

BUSINESS CASE: TARGET SQL

Introduction

Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation, and an exceptional guest experience that no other retailer can deliver.

The analysis of this business case will help us to work on multiple dimensions of the company as it has information on 100k orders from 2016 to 2018 made at Target in Brazil. Various dimensions of order mean: from order status, price, payment, and freight performance to customer location, product attributes, and finally reviews written by customers.

We will perform our analysis based on the question which will mostly cover all the dimensions of the company's performance.

Questions

1) Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

a. Data type of columns in a table.

Soln:

```
1 Select column_name,data_type
2 from target.INFORMATION_SCHEMA.COLUMNS
3 where table_name="customers"
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
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		

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

Soln:

```
1 select min(order_purchase_timestamp) as Range1,max(order_purchase_timestamp) as Range2
2 from `target.orders`
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	Range1	Range2				
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC				

c. Count the number of cities in our data set.

Soln:

```
1 select count(distinct customer_city) as total_city,count(distinct customer_state) as total_state
2 from `target.customers`
3
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	total_city	total_state				
1	4119	27				

2) In depth Exploration:

a. Is there a growing trend in the no. of orders placed over the last few years?

Soln:

Untitled 3

```
1 select distinct x.years,count(order_id) over (partition by x.years) as order_placed from
2 (select extract(year from order_purchase_timestamp) years,order_id
3 from `target.orders`) as x
4 order by x.years
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	years	order_placed				
1	2016	329				
2	2017	45101				
3	2018	54011				

Observation: Over the years we can see a clear jump in the number of order placed which shows clear influence of the store on the people.

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

Soln:

Untitled 3	RUN	SAVE	SHARE	SCHEDULE	MORE
<pre> 1 select distinct x.month,count(order_id) over (partition by x.month) as order_placed from 2 (select extract(month from order_purchase_timestamp) as month,order_id 3 from `target.orders`) as x 4 order by x.month </pre>					
Query results					
JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	month	order_placed			
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			
11	11	7544			
12	12	5674			

Observation:

- Orders placed increases decreases by considerable number up-to 8th month. It reaches peak in that month and then we see an abrupt decrease in the no of orders placed.
- There is a bit of recovery in next month but it still remains low.
- So the store performs well during first 8th month of the year and then see a drop in orders

c. During what time of the day, do the Brazilian customers mostly place their orders?

Soln:

<pre> 1 select x.time_frame,count(x.order_id) as cnt from 2 (select order_id, 3 case 4 when extract(hour from order_purchase_timestamp) between 0 and 6 5 then "Dawn" 6 when extract(hour from order_purchase_timestamp) between 7 and 12 7 then "Morning" 8 when extract(hour from order_purchase_timestamp) between 13 and 18 9 then "Afternoon" 10 else "Night" 11 end as time_frame 12 from `target.orders`) as x 13 group by x.time_frame 14 </pre>					
Query results					
JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	time_frame	cnt			
1	Morning	27733			
2	Dawn	5242			
3	Afternoon	38135			
4	Night	28331			

Observation:

- We see the most orders being placed during Afternoon time followed by Night and Morning.

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

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

Soln:

```
1 select extract(year from o.order_purchase_timestamp) as years,
2        extract(month from o.order_purchase_timestamp) as months,
3        c.customer_state,count(o.order_id) as total_orders
4 from 'target.orders' as o left join 'target.customers' as c
5 on o.customer_id=c.customer_id
6 group by months,years,c.customer_state
7 order by years,months,customer_state
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
ow	years	months	customer_state		total_orders		
1	2016	9	RR		1		
2	2016	9	RS		1		
3	2016	9	SP		2		
4	2016	10	AL		2		
5	2016	10	BA		4		
6	2016	10	CE		8		
7	2016	10	DF		6		
8	2016	10	ES		4		
9	2016	10	GO		9		
10	2016	10	MA		4		

```
1 select extract(year from o.order_purchase_timestamp) as years,
2        extract(month from o.order_purchase_timestamp) as months,
3        c.customer_state,count(o.order_id) as total_orders
4 from 'target.orders' as o left join 'target.customers' as c
5 on o.customer_id=c.customer_id
6 group by months,years,c.customer_state
7 order by years,months,customer_state
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
w	years	months	customer_state		total_orders		
26	2017	1	AC		2		
27	2017	1	AL		2		
28	2017	1	BA		25		
29	2017	1	CE		9		
30	2017	1	DF		13		
31	2017	1	ES		12		
32	2017	1	GO		18		
33	2017	1	MA		9		
34	2017	1	MG		108		
35	2017	1	MS		1		
36	2017	1	MT		11		

```

1 select extract(year from o.order_purchase_timestamp) as years,
2        extract(month from o.order_purchase_timestamp) as months,
3        c.customer_state,count(o.order_id) as total_orders
4 from 'target.orders' as o left join 'target.customers' as c
5 on o.customer_id=c.customer_id
6 group by months,years,c.customer_state
7 order by years,months,customer_state

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
low	years	months	customer_state	total_orders		
551	2018	8	RJ	745		
552	2018	8	RN	20		
553	2018	8	RO	9		
554	2018	8	RS	300		
555	2018	8	SC	206		
556	2018	8	SE	23		
557	2018	8	SP	3253		
558	2018	8	TO	13		
559	2018	9	MG	4		
560	2018	9	RJ	3		
561	2018	9	SC	1		

Observation:

- We can clearly see total orders being placed is increasing with each passing year except some month.

b. How are customers distributed across all the states?

Soln:

