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 scaler-dsml-12345.target90_sql.INFORMATION_SCHEMA.COLUMNS
where table_name='customers'
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

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: As we see,most of the Columns are of String Datatype.

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

Ans:

```
select min(order_purchase_timestamp) as first_order_date,
max(order_purchase_timestamp) as last_order_date
from `target90_sql.orders`
```

JOB INFORMATION | RESULTS | CHART PREVIEW

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

Insights: From the above table we find,the first order placed on 2016 and last on 2018.

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

Ans:

```
select count(distinct a.customer_city) as count_city

count(distinct a.customer_state) as count_state

from target90_sql.customers a
inner join target90_sql.orders b
on a.customer_id=b.customer_id
```

JOB INFORMATION | RESULTS

Row	count_city	count_state
1	4119	27

Insights:Total cities are 4119 and states are 27.

2.In-depth Exploration:

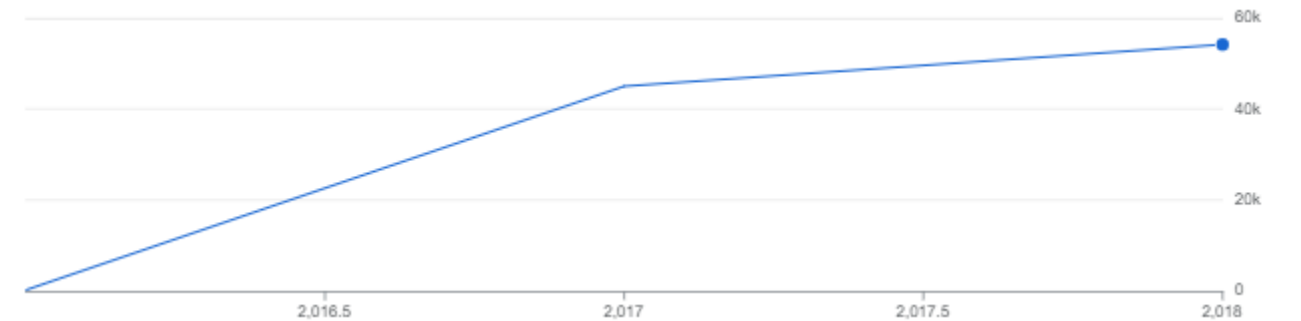
1. Is there a growing trend in the no. of orders placed over the past years?

Ans:

```
select extract(year from order_purchase_timestamp) as order_year,

count(distinct order_id)as No_of_orders
from `target90_sql.orders`
group by 1
order by 1
```

JOB INFORMATION RESULTS

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



Insights:Yes,Number of orders are increasing every year.

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

Ans:

```
select extract(month from order_purchase_timestamp) as M,

format_datetime('%B',order_purchase_timestamp) as order_month,
count(distinct order_id) as No_of_orders
from `target90_sql.orders`
group by 1,2
order by 1
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	M	order_month	No_of_orders
1	1	January	8069
2	2	February	8508
3	3	March	9893
4	4	April	9343
5	5	May	10573
6	6	June	9412
7	7	July	10318
8	8	August	10843
9	9	September	4305
10	10	October	4959
11	11	November	7544
12	12	December	5674

Insights:In May,July and August More than 10000 orders are placed.

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

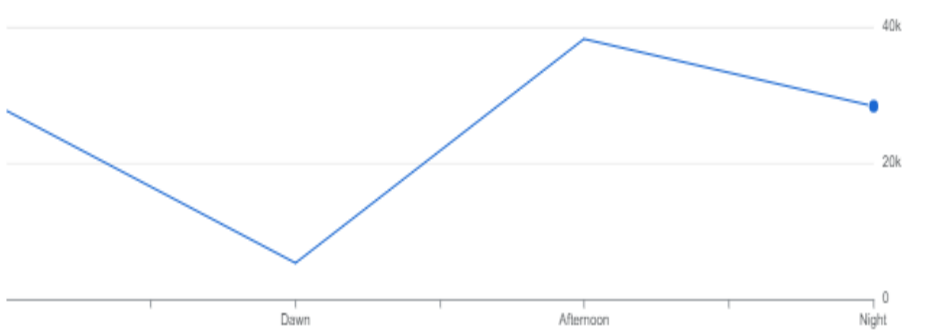
Ans:

