Business Case:
TARGET SQL

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
   a) Data type of all columns in the "customers" table.

QUERY

```
#1.1.
SELECT
TABLE_CATALOG,
TABLE_SCHEMA,
TABLE_NAME,
COLUMN_NAME,
DATA_TYPE
FROM `targetSQL.INFORMATION_SCHEMA.COLUMNS`
where TABLE_NAME = 'customers'
;
```

OUTPUT

Row	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
1	prefab-pixel-411412	targetSQL	customers	customer_id	STRING
2	prefab-pixel-411412	targetSQL	customers	customer_unique_id	STRING
3	prefab-pixel-411412	targetSQL	customers	customer_zip_code_prefix	INT64
4	prefab-pixel-411412	targetSQL	customers	customer_city	STRING
5	prefab-pixel-411412	targetSQL	customers	customer_state	STRING

INSIGHTS

- The customer table in the dataset contains information about ‘Target’s customers.
- The customers table is having five columns, namely customer_id, customer_unique_id, customer_zip_code_prefix,customer_city and customer_state
- All the data except customer_zip_code_prefix are of string datatype. customer_zip_code_prefix is integer datatype.

RECOMMENDATIONS

- The customer_zip_code_prefix information also could be stored as a string since no mathematical operations are performed on it, rather it acts as an identifier to the location of the customer.

b) Get the time range between which the orders were placed.

QUERY

```
#1.2.
SELECT
DATE(MIN(order_purchase_timestamp)) as first_date
,DATE(MAX(order_purchase_timestamp)) as last_date
FROM `targetSQL.orders`
;
```

OUTPUT

Row	first_date	last_date
1	2016-09-04	2018-10-17

INSIGHTS

- The date set contains the details of the orders received by 'Target' from customers from the date 04-09-2016 till the date 17-10-2018. i.e., span of two years, one month and thirteen days.

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

QUERY

```
#1.3
SELECT COUNT(distinct c.customer_city) as City_count
,COUNT(distinct c.customer_state) as State_count
FROM `targetSQL.orders` o
LEFT JOIN `targetSQL.customers` c
ON o.customer_id=c.customer_id
;
```

OUTPUT

Row	City_count	State_count
1	4119	27

INSIGHTS

- The customer pool of the company 'Target' spans over 4119 cities in 27 states in Brazil.

2. In-depth Exploration:

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

QUERY

```
#2.1
SELECT Year
,COUNT(order_id) as No_of_Orders
FROM(
SELECT *
,format_date('%Y',order_purchase_timestamp) as Year
FROM `targetSQL.orders`
)a
GROUP BY 1
ORDER BY 1
;
```

OUTPUT

Row	Year	No_of_Orders
1	2016	329
2	2017	45101
3	2018	54011

INSIGHTS

- There is a growing trend in the number of orders.
- From 04-09-2016, the year saw a number of 329 orders.
- In 2017 the company received 45,101 orders. Which is more than forty times than an extrapolated value for number of orders in 2016.
- Till 17-10-2018, the year saw a number of 54,011 orders from customers which is again showing an obvious growth. If the value is extrapolated for the complete year, it will be showing more than 50% growth from previous year.

Note:- Extrapolation calculation

2016 - from 04-09-2016 to the end of year – 118 days
hence, for full year, no of orders = 329/118*365 = 1017 orders
2018 – from start of year to 17-10-2018 is 289 days
hence, for full year, no of orders = 54,011/289*365 = 68,214 orders

RECOMMENDATIONS

- The growth profile of the company shows an exponential growth at the 2016-17 period and a great but reduced growth in 2017-18. It may indicate that the potency of the strategies will soon come to saturation. Implementation of newer marketing strategies recommended.