```

1 select customer_state,count(distinct customer_unique_id) as total_customer
2 from 'target.customers'
3 group by customer_state

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAP
Row	customer_state	total_customer			
1	RN	474			
2	CE	1313			
3	RS	5277			
4	SC	3534			
5	SP	40302			
6	MG	11259			
7	BA	3277			
8	RJ	12384			
9	GO	1952			
10	MA	726			
11	PE	1609			
12	PB	519			
13	ES	1964			

Observation:

- Majority of the customers are concentrated in few states, which show impact of store on the natives, but few states have few number of customers which is not a good sign.

4) Analyze the money movement by e-commerce by looking at order price, freight and others.

a. Get the % increase in the cost of orders from years 2017 to 2018(include months between Jan to Aug only)

soln:

```
1 select*from
2 (select round(((x.total-lag(x.total,1) over (order by x.years))/lag(x.total,1) over (order by x.years))*100,2) as percentage_increase from
3 (select extract(year from order_purchase_timestamp) as years, round(sum(p.payment_value),2) as total
4 from `target.orders` as o left join `target.payments` as p
5 on o.order_id=p.order_id
6 where extract(month from o.order_purchase_timestamp) between 1 and 8
7 group by years
8 order by years)as x)as y
9 where y.percentage_increase is not NULL
```

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	percentage_increase
1	136.98

Observation:

- There is whooping increase in the cost of order between 2017 and 2018 considering Jan to August month.

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

Soln:

```
1 select c.customer_state, round(sum(o1.price),2) as total_price,round(avg(o1.price),2) as avg_price,
2 from `target.customers` as c left join `target.orders` as o
3 on c.customer_id=o.customer_id
4 left join `target.order_items` as o1
5 on o.order_id=o1.order_id
6 group by (c.customer_state)
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_state	total_price	avg_price		
1	RN	83034.98	156.97		
2	CE	227254.71	153.76		
3	RS	750304.02	120.34		
4	SC	520553.34	124.65		
5	SP	5202955.05	109.65		
6	MG	1585308.03	120.75		
7	BA	511349.99	134.6		
8	RJ	1824092.67	125.12		
9	GO	294591.95	126.27		
10	MA	119648.22	145.2		

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

Soln:

```
1 select c.customer_state, round(sum(o1.freight_value),2) as total, round(avg(o1.freight_value),2) as avg_f,
2 from `target.customers` as c left join `target.orders` as o
3 on c.customer_id=o.customer_id
4 left join `target.order_items` as o1
5 on o.order_id=o1.order_id
6 group by (c.customer_state)
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
Row	customer_state	total	avg_f				
1	RN	18860.1	35.65				
2	CE	48351.59	32.71				
3	RS	135522.74	21.74				
4	SC	89660.26	21.47				
5	SP	718723.07	15.15				
6	MG	270853.46	20.63				
7	BA	100156.68	26.36				
8	RJ	305589.31	20.96				
9	GO	53114.98	22.77				

5) Analysis based on Sales, freight and sales time.

a. Find the no. of days taken to deliver each order from order purchase date as delivery time. Also calculate difference between estimated delivery time delivery time

soln:

```
1 select x.order_id,(x.delivery_date-x.order_date) as time_to_deliver,(x.day1-x.delivery_date) as diff_estimated_delivery
2 from
3 (select extract(day from order_purchase_timestamp) as order_date,order_id,
4      extract(day from order_delivered_customer_date) as delivery_date,
5      extract(day from order_estimated_delivery_date) as day1
6 from `target.orders`) as x
```

Query results

SAVE RESULT

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
Row	order_id	time_to_deliver	diff_estimated_delivery				
1	2c45c33d2f9cb8ff8b1c86cc28...	0	-1				
2	68f47f50f04c4cb6774570cfe...	0	2				
3	304e7fc7db4a67a8ab0403ce4...	-28	11				
4	c930f0fb9c6fed6ef015de48ea...	-29	12				
5	d0462d19e9c58af6416a06e62...	21	-12				
6	8d204be4884a2307f1486df72...	-27	13				
7	0d8f485ffe96c81fe3e282095e...	-29	12				
8	abe6fc40cd1fe4d8d30881130...	23	-17				
9	8576190c64f6d9d9ed5055185...	-27	13				
10	913e9a5e8da11e9a318ab2d38...	25	-24				