```
select case when purchase_time>=0 and purchase_time<=6

    then 'Dawn'
    when purchase_time>=7 and purchase_time<=12
```

```
        then 'Morning'
        when purchase_time>=13 and purchase_time<=18
        then 'Afternoon'
        when purchase_time>=19 and purchase_time<=23
        then 'Night' end as Daytime,count(distinct customer_id) as No_of_customers
 from
(select *,extract(hour from order_purchase_timestamp) as purchase_time
from target90_sql.orders)
group by Daytime
```

JOB INFORMATION | RESULTS | CHART | PR

Row	Daytime	No_of_customers
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331



Insights:Brazilian customers mostly placed their order in the afternoon.

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

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

Ans:

```
with order_table as
(select customer_state,extract(month from order_purchase_timestamp)  as
month,count(order_id)as no_of_orders
from target90_sql.customers a
inner join target90_sql.orders b
on a.customer_id=b.customer_id
group by 1,2)
select *,
lead(no_of_orders)over(partition by customer_state order by month)-no_of_orders as
Month_on_Month_orders
from order_table
order by 1
```

Row	customer_state	month	no_of_orders	Month_on_Month_or
1	AC	1	8	-2
2	AC	2	6	-2
3	AC	3	4	5
4	AC	4	9	1
5	AC	5	10	-3
6	AC	6	7	2

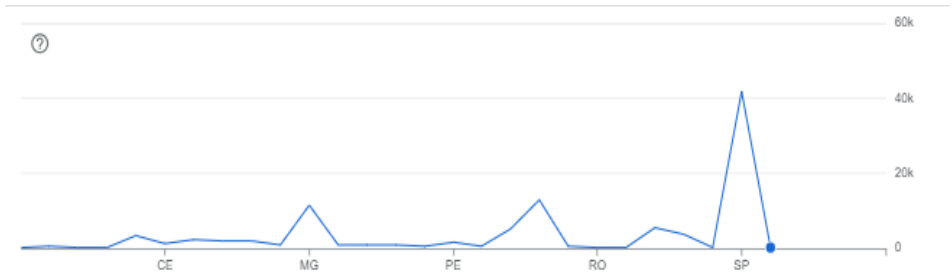
Insights: As we see,the month on month sales in every state are increasing or decreasing every month.

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

Ans:

```
select customer_state,count(distinct customer_id) as No_of_Customers
from `target90_sql.customers`
group by customer_state
order by customer_state
```

Row	customer_state	No_of_Customers
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336



Insights:From above query,we found that the maximum number customers are from state SP=41746 and minimum are from RR=46.

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).

Ans:

```
select *,((Total_cost_2018-Total_cost_2017)/Total_cost_2017)*100 as percent_increase

from
(select sum(case when format_datetime('%Y-%m',o.order_purchase_timestamp) between '2017-01' and '2017-08'
then p.payment_value end) as Total_cost_2017,
sum(case when format_datetime('%Y-%m',o.order_purchase_timestamp) between '2018-01' and '2018-08'
then p.payment_value end) as Total_cost_2018
from `target90_sql.orders` o
inner join target90_sql.payments p
on o.order_id=p.order_id)
```

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	Total_cost_2017	Total_cost_2018	percent_increase
1	3669022.12000...	8694733.83999...	136.976871646...

Insights:The cost of orders increases from 137% from year 2017 to 2018.

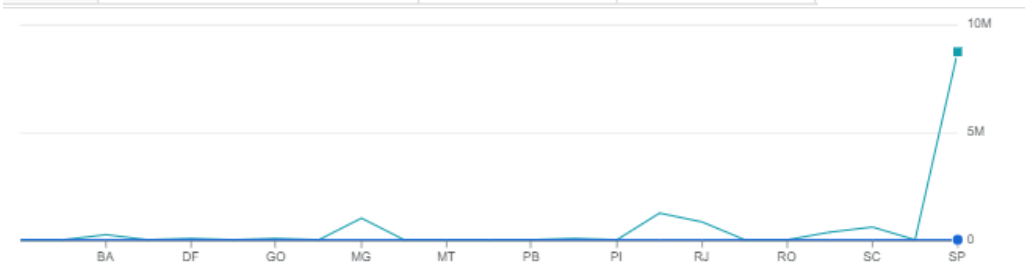
2.Calculate the Total & Average value of order price for each state.

Ans:

```
select s.seller_state as state,sum(price) as total,avg(price) as average

from `target90_sql.order_items` o
inner join `target90_sql.sellers` s
on o.seller_id=s.seller_id
group by 1
order by 1
```

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON

Row	state	total	average
1	AC	267.0	267.0
2	AM	1177.0	392.333333333...
3	BA	285561.559999...	444.108180404...
4	CE	20240.6400000...	215.325957446...
5	DF	97749.4799999...	108.731345939...
6	ES	47689.6100000...	128.197876344...



Insights:From the query,we found the maximum of total and average value Of order price are for state SP & PB AND minimum of total and average value Of order price are for state AC & MA.

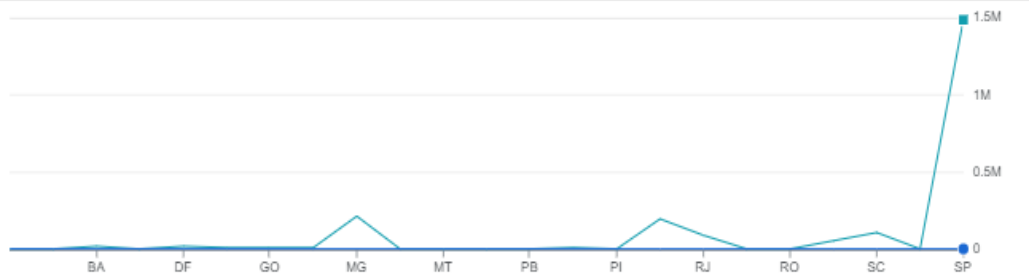
3.Calculate the Total & Average value of order freight for each state.

Ans.

```
select s.seller_state as state,sum(freight_value) as total,avg(freight_value) as average

from `target90_sql.order_items` o
inner join `target90_sql.sellers` s
on o.seller_id=s.seller_id
group by 1
order by 1
```

JOB INFORMATION | RESULTS | CHART | PREVIEW | JSON

Row	state	total	average
1	AC	32.84	32.84
2	AM	81.8	27.2666666666...
3	BA	19700.6800000...	30.6386936236...
4	CE	4359.83	46.3811702127...
5	DF	18494.0600000...	20.5718131256...
6	ES	12171.1300000...	32.7180913978...




```
Insights:From the query,we found the maximum of total and average value
        Of order freight are for state SP & RO AND minimum of total and
        average value Of order price are for state AC & SP.
```

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.

Ans:

```
select timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_deliver,

timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from target90_sql.orders
```

JOB INFORMATION RESULTS CHA

Row	time_to_deliver	diff_estimated_deliv
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	29	1

```
Insights: As we see the minimum value of time_to_deliver and diff_estimated_delivery is
          208 and -188 and maximum value of time_to_deliver and diff_estimated_delivery is
          209 d 146.
```

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

Ans:

```
select a.seller_state as highest,b.seller_state as lowest

from
(select s.seller_state,row_number()over(order by avg(oi.freight_value) desc ) as row_n
from `target90_sql.order_items` oi
inner join `target90_sql.sellers` s
on oi.seller_id=s.seller_id
group by 1) a
inner join
(select s.seller_state,row_number()over(order by avg(oi.freight_value) asc ) as row_n
from `target90_sql.order_items` oi
inner join `target90_sql.sellers` s
on oi.seller_id=s.seller_id
group by 1) b
on a.row_n=b.row_n
limit 5
```

JOB INFORMATION RESULTS CHART PREVIEW JS

Row	highest	lowest
1	RO	SP
2	CE	PA
3	PB	RJ
4	PI	DF
5	AC	PR

```
Insights:From the above table,we can see that the highest column are the states
         Whose average freight value is high and the lowest column are the states
         where average freight values are lowest than other states.
```

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

Ans:

```
select a.seller_state as highest_avg_delivery_time_state,

b.seller_state as lowest_avg_delivery_time_state
from
(select s.seller_state,
row_number()over(order by
avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)) desc ) as r
```

