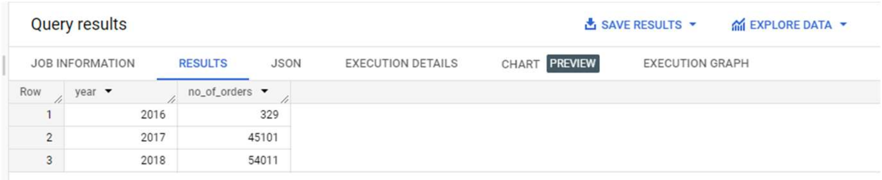
2.**In-depth Exploration:**

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

QUERY:

```
select
extract(year from order_purchase_timestamp) as year,
count(order_id) as no_of_orders
from `target.orders`
group by year
order by year
```

OUTPUT:



Row	year	no_of_orders
1	2016	329
2	2017	45101
3	2018	54011

INSIGHTS:

We can see that there is drastic increase in 2017 and little increase in 2018 in number of orders. Hence, we can see the growth of the company.

RECOMMENDATION:

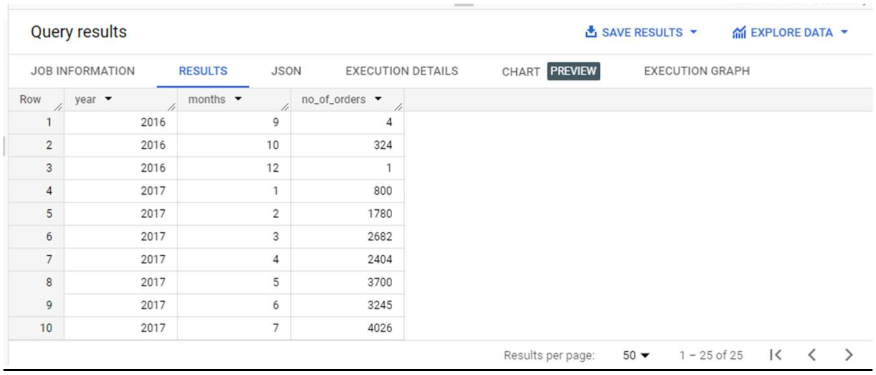
It is recommended for the company follow the same advertising strategy to increase the number of orders.

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

QUERY:

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as months , count(order_id) as no_of_orders
from `target.orders`
group by year,months
order by year,months
```

OUTPUT:



Row	year	months	no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

INSIGHTS:

We came to know there is no monthly seasonality in the given period. There is a drop in number of orders in the month of September in 2016 and 2018.

RECOMMENDATION:

So, in the month of September, the company should increase the advertising and introduce more offers.

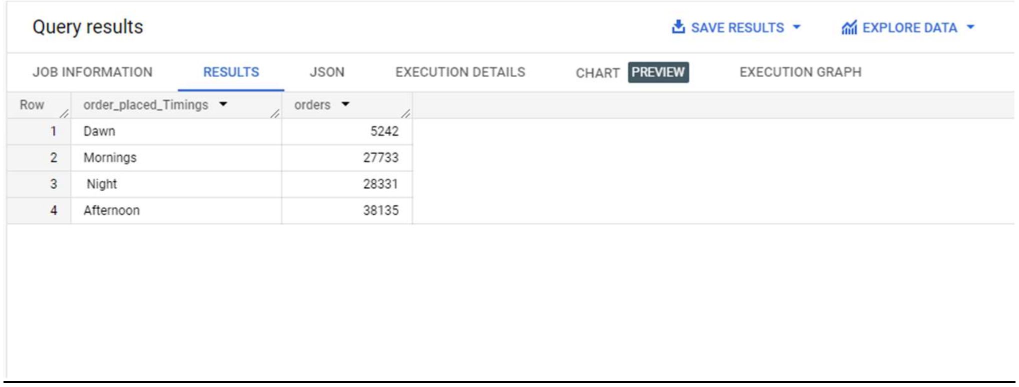
C) During what time of the day, do the Brazilian customers mostly place their orders?

QUERY :

```
select
case
when extract(hour from o. order_purchase_timestamp) between 0 and 6  then 'Dawn'
when extract(hour from o. order_purchase_timestamp) between 7 and 12  then 'Mornings'
when extract(hour from o. order_purchase_timestamp) between 13 and 18  then 'Afternoon'
else ' Night'
end as order_placed_Timings , count(*) as orders
```

