Business Case:- Target

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

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

Ans:-

```
select
column_name,data_type from target_sql.INFORMATION_SCHEMA.COLUMNS
WHERE table_name='customers';
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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:- Most columns are string type.

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

Ans:-

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

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

Insights:-This market in brazil lasted for 2years with starting in 2016 and ending in 2018.

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

Ans:-

```
select count(geolocation_city) as count_city,
count(geolocation_state) as count_state
from target_sql.geolocation;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	count_city	count_state
1	1000163	1000163

Insights:- From the result it is clear that target has its branches in almost all the city and state in brazil.

2.In-depth Exploration:

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

Ans:-

```
select EXTRACT(year from order_purchase_timestamp) as order_year,
count(*) as order_count
from target_sql.orders
group by order_year
order by order_year;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

Row	order_year	order_count
1	2016	329
2	2017	45101
3	2018	54011

Insights:- After observing the order _year and order_count ,we can say yes there is a growing trend in the no of orders placed over the past years.

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

Ans:-

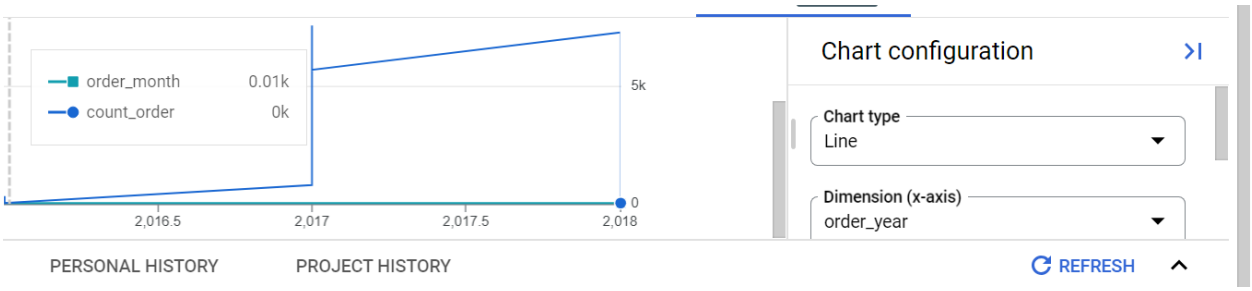
```sql
select extract(year from order_purchase_timestamp) as order_year,
extract(month from order_purchase_timestamp) as order_month,
count(*) as count_order
from target_sql.orders
group by order_year,order_month
order by order_year,order_month;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

Row	order_year	order_month	count_order
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:-



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

Ans:-

```
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'
else 'null'
end as order_time,
count(*) as count_order
from target_sql.orders
group by order_time
order by count_order desc;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION I

Row	order_time	count_order
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

PERSONAL HISTORY PROJECT HISTORY

Insights:- Maximum order is placed in the Afternoon followed by night.

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

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

Ans:-

```
select c.customer_state ,
Extract(month from o.order_purchase_timestamp) as order_month,
count (*) as count_order
from target_sql.customers c left join target_sql.orders o on
c.customer_id=o.customer_id
group by c.customer_state, order_month

order by c.customer_state asc,order_month;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	order_month	count_order
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:- No of order changes with month.

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

```
Ans:- select
customer_state,
count(*) as count_order
from target_sql.customers
group by customer_state
order by customer_state;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	count_order
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

Load more

Results per

Insights:- No of orders various with different states of brazil.

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

1. Get 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.

Ans:-