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

QUERY

```
#2.2
SELECT Month
,COUNT(order_id) as No_of_Orders
FROM(
  SELECT *
  ,format_date('%Y-%m',order_purchase_timestamp) as Month
  FROM `targetSQL.orders`
)a
GROUP BY 1
ORDER BY 1
;
```

OUTPUT

Row	Month	No_of_Orders
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026

INSIGHTS

- The monthly information about orders doesn't show any recurring tendency.
- The most number of orders received are on Nov-2017, Jan-2018,March-2017 respectively.
- The months Jan to April in 2018 and July to Nov in 2017 shows higher number of orders in the respective years.

- It is also observed that in months Nov-2016 and Oct-2018 have shown an exponential dip in the number of orders.

RECOMMENDATIONS

- The reason for abnormal reduction in orders in the above specified months should be looked into.
- The number of orders along months show a general growing trend which shall be tried to keep up.

c) 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

```
#2.3
SELECT Time_of_Day
,COUNT(order_id) as No_of_Orders
FROM(
  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 'Evening'
  when Extract(hour from order_purchase_timestamp) between 18 and 23 then 'Night'
  end as Time_of_Day
  FROM `targetSQL.orders`
)a
GROUP BY 1
ORDER BY 2 desc
;
```

OUTPUT

Row	Time_of_Day	No_of_Orders
1	Evening	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

INSIGHTS

- Evening is the time of the day when most Brazilian customers placed orders.
- Night and Morning times saw similar trend in the number of orders.
- Dawn is the least favourite time for Brazilian customers to place an order.

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

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

QUERY

```
#3.1
with Monthly as (
SELECT customer_state
,Month
,COUNT(order_id) as No_of_Orders
FROM(
  SELECT *
  ,format_date('%Y-%m',order_purchase_timestamp) as Month
  FROM `targetSQL.orders` o
  INNER JOIN  `targetSQL.customers` c
  ON o.customer_id=c.customer_id
)a
GROUP BY 1,2
)
SELECT customer_state,Month
,No_of_Orders-LAG (No_of_Orders) over (partition by customer_state order by Month) as M_O_M_orders
FROM Monthly
ORDER BY 1,2
;
```

OUTPUT

Row	customer_state	Month	M_O_M_orders
1	AC	2017-01	null
2	AC	2017-02	1
3	AC	2017-03	-1
4	AC	2017-04	3
5	AC	2017-05	3
6	AC	2017-06	-4
7	AC	2017-07	1
8	AC	2017-08	-1
9	AC	2017-09	1
10	AC	2017-10	1

Results per page: 50 1 – 50 of 565

INSIGHTS

- Every state shows a non-linear growth in the number of orders over the months.
- The Month-over-month variations show a non-consistent pattern of the years.
- Every state shows a dip towards the final month of the selected period.

RECOMMENDATIONS

- The variation in the number of orders shall be monitored and measures shall be taken for making the growth stable.
- The reasons behind the reductions in number of orders shall be investigated and taken care of.

b) How are the customers distributed across all the states?

QUERY

```
#3.2
SELECT customer_state
,COUNT(customer_id) as No_of_Customers
FROM `targetSQL.customers`
GROUP BY 1
ORDER BY 2 desc
;
```

OUTPUT

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

Results per page: 50 1 – 27 of 27

INSIGHTS

- The number of customers varies across the states with maximum number of 41,746 customers in state SP and least being 46 customers in state RR.

- Major share of customers is divided into SP, RJ and MG states.
- Three levels of popularity can be observed i.e., in ten-thousands- states like SP and RJ, in thousands- states like RS, BA and ES, in hundreds and below- states like RR, RN and SE

RECOMMENDATIONS

- Market penetration is not uniform across the states, market campaign strategies shall be devised to attract more customers from states like RR, AP, AC etc.

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

a) 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.

QUERY

```
#4.1
with Cost_table as(
SELECT o.Month,o.year
,ROUND(SUM(p.payment_value)) as Cost
FROM(
  SELECT *
  ,Extract(month from order_purchase_timestamp) as Month
  ,Extract(year from order_purchase_timestamp) as Year
  FROM `targetSQL.orders`
  WHERE Extract(year from order_purchase_timestamp) in (2017,2018)
  and Extract(Month from order_purchase_timestamp)<9
)o
INNER JOIN `targetSQL.payments` p
ON o.order_id=p.order_id
GROUP BY 1,2
ORDER BY 1,2
),
final_table as(
SELECT  s.Month as Month_Number
,s.Cost as Cost_in_2017
,e.Cost as Cost_in_2018
FROM(
  SELECT *
  FROM Cost_table
  WHERE Year=2017
)s
INNER JOIN
(
  SELECT *
  FROM Cost_table
  WHERE Year=2018
)e
ON s.Month=e.Month
ORDER BY 1
)
SELECT *
,ROUND((Cost_in_2018-Cost_in_2017)/Cost_in_2017*100,2)
as Percentage_Increase
FROM final_table
```

OUTPUT

Row	Month_Number	Cost_in_2017	Cost_in_2018	Percentage_Increase
1	1	138488.0	1115004.0	705.13
2	2	291908.0	992463.0	239.99
3	3	449864.0	1159652.0	157.78
4	4	417788.0	1160785.0	177.84
5	5	592919.0	1153982.0	94.63
6	6	511276.0	1023880.0	100.26
7	7	592383.0	1066541.0	80.04
8	8	674396.0	1022425.0	51.61

INSIGHTS

- The cost of orders increases multiple times from 2017 to 2018.
- The highest percentage increase is seen in January by 705.13 %.
- The least growth is seen in August by 51.61%.
- The change in percentage growth is contributed by growth of orders in 2017 and reduced rate of growth in orders in 2018.

RECOMMENDATIONS

- The growth in cost of orders is showing a decreasing trend when going through the year 2018. Promotion and market expansion strategies shall be implemented for increasing the growth percentage in the upcoming years.

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

QUERY

```
#4.2
SELECT c.customer_state
,ROUND(SUM(p.payment_value)) as Total_Order_Price
,ROUND(AVG(p.payment_value)) as Average_Order_Price
FROM `targetSQL.customers` c
INNER JOIN `targetSQL.orders` o
ON c.customer_id=o.customer_id
INNER JOIN `targetSQL.payments` p
ON o.order_id=p.order_id
Group BY 1
Order BY 1
```

OUTPUT

Row	customer_state	Total_Order_Price	Average_Order_Price
1	AC	19681.0	234.0
2	AL	96962.0	227.0
3	AM	27967.0	182.0
4	AP	16263.0	232.0
5	BA	616646.0	171.0
6	CE	279464.0	200.0
7	DF	355141.0	161.0
8	ES	325968.0	155.0
9	GO	350092.0	166.0
10	MA	152523.0	199.0

Results per page: 50 1 – 27 of 27

INSIGHTS

- The order price over the states varies both in overall and average aspects.
- The maximum and minimum values for total order price are from State SP and State RR respectively.
- The maximum and minimum values for average order price are from State PB and State SP respectively.
- Interestingly, State SP is first in total price and last in average price. It shows the state's large customer count and also indicates most of the orders are of lower price.
- Whereas, State RR is one of the States with high average price of orders. It shows that though the number of customers is lowest in RR (refer point no.3.b) the orders are of high price.

RECOMMENDATIONS

- The observation shows the variation of order price and also the difference in order price over different states.
- The penetration of low-value products into the states like RR is less. So, like noted in point 3.b), market penetration would be effective in these states concentrating on low value products.
- The states like SP, even though have much customer count, have lower number of high-value orders. The campaign strategies in such states should focus on to high-value products.

