

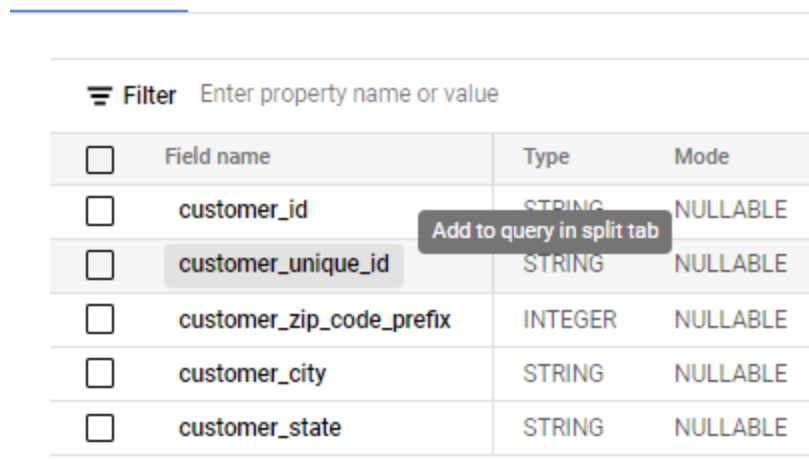
Business Case-----Target

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

Q1.1 Data type of all columns in the "customers" table.

Solution:-

By simple right click on customer table, it displays all type of columns with their datatype and constrain



<input type="checkbox"/>	Field name	Type	Mode
<input type="checkbox"/>	customer_id	STRING	NULLABLE
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE
<input type="checkbox"/>	customer_city	STRING	NULLABLE
<input type="checkbox"/>	customer_state	STRING	NULLABLE

Insights: - there are 5 columns in customer table with table name as

1. Customer_id
2. Customer_unique_id
3. customer_zip_code_prefix
4. customer_city
5. customer_state

Recommendation: - N/A

Q1.2 Get the time range between which the orders were placed.

Syntax:-

```
select min(order_purchase_timestamp) as First_order_date ,  
max(order_purchase_timestamp) as last_order_date  
from Target.orders
```

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

Insight: - The orders were place between the dates 4th of September 2016 to 17th of October 2018.

Recommendation: - N/A

Q1.3 Count the Cities & States of customers who ordered during the given period.

Syntax:-

```
select count(distinct customer_city) as no_of_cities ,
count(distinct customer_state) as no_of_states
from Target.customers as c left join `Target.orders` as o on c.customer_id = o.customer_id
```

Row	no_of_cities	no_of_states
1	4119	27

Insight: - There are 4119 cities and 27 states of customers who ordered during 4th of September 2016 to 17th of October 2018.

Recommendation: - N/A

2.In-depth Exploration

Q2.1 Is there a growing trend in the no. of orders placed over the past years?

Syntax: -

```
select extract(year from order_purchase_timestamp) as `year` , count(order_id) as  
NO_of_orders_per_year  
from Target.orders  
group by `year`  
order by `year` asc
```

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

Insight :-

Yes there is a growing trend in the no. of orders placed over the past years.

No. of orders in the year 2018 has already surpassed the No. of orders in 2017 with almost 2 and a half month to spare.

Recommendation:- N/A

Q2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Note:- Monthly seasonality has been divided on the basis of Quarter of the year i.e.

January	to	March	in	Quarter 1
April	to	June	in	Quarter 2
July	to	September	in	Quarter 3
October	to	December	in	Quarter 4

Solution: -

```
select extract(quarter from order_purchase_timestamp) as `Quarter` , count(order_id) as  
NO_of_orders_per_quarter  
from Target.orders  
group by `Quarter`  
order by `Quarter` asc
```

Row	Quarter	NO_of_orders_per_quarter
1	1	26470
2	2	29328
3	3	25466
4	4	18177

Order ordered month on month

```
select extract(month from order_purchase_timestamp) as `Month` , count(order_id) as
```

```
NO_of_orders_per_month
from Target.orders
group by `Month`
order by `Month` asc
```

Row	Month	NO_of_orders_per_month
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

Insight:-

- No. of orders seems to be falling down in the 4th quarter.
- There has been increase of orders from May to August Months
- No. of orders seems to be falling down from the month of September

Recommendation:-

- Seasonal offers shall be given to customer from the month of September to December.

Q2.3 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

Solution :-

Note :- 00:00:00 to 06:59:59 ---- Dawn
07:00:00 to 12:59:59 ---- Morning
13:00:00 to 18:59:59 ---- Dawn
19:00:00 to 23:59:59 ---- Dawn

Here time is divided in such a way that all order falls in one of the category, otherwise detail of 1638 nos. order would not group into any of the the category

```
select time_div , count(*) as `No_of_order` from
(select order_id , order_purchase_timestamp , extract(time from order_purchase_timestamp) as
`time_of_purchase` ,
( case
when extract(time from order_purchase_timestamp) between "19:00:00" and "23:59:59" then
"Night"
when extract(time from order_purchase_timestamp) between "00:00:00" and "06:59:59" then
"Dawn"
when extract(time from order_purchase_timestamp) between "07:00:00" and "12:59:59" then
"Morning"
when extract(time from order_purchase_timestamp) between "13:00:00" and "18:59:59" then
"Afternoon"
end
) as time_div
from Target.orders
order by time_of_purchase desc
)
group by time_div
order by `No_of_order` desc
```

Row	time_div	No_of_order
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insight :-

- **Most number orders have been placed in Afternoon that is between 13-18 hrs** followed by Night followed by morning and least order has been placed in Dawn.

Recommendation:-

- Order placement in dawn period that is between 0 hrs to 6 hrs is very small. Support system for placing of order can be blocked in this period for cost cutting.

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

Q3.1 Get the month on month no. of orders placed in each state.

Solution :-

```
select c.customer_state , extract(month from o.order_purchase_timestamp) as `Month`  
, count(distinct order_id) as `order_placed`  
from Target.customers as c  
left join  
Target.orders as o  
on c.customer_id = o.customer_id  
where o.order_purchase_timestamp is not null  
group by c.customer_state , `Month`  
order by c.customer_state , `Month`
```

Row	customer_state	Month	order_placed
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

Insight :-

```
select c.customer_state , count(*) as `order_placed`  
from Target.customers as c  
inner join  
Target.orders as o  
on c.customer_id = o.customer_id  
group by c.customer_state  
order by `order_placed` desc
```

Row	customer_state	order_placed
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

- Most order is placed by the State 'SP' which is more than 3 times more than the second highest order placed state that is 'RJ'.
- The most no. of order placed is in 'SP' in the month of 'August'.
- The second highest ordering placed state 'RJ' has most ordered placed in the month of 'August' also.
- Most no. of order placed is in the month of 'August' and least order placed is in the month of 'September' with order count as 12616 and 4021 respectively.

Recommendation:-

- As state 'RJ' has shown the potential of getting more orders from, Target can expand its business across the state.

Q3.2 How are the customers distributed across all the states?

Solution:-

```
select customer_state , count(distinct customer_unique_id) as No_of_customers
from Target.customers
group by customer_state
order by No_of_customers desc
```

Row	customer_state	No_of_customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

Insight :-

- State with most number of customer is 'SP' and least no. of Customer is in the state of 'RR'

Recommendation:-

- Advtisements and attractive offers can be provided in states with less no. of customers

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

Q4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only)

Solution :-

```
select d.*, ((d.Total_payment - d.diff)/d.diff)*100 as `%_increase` from
(
select t.*, lag(Total_payment , 1) over(order by Total_payment) as diff from
(
select extract(year from o.order_delivered_customer_date) as Years ,
round(sum(p.payment_value) , 2) as Total_payment
from
Target.orders as o
inner join
Target.payments as p
on o.order_id = p.order_id
where o.order_delivered_customer_date between "2017-01-01" and "2017-08-31" or
o.order_delivered_customer_date between "2018-01-01" and "2018-08-31"
group by Years
) as t
) as d
order by Total_payment asc
```

Row	Years	Total_payment	diff	%_increase
1	2017	3200049.21	null	null
2	2018	8848137.23	3200049.21	176.5000363853...

Insight: -

- **176.5% increase** in the cost of orders from January 2017 to August 2018 is seen. (This % is found out using ((Total payment in 2018 - Total payment in 2017) / Total payment in 2017)*100).
- There has been increase of cost of order Brazilian Real 5648087 from 2017 to 2018

Recommendation: - N/A

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

```
select c.customer_state , sum(p.payment_value) Total_cost , avg(p.payment_value) as avg_cost ,
from
Target.payments as p
left join
Target.orders as o
on p.order_id = o.order_id
right join
Target.customers as c
on o.customer_id = c.customer_id
group by c.customer_state
order by avg_cost desc
```

Row	customer_state	Total_cost	avg_cost
1	PB	141545.7199999...	248.3258245614...
2	AC	19680.619999999...	234.2930952380...
3	RO	60866.199999999...	233.2038314176...
4	AP	16262.799999999...	232.3257142857...
5	AL	96962.059999999...	227.0774238875...
6	RR	10064.620000000...	218.7960869565...
7	PA	218295.8500000...	215.9207220573...
8	SE	75246.25	208.4383656509...
9	PI	108523.9700000...	207.1068129770...

This table has been ordered in avg desc.

Insight:-

- State will highest cost per order is PB and highest order cost is SP.
- It is to be noted that the state will the most no. order placed and most cost of order has the lowest average payment value.

Recommendation:-

- Since our analysis suggest, even the most number of customer, orders and payment is received from the state 'SP', it has the lowest average payment. It is thus recommended to Target to apply a minimum shipping limit to the products or charge extra shipping cost in the state.
- In the state 'PB' average payment is highest which can give highest return. It is thus also suggested to target to expand their business to the 'PB'

Q4.3 Calculate the Total & Average value of order freight for each state.

Solution :-

```
select c.customer_state , round(sum(i.freight_value) , 2) as sum_of_freigh ,  
round(avg(i.freight_value) , 2) as avgerage_of_freigh  
from Target.order_items as i  
left join  
Target.orders as o  
on i.order_id = o.order_id  
right join  
Target.customers as c  
on o.customer_id = c.customer_id  
group by c.customer_state  
order by sum_of_freigh desc
```

Row	customer_state	sum_of_freigh	avgerage_of_freigh
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

The table is order in sum_of_freigh desc

Insight:-

- SP has the most value of freight, RR has the least value of freight.
- RR has the highest average of shipping where as SP has the lowest.

Recommendation:- It is recommended to Target to lower to increase its shipping charge with low freight average price and decrease its shipping charges with higher freight average price

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

Q5.1 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.

Solution:-

```
select order_id ,
timestamp_diff(order_delivered_customer_date , order_purchase_timestamp , day )
as time_to_deliver ,
timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date ,
day ) as diff_estimated_delivery
from `Target.orders`
```

Row	order_id	time_to_deliver	diff_estimated_delivery
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

Insight:-

Average Delivery and Delay Time

```
select avg(time_to_deliver) as average_deliver_time from
(
    select order_id ,
timestamp_diff(order_delivered_customer_date , order_purchase_timestamp , day ) as
time_to_deliver ,
timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date , day )
as diff_estimated_delivery
from `Target.orders`
)
```

Row	average_deliver_time
1	12.0

- Average day to deliver a order is 12 days

```
select round(avg(diff_estimated_delivery),0) as average_delay_time from
(
  select order_id ,
  timestamp_diff(order_delivered_customer_date , order_purchase_timestamp , day ) as
time_to_deliver ,
  timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date , day )
as diff_estimated_delivery
from `Target.orders`
)
```

Row	average_delay_time
1	-11.0

- Average delay Time is - 11 days. That means averagely orders are delivered 11 days before estimated day.

Recommendation:-

- As per our analysis, order is taking much less time than expected. Target could therefore decrease its estimated days for delivery for better user experience and it will also attract many other users.

1. Q5.2 Find out the top 5 states with the highest & lowest average freight value.

Top 5 States with Highest average freight value

Syntax:-

```
select * from
(
  select *, dense_rank() over(order by average_freight desc) as rankings from
(
  select c.customer_state , avg(i.freight_value) as average_freight
from
Target.order_items as i
left join
Target.orders as o
on i.order_id = o.order_id
```

```

right join
Target.customers as c
on o.customer_id = c.customer_id
group by c.customer_state
) as t
order by rankings
) as h
where rankings between 1 and 5

```

Row	customer_state	average_freight	rankings
1	RR	42.9844230769...	1
2	PB	42.7238039867...	2
3	RO	41.0697122302...	3
4	AC	40.0733695652...	4
5	PI	39.1479704797...	5

Top 5 States with lowest average freight value

Syntax:-

```

select * from
(
select *, dense_rank() over(order by average_freight asc) as rankings from
(
select c.customer_state , avg(i.freight_value) as average_freight
from
Target.order_items as i
left join
Target.orders as o
on i.order_id = o.order_id
right join
Target.customers as c
on o.customer_id = c.customer_id
group by c.customer_state
) as t
order by rankings
) as h
where rankings between 1 and 5

```

Row	customer_state	average_freight	rankings
1	SP	15.1472753904...	1
2	PR	20.5316515679...	2
3	MG	20.6301668063...	3
4	RJ	20.9609239316...	4
5	DF	21.0413549459...	5

Insights:-

- State with highest Freight value is RR followed by PB , RO , AC , PI whereas State with lowest Freight value is SP followed by PR , MG , RJ , DF