```
        with  payment1 as
 (select
 sum(payment_value )as total_2017 from target_sql.payments p join target_sql.orders o
using (order_id)
 where extract(year from o.order_purchase_timestamp)= 2017 and
 extract(month from o.order_purchase_timestamp) between 1 and 8),
  payment2 as
 (select
 sum(payment_value )as total_2018 from target_sql.payments p join target_sql.orders o
using (order_id)
 where extract(year from o.order_purchase_timestamp)= 2018 and
 extract(month from o.order_purchase_timestamp) between 1 and 8)
select
((payment2.total_2018 - payment1.total_2017)*100)/payment1.total_2017 as
increase_cost_order
from payment1,payment2;
```

Insights:- From the result it is observed 136% % increase in the cost of orders from year 2017 to 2018

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

Ans:-

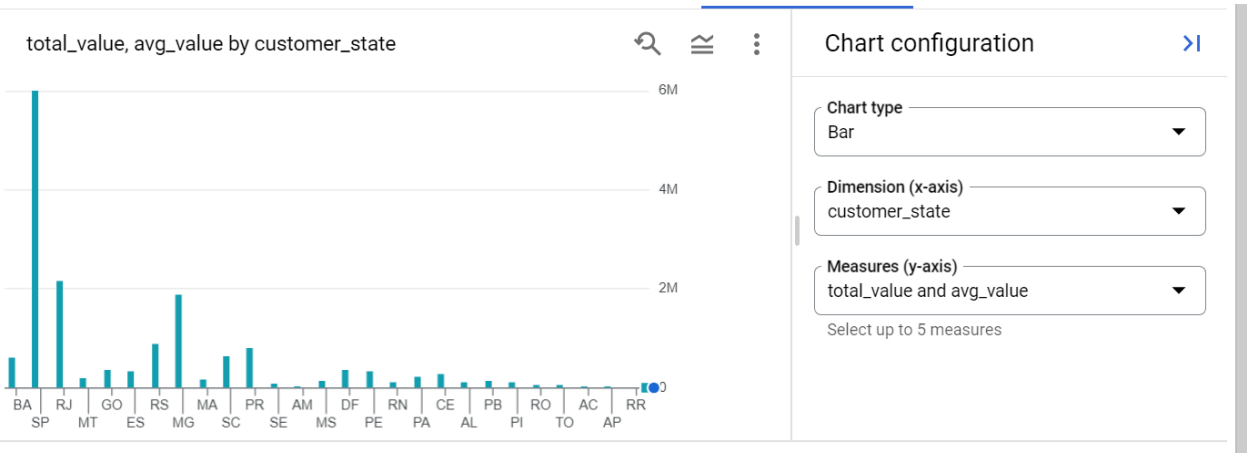
```
select c.customer_state,

sum(p.payment_value) as total_value,
avg(p.payment_value) as avg_value
from target_sql.payments p join target_sql.orders o using (order_id)
join target_sql.customers c using (customer_id)
group by c.customer_state;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	total_value	avg_value
1	BA	616645.8200000...	170.8160166204...
2	SP	5998226.959999...	137.5046297739...
3	RJ	2144379.689999...	158.5258882235...
4	MT	187029.29	195.2289039665...
5	GO	350092.3099999...	165.7634043560...
6	ES	325967.55	154.7069530137...
7	RS	890898.5400000...	157.1804057868...
8	MG	1872257.259999...	154.7064336473...
9	MA	152523.02	198.8566101694...

Insights:- This market in brazil is doing successful business in almost ll the states.



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

Ans:-

```
select
c.customer_state,
sum(oi.freight_value) as total_value,
avg(oi.freight_value) as avg_value from target_sql.order_items oi join
target_sql.orders o using (order_id)
join target_sql.customers c using (customer_id)
group by c.customer_state;
```

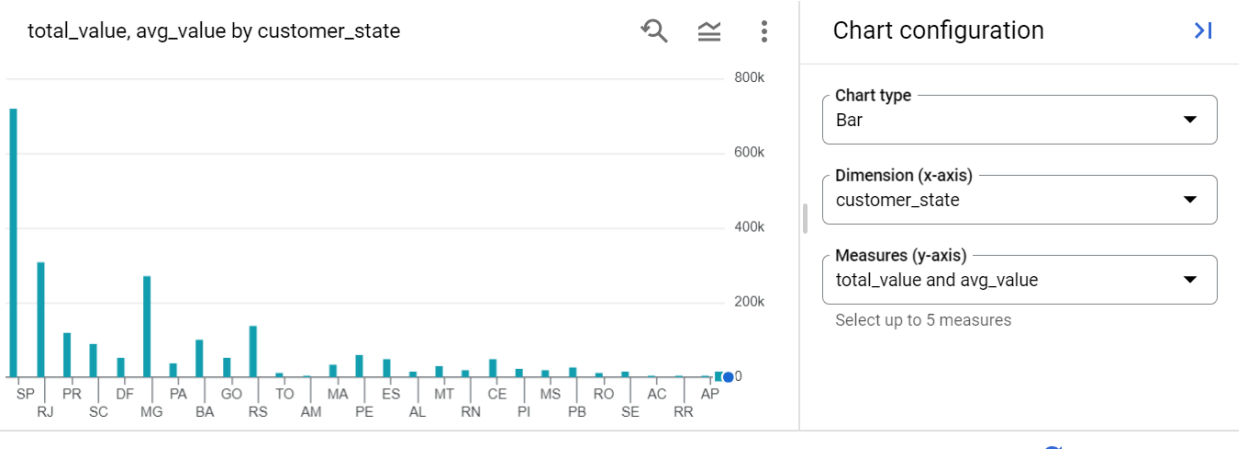
Query results

SAVE R

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

Row	customer_state	total_value	avg_value
1	SP	718723.0699999...	15.14727539041...
2	RJ	305589.3100000...	20.96092393168...
3	PR	117851.6800000...	20.53165156794...
4	SC	89660.26000000...	21.47036877394...
5	DF	50625.49999999...	21.04135494596...
6	MG	270853.4600000...	20.63016680630...
7	PA	38699.30000000...	35.83268518518...
8	BA	100156.6799999...	26.36395893656...
9	GO	53114.97999999...	22.76681525932...

Insights:-



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

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.

   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

Ans:-

```
select order_id,

 (extract(day from order_delivered_customer_date)) - (extract(day from
order_purchase_timestamp)) as time_to_deliver,
 (extract(day from order_estimated_delivery_date)) -(extract(day from
order_delivered_customer_date)) as diff_estimated_delivery
 from target_sql.orders;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHA

Row	order_id	time_to_deliver	diff_estimated_delive
1	2c45c33d2f9cb8ff8b1c86cc28...	0	-1
2	68f47f50f04c4cb6774570cfde...	0	2
3	304e7fc7db4a67a8ab0403ce4...	-28	11
4	c930f0fb9c6fed6ef015de48ea...	-29	12
5	d0462d19e9c58af6416a06e62...	21	-12
6	8d204be4884a2307f1486df72...	-27	13
7	0d8f485ffe96c81fe3e282095e...	-29	12
8	abe6fc40cd1fe4d8d30881130...	23	-17
9	8576190c64f6d9d9ed5055185...	-27	13
10	913e9a5e8da11e9a318ab2d38...	25	-24

Insights:- orders are delivered to the customer on time.

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

```
(select c.customer_state,
Avg(oi.freight_value) as freight_value,"highest" as comparison from
target_sql.order_items oi join target_sql.orders o using (order_id)
join target_sql.customers c using (customer_id)
group by c.customer_state
order by freight_value desc
LIMIT 5)
union all
(select c.customer_state,
Avg(oi.freight_value) as freight_value ,"lowest" as comparison from
target_sql.order_items oi join target_sql.orders o using (order_id)
join target_sql.customers c using (customer_id)
group by c.customer_state
order by freight_value asc
limit 5);
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | C

ow	customer_state	freight_value	comparison
1	RR	42.98442307692...	highest
2	PB	42.72380398671...	highest
3	RO	41.06971223021...	highest
4	AC	40.07336956521...	highest
5	PI	39.14797047970...	highest
6	SP	15.14727539041...	lowest
7	PR	20.53165156794...	lowest
8	MG	20.63016680630...	lowest
9	RJ	20.96092393168...	lowest
10	DF	21.04135494596...	lowest

Insights:- For some state company has to pay high freight _value(like RR,PB,RO,AC,PI) and for some state lowest freight_value(like SP,PR,MG,RJ,DF)

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

```
(select c.customer_state,
avg (date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as
delivery_time,"highest" as comparison from target_sql.orders o join
target_sql.customers c using (customer_id)
group by c.customer_state
order by delivery_time desc
limit 5
)
union all
(select c.customer_state,
avg (date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as
delivery_time,"lowest" as comparison from target_sql.orders o join
target_sql.customers c using (customer_id)
group by c.customer_state
order by delivery_time asc
limit 5
);
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	delivery_time	comparison
1	RR	28.97560975609...	highest
2	AP	26.73134328358...	highest
3	AM	25.98620689655...	highest
4	AL	24.04030226700...	highest
5	PA	23.31606765327...	highest
6	SP	8.298061489072...	lowest
7	PR	11.52671135486...	lowest
8	MG	11.54381329810...	lowest
9	DF	12.50913461538...	lowest
10	SC	14.47956019171...	lowest

Insights:- States with codes(RR,AP,M,AL,PA) has more delivery time in compare to states with codes(SP,PR,MG,DF,SC)

4.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.

Ans:-

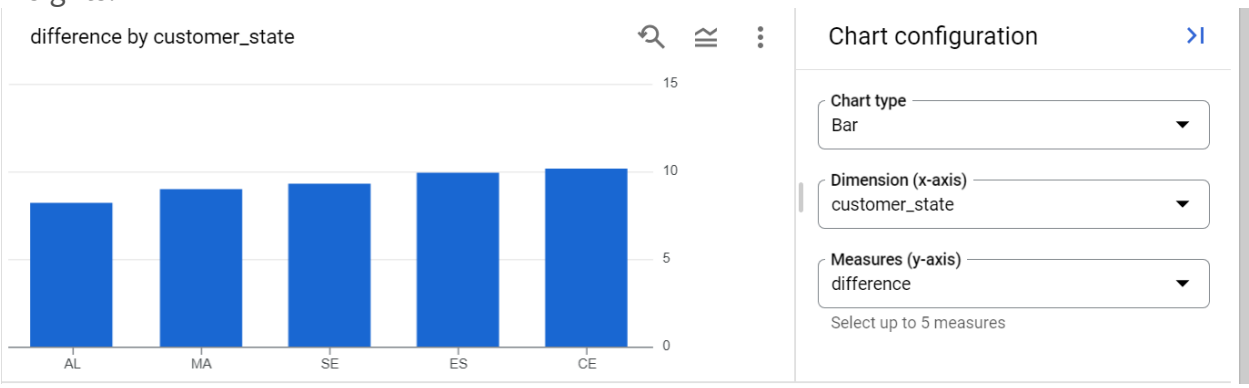
```
with cte as
(select c.customer_state,
avg (date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as
delivery_time,
 avg(date_diff(o.order_estimated_delivery_date,o.order_purchase_timestamp,day)) as
estimate_delivery_time
 from target_sql.orders o join target_sql.customers c using (customer_id)
 group by c.customer_state)
 select cte.customer_state,(cte.estimate_delivery_time- cte.delivery_time) as
difference from
 cte
 order by difference asc
 limit 5;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	difference
1	AL	8.184879331060...
2	MA	8.995294987481...
3	SE	9.321577825159...
4	ES	9.941657883025...
5	CE	10.11929932160...

Insights:-



6.Analysis based on the payments:

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

Ans:-

```
select
extract( month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year,
p.payment_type,
count(*) as num_orders
from target_sql.orders o join target_sql.payments p using (order_id)
group by year,month,p.payment_type
order by year,month,payment_type;
```

