BUSINESS CASE STUDY

TARGET RETAIL IN BRAZIL

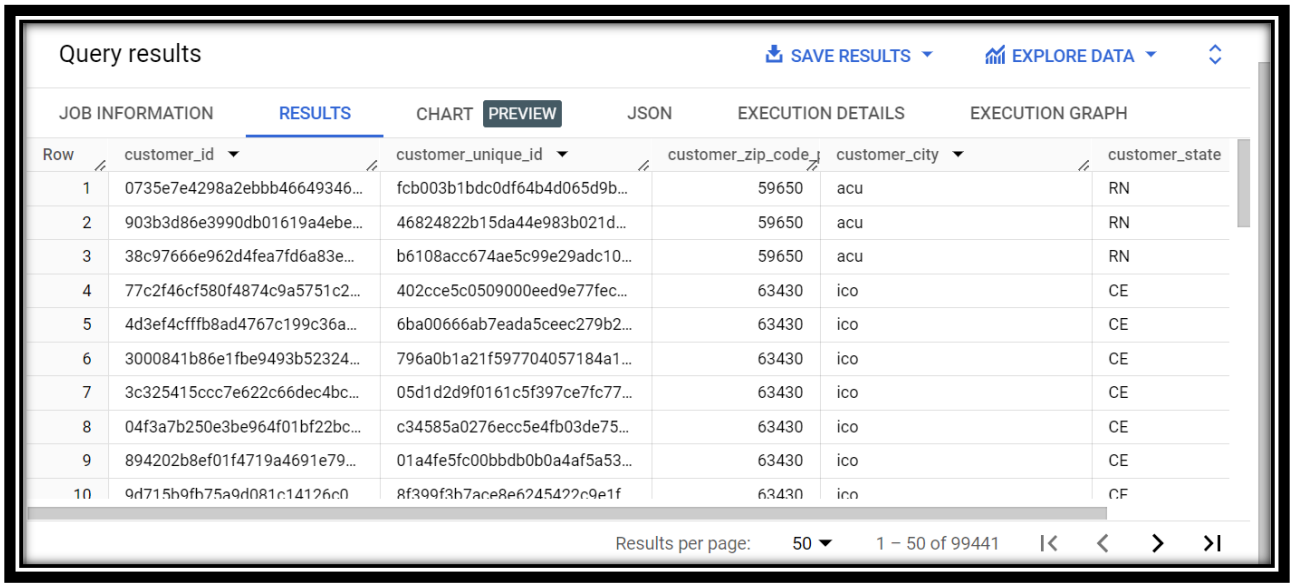
(2016-2018)

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

select * from Target.customers;

correct Answer:

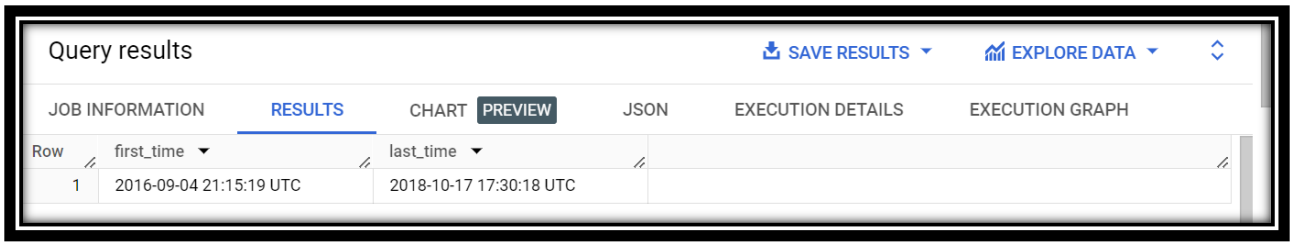
```
select column_name,data_type from verdant-tempest-399415.Target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
```



