

TARGET SQL – BUSINESS CASE

Q NO-1) Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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

```
select column_name, data_type
from `employess-452419.target.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers'
```

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

→ There are 4 string data types and one integer data type in customers table

1b- GET THE RANGE BETWEEN WHICH THE ORDERS WERE PLACED.

```
select min(order_purchase_timestamp) as first_order,
       max(order_purchase_timestamp) as last_order
from `employess-452419.target.orders`
```

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

➔ The first ever order was placed on 4th September 2016 and last order was placed on 17th October 2018

1c- Count the Cities & States of customers who ordered during the given period.

```
select count(distinct customer_state) as number_of_states,
       count(distinct customer_city) as number_of_cities
from `employess-452419.target.customers`
```

Row	number_of_states	number_of_cities	
1	27	4119	

➔ Target has it's market over 27 states and 4119 cities

2Q) In-depth Exploration:

2a - Is there a growing trend in the no. of orders placed over the past years?

```
|
select extract(year from order_purchase_timestamp) as year, count(order_id) as no_of_orders
from `employess-452419.target.orders`
group by 1
order by no_of_orders
```

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

- ➔ We can clearly see that there is a growing trend in orders placed over past years. Here we can observe that from 2016 to 2017 numbers of orders took almost 150 times by the end of 2017.
- ➔ From 2017 to 2018 also number of orders placed were increased by 120%.

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

```
select extract(month from order_purchase_timestamp) as month, count(order_id) as number_of_orders
from `employess-452419.target.orders`
group by 1
order by month
```

Row	month	number_of_orders
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

- ➔ Here we can see from the start of the year there is a growth in number of orders placed by the following months till august. Then the number of orders placed for the 9th and 10th months decreases rapidly , this may cause because of the seasonal change as it is rainy season some customers prefer to stay home and only get things which are necessary and daily using products .

2c - 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 daily_hours, count(order_id) as no_of_orders,
from `employess-452419.target.orders`
group by daily_hours
order by no_of_orders
```

Row	daily_hours ▼	no_of_orders ▼
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

➔ It is observed that in dawn as it is probably sleeping time there are not many orders in that hour, but in the afternoon hours from 13 to 18 there is a huge increase in orders followed by night hours and morning hours

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

3a - Get the month on month no. of orders placed in each state.

```
select c.customer_state ,extract(month from o.order_purchase_timestamp) as months, count(o.order_id) as no_of_orders
from `employess-452419.target.orders` o
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1,2
order by no_of_orders desc
```

➔ We can clearly observe from the below the state coded SP has the highest count of orders in each month. Hence implementing more strategies and combo offers in SP state would also help in increasing more number of orders.

Row	customer_state ▼	months ▼	no_of_orders ▼
1	SP	8	4982
2	SP	5	4632
3	SP	7	4381
4	SP	6	4104
5	SP	3	4047
6	SP	4	3967
7	SP	2	3357
8	SP	1	3351
9	SP	11	3012
10	SP	12	2357
11	SP	10	1908
12	SP	9	1648

3b - How are the customers distributed across all the states?

```
select customer_state,count(distinct customer_id) as no_of_customers
from `employess-452419.target.customers`
group by 1
order by 2 desc
```

➔ Here we can see from the below table that SP state has highest of customers present who placed an order followed by RJ and MG. From the result table focusing on SP state's customers alone can increase the

number of orders by 30% to 40% . And it is also necessary increase the number of users in other states also.

Row	customer_state	no_of_customers
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

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

4a - 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.

```
--4a)
with year_sales as
(select extract(year from o.order_purchase_timestamp) as years, sum(p.payment_value) as total_amount,
    lag(sum(p.payment_value),1) over(order by sum(p.payment_value)) as prev_year_sales
from `employess-452419.target.orders` o
inner join `employess-452419.target.payments` p
on o.order_id = p.order_id
where extract(year from o.order_purchase_timestamp) between 2017 and 2018 and
    extract(month from o.order_purchase_timestamp) between 1 and 8
group by 1
order by 2)

select *, round((total_amount - prev_year_sales)/ prev_year_sales * 100, 2) as percentage
from year_sales
```

Row	years	total_amount	prev_year_sales	percentage
1	2017	3669022.119999...	null	null
2	2018	8694733.840000...	3669022.119999...	136.98

➔ From the obtained table it is observed that there is huge percentage increase of nearly 137% for cost of orders from 2017 to 2018 including months from January to august .

4b - Calculate the Total & Average value of order price for each state.