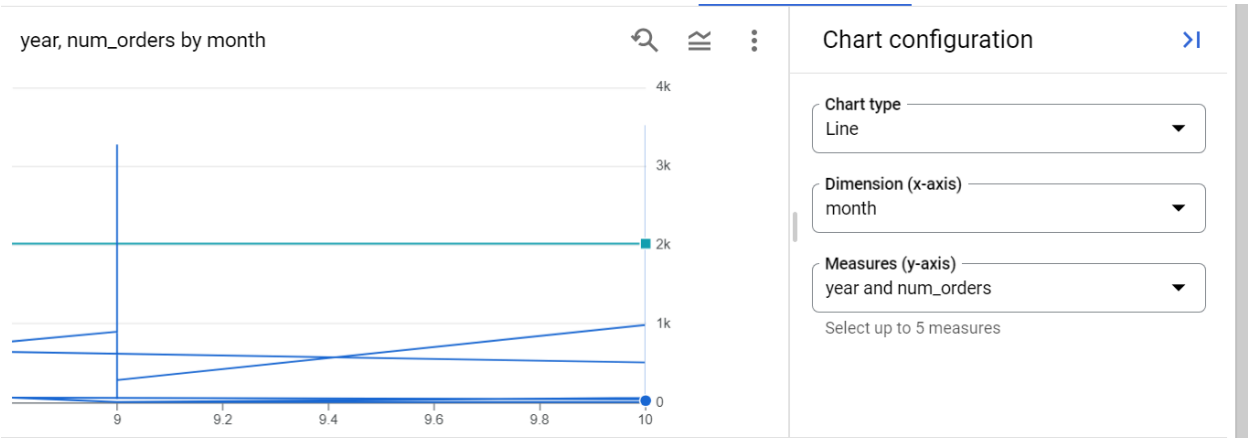
Query results

SAVE RESULTS

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

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

Insights:-



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

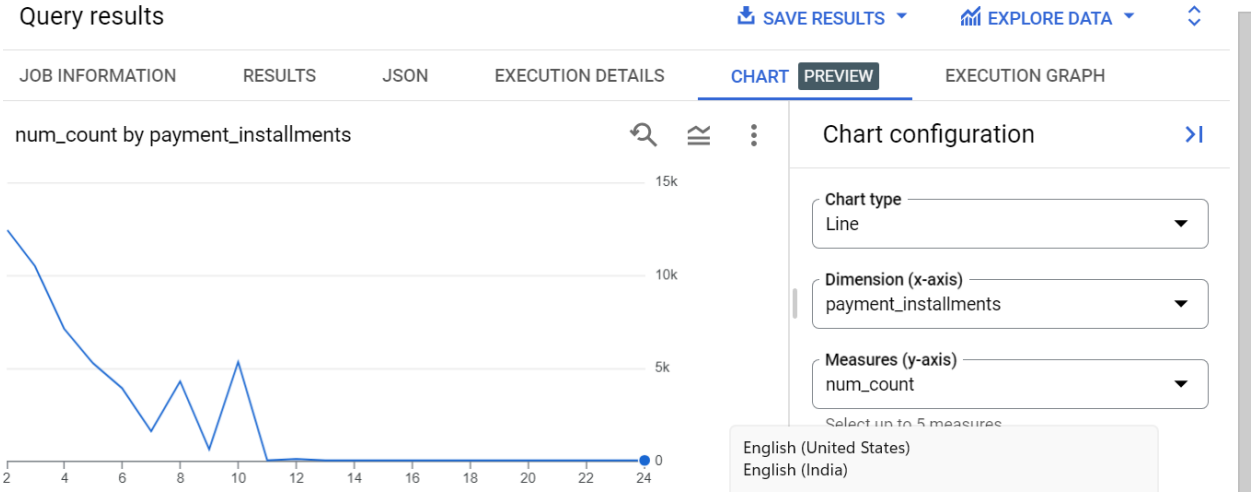
```
select payment_installments,
 count(*) as num_count
 from target_sql.payments
 where payment_installments>1
 group by payment_installments;
```

Query results

JOB INFORMATION | RESULTS | JSON | EX

Row	payment_installment	num_count
1	2	12413
2	3	10461
3	4	7098
4	5	5239
5	6	3920
6	7	1626
7	8	4268
8	9	644
9	10	5328

Insights:-



THANK YOU