Row	customer_id	customer_unique_id	customer_zip_code_	customer_city	customer_state
1	0735e7e4298a2ebbb46649346...	fcb003b1bdc0df64b4d065d9b...	59650	acu	RN
2	903b3d86e3990db01619a4ebe...	46824822b15da44e983b021d...	59650	acu	RN
3	38c97666e962d4fea7fd6a83e...	b6108acc674ae5c99e29adc10...	59650	acu	RN
4	77c2f46cf580f4874c9a5751c2...	402cce5c0509000eed9e77fec...	63430	ico	CE
5	4d3ef4cfffb8ad4767c199c36a...	6ba00666ab7eada5ceec279b2...	63430	ico	CE
6	3000841b86e1fbe9493b52324...	796a0b1a21f597704057184a1...	63430	ico	CE
7	3c325415ccc7e622c66dec4bc...	05d1d2d9f0161c5f397ce7fc77...	63430	ico	CE
8	04f3a7b250e3be964f01bf22bc...	c34585a0276ecc5e4fb03de75...	63430	ico	CE
9	894202b8ef01f4719a4691e79...	01a4fe5fc00bbdb0b0a4af5a53...	63430	ico	CE
10	9d715b9fb75a9d081c14126c0	8f399f3b7ace8e6245422c9e1f	63430	ico	CE

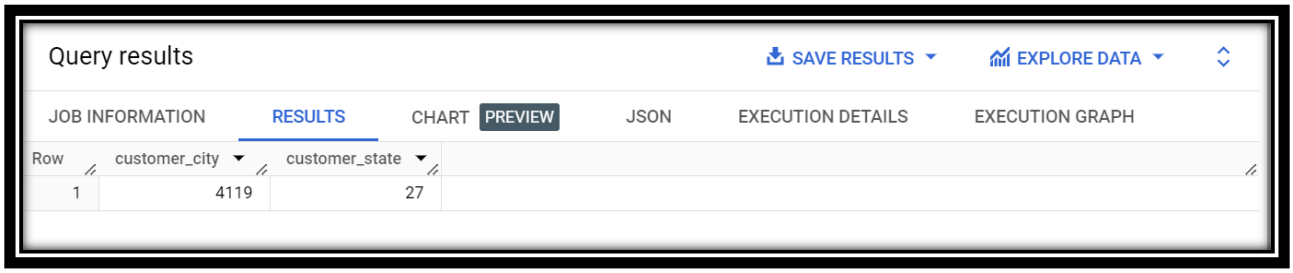
1.2. Get the time range between which the orders were placed.

```
select  min(order_purchase_timestamp) as first_time,
max(order_purchase_timestamp) as last_time from Target.orders;
```



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

```
select
count(distinct customer_city) as customer_city,
count(distinct customer_state) as customer_state
from Target.customers;
```



2.In-depth Exploration:

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

```
select
FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp) as month,
count(FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp)) as no_of_orders
from Target.orders
GROUP BY 1
ORDER BY 1
```

