BUSINESS ANALYSIS REPORT

OVERVIEW:

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews. By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

1. **Exploratory analysis steps like checking the structure & characteristics of the dataset:**

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

select the “customers” table and choose the information_schema to get the data types of all columns.

Result:

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default value	Policy tags	Description
customer_id	STRING	NULLABLE	-	-	-	-	-
customer_unique_id	STRING	NULLABLE	-	-	-	-	-
customer_zip_code_prefix	INTEGER	NULLABLE	-	-	-	-	-
customer_city	STRING	NULLABLE	-	-	-	-	-
customer_state	STRING	NULLABLE	-	-	-	-	-

Insight:

The customer table contains details of customer that are represented by five columns. The columns fields are customer_id, customer_unique_id, customer_city, customer_state with data type of 'string' and customer_zip_code_prefix with data type of 'integer'.

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

```
select order_purchase_timestamp from mp_.orders;
```

Result:

Row	order_purchase_timestamp
1	2017-11-25 11:10:33 UTC
2	2017-12-05 01:07:58 UTC
3	2017-12-05 01:07:52 UTC
4	2018-02-09 17:21:04 UTC
5	2017-11-06 13:12:34 UTC
6	2017-04-20 12:45:34 UTC
7	2017-07-13 11:03:05 UTC
8	2017-07-11 13:36:30 UTC
9	2017-07-29 18:05:07 UTC
10	2017-07-13 10:02:47 UTC
11	2017-07-19 12:44:59 UTC
12	2018-05-11 18:24:01 UTC
13	2018-05-20 18:58:04 UTC

Insight:

The field ‘ order_purchase_timestamp’ gives the date and time of all orders that are placed by the customers in the time range given from the dataset.

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

```
select count(distinct customer_city ) as cities_count,count(distinct customer_state ) as state_count from mp_.customers;
```

Result:

Row	cities_count	state_count
1	4119	27

Insight:

From the above result, we clearly cited that our customers are from 27 states of ‘Brazil’ which includes about 4119 cities of the country. This shows that our product went widespread all over the country and gain a huge amount of customers.

2. In-depth Exploration:

1. 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 `mp_.orders` group by year_  order by year_ desc ;
```

Result:

year_	no_of_orders
2018	54011
2017	45101
2016	329

Insight:

From year 2016 to 2017, there is a rapid growth in orders as well as in 2018 the growth is significant and consistent. It clearly shows that number of orders placing in each year is growing tremendously. We have to make this trend consistent for the upcoming years by doing the right things.

2. 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 no_of_orders from `mp_.orders` group by month_  order by month_
desc ;
```

Result:

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

Insight:

From the above observed data, we find that the maximum order was placed in 8th month which is august is 10843 orders. There are three months in a year where number of orders exceeds 10,000 .Except two months ,the orders placed in every month are above 5000.Almost throughout the year, the number of orders placed are consistent and the figures are good.

3. During what time of the day, do the Brazilian customers mostly place their orders ? (Dawn, Morning, Afternoon or Night)

```
select

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',

if (extract(hour from order_purchase_timestamp ) between 19 and 23, 'night','other'))))) as
period,

count(order_id) as order_count

from mp_.orders  group by period order by order_count desc;
```

Result:

period	order_count
afternoon	38135
night	28331
morning	27733
dawn	5242

Insight:

It clearly tells the maximum orders are placed in the afternoon of a day. We find that the Brazilians do not prefer to make orders in dawn. We recommend our client to push the advertisements more on afternoon and based on the period of day to make Brazilian people order more.

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

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

```
 with monthonmonth_orders as(

select c.customer_state, extract(month from o.order_purchase_timestamp) as month_,
count(o.order_id) as no_of_orders

from mp_.customers c  join mp_.orders o on c.customer_id=o.customer_id

group by c.customer_state,month_  order by c.customer_state ,month_)

select customer_state, month_, no_of_orders, lag(no_of_orders)over(partition by
customer_state  order by month_) as previousmonth_orders, no_of_orders-
```

lag(no_of_orders)over(partition by customer_state order by month_) as month_on_month_difference

from monthonmonth_orders order by customer_state,month_

Result:

Row	customer_state	month_	no_of_orders	previousmonth_orde	month_on_month_dif
1	AC	1	8	*null*	*null*
2	AC	2	6	8	-2
3	AC	3	4	6	-2
4	AC	4	9	4	5
5	AC	5	10	9	1
6	AC	6	7	10	-3
7	AC	7	9	7	2
8	AC	8	7	9	-2
9	AC	9	5	7	-2
10	AC	10	6	5	1
11	AC	11	5	6	-1
12	AC	12	5	5	0
13	AL	1	39	*null*	*null*

Insight:

We provide the data for number of orders placed in each month in every state of the country. Also we provide some additional information on month on month comparison of orders.

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

select customer_state, count(customer_id) as no_of_customers from mp_.customers group by customer_state order by no_of_customers desc

Result:

	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
11	PE	1652
12	CE	1336
13	PA	975
14	MT	907

Insight:

It shows the number of customers from various states of Brazil. We find that the maximum number of customers are from the state 'SP'. It also tells the truth that the customers from other states are comparatively very lower. We need to focus on all other states to increase the number of customers which further results in increase in orders.

4. 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). You can use the "payment_value" column in the payments table to get the cost of orders:

```
with cost_percentage as

(with cost_of_orders_ as

(select extract(year from o.order_purchase_timestamp) as year_, extract(month from
o.order_purchase_timestamp) as month_,

sum(p.payment_value) as cost_of_orders

from mp_.orders o join mp_.payments p on o.order_id=p.order_id

where extract(month from o.order_purchase_timestamp) between 1 and 7

and extract(year from o.order_purchase_timestamp) in (2018,2017)
```

```
group by year_,month_  order by year_ desc,month_ asc)

select year_,sum(cost_of_orders) as totalcost_of_orders,

from cost_of_orders_ group by year_)

select year_, totalcost_of_orders, lead(totalcost_of_orders)over(order by year_ desc)as
prev_year_orders, ((totalcost_of_orders-(lead(totalcost_of_orders)over(order by year_
desc)))/(lead(totalcost_of_orders)over(order by year_ desc))*100) as increase_percentage

from cost_percentage order by year_ desc
```

Result:

Row	year_	totalcost_of_orders	prev_year_orders	increase_percentage
1	2018	7672308.519999...	2994625.799999...	156.2025786326...
2	2017	2994625.799999...	*null*	*null*

Insight:

From the above data, We get year wise total cost of the orders between the month of January and August for the years 2017,2018. Also, it shows the percentage increase of total cost of orders. There is a rapid increase percentage of 156 from year 2017 to 2018.

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

```
select c.customer_state, avg(oi.price) as avg_value_orderprice, sum(oi.price) as
Total_value_orderprice  from mp_.order_items oi left join mp_.orders o

on oi.order_id=o.order_id join mp_.customers c on o.customer_id = c.customer_id

group by c.customer_state  order by c.customer_state
```

Result:

Row	customer_state	avg_value_orderprice	Total_value_orderpric
1	AC	173.7277173913...	15982.94999999...
2	AL	180.8892117117...	80314.81
3	AM	135.4959999999...	22356.84000000...
4	AP	164.3207317073...	13474.29999999...
5	BA	134.6012082126...	511349.9900000...
6	CE	153.7582611637...	227254.7099999...
7	DF	125.7705486284...	302603.9399999...
8	ES	121.9137012411...	275037.3099999...
9	GO	126.2717316759...	294591.9499999...
10	MA	145.2041504854...	119648.2199999...
11	MG	120.7485741488...	1585308.029999...
12	MS	142.6283760683...	116812.6399999...
13	MT	148.2971848341...	156453.5299999...

Insight:

It gives the information on average of order price and total value of order price of each state of the country.

