I. 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.

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

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

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

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

```
select min (order_purchase_timestamp) start_time ,
max(order_purchase_timestamp)end_time
from `target.orders`
```

Query results

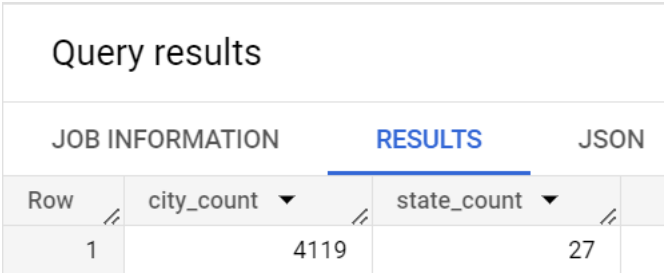
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

Assumption = Order_purchase_timestamp taken as the time for order placed with the given data.

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

```
select count(distinct customer_city)city_count, count(distinct
customer_state)state_count
from `target.customers` c
join `target.orders` o
on c.customer_id= o.customer_id
where order_purchase_timestamp between '2016-09-04 21:15:19' and '2018-
10-17 17:30:18'
```



Assumption = Here, time range in 'where' clause is taken from the output of 1.B.

Insight = Total states in Brazil are 26 and federal district and Total municipalities are 5,564 . So, company got its reach in every state and the federal district. But the reach in municipalities is 4119, i.e.1445 municipalities are still left .

2)In-depth Exploration

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

```
with cte as (select extract(year from order_purchase_timestamp) as year,
count(*) as order_count
from `target.orders`
group by year)
select a.year,a.order_count, b.order_count AS prev_yr_order_count,
round(((a.order_count - b.order_count) / b.order_count)*100,2) AS
yearly_growth_percent
from cte a
left join cte b
on a.year = b.year + 1 order by a.year
```

Row	year	order_count	prev_yr_order_count	yearly_growth_percer
1	2016	329	*null*	*null*
2	2017	45101	329	13608.51
3	2018	54011	45101	19.76

Assumption = order_purchase_timestamp taken as the order places and counted as per required.

Insights = 1)The order count in year 2016 is comparatively low either because company started it's operation from September, 2016 or due to lack of previous data.

2)The company got a huge response from customers in 2017 with a growth of 13608.51%

3) in 2018, though there was a growth of 19.67% but it's pretty less compared to previous year.

Recommendation = In 2018, company may have retained it's customers from previous years but more new customers and municipalities need to be target to increase the order count.

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

```
with cte as(select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(*) as order_count
from `target.orders`
group by year,month)
select year, month, order_count,
lag(order_count) over(partition by year order by year, month) as
prev_month_order_count,
((order_count - lag(order_count) over(partition by year order by year, month))
/ lag(order_count) over(partition by year order by year, month))*100 as
monthly_growth_percent
from cte order by year, month
```

Query results

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

Row	year	month	order_count	prev_month_order_cc	monthly_growth_perc
1	2016	9	4	null	null
2	2016	10	324	4	8000.0
3	2016	12	1	324	-99.6913580246...
4	2017	1	800	null	null
5	2017	2	1780	800	122.5000000000...
6	2017	3	2682	1780	50.67415730337...
7	2017	4	2404	2682	-10.3653989560...
8	2017	5	3700	2404	53.91014975041...
9	2017	6	3245	3700	-12.2972972972...
10	2017	7	4026	3245	24.06779661016...
11	2017	8	4331	4026	7.575757575757...

Insight = 1)The orders count starts accelerating in March month and get peak in October and November as people preparing for Christmas and New year in December in 2017.

2) A seasonality in the orders can be seen between March and August in 2017 and 2018.

Recommendations= As per the above insights:

1)During High growth month promotions and special offers to be given.

2)During low growth month marketing strategies , issues with the customers reach and inventory management needs to be done

C. 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 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 day_time,
count(*) as order_count
from `target.orders`
group by day_time
order by order_count desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	day_time	order_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights= 1) The orders are more concentrated in Afternoon reflecting their shopping habits and routines. The orders placed during morning shows people take decision in night and orders in morning as there is not much difference in order count between night and day.