Ans: yes the orders are growing for the past years

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW

Row	month	no_of_orders
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026

Query results

SAVE RESULTS | EXPLORE DATA

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

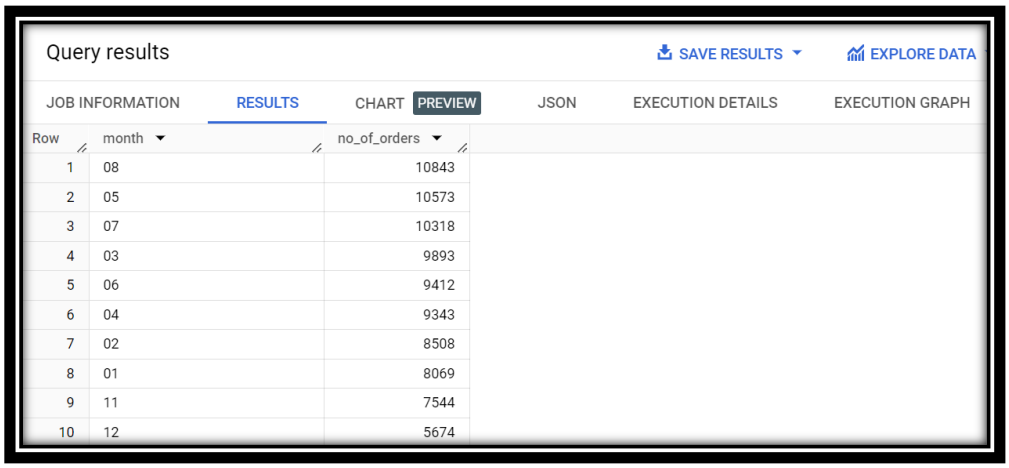
Row	no_of_orders	year
1	329	year 2016
2	45101	year 2017
3	54011	year 2018

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

```
select

t.month,

t.no_of_orders
 from
(
select
FORMAT_TIMESTAMP('%m',order_purchase_timestamp) as month,
count(FORMAT_TIMESTAMP('%m',order_purchase_timestamp)) as no_of_orders
from Target.orders
group by FORMAT_TIMESTAMP('%m',order_purchase_timestamp)) t
order by t.no_of_orders desc;
```



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	month	no_of_orders
1	08	10843
2	05	10573
3	07	10318
4	03	9893
5	06	9412
6	04	9343
7	02	8508
8	01	8069
9	11	7544
10	12	5674

In month of “August,may,July” the purchase rate is high

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

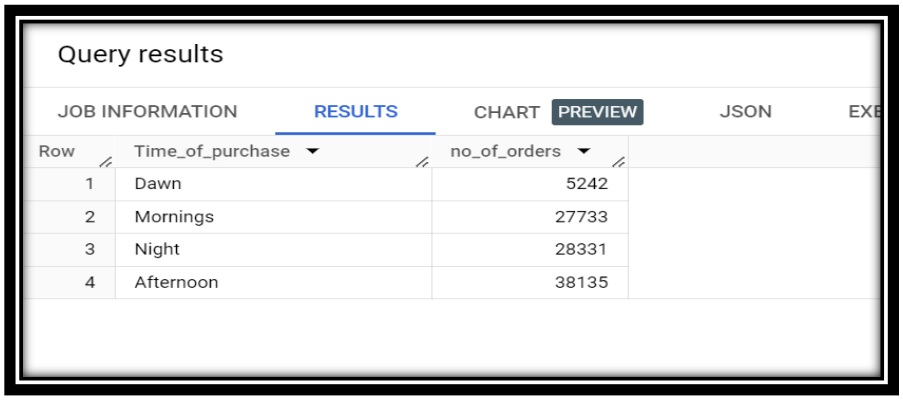
0-6 hrs : Dawn

7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night

```
select
case
   when FORMAT_TIMESTAMP('%H',order_purchase_timestamp) between '00' and '06' then 'Dawn'
   when FORMAT_TIMESTAMP('%H',order_purchase_timestamp) between '07' and '12' then 'Mornings'
   when FORMAT_TIMESTAMP('%H',order_purchase_timestamp) between '13' and '18' then 'Afternoon'
   when FORMAT_TIMESTAMP('%H',order_purchase_timestamp) between '19' and '23' then 'Night'
end as Time_of_purchase,
count(*) as no_of_orders
from Target.orders
group by 1
order by no_of_orders;
```



Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON | EXE

Row	Time_of_purchase	no_of_orders
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

Brazilian customers mostly place their orders in "Afternoon"

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
t.customer_state,
t.month,
t.no_of_orders from (
select
c.customer_state,
FORMAT_TIMESTAMP('%Y-%m',o.order_purchase_timestamp) as month,
count(FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp)) as no_of_orders
from Target.customers c
left join Target.orders o
using(customer_id)
group by FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp),c.customer_state)t
order by 2,1;
```

Query results

SAVE RESULTS

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

Row	customer_state	month	no_of_orders
21	RS	2016-10	24
22	SC	2016-10	11
23	SE	2016-10	3
24	SP	2016-10	113
25	PR	2016-12	1
26	AC	2017-01	2
27	AL	2017-01	2
28	BA	2017-01	25
29	CE	2017-01	9
30	DF	2017-01	13

3.2How are the customers distributed across all the states?

```
select
```

```
distinct customer_state,
count(*) as no_of_customers
from Target.customers
group by customer_state
order by no_of_customers desc;
```

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

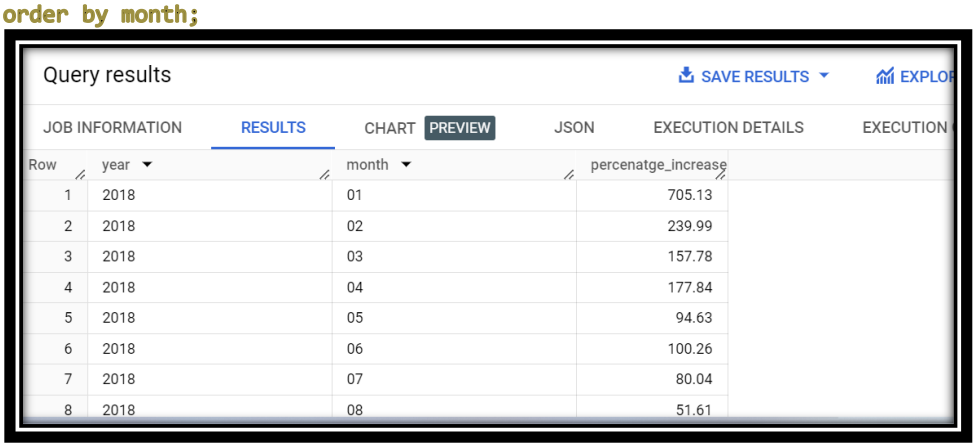
Target in Brazil have there customers more in 'SP','RJ','MG'

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

4.1Get 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.

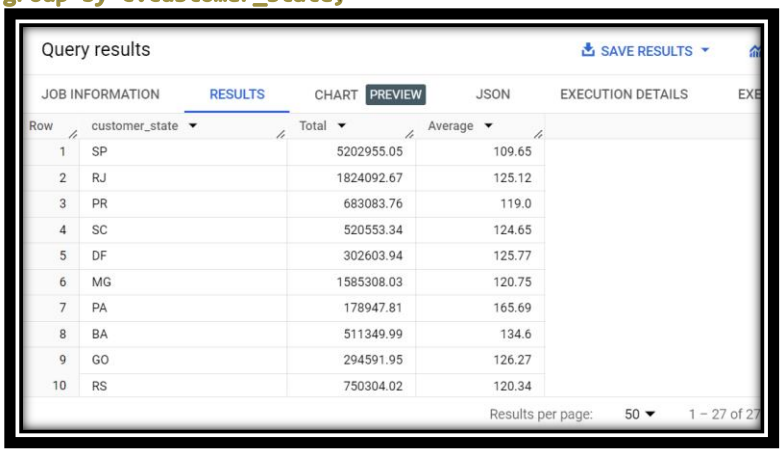
```
create view Target.cte1 as
(select
year,
month,
payment_value from
(select
FORMAT_TIMESTAMP('%Y',o.order_purchase_timestamp) as year,
FORMAT_TIMESTAMP('%m',o.order_purchase_timestamp) as month,
round(sum(p.payment_value),2) as payment_value
from Target.payments p
join Target.orders o
using(order_id)
group by 1,2)t
where t.year between '2017' and '2018' and
t.month between '01' and '08'
order by t.month);
select year,month,percenatge_increase from
(select
year,
month,
payment_value,
round((payment_value - lead(payment_value,1) over(partition by month order by payment_value
desc))
/(lead(payment_value,1) over(partition by month order by payment_value desc) * .01),2) as
percenatge_increase
from Target.cte1)t
where t.percenatge_increase is not null
order by month;
```



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

```
select
c.customer_state,
round(Sum(ot.price),2) as Total,
round(Sum(distinct ot.price)/count(distinct ot.price),2) as Average,
 from Target.orders o
```

```
join Target.order_items ot using(order_id)
left join Target.customers c using (customer_id)
group by c.customer_state;
```



Query results

SAVE RESULTS

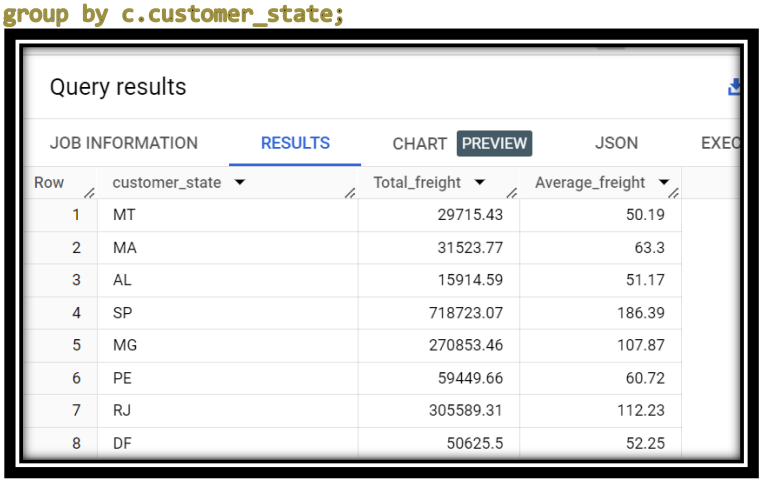
JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXE

Row	customer_state	Total	Average
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	PR	683083.76	119.0
4	SC	520553.34	124.65
5	DF	302603.94	125.77
6	MG	1585308.03	120.75
7	PA	178947.81	165.69
8	BA	511349.99	134.6
9	GO	294591.95	126.27
10	RS	750304.02	120.34

Results per page: 50 1 – 27 of 27

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

```
select
c.customer_state,
round(Sum(ot.freight_value),2) as Total_freight,
round(Avg(ot.freight_value),2) as Average_freight,
 from Target.orders o
join Target.order_items ot using(order_id)
left join Target.customers c using (customer_id)
group by c.customer_state;
```



Query results

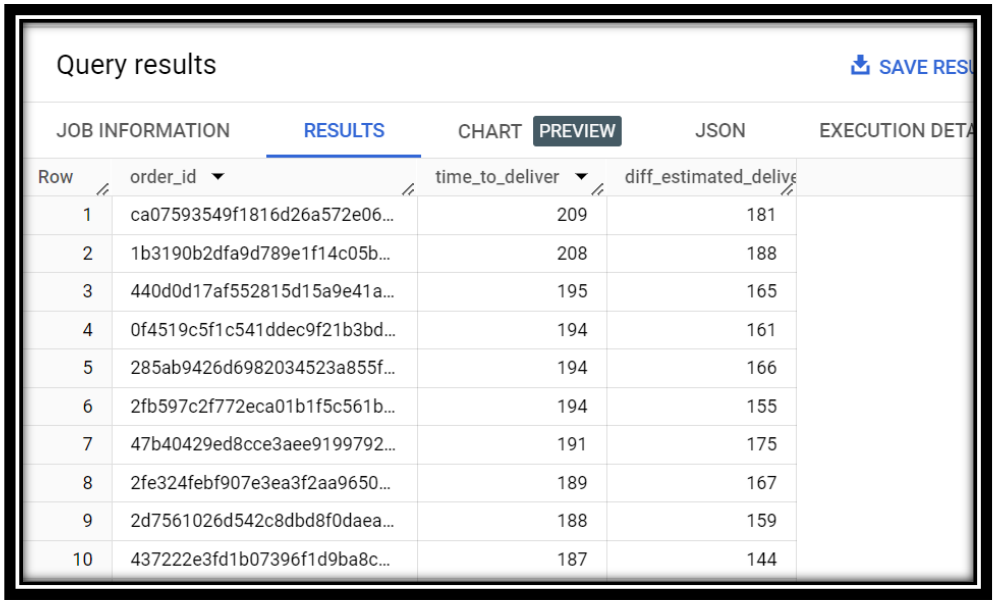
JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXEC

Row	customer_state	Total_freight	Average_freight
1	MT	29715.43	50.19
2	MA	31523.77	63.3
3	AL	15914.59	51.17
4	SP	718723.07	186.39
5	MG	270853.46	107.87
6	PE	59449.66	60.72
7	RJ	305589.31	112.23
8	DF	50625.5	52.25

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.