```
 from `target90_sql.orders` o
 inner join `target90_sql.order_items` oi
 on o.order_id=oi.order_id
inner join `target90_sql.sellers` s
on oi.seller_id=s.seller_id
group by 1) a
inner join
(select s.seller_state,
row_number()over(order by
avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day)) asc ) as r
from `target90_sql.orders` o
inner join `target90_sql.order_items` oi
on o.order_id=oi.order_id
inner join `target90_sql.sellers` s
on oi.seller_id=s.seller_id
group by 1) b
on a.r=b.r
limit 5
```

JOB INFORMATION | RESULTS | CHART | PREVIEW | JS

Row	highest_avg_delivery_time_state	lowest_avg_delivery_time_state
1	AM	AC
2	CE	RS
3	MA	RJ
4	RO	SP
5	MT	MS

```
Insights:From the above table,AM and AC are the states that have most highest and most
         Lowest average delivery time.
```

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

Ans:

```
select s.seller_state as state,

avg(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day)) as
comparison
from `target90_sql.orders` o
inner join `target90_sql.order_items` oi
on o.order_id=oi.order_id
inner join `target90_sql.sellers` s
on oi.seller_id=s.seller_id
group by 1
order by 2 desc
limit 5
```

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	state	comparison
1	RO	23.5
2	PB	18.8378378378...
3	MS	16.4599999999...
4	SE	16.3
5	RS	15.3702166897...

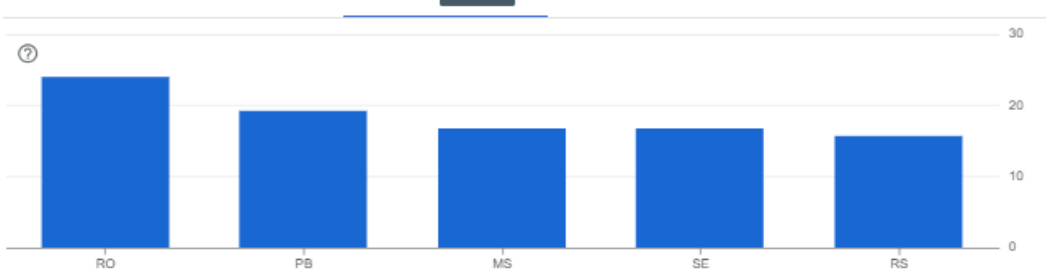
OR

```
 with table1 as

 (select
s.seller_state,avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day))
as time_to_deliver,
 avg(timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day)) as
estimated_delivery_time
 from `target90_sql.orders` o
 inner join `target90_sql.order_items` oi
 on o.order_id=oi.order_id
 inner join `target90_sql.sellers` s
 on oi.seller_id=s.seller_id
 group by 1)

 select seller_state as state,(estimated_delivery_time-time_to_deliver) as comparison
 from table1
 order by 2 desc
 limit 5
```

JOB INFORMATION | RESULTS | CHART | PREVIE

Row	state	comparison
1	RO	24.0
2	PB	19.1536273115...
3	MS	16.78
4	SE	16.7000000000...
5	RS	15.7522663702...

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



Insights:As we Found from above query that the top 5 States in which order Delivery is really fast as compared to the estimated date of delivery Are RO,PB,MS,SE and RS.

6.Analysis based on the payments:

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

Ans:

```
with pay_details as

(select p.payment_type,
extract(month from o.order_purchase_timestamp) as month,count(o.order_id) as No_of_orders,
from `target90_sql.orders` o
inner join `target90_sql.payments` p
on o.order_id=p.order_id
group by 1,2)

select *,
lead(No_of_orders)over(partition by payment_type order by month)  -No_of_orders as
month_on_month_orders
from pay_details
order by 1
```

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

Row	payment_type	month	No_of_orders	month_on_month_or
1	UPI	1	1715	8
2	UPI	2	1723	219
3	UPI	3	1942	-159
4	UPI	4	1783	252
5	UPI	5	2035	-228
6	UPI	6	1807	267

Insights:We found the payment are done using 4 different payment types-UPI,credit card,
Debit card and voucher and the highest number of orders=8350 are pay through credit card and
Lowest=43 through debit card.

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

Ans:

```
select count(order_id) as No_of_orders

from `target90_sql.payments`
where payment_installments=1
```

JOB INFORMATION | RESULTS

Row	No_of_orders
1	52546

Insights:No. of orders placed on the basis of the payment installments
that have been paid are 52546.