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

QUERY

```
SELECT c.customer_state
,ROUND(SUM(oi.freight_value)) as Total_Fright_Value
,ROUND(AVG(oi.freight_value)) as Average_Fright_Value
FROM `targetSQL.customers` c
INNER JOIN `targetSQL.orders` o
ON c.customer_id=o.customer_id
INNER JOIN `targetSQL.order_items` oi
ON o.order_id=oi.order_id
Group BY 1
Order BY 1
;
```

OUTPUT

Row	customer_state	Total_Fright_Value	Average_Fright_Value
1	AC	3687.0	40.0
2	AL	15915.0	36.0
3	AM	5479.0	33.0
4	AP	2789.0	34.0
5	BA	100157.0	26.0
6	CE	48352.0	33.0
7	DF	50625.0	21.0
8	ES	49765.0	22.0
9	GO	53115.0	23.0
10	MA	31524.0	38.0

Results per page: 50 1 – 27 of 27

INSIGHTS

- Total Fright Value varies from state to state with Maximum in State SP and minimum in State RR.
- Average fright value has maximum values in States PB and RR and minimum value in State SP.
- As per insights of 4.a), the orders to State SP are low-value products which are smaller in size and also large in number. Hence the freight value is higher totally and low in average.
- Similarly, high-value larger size orders from States RR and PB are less in number and hence the low total freight value and high average freight value.

RECOMMENDATIONS

- The penetration of small-size (hence low-freight value) products into the states like RR is less. Hence, like noted in point 3.b), market penetration would be effective in these states concentrating on small-size products.
- The states like SP, even though have much customer count, have lower number of large-size orders. The campaign strategies in such states should focus on to large-size products.

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

a) 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:

time_to_deliver = order_delivered_customer_date - order_purchase_timestamp

diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date

QUERY

```
#5.1
SELECT order_id
,date_diff(order_delivered_customer_date,order_purchase_timestamp ,day) as time_to_deliver
,date_diff(order_delivered_customer_date, order_estimated_delivery_date,day) as diff_estimated_delivery
FROM `targetSQL.orders`
WHERE order_delivered_customer_date is not null
;
```

OUTPUT

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

INSIGHTS

- The time to deliver varies from 201 days to less than a day.
- The difference in estimated delivery shows both positive values and negative values.
- Positive values of difference in estimated delivery shows the orders are received later than the estimated date.
- Negative values of difference in estimated delivery shows that the order is delivered before the estimated date.
- The difference in estimated delivery varies from -146 to 188 days.

RECOMMENDATIONS

- The delivery time taken for orders are showing an inconsistency.
- Even though delivering before expected date is an advantage, the estimation should be corrected to show near to real value for the estimated date of delivery. Hence the estimation algorithms should be corrected for precision.
- The latency in delivery is a major disadvantage for the company. These areas should be identified and delivery network should be enhanced for speedy deliveries in such areas.

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

QUERY

```
with  avg_freight_table as(
SELECT c.customer_state
,ROUND(AVG(oi.freight_value)) as Average_Freight_Value
FROM `targetSQL.customers` c
INNER JOIN `targetSQL.orders` o
ON c.customer_id=o.customer_id
INNER JOIN `targetSQL.order_items` oi
ON o.order_id=oi.order_id
Group BY 1
)
SELECT l.rnk as Rank
,l.customer_state as Lowest_Freight_Value_States
,h.customer_state as Highest_Freight_Value_States
FROM(
  SELECT *
  ,row_number() over(order by Average_Freight_Value) as rnk
  FROM avg_freight_table
)l
INNER JOIN (
  SELECT *
  ,row_number() over(order by Average_Freight_Value desc) as rnk
  FROM avg_freight_table
)h
ON l.rnk=h.rnk
WHERE l.rnk<6
;
```

OUTPUT

Row	Rank	Lowest_Freight_Value_States	Highest_Freight_Value_States
1	1	SP	PB
2	2	SC	RR
3	3	MG	RO
4	4	RJ	AC
5	5	PR	PI

INSIGHTS

- Lowest freight values are in states SP, SC, MG, RJ and PR
- Highest freight values are in states PB, RR, RO, AC and PI