```
from `target.orders` o inner join `target.customers` c
on o.customer_id = c.customer_id
group by order_placed_Timings
order by orders
```

OUTPUT:



Row	order_placed_Timings	orders
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

INSIGHTS:

It can be seen that large number of orders are placed during afternoon by Brazilian customers.

RECOMMENDATION:

So, in the period of dawn, they should re-stock the products so the buyers who purchase in the afternoon can get fresh products.

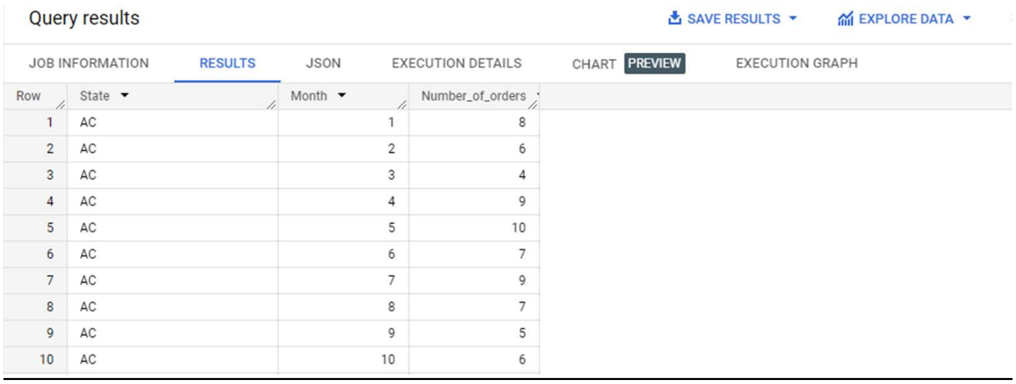
3. Evolution of E-commerce orders in the Brazil region:

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

QUERY:

```
select c.customer_state as State,
extract(month from o.order_purchase_timestamp) as Month,
count(o.order_id) as  Number_of_orders
from `target.customers` c inner join `target.orders` o
on c.customer_id = o.customer_id
group by State,Month
order by State,Month
```

OUTPUT:



Row	State	Month	Number_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

INSIGHTS:

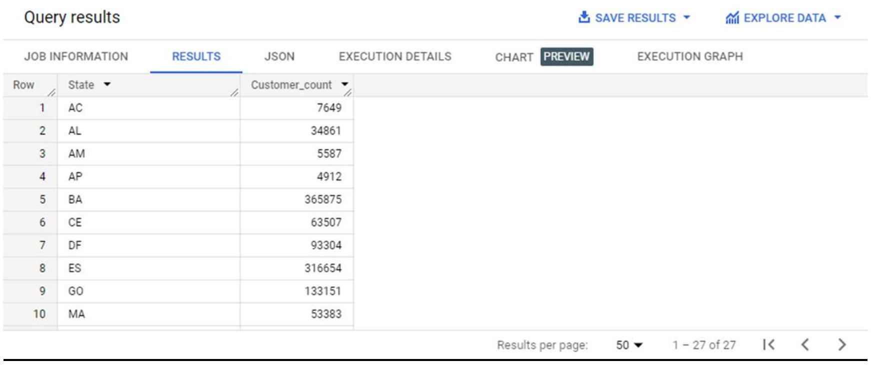
The maximum number of orders (4982) are placed in the month of August in the state of Sau Paulo (SP) and the minimum (2) in the month of September in the state Amapa (AP).

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

QUERY:

```
select customer_state as State, count(customer_unique_id) as Customer_count
from `target.customers` c inner join `target.geolocation` g
on c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
Group by c.customer_state
order by c.customer_state, Customer_count
```

OUTPUT:



Row	State	Customer_count
1	AC	7649
2	AL	34861
3	AM	5587
4	AP	4912
5	BA	365875
6	CE	63507
7	DF	93304
8	ES	316654
9	GO	133151
10	MA	53383

INSIGHTS:

The customers are not distributed evenly across all the states. We can see that the maximum number of customers are present in the state of Sau Paulo (SP) and minimum in the state of Roraima (RR).

RECOMMENDATION:

It is recommended for the company to have enough manpower in the states with large number of customers to manage the business more effectively.

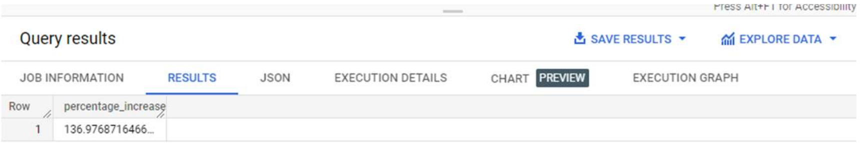
4.Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

QUERY:

```
with cte as  (
select
extract(year from o.order_purchase_timestamp) as year,
extract(month FROM o.order_purchase_timestamp) as month,
sum(p.payment_value) as total_cost
from `target.orders` o inner join `target.payments` p
on o.order_id = p.order_id
where
extract(year from o.order_purchase_timestamp) in (2017, 2018)
and
extract(month from o.order_purchase_timestamp) between 1 and 8
group by year, month)
select
 (sum(case when year = 2018 then total_cost end) -
   sum(case when year = 2017 then total_cost end)) /
   sum(case when year = 2017 then total_cost end) * 100 as percentage_increase
from cte
```

OUTPUT:



Row	percentage_increase
1	136.9768716466...

INSIGHTS:

From the output the cost of orders has increased 136.98% from the year 2017 to 2018 considering the months of January to august only

RECOMMENDATION:

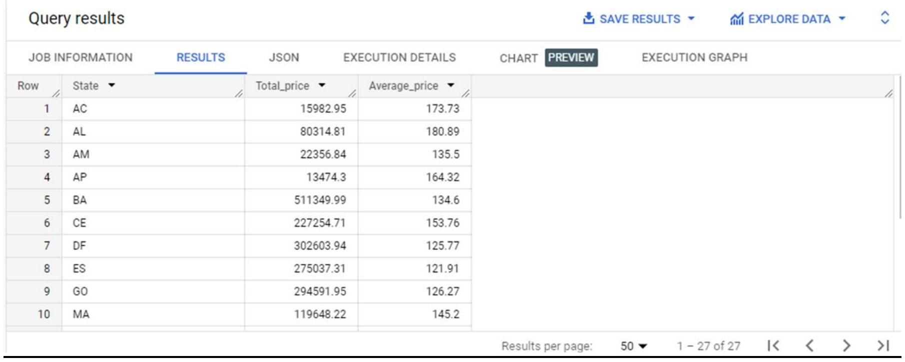
It is suggested the company follow the same business strategy that is followed in the year 2018, since we can see a great growth in the numbers.

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

QUERY:

```
select
c.customer_state as State,
round(sum(oi.price),2) as Total_price,
round(avg(oi.price),2) as Average_price
from `target.orders` o inner join  `target.customers` c
on c.customer_id = o.customer_id
inner join `target.order_items`  oi
on o.order_id = oi.order_id
group by c.customer_state
order by c.customer_state
```

OUTPUT:



Row	State	Total_price	Average_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

INSIGHTS:

It is observed that (SP) has the maximum total price (5202955.05). (RR) has the minimum total price (7829.43)

RECOMMENDATION:

The company must focus more on the states with minimum total price.

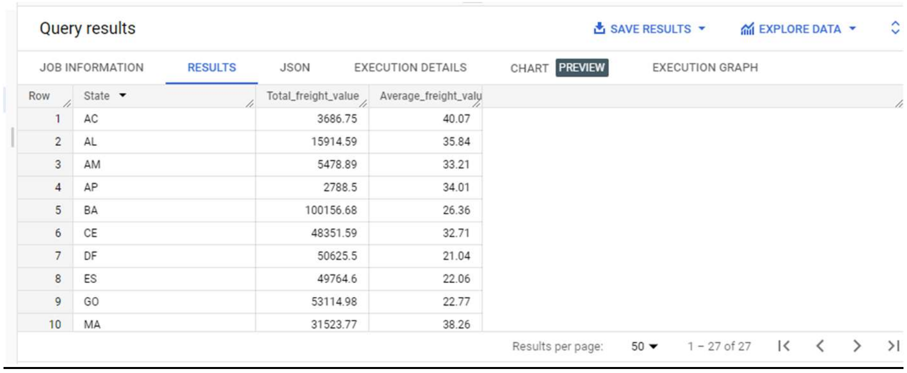
C) Calculate the Total & Average value of order freight for each state

QUERY:

```
select
c.customer_state as State,
round(sum(oi.freight_value),2) as Total_freight_value,
round(avg(oi.freight_value),2) as Average_freight_value
from `target.orders` o inner join  `target.customers` c
on c.customer_id = o.customer_id
inner join `target.order_items`  oi
on o.order_id = oi.order_id
group by c.customer_state
```