```
select
distinct order_id,
DATETIME_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) as time_to_deliver,
DATETIME_DIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY) as
diff_estimated_delivery
from Target.orders
where order_status = 'delivered'
ORDER BY time_to_deliver DESC;
```



Query results

SAVE RESU

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETA

Row	order_id	time_to_deliver	diff_estimated_delive
1	ca07593549f1816d26a572e06...	209	181
2	1b3190b2dfa9d789e1f14c05b...	208	188
3	440d0d17af552815d15a9e41a...	195	165
4	0f4519c5f1c541ddec9f21b3bd...	194	161
5	285ab9426d6982034523a855f...	194	166
6	2fb597c2f772eca01b1f5c561b...	194	155
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324febf907e3ea3f2aa9650...	189	167
9	2d7561026d542c8dbd8f0daea...	188	159
10	437222e3fd1b07396f1d9ba8c...	187	144

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

```
(select * from (
select
c.customer_state,
round(sum(freight_value)/count(freight_value),2) as Avg_freight_value,
dense_rank() over(order by round(sum(freight_value)/count(freight_value),2) desc ) as rank
from Target.orders o
join Target.order_items ot using(order_id)
join Target.customers c using(customer_id)
group by c.customer_state
order by Avg_freight_value desc) t
where t.rank <= 5)
union all
(select * from
(select
c.customer_state,
round(sum(freight_value)/count(freight_value),2) as Avg_freight_value,
dense_rank() over(order by round(sum(freight_value)/count(freight_value),2)) as rank
from Target.orders o
join Target.order_items ot using(order_id)
join Target.customers c using(customer_id)
group by c.customer_state
order by Avg_freight_value) t
where t.rank <=5)
order by Avg_freight_value;
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON

Row	customer_state	Avg_freight_value	rank
1	SP	15.15	1
2	PR	20.53	2
3	MG	20.63	3
4	RJ	20.96	4
5	DF	21.04	5
6	PI	39.15	5
7	AC	40.07	4
8	RO	41.07	3
9	PB	42.72	2
10	RR	42.98	1

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

```
(select * from
(select
c.customer_state,
round(avg(DATETIME_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY)),2) as
Avg_delivery_time,
dense_rank() over(order by
round(avg(DATETIME_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY)),2)
desc ) as rank
from Target.customers c
join Target.orders o
using(customer_id)
group by c.customer_state
order by Avg_delivery_time desc) t
where t.rank <= 5)
union all
(select * from
(select
c.customer_state,
round(avg(DATETIME_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY)),2) as
Avg_delivery_time,
dense_rank() over(order by
round(avg(DATETIME_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY)),2))
as rank
from Target.customers c
join Target.orders o
using(customer_id)
group by c.customer_state
order by Avg_delivery_time) t
where t.rank <= 5)
order by Avg_delivery_time;
```