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

```
select c.customer_state, avg(oi.freight_value) as avg_value_freightprice,
sum(oi.freight_value) as Total_value_freightprice

from mp_.order_items oi left join mp_.orders o on oi.order_id=o.order_id

join mp_.customers c  on o.customer_id = c.customer_id

group by c.customer_state order by c.customer_state
```

Result:

Row	customer_state	avg_value_freightpric	Total_value_freightpr
1	AC	40.07336956521...	3686.750000000...
2	AL	35.84367117117...	15914.58999999...
3	AM	33.20539393939...	5478.890000000...
4	AP	34.00609756097...	2788.500000000...
5	BA	26.36395893656...	100156.6799999...
6	CE	32.71420162381...	48351.58999999...
7	DF	21.04135494596...	50625.49999999...
8	ES	22.05877659574...	49764.59999999...
9	GO	22.76681525932...	53114.97999999...
10	MA	38.25700242718...	31523.77000000...
11	MG	20.63016680630...	270853.4600000...
12	MS	23.37488400488...	19144.03000000...
13	MT	28.16628436018...	29715.43000000...

Insight:

It gives the information on average of freight price and total value of freight value of all states.

5. 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:

```
select order_id, customer_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp, day) as
delivery_days,

timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date, day) as
diff_from_estimatedelivery

from mp_.orders
```

Result:

Row	order_id	customer_id	delivery_days	diff_from_estimated
1	1950d777989f6a877539f5379...	1bccb206de9f0f25adc6871a1...	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	de4caa97afa80c8eeac2ff4c8d...	30	-28
3	65d1e226dfaeb8cdc42f66542...	70fc57eeae292675927697fe0...	35	-16
4	635c894d068ac37e6e03dc54e...	7a34a8e890765ad6f90db76d0...	30	-1
5	3b97562c3aee8bdedcb5c2e45...	065d53860347d845788e041c...	32	0
6	68f47f50f04c4cb6774570cfde...	0378e1381c730d4504ebc07d2...	29	-1
7	276e9ec344d3bf029ff83a161c...	d33e520a99eb4cfc0d3ef2b6ff...	43	4
8	54e1a3c2b97fb0809da548a59...	a0bc11375dd3d8bdd0e0bfcbc...	40	4
9	fd04fa4105ee8045f6a0139ca5...	8fe0db7abbccaf2d788689e91...	37	1
10	302bb8109d097a9fc6e9cefc5...	22c0028cdec95ad1808c1fd50...	33	5
11	66057d37308e787052a32828...	dca924c5e55e17bdba2ad42ae...	38	6
12	19135c945c554eebfd7576c73...	1c7a9b908094192a2dfae2819...	36	2
13	4493e45e7ca1084efcd38ddeb...	a1fa003a1a17fc47164251e0e...	34	0

Insight:

We observed the above data and it shows the number of days taken to deliver an order and the difference between the actual delivery date and the estimated delivery date. We need to cut down the difference of days between actual delivery and estimated delivery. We recommend our client to deliver the order on perfect on estimated delivery date which gain more reputation to the company and satisfies our customers resulting in the growth of the business.

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

```
select * from

(with freight_ as(

select c.customer_state,avg(oi.freight_value) as highest_avg_freight_value,

from mp_.customers c join mp_.orders o on c.customer_id=o.customer_id

join mp_.order_items oi on o.order_id=oi.order_id

group by c.customer_state)

select customer_state,highest_avg_freight_value,dense_rank()over( order by
highest_avg_freight_value desc) as rank_

from freight_ order by rank_ limit 5) as a

 join

(with freight_1 as(
```

```
select c.customer_state,avg(oi.freight_value) as lowest_avg_freight_value

from mp_.customers c join mp_.orders o on c.customer_id=o.customer_id

join mp_.order_items oi on o.order_id=oi.order_id

group by c.customer_state)

select customer_state,lowest_avg_freight_value,dense_rank()over( order by
lowest_avg_freight_value asc) as rank_

from freight_1 order by rank_ asc limit 5) as b

on a.rank_=b.rank_
```

Result:

customer_state	highest_avg_freight_	rank_	customer_state_1	lowest_avg_freight_v	rank__1
RR	42.98442307692...	1	SP	15.14727539041...	1
PB	42.72380398671...	2	PR	20.53165156794...	2
RO	41.06971223021...	3	MG	20.63016680630...	3
AC	40.07336956521...	4	RJ	20.96092393168...	4
PI	39.14797047970...	5	DF	21.04135494596...	5

Insight:

We get the top five states with highest average freight value and top five states with lowest average freight value. We need to focus on these five states with lowest average freight value and wanted to make it higher.