```
order by c.customer_state
```

OUTPUT:



Row	State	Total_freight_value	Average_freight_valu
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

INSIGHTS:

It is observed that (SP) has the maximum total freight (718723.07) and minimum average freight value (15.15). (RR) has the minimum total price (2235.19) and maximum freight value (42.98). This shows that it takes 3 times the cost of delivery in (RR) compared to (SP).

RECOMMENDATION:

The company must try to decrease the freight value in states with high average freight value by changing the delivery policy.

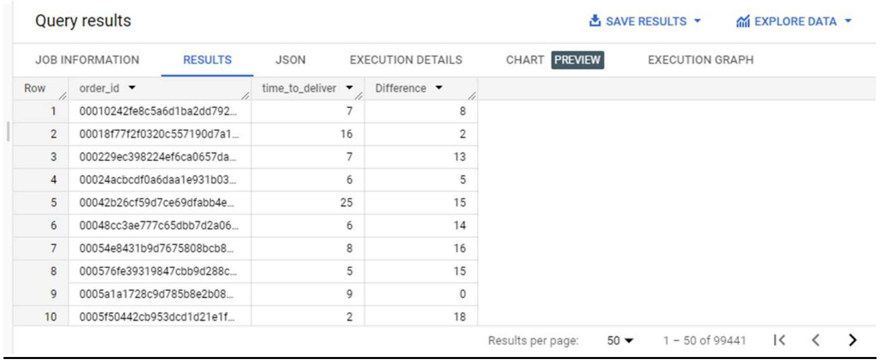
5.Analysis based on sales, freight and delivery time.

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.

QUERY:

```
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 Difference
from `target.orders`
order by order_id
```

OUTPUT:



Row	order_id	time_to_deliver	Difference
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

INSIGHTS:

Most of the orders are delivered within 10 days (about 1 and a half weeks) of order being placed, a very few orders take a long duration to be delivered. The difference column shows that most of the orders are delivered before the estimated date of delivery.

RECOMMENDATION:

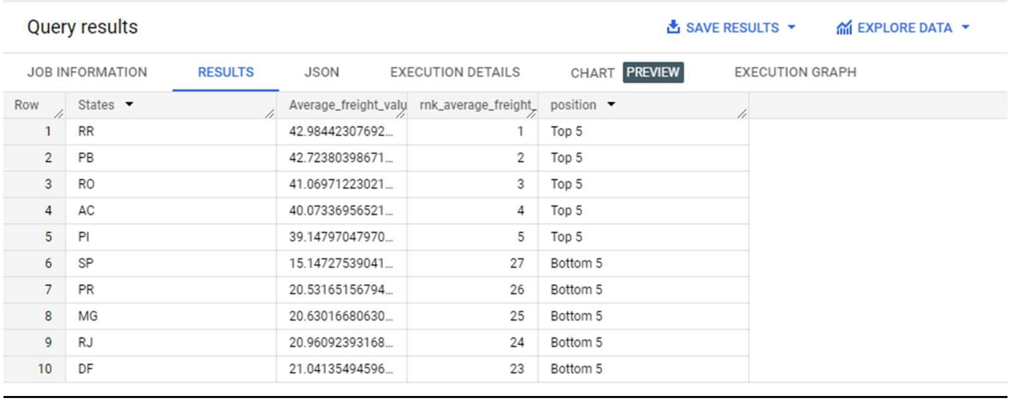
The company must look into the orders with large time to deliver and the place, distance covered and improvise the delivery system.

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

QUERY:

```
(select  c.customer_state as States,
avg(oi.freight_value) as Average_freight_value ,
dense_rank() over (order by avg(oi.freight_value) desc) as rnk_average_freight_value, "Top 5" as position
from `target.orders` o inner join `target.customers` c
on c.customer_id = o.customer_id
inner join `target.order_items`  oi
on o.order_id = oi.order_id
group by c.customer_state
order by rnk_average_freight_value
limit 5)
union distinct
(select  c.customer_state as States,
avg(oi.freight_value) as Average_freight_value ,
dense_rank() over (order by avg(oi.freight_value) desc) as rnk_average_freight_value,
"Bottom 5" as position
from `target.orders` o inner join `target.customers` c
on c.customer_id = o.customer_id
inner join `target.order_items`  oi
on o.order_id = oi.order_id
group by c.customer_state
order by rnk_average_freight_value desc
limit 5)
```

OUTPUT:



Row	States	Average_freight_valu	rnk_average_freight_	position
1	RR	42.98442307692...	1	Top 5
2	PB	42.72380398671...	2	Top 5
3	RO	41.06971223021...	3	Top 5
4	AC	40.07336956521...	4	Top 5
5	PI	39.14797047970...	5	Top 5
6	SP	15.14727539041...	27	Bottom 5
7	PR	20.53165156794...	26	Bottom 5
8	MG	20.63016680630...	25	Bottom 5
9	RJ	20.96092393168...	24	Bottom 5
10	DF	21.04135494596...	23	Bottom 5

INSIGHTS:

We can see the top 5 states which have maximum average freight value.

RECOMMENDATION:

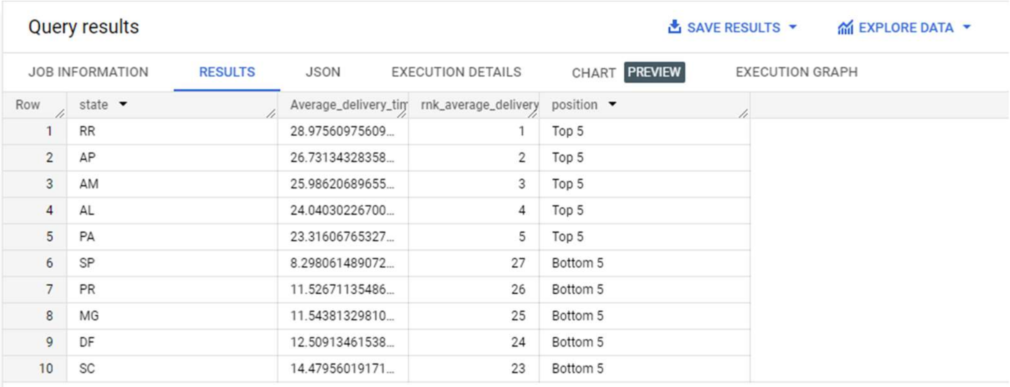
The company must try to decrease the freight value in states with high average freight value by changing the delivery policy.

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

QUERY:

```
with cte as(
  select c.customer_state,
  date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as delivery_time
  from `target.orders` o inner join  `target.customers` c
  on c.customer_id = o.customer_id
  order by delivery_time
)
(select customer_state as state,
avg(delivery_time) as Average_delivery_time,
dense_rank() over(order by avg(delivery_time) desc ) as rnk_average_delivery_time,
"Top 5" as position
from cte
group by customer_state
order by rnk_average_delivery_time
limit 5)
UNION DISTINCT
(select customer_state as state,
avg(delivery_time) as Average_delivery_time,
dense_rank() over(order by avg(delivery_time) desc ) as rnk_average_delivery_time,
"Bottom 5" as position
from cte
group by customer_state
order by rnk_average_delivery_time desc
limit 5)
```

OUTPUT:



Row	state	Average_delivery_tim	rnk_average_delivery	position
1	RR	28.97560975609...	1	Top 5
2	AP	26.73134328358...	2	Top 5
3	AM	25.98620689655...	3	Top 5
4	AL	24.04030226700...	4	Top 5
5	PA	23.31606765327...	5	Top 5
6	SP	8.298061489072...	27	Bottom 5
7	PR	11.52671135486...	26	Bottom 5
8	MG	11.54381329810...	25	Bottom 5
9	DF	12.50913461538...	24	Bottom 5
10	SC	14.47956019171...	23	Bottom 5

INSIGHTS:

We can see the top 5 of highest and lowest states which have maximum average delivery time.

RECOMMENDATION:

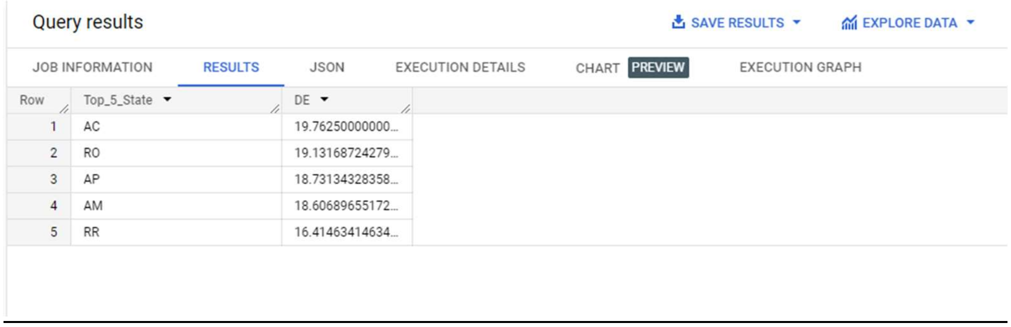
The company has to concentrate on the cities with maximum average delivery time. It has to look into the average distance covered to enhance the delivery service.

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

QUERY:

```
select c.customer_state as Top_5_State,
avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)) as DE
from `target.orders` o inner join  `target.customers` c
on c.customer_id = o.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state
order by DE desc
limit 5
```

OUTPUT:



Row	Top_5_State	DE
1	AC	19.76250000000...
2	RO	19.13168724279...
3	AP	18.73134328358...
4	AM	18.60689655172...
5	RR	16.41463414634...

INSIGHTS:

We can see the states where the delivery is fast compared to the estimated date of delivery.

RECOMMENDATION:

The company must increase the number of vehicles used for transportation to enhance delivery service.

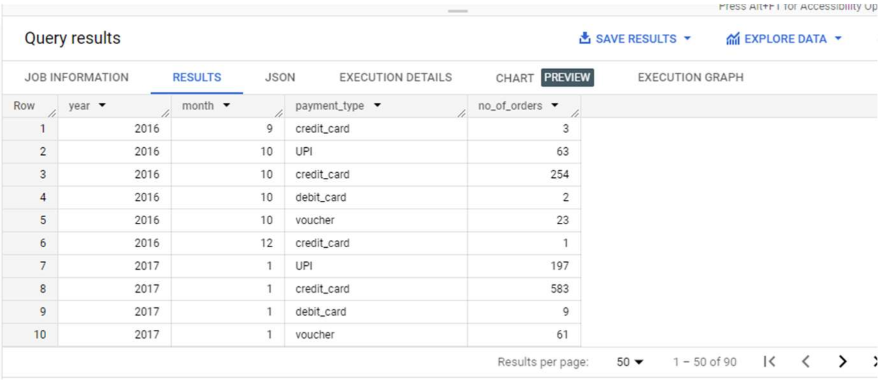
6) Analysis based on the payments:

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

QUERY:

```
select
extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
p.payment_type as payment_type,
count(o.order_id) as no_of_orders
from `target.orders` o inner join  `target.payments` p
on o.order_id = p.order_id
group by year,month,payment_type
order by year,month,payment_type
```

OUTPUT:



Query results

Row	year	month	payment_type	no_of_orders
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	583
9	2017	1	debit_card	9
10	2017	1	voucher	61

INSIGHTS:

It can be noticed from the output that maximum payments have been made through credit cards and min with debit cards.

RECOMMENDATION:

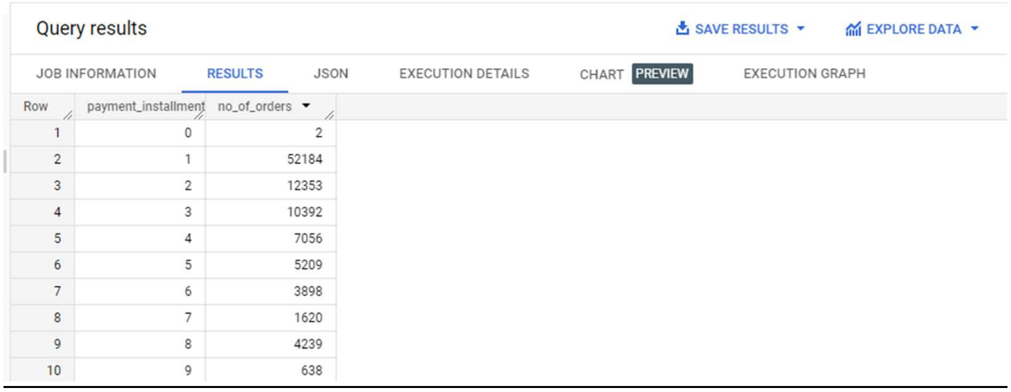
Since most of the payments are through credit cards the company can avail offer for the customer who pay through credit card.

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

QUERY:

```
select
p.payment_installments,count(o.order_id) as no_of_orders
from `target.orders` o inner join `target.payments` p
on o.order_id = p.order_id
where o.order_status != 'canceled'
group by p.payment_installments
order by p.payment_installments,no_of_orders desc
```

OUTPUT:



Row	payment_installment	no_of_orders
1	0	2
2	1	52184
3	2	12353
4	3	10392
5	4	7056
6	5	5209
7	6	3898
8	7	1620
9	8	4239
10	9	638

INSIGHTS:

It can be noticed from the output that maximum orders have been made through in a single installment.

RECOMMENDATION: