Name - Soniya Raviraj Ghatage
Business Case:Target SQL

A. 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.

Query:

```
select
  column_name,data_type
from `targetsbc`.target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name='customers';
```

Output:

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

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

Query:

```
select
min(order_purchase_timestamp) as `First_order`,
max(order_purchase_timestamp) as `last_order`,
DATE_DIFF(max(order_purchase_timestamp),
            min(order_purchase_timestamp) ,day) as `total_days`
from `targetsbc.target.orders`;
```

Output:

Row	First_order	last_order	total_days
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

Insight :

Here we can see, the first order was made at "2016-09-04 21:15:19 UTC" and last order was made at " 2018-10-17 17:30:18 UTC".date range between these Two timestamps is approximately " 772" days.

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

Query:

```
select
count(distinct customer_city) as `count_city`,
count(distinct customer_state) as `count_state`
from `target.customers` c
inner join `target.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";
```

Output :

Row	count_city	count_state
1	4119	27

Insight :

There are "4119" Cities and "27" States from where customers made orders during the given time period.

Recommendations:

- We can do some promotions or attend events in other cities.
- We can create web contents where more people would know about our business.

B. In-depth Exploration:

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

Query:

```
 select t.* ,
if(lead(`count of orders placed`) over(order by `year`,`month`)> `count of orders
placed`,"Yes","No") as `sales increased`
from
(
select extract(month from `order_purchase_timestamp`) as `month`,
        extract(year from `order_purchase_timestamp`) as `year`,
        count(order_id) as `count of orders placed`,
from `targetsbc.target.orders` as o
inner join `target.customers` c
on o.customer_id=c.customer_id
group by `month`,`year`
order by 2,1) as t
order by `year`,`month`;
```

Output:

Row	month	year	count of orders place	sales increased
1	9	2016	4	Yes
2	10	2016	324	No
3	12	2016	1	Yes
4	1	2017	800	Yes
5	2	2017	1780	Yes
6	3	2017	2682	No
7	4	2017	2404	Yes
8	5	2017	3700	No
9	6	2017	3245	Yes
10	7	2017	4026	Yes

Insight :

After 2016, sales can be seen increased.

In 2017, sales has been pretty good. In 11th month i.e November there is highest sales but in december the sales seemed to be fall down.

In 2018, though it increased in 1st month , there were some ups and downs . But after 3rd month the sales can be seen decreasing. (Note: “Sales increased ” column represents if the sales increased in the following 1 month or not.)

Recommendations:

- After November i.e festive season offer ended , we can give discount on less selling item so that we can clear the stock as well as increase the sales.
- If we want to increase the sales, we can give offers during festival's which might attract more customers.
- We can give out coupons which should be valid for at least 2 months , so that we could see some consistency in sales for 2-3 months.

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

Query :

```
select extract(month from `order_purchase_timestamp`) as `month`,
        count(distinct order_id) as `number of order placed`
from `targetsbc.target.orders`
group by `month`
order by 1
```

Output:

Row	month	number of order plac
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

Insight :

We can see monthly seasonality in number of orders being placed. From above we can see that most order are placed in August , May ,July. After August, there is decrease in orders placed. Though it increased in 11th month it decreased again at the end of the year.

Recommendations:

- We would need more product's stock so that we don't run out off products.
- We can also launch new products .
- During peak months of the year we can give offers on least selling products .

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**

Query :

```
select count(order_id) as `Number of orders`,
 if(extract(hour from `order_purchase_timestamp`) between 0 and 6,"Dawn",
              if(extract(hour from `order_purchase_timestamp`) between 7 and
12,"Morning",
              if(extract(hour from `order_purchase_timestamp`) between 13 and 18,
               "Afternoon","Night"
              )
              )
) as`time`
from `targetsbc.target.orders`
group by `time`
order by 1 desc
```

Output:

Row	Number of orders	time
1	38135	Afternoon
2	28331	Night
3	27733	Morning
4	5242	Dawn

Insight :
We can see that most of the orders are placed during Afternoon and Night. And least orders is placed at Dawn.

Recommendations:

- We can also increase price of few products during peak time on top selling products.
- We can launch new products during peak time period.

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

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

Query :

```
select count(ot.order_id) as `no of orders`,
extract(month from ot.order_purchase_timestamp) as `Month`,
ct.customer_state
from `target.orders` as ot
inner join `target.customers` ct
on ot.customer_id=ct.customer_id
group by 2,3
order by 3,2;
```

Output:

Row	no of orders	Month	customer_state
1	8	1	AC
2	6	2	AC
3	4	3	AC
4	9	4	AC
5	10	5	AC
6	7	6	AC
7	9	7	AC
8	7	8	AC
9	5	9	AC
10	6	10	AC

Insight :

These are the no. of orders placed in each state, in each month by our customers.We can compare which states have ordered more orders in a year.

'SP' state is highest selling state throughout the year. `RR`,`AP` are the states where sales are lowest for few months.

Recommendations:

- The highest selling state can be compared with lowest selling state and we can come up with ideas like what is the reason of high selling, what needs to be added to the lowest selling state and so on.
- Different states have different cultures based on that we can come up with Cultural theme promotion to attract more customers.

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

Query :

```
select
count(distinct customer_id) as `no of customers`,
customer_state
from `target.customers`
group by customer_state
```

```
order by 1 desc;
```

Output:

Row	no of customers	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO

Insight :

`SP` state has highest number of customers of approximately 41746 customers. While 'RR', 'AP', 'AC' states have less than 100 customers.

Recommendations:

- We can introduce our customer to "Refer and Earn" policy.
- We can ask the customers to give us feedback so that we can satisfy customers need.
- In states like 'RR', 'AP', 'AC' we can do more survey about what they need or what kind of product will suit the locals of that area.
- And , to keep the consistency in state like 'SP' or other state we can use offer spin-wheel like lucky draw .

D.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).

Query :

```
WITH first_year as
(
  select sum(`2017`) as `total_in_2017`,
  `year`
  from
  (
     select round(sum(p.payment_value)) `2017`,
     extract(month from `order_purchase_timestamp`) as `month`,
     extract(year from `order_purchase_timestamp`) as `year`
     from `target.payments` as p
     inner join `target.orders` o
     on p.order_id=o.order_id
     group by 2,3
  ) as a
  where `month` between 1 and 8 and
  `year` = 2017
  group by 2
),
second_year as
(
   select sum(`2018`) as `total_in_2018`,
   `year`
   from
   (
     select round(sum(p.payment_value)) `2018`,
     extract(month from `order_purchase_timestamp`) as `month`,
     extract(year from `order_purchase_timestamp`) as `year`
     from `target.payments` as p
     inner join `target.orders` o
     on p.order_id=o.order_id
     group by 2,3
    ) as b
   where `month` between 1 and 8 and
   `year` = 2018
   group by 2
)
select a1.total_in_2017,a2.total_in_2018,
```

```
round((a2.total_in_2018 - a1.total_in_2017)/a1.total_in_2017 * 100,2) as `Percent
increased`
from first_year a1,second_year a2;
```

Output:

Row	total_in_2017	total_in_2018	Percent increased
1	3669022.0	8694732.0	136.98

Insight :

We can see that the sales percent has been increased by 136.98% from year 2017 to 2018.

Recommendations:

- We can keep the same strategies used so far so that sales keeps rising in the upcoming year as well.

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

Query :

```
select round(sum(p.payment_value)) as `Total price`,
       round(avg(p.payment_value)) as `Avg price`,
       c.customer_state
from `target.payments` p
inner join `target.orders` o
on p.order_id=o.order_id
inner join `target.customers` c
on o.customer_id=c.customer_id
group by 3
order by 1,2 desc;
```

Output:

Row	Total price	Avg price	customer_state
1	10065.0	219.0	RR
2	16263.0	232.0	AP
3	19681.0	234.0	AC
4	27967.0	182.0	AM
5	60866.0	233.0	RO
6	61485.0	204.0	TO
7	75246.0	208.0	SE
8	96962.0	227.0	AL
9	102718.0	197.0	RN
10	108524.0	207.0	PI

Insight :

'SP' and 'RJ' state has highest total price of orders , while 'RR' and 'AP' has lowest total price of orders. Likewise, 'PB' and 'AC' states , 'RR' and 'SP' states are highest and lowest states with respect to average price of orders.