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

QUERY

```
#5.3
With avg_delivery_table as(
SELECT c.customer_state
,AVG(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp ,day))
 as avg_delivery_time
FROM `targetSQL.orders` o
INNER JOIN `targetSQL.customers` c
ON o.customer_id=c.customer_id
WHERE o.order_delivered_customer_date is not null
GROUP BY 1
)
SELECT l.rnk as Rank
,l.customer_state as Lowest_Delivery_Time_States
,h.customer_state as Highest_Delivery_Time_States
FROM(
  SELECT *
  ,row_number() over(order by avg_delivery_time) as rnk
  FROM avg_delivery_table
)l
INNER JOIN (
  SELECT *
  ,row_number() over(order by avg_delivery_time desc) as rnk
  FROM avg_delivery_table
)h
ON l.rnk=h.rnk
WHERE l.rnk<6
;
```

OUTPUT

Row	Rank	Lowest_Delivery_Time_States	Highest_Delivery_Time_States
1	1	SP	RR
2	2	PR	AP
3	3	MG	AM
4	4	DF	AL
5	5	SC	PA

INSIGHTS

- Fastest delivery states are in states SP, PR, MG, DF and SC
- Longest delivery times are in states RR, AP, AM, AL and PA

RECOMMENDATIONS

- The delivery network in states like SP, PR, MG, DF and SC should be improved.

d) 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.

QUERY

```
#5.4
WITH average_table as(
SELECT c.customer_state
,AVG(date_diff(order_delivered_customer_date,order_purchase_timestamp ,day)) as avg_delivery_time
,AVG(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day)) as avg_est_time
FROM `targetSQL.orders` o
INNER JOIN `targetSQL.customers` c
ON o.customer_id=c.customer_id
WHERE o.order_delivered_customer_date is not null
GROUP BY 1
)
SELECT customer_state
,ROUND(avg_delivery_time-avg_est_time,1) as diff_of_avg_days
FROM average_table
ORDER BY 2
LIMIT 5
;
```

OUTPUT

Row	customer_state	diff_of_avg_days
1	AC	-20.1
2	RO	-19.5
3	AP	-19.1
4	AM	-18.9
5	RR	-16.7

INSIGHTS

- The States with fastest delivery as compared to the estimated delivery time are States AC, RO, AP, AM and RR.
- The negative values shows that the order got delivered before the estimated time in an average.

RECOMMENDATIONS

- The delivery network on the slower states and the delivery time estimation should be improved.

6. Analysis based on the payments:

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

QUERY

```
#6.1
WITH Count_table as(
SELECT p.payment_type
,format_date('%Y-%m',order_purchase_timestamp) as Month
,COUNT(o.order_id) as order_count
FROM `targetSQL.orders` o
INNER JOIN `targetSQL.payments` p
ON o.order_id=p.order_id
GROUP BY 1,2
)
SELECT payment_type,Month,order_count
,order_count-LAG(order_count) over(partition by payment_type order by Month)
 as M_O_M_order_count
FROM Count_table
;
```

OUTPUT

Row	payment_type	Month	order_count	M_O_M_order_count
1	debit_card	2016-10	2	null
2	debit_card	2017-01	9	7
3	debit_card	2017-02	13	4
4	debit_card	2017-03	31	18
5	debit_card	2017-04	27	-4
6	debit_card	2017-05	30	3
7	debit_card	2017-06	27	-3
8	debit_card	2017-07	22	-5
9	debit_card	2017-08	34	12
10	debit_card	2017-09	43	9

INSIGHTS

- Most of the people use either voucher or UPI for transaction.
- The month-over-month variation of the number of orders per different payment types is non-consistent.

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

QUERY

```
#6.2
SELECT payment_installments
,COUNT(order_id) as No_of_Orders
FROM `targetSQL.payments`
GROUP BY 1
ORDER BY 1
;
```

OUTPUT

Row	payment_installment	No_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
9	8	4268
10	9	644

Results per page: 50 1 – 24 of 24

INSIGHTS

- Most number of people paid the orders in one instalment.
- The number of people availed higher number of instalments is on a decreasing trend overall.