Recommendations = 1) The company should try to schedule promotions, customer engagement strategies and advertisements during afternoon and night.

2)Operational processes , customer services and order fulfilments to be effectively optimised during afternoon and night.

III. 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 order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
customer_state,
count(*) as order_count
from `target.orders`o
join `target.customers`c
on o.customer_id=c.customer_id
group by year, month, customer_state
order by year ,month, customer_state
```

Query results

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

Row	year	month	customer_state	order_count
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

Insights = 1)with the above query states with maximum order count and minimum order count will help with operational planning and marketing strategies based on customers behaviour change over time .

Recommendations =1) State specific promotions and marketing strategies to be made based on the regional preferences and demand.

2) Launching targeted campaigns and business strategies required at the states with low order count.

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

```
select customer_state,
count(distinct customer_id) as unique_customer_count
from `target.customers`
group by customer_state
order by unique_customer_count desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	unique_customer_co
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

Insights= Largest customer bases are in the states of SP , RJ and MG and lowest in RR ,AC ,AP.

Recommendations = 1) A good engagement strategies in states with large customer bases would be vital in customer satisfaction and loyalty in these states.

2)States with lower customer bases need to be explored for opportunities for growth and market penetration.

IV. Impact on Economy: Analyze the money movemen by e-commerce by looking at order prices, freight a others

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

```
with
cte as
(select extract(year from o.order_purchase_timestamp) as order_year,
extract(month from o.order_purchase_timestamp) as order_month,
sum(p.payment_value) total_order_cost
from `target.orders` o
join `target.payments` p
on o.order_id=p.order_id
where o.order_purchase_timestamp >=timestamp('2017-01-01') and
o.order_purchase_timestamp < timestamp('2018-09-01')
group by order_year,order_month),
cte2 as
(select order_year, order_month, total_order_cost as previous_year_cost
from cte
where order_year = 2017)
select c1.order_year, c1.order_month,
c1.total_order_cost,c2.previous_year_cost,
(round(((c1.total_order_cost - c2.previous_year_cost) /c2.previous_year_cost)
* 100, 2)) as percentage_increase
from cte as c1
join cte2 as c2
on c1.order_year = c2.order_year+1 and c1.order_month = c2.order_month
order by c1.order_year, c1.order_month
```

Query results

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

Row	order_year	order_month	total_order_cost	previous_year_cost	percentage_increase
1	2018	1	1115004.180000...	138488.0399999...	705.13
2	2018	2	992463.3400000...	291908.0099999...	239.99
3	2018	3	1159652.119999...	449863.6000000...	157.78
4	2018	4	1160785.479999...	417788.0300000...	177.84
5	2018	5	1153982.149999...	592918.8200000...	94.63
6	2018	6	1023880.499999...	511276.3800000...	100.26
7	2018	7	1066540.750000...	592382.9200000...	80.04
8	2018	8	1022425.320000...	674396.3200000...	51.61

Assumption= Here previous_year_cost is the cost of same month in previous year.

Insights = 1) The highest growth in order cost has been observed in January,2018 with growth of 705.13%, followed by February with growth of 239.99% which is quite impressive.

2) The lowest growth in order cost observed in August,2018 with growth of 51.61%.

3) Overall, there have been growth in the total_order_cost of 2018 as compared to 2017, which shows the marketing strategies and customer engagement is effectively operationalised.

Recommendations= 1) Adjusting business strategies, pricing and marketing efforts based on monthly growth to optimise revenue growth.

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

```
select customer_state,
sum(order_price) total_order_price,
round(avg(order_price),2) as average_order_price
from (select c.customer_state, p.order_id,
sum(p.payment_value) as order_price
from `target.customers` c
join `target.orders` o
```

```
on c.customer_id=o.customer_id
join `target.payments` p
on o.order_id = p.order_id
group by c.customer_state, p.order_id
)tbl
group by customer_state
order by total_order_price desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	total_order_price	average_order_price
1	SP	5998226.959999...	143.69
2	RJ	2144379.69	166.85
3	MG	1872257.260000...	160.92
4	RS	890898.5399999...	162.99
5	PR	811156.3799999...	160.78
6	SC	623086.4299999...	171.32
7	BA	616645.8200000...	182.44
8	DF	355141.0800000...	165.95
9	GO	350092.3099999...	173.31
10	ES	325967.5500000...	160.34

Insights= 1)The states with highest total_order_price and having lower Average_order_price shows that number of customers are quite large in these states but are placing lower value orders .

2) The states with higher Average_order_price and having lower total_order_price shows that these states have small number of customers but placing high value orders indicating affluent customer bases.

Recommendations= 1) States with lower Average_order_price reveals potential market opportunities for pricing adjustments, promotions and targeted marketing campaigns based on order price insights.

2)Efficient strategies required against the competitors in different states based on various states' order price .

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

```
select c.customer_state,
sum(oi.freight_value)as total_freight_value,
round(avg(oi.freight_value),2)as average_freight_value
from `target.customers` c
join`target.orders` o
on c.customer_id=o.customer_id
join `target.order_items` oi
on o.order_id=oi.order_id
group by c.customer_state
order by total_freight_value DESC;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	total_freight_value	average_freight_valu
1	SP	718723.0699999...	15.15
2	RJ	305589.3100000...	20.96
3	MG	270853.4600000...	20.63
4	RS	135522.7400000...	21.74
5	PR	117851.6800000...	20.53
6	BA	100156.6799999...	26.36
7	SC	89660.26000000...	21.47
8	PE	59449.65999999...	32.92
9	GO	53114.97999999...	22.77
10	DF	50625.49999999...	21.04

Insights= 1)The states with highest total_freight_value and having lower Average_freight_value shows that number of customers are quite large in these states and also the Freight_cost is low .

2) The states with higher Average_freight_value and having lower total_order_price indicating affluent customer bases but the freight_cost is more either due to distance from the supply chain or the varied freight_price slabs in these states.

Recommendations= 1)Regional pricing strategies are required to ensure that shipping costs are appropriately factored into product pricing .

2) Logistics , allocation of resources and optimised supply chain to be operationalised based of state's freight cost to manage customers' expectations regarding shipping fees.

V. 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.

```
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 diff_estimated_delivery
from `target.orders`
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_deliver	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

Assumptions= 1) Here negative sign in diff_estimated_delivery implies that order delvered before the expected delivery date. So, the more the negative value in diff_estimated_delivery , the more earlier the order delivered compared to estimated delivery time. For instance: In row 1, time_to_deliver=30 means it took 30 days to deliver the order and diff_estimated_delivery= -12, means it came 12 days earlier than estimated delivery

Insights = The diff_estimated_delivery indicates the accuracy of alignment between delivery time and estimated delivery , impacting customer satisfaction and helping the company to address any discrepancies .

Recommendations= 1) Optimise order fulfilment and delivery processes based on delayed deliveries and fast deliveries .

2)Regular analysis of delivery time and estimated vs actual differences to evaluate the effectiveness of delivery strategies and improvements over time.

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

```
with cte as(
select customer_state,
round(avg(freight_value),2) as avg_freight
from `target.customers` c
join `target.orders` o
on c.customer_id=o.customer_id
join `target.order_items` oi
on o.order_id=oi.order_id
group by customer_state),

top as(
select customer_state,avg_freight,
row_number() over(order by avg_freight desc) top5
from cte
limit 5),
```

```
bottom as(
select customer_state,avg_freight,
row_number() over (order by avg_freight) bottom5
from cte
limit 5)

select ts.customer_state as top_state,ts.avg_freight as
top_avg_freight,bs.customer_state as bottom_state,
bs.avg_freight as bottom_avg_freight
from top ts
JOIN bottom bs
ON ts.top5=bs.bottom5
```

Query results

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

Row	top_state	top_avg_freight	bottom_state	bottom_avg_freight
1	RR	42.98	SP	15.15
2	PB	42.72	PR	20.53
3	RO	41.07	MG	20.63
4	AC	40.07	RJ	20.96
5	PI	39.15	DF	21.04

Insights = Here, top_states are those having highest avg_freight_cost and bottom_states are those having lowest avg_freight_cost.

Recommendations= 1) Regional pricing strategies are required to ensure that shipping costs are appropriately factored into product pricing.

2) Logistics, allocation of resources and optimised supply chain to be operationalised based of state's freight cost to manage customers' expectations regarding shipping fees.

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