Recommendations:

- We can use the same strategies as the states where total and average price is high .
- We can focus more on the states where total and average price is low to increase the sales.

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

Query :

```
select round(sum(ot.freight_value)) as `Total freight_value`,
       round(avg(ot.freight_value)) as `Avg freight_value`,
       c.customer_state
from `target.order_items` as ot
inner join `target.orders` o
on ot.order_id=o.order_id
inner join `target.customers` c
on o.customer_id=c.customer_id
group by 3
```

```
order by 1 desc;
```

Output:

Row	Total freight_value	Avg freight_value	customer_state
1	718723.0	15.0	SP
2	305589.0	21.0	RJ
3	270853.0	21.0	MG
4	135523.0	22.0	RS
5	117852.0	21.0	PR
6	100157.0	26.0	BA
7	89660.0	21.0	SC
8	59450.0	33.0	PE
9	53115.0	23.0	GO
10	50625.0	21.0	DF

Insight :

'SP' and 'RJ' state has highest total freight of orders , while 'AP' and 'RR' has lowest total freight of orders. Likewise, 'PB' and 'RR' states , 'MG' and 'SP' states are highest and lowest states with respect to average freight of orders.

E.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.

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`
where `order_delivered_customer_date` is not null
order by 2,3;
```

Output:

Row	order_id	time_to_deliver	diff_estimated_delive
1	d5fbeedc85190ba88580d6f82...	0	7
2	79e324907160caea526fd8b94...	0	8
3	e65f1eeee1f52024ad1dcd034...	0	9
4	b70a8d75313560b4acf607739...	0	9
5	1d893dd7ca5f77ebf5f59f0d20...	0	10
6	d3ca7b82c922817b06e5ca211...	0	11
7	f3c6775ba3d2d9fe2826f93b71...	0	11
8	21a8ffca665bc7a1087d31751...	0	11
9	f349cdb62f69c3fae5c4d7d3f3...	0	12
10	38c1e3d4ed6a13cd0cf612d4c...	0	16

Insight :

From above output we can conclude that few orders were delivered on same day of purchase. And about approximately thousands of orders are delivered within 1-2 days which is a good sign but there are very few orders that took almost 200 days to be delivered.(from time_to_deliver column)

About the difference (in days) between the estimated & actual delivery date of an order , we can say that , some orders were delivered before 80-100 days and while some orders were delivered after 100 days.

(Note : negative days denotes delay, positive days denote before time)

Recommendations:

- Firstly, we need to make sure most of the orders should be delivered within estimated time.
- Secondly , once most of the orders are seen to be delivered within estimated time , we should try to reduce the estimated day from 200 to 180 and then 180 to 150 and so on .
- We should also increase the modes of transportation where customers or orders are high so that won't lose any customer for delaying.

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

Query :

```
(select
       round(avg(ot.freight_value)) as `Avg freight_value`,
       c.customer_state
from `target.order_items` as ot
inner join `target.orders` o
on ot.order_id=o.order_id
inner join `target.customers` c
on o.customer_id=c.customer_id
group by 2
order by 1 asc limit 5 offset 22)

UNION ALL

(select
       round(avg(ot.freight_value)) as `Avg freight_value`,
       c.customer_state
from `target.order_items` as ot
inner join `target.orders` o
on ot.order_id=o.order_id
inner join `target.customers` c
on o.customer_id=c.customer_id
group by 2
order by 1
limit 5 );
```

Output:

Row	Avg freight_value	customer_state
1	39.0	PI
2	40.0	AC
3	41.0	RO
4	43.0	PB
5	43.0	RR
6	15.0	SP
7	21.0	PR
8	21.0	RJ
9	21.0	DF
10	21.0	MG

Insight :

Above are the top 5 & the bottom 5 states arranged in increasing order of the average freight value.

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

Query :