```
select c.customer_state, round(sum(p.payment_value),2) as total_amount,
    round(sum(p.payment_value)/ count(distinct o.order_id),2) as avg_amount
from `employess-452419.target.orders` o
inner join `employess-452419.target.payments` p
on o.order_id = p.order_id
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 desc
```

Row	customer_state ▼	total_amount ▼	avg_amount ▼
1	SP	5998226.96	143.69
2	RJ	2144379.69	166.85
3	MG	1872257.26	160.92
4	RS	890898.54	162.99
5	PR	811156.38	160.78
6	SC	623086.43	171.32
7	BA	616645.82	182.44
8	DF	355141.08	165.95
9	GO	350092.31	173.31
10	ES	325967.55	160.34

-→ From the above table it is observed that SP state has highest amount order price state wise but the average is lesser than the other states, one can presume that in this state more number orders being purchased with lesser amount while other states like GO having average cost of order as 182 , in this case number customers in this state are less but they are buying expensive products than the other states

4c - Calculate the Total & Average value of order freight for each state.

```
select c.customer_state, round(sum(i.freight_value),2) as total_amount,
       round(sum(i.freight_value)/ count(distinct o.order_id),2) as avg_amount
from `employess-452419.target.orders` o
inner join `employess-452419.target.order_items` i
on o.order_id = i.order_id
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 3 desc
```


Row	customer_state ▼	total_amount ▼	avg_amount ▼
1	RR	2235.19	48.59
2	PB	25719.73	48.35
3	RO	11417.38	46.22
4	AC	3686.75	45.52
5	PI	21218.2	43.04
6	MA	31523.77	42.6
7	TO	11732.68	42.05
8	AP	2788.5	41.01
9	SE	14111.47	40.9
10	PA	38699.3	39.9

➔ In RR and PB state the freight value is higher compared to other states, one can assume that here in these states the number of goods transported are less hence the heavy price when there are plenty of goods to carry the freight value decreases according to the business conditions. Look at SP state data which has shown highest amount of orders and cost of orders have the lowest freight value

Row	customer_state ▼	total_amount ▼	avg_amount ▼ ↑
1	SP	718723.07	17.37
2	MG	270853.46	23.46

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

5a - 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:

- $\text{time_to_deliver} = \text{order_delivered_customer_date} - \text{order_purchase_timestamp}$
- $\text{diff_estimated_delivery} = \text{order_delivered_customer_date} - \text{order_estimated_delivery_date}$

```
--5a)
select order_id,date_diff(timestamp(order_delivered_customer_date), timestamp(order_purchase_timestamp), day)as
time_to_deliver,
| date_diff(timestamp(order_estimated_delivery_date), timestamp(order_delivered_customer_date), day) as
diff_estimated_delivery
from `employess-452419.target.orders`
order by 2 desc
```

Row	order_id ▼	time_to_deliver ▼	diff_estimated_delivery ▼
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	285ab9426d6982034523a855f...	194	-166
5	0f4519c5f1c541ddec9f21b3bd...	194	-161
6	2fb597c2f772eca01b1f5c561b...	194	-155
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324feb907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	c27815f7e3dd0b926b5855262...	187	-162

➔ from the above table there were many instances that delivery took more than 6 months and there are some other instances where delivery was earlier than expected delivery time. This happens due to the different

business conditions. Having lesser difference estimated and actual delivery time will satisfy the customers most.

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

→ For the lowest average freight value

```
select c.customer_state, sum(i.freight_value)/count(distinct o.order_id) as avg_freight
from `employess-452419.target.orders` o
inner join `employess-452419.target.order_items` i
on o.order_id = i.order_id
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2
limit 5
```

Row	customer_state	avg_freight
1	SP	17.37095033232...
2	MG	23.46270443520...
3	PR	23.57976790716...
4	DF	23.82376470588...
5	RJ	23.94525231155...

→ For the highest average freight value:

