Target SQL Business Case Study

Description:
Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver. 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 analysing 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.

I. 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.

```
select column_name,data_type
from `TARGET_SQL.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
```

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTIO

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

Observation- We conducted a validation of the column data types within the 'customers' table by executing the provided SQL query in BigQuery.

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

```
select min(order_purchase_timestamp) as min ,max(order_purchase_timestamp) as max
from `TARGET_SQL.Orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

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

Observation- We used this to find out when the data started and ended, which was from 2016-09-4 to 2018-10-17. This helps us understand the timeframe we're analysing.

C. Count the number of Cities and States in our dataset.

```
select distinct customer_city, customer_state,
from `TARGET_SQL.Orders`  o inner join `TARGET_SQL.Customers` c
```

```
on o.customer_id=c.customer_id limit 10
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT

Observation- Observing the data, we notice that Sao Paulo city, located in the state of Sao Paulo (SP), has more orders compared to the combined orders of the following five cities. This is largely attributed to Sao Paulo being the most populous and affluent state in Brazil.

II. In-depth Exploration:

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

```
select extract( year from order_purchase_timestamp) as year,extract( month from
order_purchase_timestamp) as month,count(order_id) as No_of_orders
from `TARGET_SQL.Orders`
group by 1,2
order by year,month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAI

Row	year	month	No_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728
18	2018	3	7211
19	2018	4	6939
20	2018	5	6873
21	2018	6	6167
22	2018	7	6292
23	2018	8	6512
24	2018	9	16
25	2018	10	4

Observation- Based on the analysis of order count there is a increasing trend in the orders placed from 2016 till 2018 with little deterioration in the middle .We can see consistent upward trend in order count, suggesting growing consumer engagement.

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

```
select count(distinct order_id) as No_of_orders,extract( month from
order_purchase_timestamp) as month
from `TARGET_SQL.Orders`
group by 2
order by 2
```

Query results

JOB INFORMATION | RESULTS | JSON

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

Observation- The analysis indicates noticeable monthly seasonality, with elevated order volumes observed consistently from March to August

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

```
select count(order_id)asNo_of_orders,
case
when extract( hourfrom order_purchase_timestamp)between(0)and(6)then 'Dawn'
when extract( hourfrom order_purchase_timestamp)between(7)and(12)then 'Mornings'
when extract( hourfrom order_purchase_timestamp)between(13)and(18)then'Afternoon'
when extract( hourfrom order_purchase_timestamp)between(19)and(23)then 'Night'
END as intervals
from `TARGET_SQL.Orders` o join `TARGET_SQL.Customers` C
on o.customer_id=C.customer_id
group by intervals
order by No_of_orders desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION

Row	No_of_orders	intervals
1	38135	Afternoon
2	28331	Night
3	27733	Mornings
4	5242	Dawn

Observation- Brazilian customers show a preference for daytime online shopping, notably in the afternoon and evening hours. Understanding these buying patterns enables e-commerce businesses to allocate resources efficiently and tailor marketing strategies to capitalize on peak demand periods, ultimately enhancing customer satisfaction and driving sales.

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

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

```
select c.customer_state,count(order_id)as No_of_orders,extract(month from
order_purchase_timestamp) as month
from `TARGET_SQL.Orders`  o inner join `TARGET_SQL.Customers` c
on o.customer_id=c.customer_id
group by 1,3
order by 1,3
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	No_of_orders	month
1	AC	8	1
2	AC	6	2
3	AC	4	3
4	AC	9	4
5	AC	10	5
6	AC	7	6
7	AC	9	7
8	AC	7	8
9	AC	5	9
10	AC	6	10
11	AC	5	11

Observation- The results are in month-on-month order counts in each state. This insight identifies order count in different regions with respective of month.

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

```
select c.customer_state,count(distinct customer_unique_id) as No_of_customer
from `TARGET_SQL.Orders`  o inner join `TARGET_SQL.Customers` c
on o.customer_id=c.customer_id
group by 1
order by 1
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECU

Row	customer_state	No_of_customer
1	AC	77
2	AL	401
3	AM	143
4	AP	67
5	BA	3277
6	CE	1313
7	DF	2075
8	ES	1964
9	GO	1952
10	MA	726
11	MG	11259
12	MS	694
13	MT	876
14	PA	949
15	PB	519
16	PE	1609
17	PI	482
18	PR	4882
19	RJ	12384
20	RN	474
21	RO	240
22	RR	45
23	RS	5277
24	SC	3534
25	SE	342
26	SP	40302
27	TO	273

Observation- The query captures the total number of customers, the number of customers who made more than one purchase, and the total orders placed state-wise.