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

```
select * from (with delivery as(  select c.customer_state,
avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))
as highest_avg_delivery

from mp_.customers c join mp_.orders o on c.customer_id=o.customer_id

group by c.customer_state)

select customer_state,highest_avg_delivery,dense_rank()over(order by highest_avg_delivery
desc) as rank_ from delivery order by rank_ limit 5) as a

join
```

```
(with delivery1 as(  select c.customer_state,
avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))
as lowest_avg_delivery

from mp_.customers c join mp_.orders o on c.customer_id=o.customer_id

group by c.customer_state)

select customer_state, lowest_avg_delivery ,dense_rank()over(order by lowest_avg_delivery
asc) as rank_ from delivery1 order by rank_ limit 5) as b

on a.rank_=b.rank_
```

Result:

	customer_state	highest_avg_delivery	rank_	customer_state_1	lowest_avg_delivery	rank__1
1	RR	28.97560975609...	1	SP	8.298061489072...	1
2	AP	26.73134328358...	2	PR	11.52671135486...	2
3	AM	25.98620689655...	3	MG	11.54381329810...	3
4	AL	24.04030226700...	4	DF	12.50913461538...	4
5	PA	23.31606765327...	5	SC	14.47956019171...	5

Insight:

The above data shows the top five states with highest average delivery time of order and the top five states with lowest average delivery time of order. We have to reduce the average delivery time in states having highest average delivery time.

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

```
 with delivery as ( select customer_state,

avg(timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day)) as
fastest_delivery_than_actual

from mp_.orders o join mp_.customers c on o.customer_id=c.customer_id

group by customer_state )

select customer_state, fastest_delivery_than_actual, dense_rank()over( order by
fastest_delivery_than_actual desc) as rank_  from delivery

order by fastest_delivery_than_actual desc   limit 5;
```

Result:

customer_state	fastest_delivery_thar	rank_
AC	19.76250000000...	1
RO	19.13168724279...	2
AP	18.73134328358...	3
AM	18.60689655172...	4
RR	16.41463414634...	5

Insight:

We got data of top five states with fastest delivery time than the estimated delivery time. This thing which is very good to see. We have to extend this delightful thing to all other states of the country for the growth of the business.

6. Analysis based on the payments:

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

```
with payment_types_with_orders as(

select p.payment_type,extract(month from o.order_purchase_timestamp)as
month_,count(o.order_id) as no_of_orders

from mp_.payments p join mp_.orders o on p.order_id=o.order_id

group by payment_type,month_)

select payment_type, month_, no_of_orders ,lag(no_of_orders)over(partition by
payment_type order by month_) as prevmonth_orders,no_of_orders-
lag(no_of_orders)over(partition by payment_type order by month_) as
difference_from_prevmonth_orders

from payment_types_with_orders
```

Result:

Row	payment_type	month_	no_of_orders	prevmonth_orders	difference_from_prev
1	debit_card	1	118	null	null
2	debit_card	2	82	118	-36
3	debit_card	3	109	82	27
4	debit_card	4	124	109	15
5	debit_card	5	81	124	-43
6	debit_card	6	209	81	128
7	debit_card	7	264	209	55
8	debit_card	8	311	264	47
9	debit_card	9	43	311	-268
10	debit_card	10	54	43	11
11	debit_card	11	70	54	16
12	debit_card	12	64	70	-6
13	voucher	1	477	null	null
14	voucher	2	424	477	-53

Insight:

From the above result, We got the data of number of orders on each month based on different payment types. Also, we provide some information on month on month comparison of orders of every payment type.

2. 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 `mp_.payments` group by payment_installments
```

Result:

payment_installment	no_of_orders
0	2
1	52546
2	12413
3	10461
4	7098
5	5239
6	3920
7	1626
8	4268
9	644
10	5328
11	23
12	133
13	16

Insight:

It shows the number of orders based on the frequency of installment of payments. The Brazilian people do not prefer to make their payment in more than 10 installments. The numbers are very high for payments made in less than 3 installments. We have to encourage people by giving some offers or make a new scheme for those who made payments in less than 3 installments which is a good thing for the company.