```
select c.customer_state, sum(i.freight_value)/count(distinct o.order_id) as avg_freight
from `employess-452419.target.orders` o
inner join `employess-452419.target.order_items` i
on o.order_id = i.order_id
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5
```

Row	customer_state ▼	avg_freight ▼
1	RR	48.59108695652...
2	PB	48.34535714285...
3	RO	46.22421052631...
4	AC	45.51543209876...
5	PI	43.03894523326...

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

```
select c.customer_state, round(avg(date_diff(timestamp(order_delivered_customer_date), timestamp
(order_purchase_timestamp), day)), 2) as avg_delivery_time
from `employess-452419.target.orders` o
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2
limit 5
```

➔ The below states are the 5 states with lowest average of deliver time

Row	customer_state ▼	avg_delivery_time ▼
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

➔ The below states are top 5 states with highest average delivery time

```
select c.customer_state,round(avg(date_diff(timestamp(order_delivered_customer_date), timestamp
(order_purchase_timestamp), day)), 2) as avg_delivery_time
from `employess-452419.target.orders` o
inner join `employess-452419.target.customers` c
on o.customer_id = c.customer_id
group by 1
order by 2 desc
limit 5
```

Row	customer_state ▼	avg_delivery_time ▼
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

5d - 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.

```
--  
select c.customer_state, round(avg(date_diff(timestamp(order_estimated_delivery_date), timestamp  
(order_delivered_customer_date), day)), 2) as avg_delivery_time  
from `employess-452419.target.orders` o  
inner join `employess-452419.target.customers` c  
on o.customer_id = c.customer_id  
group by 1  
order by 2 desc  
limit 5
```

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

➔ AC, RO, AP, AM, RR are the top 5 states where delivery happens before the estimated delivery time

6Q) Analysis based on the payments

6a - Find the month on month no. of orders placed using different payment types.

```

select p.payment_type ,extract(month from o.order_purchase_timestamp) as months,
       count(o.order_id) as no_of_orders
from `employess-452419.target.orders` o
inner join `employess-452419.target.payments` p
on o.order_id = p.order_id
group by 1,2
order by months

```

Row	payment_type ▼	months ▼	no_of_orders ▼
1	credit_card	1	6103
2	UPI	1	1715
3	voucher	1	477
4	debit_card	1	118
5	credit_card	2	6609
6	UPI	2	1723
7	voucher	2	424
8	debit_card	2	82
9	credit_card	3	7707
10	UPI	3	1942
11	debit_card	3	109
12	voucher	3	591

➔ There were four different payment types used for placing an order they are UPI, voucher , credit_card and debit_card, by checking the number of orders monthly wise we can see that more number of orders are being placed by credit_card and less from the credit card

6b - Find the no. of orders placed on the basis of the payment installments that have been paid.

```
select payment_installments , count(order_id) as no_of_orders
from `employess-452419.target.payments`
group by 1
order by no_of_orders desc
```

Row	payment_installment	no_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644
11	12	133

- ➔ From the above table it is observed that number of orders are higher for lesser installments like 1 or 2. Here 1 installment means paying full at one instance and 2 is paying half in first installment and the rest in second one. It seems customer prefer 2 or 3 installments if not paying fully on purchase time.

Actionable Insights & Recommendations :

- ➔ From the count of orders as per state record it has been observed that SP state has highest number of orders overall , therefore focusing more

on this state like giving them reasonable offers and monthly subscription would make them engage more on target platform and orders will flow accordingly

- ➔ As per the payment type wise number of orders, it has been observed that number of orders using debit card are very less , here what we can offer is giving some discount on using specific debit cards will help the orders grow and also by partnering up with debit card companies will benefit Target more.
- ➔ The states RR , PB and RO has the highest freight value which will increase cost to company, here we can increase inventory so that freight value can reduced and that money can be used in marketing strategies.
- ➔ Implement the same strategies used in SP state for RR, AP and AC states will increase the sales more.