```
with cte as(
select customer_state,
round(avg(date_diff(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) as avg_delivery_time
from `target.customers` c
join `target.orders` o
on c.customer_id=o.customer_id
group by customer_state),

top as(
select customer_state,avg_delivery_time,
row_number() over(order by avg_delivery_time desc) top5
from cte
limit 5),

bottom as(
select customer_state,avg_delivery_time,
row_number() over (order by avg_delivery_time) bottom5
from cte
limit 5)

select ts.customer_state as Slow_delivery_states,ts.avg_delivery_time as
long_avg_delivery_days,bs.customer_state as fast_delivery_states,
bs.avg_delivery_time as short_avg_delivery_days
from top ts
join bottom bs
on ts.top5=bs.bottom5
```

Query results

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

Row	Slow_delivery_states	long_avg_delivery_da	fast_delivery_states	short_avg_delivery_d
1	RR	28.98	SP	8.3
2	AP	26.73	PR	11.53
3	AM	25.99	MG	11.54
4	AL	24.04	DF	12.51
5	PA	23.32	SC	14.48

Insights = Here, slow_delivery_states are those having long average delivery days and fast_delivery_states are those having short average delivery days.

Recommendations = Operational improvements in logistics and delivery processes by comparing top and bottom states based on delivery time will aid in optimising delivery time, market segmentation and positive customer satisfaction.

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

```
with cte as (
select customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,
day))- avg(date_diff(order_estimated_delivery_date,
order_delivered_customer_date, day)) as delivery_speed
from `target.customers` c
join `target.orders` o
on c.customer_id=o.customer_id
where order_delivered_customer_date is not null
group by customer_state)

select customer_state, delivery_speed
from cte
order by delivery_speed
limit 5
```

Query results

JOB INFORMATION | RESULTS | JSON | EXE

Row	customer_state	delivery_speed
1	SP	-1.83726385973...
2	PR	-0.83749746089...
3	MG	-0.75314839277...
4	RO	-0.21810699588...
5	AC	0.875000000000...

Insights= 1) Here, delivery_speed is in DAY format. Negative sign indicates faster delivery speed compared to estimated delivery. For instance = in row 1, delivery speed -1.83 indicates that the order was delivered 1.83 days before estimated delivery .

2)The above Top 5 states have the fastest delivery in respective order, exhibiting relatively high level of delivery efficiency and may have optimised logistics and delivery processes.

3)Fast delivery contribute to higher customer satisfaction and positive shopping experience .

Recommendations= 1)Insights from fast delivery states will be helpful for operational strategies to improve delivery time in other regions.

2)These insights of fast delivery states followed by slower delivery states will give competitive advantage in attracting and retaining customers in these regions.

VI. Analysis based on the payments:

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

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

Query results

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

Row	year	month	payment_type	order_count
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= 1) It's been observed that maximum customers use Credit card payment type and compared to all pretty few customers uses debit card.

Recommendations = 1) Adequate support and convenience of payment to be provided for customers using credit card and UPI payment.

2)The operational and marketing strategies involving gift vouchers or coupons to be based on credit card payment type to attract more customers.

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

```
with cte as
(select o.order_id,
max(p.payment_installments) as payment_installments
from `target.orders` o
join `target.payments` p
on o.order_id = p.order_id
where p.payment_installments > 0
group by o.order_id)

select payment_installments,
count(*) as order_count
from cte
group by payment_installments
order by payment_installments
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	payment_installment	order_count
1	1	48268
2	2	12363
3	3	10429
4	4	7070
5	5	5227
6	6	3908
7	7	1622
8	8	4251
9	9	644
10	10	5315

Insights= The above output shows the customer's financial preferences and needs.

Recommendations= 1) Customer support with efficient customer grievance redressal mechanism to be strategize based on the instalments usages .

2)Marketing strategies and promotions for customers who prefer installment payments.