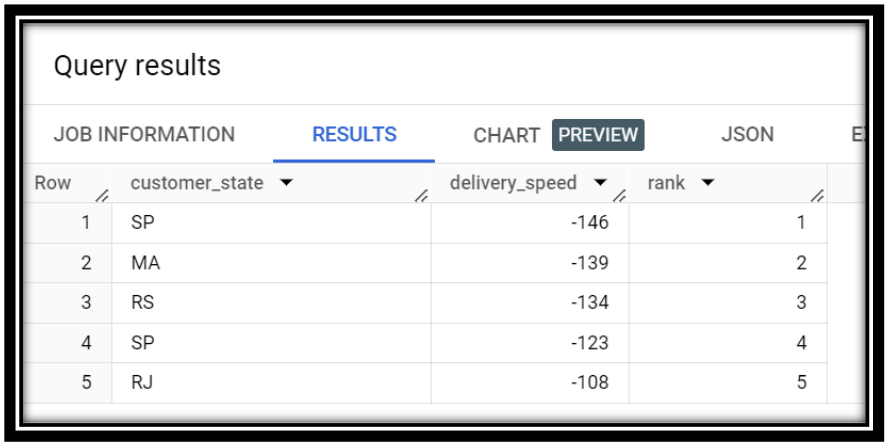
Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON | EXECU

Row	customer_state	Avg_delivery_time	rank
1	SP	8.3	1
2	PR	11.53	2
3	MG	11.54	3
4	DF	12.51	4
5	SC	14.48	5
6	PA	23.32	5
7	AL	24.04	4
8	AM	25.99	3
9	AP	26.73	2
10	RR	28.98	1

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

```
select * from
(select
c.customer_state,
DATETIME_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date,DAY) as
delivery_speed,
dense_rank() over(order by
DATETIME_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date,DAY)) as rank
from Target.customers c
join Target.orders o
using(customer_id)
order by delivery_speed) t
where t.delivery_speed is not null
and t.rank<=5;
```



Row	customer_state	delivery_speed	rank
1	SP	-146	1
2	MA	-139	2
3	RS	-134	3
4	SP	-123	4
5	RJ	-108	5

6.Analysis based on the payments:

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

```
select
FORMAT_TIMESTAMP('%m',o.order_purchase_timestamp) as month,
p.payment_type,
count(distinct o.order_id) as no_of_orders
from Target.payments p
join Target.orders o
using(order_id)
group by 2,1
order by month,no_of_orders desc;
```

Query results

SAVE RESULTS

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

Row	month	payment_type	no_of_orders
1	01	credit_card	6093
2	01	UPI	1715
3	01	voucher	337
4	01	debit_card	118
5	02	credit_card	6582
6	02	UPI	1723
7	02	voucher	288
8	02	debit_card	82
9	03	credit_card	7682
10	03	UPI	1942

Results per page: 50 1 – 50

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

```
select
```

```
payment_installments,
count(distinct order_id) as no_of_orders
from Target.payments
where payment_installments > 0 and
payment_sequential > 0
group by payment_installments;
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	payment_installment	no_of_orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315

Insights from the data

- São Paulo, Gerais, Rio have the greater number of orders placed by the customer, So the keep increasing the outlet in these areas can be able to attract customers, boost sales, and enhance the brand image.
- Promote with more activities and attract customers with more privilege at afternoon.
- Retails can establish more discount and best price over the month period from May to August every year.
- Freight insurance policy can implement in the high revenue states and continued to all other secondary revenue states.
- PA, AL,AM,AP,RR are the state having high delivery time, so we can keep local warehouse and Inventory Management
- for these regions to reduce the delivery time.
- Since more customers chose credit card as their mode of payment, we can offer incentives such as airline miles, hotel room rentals, gift certificates and cash back on purchases.
- And we can keep a credit card outlet to get new credit cards for customer who have not use credit card as their mode of payment.

End of Case study