```
(select
       round(avg(date_diff(`order_delivered_customer_date`,
`order_purchase_timestamp`, day))) as `Avg delivery`,
       c.customer_state
from `target.order_items` as ot
inner join `target.orders` o
on ot.order_id=o.order_id
inner join `target.customers` c
on o.customer_id=c.customer_id
```

```
where `order_delivered_customer_date` is not null
group by 2
order by 1 asc limit 5 offset 22)

UNION ALL

(select
       round(avg(date_diff(`order_delivered_customer_date`,
`order_purchase_timestamp`, day))) as `Avg delivery`,
       c.customer_state
from `target.order_items` as ot
inner join `target.orders` o
on ot.order_id=o.order_id
inner join `target.customers` c
on o.customer_id=c.customer_id
where `order_delivered_customer_date` is not null
group by 2
order by 1
limit 5 );
```

Output:

Row	Avg delivery	customer_state
1	23.0	PA
2	24.0	AL
3	26.0	AM
4	28.0	AP
5	28.0	RR
6	8.0	SP
7	11.0	PR
8	12.0	MG
9	13.0	DF
10	15.0	SC

Insight :

We want you to find the top 5 & the bottom 5 states arranged in increasing order of the average delivery time.

Recommendations:

- We can compare top and bottom states to improve our average delivery time.
- We should improve our delivery time in bottom 5 states so that we won't lose customers and to increase more customers.

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

Query :

```
select
  round(avg(date_diff(`order_estimated_delivery_date`,
`order_delivered_customer_date`, day)))
   as `avg_estimated_delivery`,
  c.customer_state
from `target.orders` as o
inner join `target.customers` c
on o.customer_id=c.customer_id
where `order_delivered_customer_date` is not null
group by 2
order by 1 desc limit 5;
```

Output:

Row	avg_estimated_delive	customer_state
1	20.0	AC
2	19.0	AM
3	19.0	RO
4	19.0	AP
5	16.0	RR

Insight :

These are the top 5 states where the average order delivery is really fast as compared to the estimated date of delivery .

Recommendations:

- We should see whether there are more number of transportations or see how these states manages to deliver before estimated time so that we can apply same methods on other states as well.

F.Analysis based on the payments:

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

Query :

```
select
  extract(year from o.order_purchase_timestamp) as `Year`,
  extract(month from o.order_purchase_timestamp) as `Month`,
  count(distinct(p.order_id)) as `no of orders`,
  p.payment_type
from `target.payments` as p
inner join `target.orders` as o
on p.order_id=o.order_id
group by 1,2,4
order by 3 desc;
```

Output:

Row	Year	Month	no of orders	payment_type
1	2017	11	5867	credit_card
2	2018	3	5674	credit_card
3	2018	1	5511	credit_card
4	2018	5	5475	credit_card
5	2018	4	5441	credit_card
6	2018	2	5235	credit_card
7	2018	8	4963	credit_card
8	2018	6	4796	credit_card
9	2018	7	4738	credit_card
10	2017	12	4363	credit_card

Insight :

We can see that most of the customers used credit card and UPI payment mode in 2017 , 2018. Whereas , debit card and vouchers are least used.

Recommendations:

- We can collab with banks so that we can give more offers to customers when they uses credit card. Like, 5% cashback on credit card.

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

Query :

```
select
count(distinct(o.order_id)) as `no of orders`,
  p.payment_installments
from `target.payments`  as p
inner join `target.orders` as o
on p.order_id=o.order_id
where p.payment_installments >=1
  and o.order_status != 'canceled'
group by 2
```

Output:

Row	no of orders	payment_installment
1	48732	1
2	12329	2
3	10374	3
4	7046	4
5	5204	5
6	3894	6
7	1617	7
8	4224	8
9	638	9
10	5279	10

Insight :

Most of the number of orders have done 2-3 payment instalments.