IV. Impact on Economy: Analyze 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).

```
select extract(month from order_purchase_timestamp)as month,
(
(sum(case when extract(year from order_purchase_timestamp) =2018 and extract (month
from order_purchase_timestamp)between 1 and 8 then p.payment_value END ) - sum
(case when extract(year from order_purchase_timestamp)=2017 and extract (month from
order_purchase_timestamp)between 1 and 8 then p.payment_value END )
)
/sum (case when extract(year from order_purchase_timestamp)=2017 and extract (month
from order_purchase_timestamp)between 1 and 8 then p.payment_value END )
)*100 as Percent_increase
from `TARGET_SQL.Orders` o join `TARGET_SQL.Payments` p
on o.order_id=p.order_id
WHERE
EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY 1
ORDER BY 1
```

JOB INFORMATION | RESULTS | JSON | EXE

Row	month	Percent_increase
1	1	705.1266954171...
2	2	239.9918145445...
3	3	157.7786066709...
4	4	177.8407701149...
5	5	94.62734375677...
6	6	100.2596912456...
7	7	80.04245463390...
8	8	51.60600520477...

Observation- The total cost of orders experienced a significant increase of 138.53% from 2017 to 2018, focusing solely on the period from January to August. January recorded the highest month-on-month percentage increase, followed by February and April.

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

```
select round(sum(o1.price),2) as Total_value,round(avg(o1.price),2)as Average_value
,c.customer_state
from `TARGET_SQL.Order_Items` o1
join `TARGET_SQL.Orders` o2
on o1.order_id=o2.order_id
join `TARGET_SQL.Customers` c
on o2.customer_id=c.customer_id
group by 3
order by 3
```

Row	Total_value	Average_value	customer_state
1	15982.95	173.73	AC
2	80314.81	180.89	AL
3	22356.84	135.5	AM
4	13474.3	164.32	AP
5	511349.99	134.6	BA
6	227254.71	153.76	CE
7	302603.94	125.77	DF
8	275037.31	121.91	ES

Observation- The results include both the total order price value and the average order price for each respective state.This insights provides insights into regional customer behaviour and preferences.

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

```
select round(sum(o1.freight_value),2) as
Total_value,round(avg(o1.freight_value),2)as Average_value ,c.customer_state
from `TARGET_SQL.Order_Items` o1
join `TARGET_SQL.Orders` o2
on o1.order_id=o2.order_id
join `TARGET_SQL.Customers` c
on o2.customer_id=c.customer_id
group by 3
order by 3
```

JOB INFORMATION | RESULTS | JSON | EXECUTIOI

Row	Total_value	Average_value	customer_state
1	3686.75	40.07	AC
2	15914.59	35.84	AL
3	5478.89	33.21	AM
4	2788.5	34.01	AP
5	100156.68	26.36	BA
6	48351.59	32.71	CE
7	50625.5	21.04	DF
8	49764.6	22.06	ES
9	53114.98	22.77	GO
10	31523.77	38.26	MA
11	270853.46	20.63	MG
12	19144.03	23.37	MS
13	29715.43	28.17	MT
14	38699.3	35.83	PA

Observation- The results as total freight_value and average freight_value for respective state. These insights can assist in optimizing logistics operations and understanding regional shipping trends.

V. 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.

Hint: 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_estimated_delivery_date - order_delivered_customer_date

```
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_SQL.orders`
where date_diff(order_delivered_customer_date,order_purchase_timestamp,day) is not
null
order by 2
```

JOB INFORMATION | RESULTS | CHART | JSON | EXECUT

Row	ORDER_ID	time_to_deliver	diff_estimated_deliv
1	e65f1eeee1f52024ad1dcd034...	0	9
2	bb5a519e352b45b714192a02f...	0	25
3	434cecee7d1a65fc65358a632...	0	19
4	d3ca7b82c922817b06e5ca211...	0	11
5	1d893dd7ca5f77ebf5f59f0d20...	0	10
6	d5fbeedc85190ba88580d6f82...	0	7
7	79e324907160caea526fd8b94...	0	8
8	38c1e3d4ed6a13cd0cf612d4c...	0	16
9	8339b608be0d84fca9d8da68b...	0	27
10	f349cdb62f69c3fae5c4d7d3f3...	0	12