Recommendations:-

- It is to be noted that state with highest freight value has the lowest no. of orders and state with lowest freight value has the highest orders

1. Q5.3 Find out the top 5 states with the highest & lowest average delivery time.

Solution:-

Top 5 states with Highest average delivery time

Syntax:-

```
select * from
(
select * , dense_rank() over(order by `average delivery day` desc) as rankings from
(
select c.customer_state , avg(timestamp_diff(o.order_delivered_customer_date ,
o.order_purchase_timestamp , day)) as `average delivery day`
from
Target.orders as o
inner join
Target.customers as c
on o.customer_id = c.customer_id
group by c.customer_state
) as t
) as h
where rankings between 1 and 5
order by rankings
```

Row	customer_state	average delivery day	rankings
1	RR	28.97560975609...	1
2	AP	26.73134328358...	2
3	AM	25.98620689655...	3
4	AL	24.04030226700...	4
5	PA	23.31606765327...	5

Top 5 states with Lowest average delivery time

Syntax:-

```

Select * from
(
select * , dense_rank() over(order by `average delivery day` asc) as rankings from
(
select c.customer_state , avg(timestamp_diff(o.order_delivered_customer_date ,
o.order_purchase_timestamp , day)) as `average delivery day`
from
Target.orders as o
inner join
Target.customers as c
on o.customer_id = c.customer_id
group by c.customer_state
) as t
) as h
where rankings between 1 and 5
order by rankings

```

Row	customer_state	average delivery day	rankings
1	SP	8.298061489072...	1
2	PR	11.52671135486...	2
3	MG	11.54381329810...	3
4	DF	12.50913461538...	4
5	SC	14.47956019171...	5

Insight:-

- Top 5 states with Highest average delivery time
 1. RR
 2. AP
 3. AM
 4. AL
 5. PA
- Top 5 states with Lowest average delivery time
 1. SP
 2. PR
 3. MG
 4. DF
 5. SC

Recommendation:-

- It is to be noted that cost of freight is expensive , Average delivery time is very high and number of order is the least in the state of RR. It is thus advised to Target for improving the status in the state

Q5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Solution:-

```
select c.customer_state ,  
round(avg(timestamp_diff(order_estimated_delivery_date ,  
order_delivered_customer_date , day)),2) as avg_delivery_expected_day  
from Target.orders as o  
inner join  
Target.customers as c  
on o.customer_id = c.customer_id  
where order_status = "delivered"  
group by c.customer_state  
order by avg_delivery_expected_day desc
```

Row	customer_state	avg_delivery_expected_day
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Insight:-

- Top State where orders are delivered before estimated time are:-
 1. AC
 2. RO
 3. AP
 4. AM
 5. RR

Recommendation: - NA

6. Analysis based on the payments:

Q6.1 Find the month on month no. of orders placed using different payment types.

Solution:-

```
select extract(month from order_purchase_timestamp) as `months` , payment_type , count(*) as  
`count_of payments`  
from  
Target.orders as o  
inner join  
`Target.payments` as p  
on o.order_id = p.order_id  
group by months , payment_type  
order by months , `count_of payments`
```

Row	months	payment_type	count_of payments
1	1	debit_card	118
2	1	voucher	477
3	1	UPI	1715
4	1	credit_card	6103
5	2	debit_card	82
6	2	voucher	424
7	2	UPI	1723
8	2	credit_card	6609
9	3	debit_card	109
10	3	voucher	591

Insight:-

- There are 4 mode of transaction namely
 1. Debit Card
 2. Voucher
 3. UPI
 4. Credit Card

Row	months	payment_type	count_of payments
1	5	credit_card	8350
2	8	credit_card	8269
3	7	credit_card	7841
4	3	credit_card	7707
5	4	credit_card	7301
6	6	credit_card	7276
7	2	credit_card	6609
8	1	credit_card	6103
9	11	credit_card	5897
10	12	credit_card	4378

Table ordered by count_of_payment

- Most Payment is made using credit Card.
- Most Transaction is made on the month of May using credit Card

Row	months	payment_type	count_of payments
1	9	not_defined	1
2	8	not_defined	2
3	9	debit_card	43

- Least Transaction is made in the month of September using Debit Card

Recommendation: -

- With the growing trend, target should be prepared of accepting more payments from credit Card.

Q6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

Solution:-

```
select count(distinct order_id) as `count of order paid`
from Target.payments
where payment_installments >= 1
```

Row	count of order paid
1	99438

Insight: -

- There are 99438 number of orders that have been paid

Recommendation: - N/A