Q1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

1. Data type of columns in a table

1. data type of customer Table….example…and so on

Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

2. Time period for which the data is given

```
select max(order_purchase_timestamp) as maximum_timeperiod,min(order_purchase_timestamp) as minimum_t
imeperiod
from `target_case.orders`
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

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

3 Cities and States of customers ordered during the given period

```
select c.customer_state,c.customer_city

from `target_case.customers` c
join `target_case.orders` o
on c.customer_id=o.customer_id
where order_purchase_timestamp between "2016-09-04 21:15:19 UTC" and "2018-10-17 17:30:18 UTC"
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

Row	customer_state	customer_city
1	AL	maceio
2	SE	aracaju
3	SE	aracaju
4	AL	maceio
5	PI	teresina
6	AL	pau d'arco
7	RN	natal

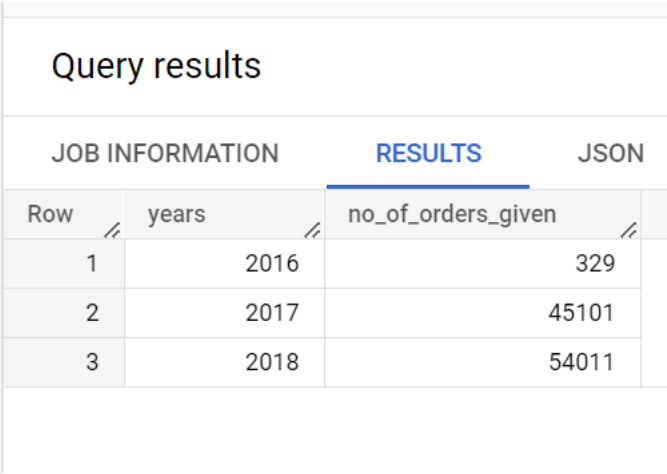
Q2. In-depth Exploration:

1.(a)Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?

```
with M as
(select order_id,extract(year from order_purchase_timestamp) as years
from target_case.orders)

select years,count(order_id) as no_of_orders_given
from M
```

```
group by years
order by years;
```



Query results

JOB INFORMATION | RESULTS | JSON

Row	years	no_of_orders_given
1	2016	329
2	2017	45101
3	2018	54011

→ From the above we can find the growth of e-commers as the no of orders growth….

1.(b)Can we see some seasonality with peaks at specific months?

```
with M as
(select order_id,
extract(year from order_purchase_timestamp) as years,
extract(month from order_purchase_timestamp) as months
from target_case.orders),

M1 as(select years,months,count(order_id) as no_of_orders_given
from M
group by years,months
order by years,months)


select *
from M1
;
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	years	months	no_of_orders_giv
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
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673

2.What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
with A as (select order_id,
extract(Time from order_purchase_timestamp) as times
from target_case.orders)
,
A1 as (select times,
case
when times between '06:00:00' and '12:00:00' then "Morning"
when times between '12:00:00' and '17:00:00' then "Afternoon"
when times between '17:00:00' and '22:00:00' then "Evening"
else "Night"
end as time_phase
from A)

select time_phase,count(times) as no_of_orders
from A1
group by time_phase;
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	time_phase	no_of_orders
1	Morning	22240
2	Night	14678
3	Evening	30311
4	Afternoon	32212

→Insights::Thus conclution is at Afternoon Brazil customer buy more…. Which is 32.3%(aprox)

Total =99441 no of orders

→Thus need to have more no of employees at this hour…

→Need to make back up ready at this time … back up as team to handle small technical issue.

→Need to have storage back up if needed…

Q3. Evolution of E-commerce orders in the Brazil region:

1.Get month on month orders by states

```
with A as
(select
c.customer_state,o.customer_id,
extract(year from order_purchase_timestamp) as years,
extract(month from order_purchase_timestamp) as month
from target_case.orders o
join target_case.customers c
on o.customer_id=c.customer_id)

select customer_state,years,month,count(customer_id) as no_of_orders
from A
group by customer_state,years,month
order by customer_state,years,month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTIO

Row	customer_state	years	month	no_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6

2.Distribution of customers across the states in Brazil

```
select customer_state,count(customer_id) as sign_in_cust

from target_case.customers
group by customer_state
```

Query results

JOB INFORMATION | RESULTS | JSON | EXE

ow	customer_state	sign_in_cust
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747
11	PE	1652
12	PB	536
13	ES	2033
14	PR	5045

Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

1.Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table

```
with A as (select p.payment_value,
extract(year from order_purchase_timestamp) as years,
extract(month from order_purchase_timestamp) as months
from `target_case.payments`  p
join `target_case.orders` o
on p.order_id=o.order_id)

select years,sum(payment_value) as  yearly_sum_
from A
where
(months between 1 and 8) and(years=2017 or years=2018)
group by years
order by years
```

Query results

JOB INFORMATION | RESULTS

Row	years	yearly_sum_
1	2017	3669022.11...
2	2018	8694733.83...

#insightes

→yearly sum Difference=yearly sum(2018)-yearly sum(2017)=**5025711.72**

→**%** increase =(5025711.72/8694733.83)*100=**57.8%(**approx.…)

2.Mean & Sum of price and freight value by customer state

```
select c.customer_state,avg(price) avg_of_price,
avg(freight_value) avg_of_freight,
sum(price) sum_of_price,
sum(freight_value) sum_of_freight
from `target_case.order_items` oi
join `target_case.orders` o
on oi.order_id=o.order_id
```

```
join `target_case.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by c.customer_state
```

Query results

SAVE RES

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

Row	customer_state	avg_of_price	avg_of_freight	sum_of_price	sum_of_freight
1	AC	173.727717...	40.0733695...	15982.9499...	3686.74999...
2	AL	180.889211...	35.8436711...	80314.81	15914.5899...
3	AM	135.495999...	33.2053939...	22356.8400...	5478.88999...
4	AP	164.320731...	34.0060975...	13474.2999...	2788.50000...
5	BA	134.601208...	26.3639589...	511349.990...	100156.679...
6	CE	153.758261...	32.7142016...	227254.709...	48351.5899...
7	DF	125.770548...	21.0413549...	302603.939...	50625.4999...
8	ES	121.913701...	22.0587765...	275037.309...	49764.5999...
9	GO	126.271731...	22.7668152...	294591.949...	53114.9799...
10	MA	145.204150...	38.2570024...	119648.219...	31523.7700...
11	MG	120.748574...	20.6301668...	1585308.02...	270853.460...
12	MS	142.628376...	23.3748840...	116812.639...	19144.0300...

Q5. Analysis on sales, freight and delivery time

Calculate days between purchasing, delivering and estimated delivery

```
-- Calculate days between purchasing, delivering
with A as
(select
extract(date from order_purchase_timestamp ) as purchasing,
extract(date from order_delivered_carrier_date ) as delivering,
extract(date from order_estimated_delivery_date) as estimated_delivery
from `target_case.orders`
)

select purchasing,delivering,date_diff(delivering,purchasing,day) day_bet_pur_del
from A
```

Row	purchasing	delivering	day_bet_pur_del
1	2017-02-04	2017-03-03	27
2	2018-07-11	2018-07-31	20
3	2018-03-22	2018-04-05	14
4	2018-01-19	2018-02-03	15
5	2016-10-07	2016-10-30	23
6	2016-10-05	2016-11-07	33
7	2016-10-07	2016-10-26	19
8	2016-10-05	2016-11-14	40
9	2018-03-22	2018-04-06	15
10	2017-11-24	2018-01-04	41
11	2018-05-18	2018-06-06	19

--days between estimated_delivery and delivering

```
with A as
(select
extract(date from order_purchase_timestamp ) as purchasing,
extract(date from order_delivered_carrier_date ) as delivering,
extract(date from order_estimated_delivery_date) as estimated_delivery
from `target_case.orders`
)

select delivering,estimated_delivery,date_diff(estimated_delivery,delivering,day) day_bet_del_est_del
from A
```

	delivering	estimated_delive	day_bet_del_est
1	2018-07-31	2018-08-01	1
2	2017-12-18	2018-01-29	42
3	2018-06-14	2018-07-24	40
4	2018-08-13	2018-08-17	4
5	2017-05-20	2017-06-27	38
6	2018-03-09	2018-04-19	41
7	2018-01-04	2017-12-20	-15
8	2018-08-06	2018-08-09	3
9	2018-08-06	2018-08-09	3
10	2018-05-18	2018-06-25	38
11	2018-07-04	2018-08-20	47

note that here + value of day_bet_del_est_del means deliver before estimated_delivery and -ve means it take extra days

#Insights...late delivery 336 and deliver before time 97185

Thus late_delivery is 0.345%....
#recomendation... which can we be consider good...but can be improved with few more employees...

- Find time_to_delivery & diff_estimated_delivery. Formula for the same given below

-- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

-- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
select
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
,date_diff(order_delivered_customer_date,order_purchase_timestamp,  day) as time_to_delivery
from `target_case.orders`
order by order_estimated_delivery_date ;
```

JOB INFORMATION | RESULTS | JSON

Row	diff_estimated_d	time_to_delivery
101	28	26
102	28	25
103	33	18
104	31	20
105	45	7
106	42	12
107	28	26
108	31	23
109	44	7
110	12	40
111	43	10

- Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
with diff as (select
order_id,
customer_id, order_estimated_delivery_date,order_delivered_customer_date,order_purchase_timestamp,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
,date_diff(order_delivered_customer_date,order_purchase_timestamp,  day) as time_to_delivery
from `target_case.orders`
)

select c.customer_state,avg(freight_value) as freight_value,avg(time_to_delivery) as time_to_delivery,avg(diff_es
timated_delivery) as diff_estimated_delivery
    from `target_case.customers` c
    join diff d
    on d.customer_id=c.customer_id
    join `target_case.order_items` oi
    on d.order_id=oi.order_id
```

```
group by c.customer_state;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTIO

Row	customer_state	freight_value	time_to_delivery	diff_estimated_c
1	MT	28.1662843...	17.5081967...	13.6393442...
2	MA	38.2570024...	21.2037500...	9.10999999...
3	AL	35.8436711...	23.9929742...	7.97658079...
4	SP	15.1472753...	8.25960855...	10.2655943...
5	MG	20.6301668...	11.5155221...	12.3971510...
6	PE	32.9178626...	17.7920962...	12.5521191...
7	RJ	20.9609239...	14.6893821...	11.1444931...
8	DF	21.0413549...	12.5014861...	11.2747346...
9	RS	21.7358043...	14.7082993...	13.2030001...
10	SE	36.6531688...	20.9786666...	9.16533333...
11	PR	20.5316515...	11.4807930...	12.5338998...

```
-- Sort the data to get the following:

-- Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
with diff as (select
order_id,
customer_id, order_estimated_delivery_date,order_delivered_customer_date,order_purchase_timestamp,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
,date_diff(order_delivered_customer_date,order_purchase_timestamp,  day) as time_to_delivery
from target_case.orders
)
,
A1 as(select c.customer_state,avg(freight_value) as freight_value,avg(time_to_delivery) as time_to_delivery,avg(
diff_estimated_delivery) as diff_estimated_delivery
    from `target_case.customers` c
    join diff d
    on d.customer_id=c.customer_id
    join `target_case.order_items` oi
    on d.order_id=oi.order_id
    group by c.customer_state)
# lowest 5 state with freight value
select customer_state,freight_value
from A1
order by A1.freight_value limit 5;
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	freight_value
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

```
#highest 5 state with freight value
select customer_state,freight_value
from A1
order by A1.freight_value desc
limit 5;
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	freight_value
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

```
-- Top 5 states with highest/lowest average time to delivery

with diff as (select
order_id,
customer_id, order_estimated_delivery_date,order_delivered_customer_date,order_purchase_timestamp,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
,date_diff(order_delivered_customer_date,order_purchase_timestamp,  day) as time_to_delivery
from target_case.orders
)
,
A1 as(select c.customer_state,avg(freight_value) as freight_value,avg(time_to_delivery) as time_to_delivery,avg(
diff_estimated_delivery) as diff_estimated_delivery
    from `target_case.customers` c
    join diff d
    on d.customer_id=c.customer_id
    join `target_case.order_items` oi
    on d.order_id=oi.order_id
    group by c.customer_state)
-- Top 5 states with lowest average time to delivery
select customer_state,time_to_delivery
from A1
order by A1.time_to_delivery
```

```
limit 5 ;
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	SP	8.25960855...
2	PR	11.4807930...
3	MG	11.5155221...
4	DF	12.5014861...
5	SC	14.5209858...

```
--Top 5 state with highest average time to delivery
select customer_state,time_to_delivery
from A1
order by A1.time_to_delivery desc
limit 5 ;
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	time_to_delivery
1	RR	27.8260869...
2	AP	27.7530864...
3	AM	25.9631901...
4	AL	23.9929742...
5	PA	23.3017077...

```
-- Top 5 states where delivery is really fast/ not so fast compared to estimated date\

with diff as (select
order_id,
customer_id, order_estimated_delivery_date,order_delivered_customer_date,order_purchase_timestamp,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
,date_diff(order_delivered_customer_date,order_purchase_timestamp,  day) as time_to_delivery
from target_case.orders
)
,
A1 as(select c.customer_state,avg(freight_value) as freight_value,avg(time_to_delivery) as time_to_delivery,avg(
diff_estimated_delivery) as diff_estimated_delivery
    from target_case.customers c
    join diff d
    on d.customer_id=c.customer_id
    join target_case.order_items oi
    on d.order_id=oi.order_id
    group by c.customer_state)
--Top 5 states where delivery is really not so fast compared to estimated date
select customer_state,diff_estimated_delivery
from A1
order by A1.diff_estimated_delivery
limit 5 ;
```

Query results

JOB INFORMATION | RESULTS | JSON

Row	customer_state	diff_estimated_c
1	AL	7.97658079...
2	MA	9.10999999...
3	SE	9.16533333...
4	ES	9.76853932...
5	BA	10.1194678...

```
--Top 5 states where delivery is really fast  compared to estimated date
select customer_state,diff_estimated_delivery
from A1
order by A1.diff_estimated_delivery desc
limit 5 ;
```

JOB INFORMATION | RESULTS | JSON | E

Row	customer_state	diff_estimated_c
1	AC	20.0109890...
2	RO	19.0805860...
3	AM	18.9754601...
4	AP	17.4444444...
5	RR	17.4347826...

Q6. Payment type analysis:

1.Month over Month count of orders for different payment types

```
with A as(select p.payment_type,o.order_id,extract(month from order_purchase_timestamp) as months,
extract(year from order_purchase_timestamp) as years
from `target_case.payments` p
join `target_case.orders` o
on p.order_id=o.order_id)

select payment_type,years,months,count(order_id) count_of_order
from A
group by payment_type,years,months
order by payment_type,years,months ;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

Row	payment_type	years	months	count_of_order
1	UPI	2016	0	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845

months

Results per pa

#insights…In 2016 ,2017,2018 orders given by (credit-card) are more then any other …..and followed by upi…

as upi order is 19784 and credit card order is 76795…thus upi order is just 25.76% of credit card

→ Recommendations… 1.Thus we can give offer to upi …to increase orders by upi…like 5% off using paytm etc…to attract youngsters with no credit card…….

2.Count of orders based on the no. of payment installments

```
select payment_installments,COUNT(order_id) as count_of_orders
from `target_case.payments`
group by payment_installments
```

JOB INFORMATION | RESULTS

Row	payment_installr	count_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626

#insights...select *from A where count_of_orders>10000 ;

JOB INFORMATION | RESULTS | JSON

Row	payment_installments	count_of_orders
1	1	52546
2	2	12413
3	3	10461

➔ Thus payment installments mostly used are 1,2,3

Some more insights …

```
select review_score,count(order_id) as count_order
from `target_case.order_reviews`
group by review_score
```

JOB INFORMATION | RESULTS

Row	review_score	count_order
1	1	11424
2	2	3151
3	3	8179
4	4	19142
5	5	57328

Total count order=99224 thus review score 1,2 which is worst are 11.51% and 3.17%
Which need to improve by decreasing rating 1 and 2

→Another problem is noo of comments
According to data we have

```
select count(*) as total_no,count(review_comment_title) actual_commented,count(*)-
count(review_comment_title) as not_comments
from `target_case.order_reviews`
```

JOB INFORMATION | RESULTS | JSON

Row	total_no	actual_comment	not_comments
1	99224	11549	87675

RECOMADATIONS:: Need to send special email to user how buy product but did not give review or comments … “SO THAT WE CAN IMPROVE OUR SERVICE”……