Observation – From the observed date we can find the no. of days taken to deliver each order and the time to deliver days arranged from ascending order. Adding to it we can get the difference of estimated delivery . Through data analysis, we can pinpoint orders with longer delivery times and compare each delivery duration against the average to evaluate delivery efficiency. Negative values in the 'diff_estimated_delivery' column signify delays, whereas positive values indicate early deliveries

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

```
with avg_freight_per_state as (
select c.customer_state,avg(o1.freight_value) as avg_freight
from `TARGET_SQL.order_items` o1 join `TARGET_SQL.orders` o2 on
o1.order_id=o2.order_id
join `TARGET_SQL.customers` c on o2.customer_id=c.customer_id
group by 1)

SELECT customer_state,avg_freight
FROM (
      SELECT customer_state,avg_freight,
           ROW_NUMBER() OVER (ORDER BY avg_freight DESC) AS highest_rank,
           ROW_NUMBER() OVER (ORDER BY avg_freight ASC) AS lowest_rank
      FROM avg_freight_per_state
      ) ranked_freight
WHERE highest_rank <= 5 OR lowest_rank <= 5
```

Row	customer_state	avg_freight
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...
6	PI	39.14797047970...
7	AC	40.07336956521...
8	RO	41.06971223021...
9	PB	42.72380398671...
10	RR	42.98442307692...

Observation – From the results, we can observe the state having lowest freight value are SP&PR and highest freight value are PB & RR. Adding to this we can observe certain states have higher average freight values, suggesting unique characteristics or logistical challenges that contribute to increased shipping costs.

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

```
with avg_delivery_time_per_state as
(
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)
as avg_delivery_time
from `TARGET_SQL.customers` c join `TARGET_SQL.orders`o on
c.customer_id=o.customer_id
group by 1)

select  customer_state,avg_delivery_time
from
(select customer_state ,avg_delivery_time,
        row_number()over(order by avg_delivery_time desc)high_rank,
        row_number()over(order by avg_delivery_time asc)low_rank
```

```
  from avg_delivery_time_per_state) rank_delivery_time
where high_rank<= 5 or low_rank <= 5
```

JOB INFORMATION | RESULTS | CHART | JSON

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
6	PA	23.32
7	AL	24.04
8	AM	25.99
9	AP	26.73
10	RR	28.98

Observation- From the results, we can observe the state having lowest average delivery time is SP and average delivery time is RR. For the states that has lowest average delivery time their logistics need to be considered and corrective measures has to take in order to increase the speed of the delivery time.

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

```
select c.customer_state ,
round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day
)),2) as avg_speed_delivery
from `TARGET_SQL.customers` c join `TARGET_SQL.orders`o on
c.customer_id=o.customer_id
group by 1
order by 2 desc limit 5
```

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

Observation- Analysis of the query shows that the top 5 states, including Amazonas (AM), Amapá (AP), Roraima (RR), Acre (AC), and Rondônia (RO), have average delivery speeds compared to the estimated dates indicating efficient delivery processes.

VI. Analysis based on the payments:

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

```
select extract(month from order_purchase_timestamp) as
month,p.payment_type,count(o.order_id) as No_of_orders
from `TARGET_SQL.Orders`  o inner join `TARGET_SQL.Payments` p
on o.order_id=p.order_id
group by 1,2
order by 1,2
limit 8
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	month	payment_type	No_of_orders
1	1	UPI	1715
2	1	credit_card	6103
3	1	debit_card	118
4	1	voucher	477
5	2	UPI	1723
6	2	credit_card	6609
7	2	debit_card	82
8	2	voucher	424

Observation- Analysis shows that customer payment preferences by examining the monthly distribution of payment types. This analysis helps businesses identify trends in payment methods, optimize the checkout experience, expand payment options, and enhance overall customer satisfaction

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

```
select p.payment_installments,count(o.order_id) as no_of_orders
from `TARGET_SQL.Payments` p join `TARGET_SQL.Orders` o
on p.order_id=o.order_id
where payment_installments>0
group by 1
```

JOB INFORMATION | RESULTS | JSON | EXI

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

Observation- Results show the count of distinct orders for each payment installment option, indicating the distribution of customers who choose different installment plans. The most preferred installment option is a single payment, or at least the first installment has been paid in most cases.