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

Soln:

```
1 select c.customer_state,round(avg(o1.freight_value),2) as avg_f,
2 from `target.customers` as c left join `target.orders` as o
3 on c.customer_id=o.customer_id
4 left join `target.order_items` as o1
5 on o.order_id=o1.order_id
6 group by (c.customer_state)
7 order by avg_f desc
8 limit 5
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	avg_f		
1	RR	42.98		
2	PB	42.72		
3	RO	41.07		
4	AC	40.07		
5	PI	39.15		

Untitled 4 RUN SAVE SHARE SCHEDULE

```
1 select c.customer_state,round(avg(o1.freight_value),2) as avg_f,
2 from `target.customers` as c left join `target.orders` as o
3 on c.customer_id=o.customer_id
4 left join `target.order_items` as o1
5 on o.order_id=o1.order_id
6 group by (c.customer_state)
7 order by avg_f asc
8 limit 5
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	customer_state	avg_f		
1	SP	15.15		
2	PR	20.53		
3	MG	20.63		
4	RJ	20.96		
5	DF	21.04		

c. Find out the top 5 states with the highest and lowest delivery time.

Soln: Highest delivery time:

```
1 select distinct x.customer_state, round(avg(x.day1-x.day2) over (partition by x.customer_state),2) as average from
2 (select
3   extract(day from o.order_delivered_customer_date) as day1,
4   extract(day from o.order_purchase_timestamp) as day2,c.customer_state
5 from `target.orders` as o left join `target.customers` as c
6   on o.customer_id=c.customer_id) as x
7 order by average DESC
8 limit 5
```

Query results

[SAVE RE](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
low	customer_state	average			PREVIEW
1	AP	4.31			
2	RR	1.9			
3	PE	1.4			
4	PI	1.31			
5	CE	1.2			

Lowest delivery time:

```
1 select distinct x.customer_state, round(avg(x.day1-x.day2) over (partition by x.customer_state),2) as average from
2 (select
3   extract(day from o.order_delivered_customer_date) as day1,
4   extract(day from o.order_purchase_timestamp) as day2, c.customer_state
5 from `target.orders` as o left join `target.customers` as c
6   on o.customer_id=c.customer_id) as x
7 order by average asc
8 limit 5
```

Query results

[SAVE R](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_state	average				
1	AM	-1.67				
2	RO	0.07				
3	RN	0.11				
4	MT	0.15				
5	AL	0.16				

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

Soln:

```
1 select y.customer_state from
2 (select distinct x.customer_state, round(avg(x.day1-x.day2) over (partition by x.customer_state),2) as average1,
3   round(avg(x.day3-x.day2) over (partition by x.customer_state),2) as average2 from
4   (select
5     extract(day from o.order_delivered_customer_date) as day1,
6     extract(day from o.order_purchase_timestamp) as day2,
7     extract(day from o.order_estimated_delivery_date ) as day3,
8     c.customer_state
9   from `target.orders` as o left join `target.customers` as c
10    on o.customer_id=c.customer_id) as x) as y
11 where average1>average2
12 order by average1 desc
13 limit 5
14
15
```

Query results

[SAVE](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_state					
1	RR					
2	PE					
3	PI					
4	CE					
5	PB					

6) Analysis based on the payments

a. Find the month on month no of order placed using different payment type.

Soln:

```
1 select distinct extract(month from order_purchase_timestamp) as months,
2                extract(year from order_purchase_timestamp) as years,
3                count(o.order_id) as no_of_orders, p.payment_type
4 from `target.payments` as p left join `target.orders` as o
5 on p.order_id=o.order_id
6 group by years, months, p.payment_type
7 order by years, months ASC
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
Row	months	years	no_of_orders	payment_type			
1	9	2016	3	credit_card			
2	10	2016	254	credit_card			
3	10	2016	23	voucher			
4	10	2016	2	debit_card			
5	10	2016	63	UPI			
6	12	2016	1	credit_card			
7	1	2017	61	voucher			
8	1	2017	197	UPI			
9	1	2017	583	credit_card			
10	1	2017	9	debit_card			

b. Find the no. of order placed on the basis of the payment installments that have been paid.

Soln:

```
1 select count(x.order_id) as Payment_done from
2 (select (o.price+o.freight_value) as price, p.payment_value, p.order_id
3 from `target.order_items` as o left join `target.payments` as p
4 on o.order_id=p.order_id) as x
5 where x.price=x.payment_value
6
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION
Row	Payment_done					
1	69631					

Overall Observation:

- Company have reach over all the 27 states in the country.
- Data shows considerable increase in the number of customers and orders being placed over the years.
- First 8 month sees a good amount of order being placed but it decreases abruptly and the trend continue for next 4 month.
- Some states have very good delivery time as compared to others and there we can clearly see number of customers being increased.
- Variety of Payment methods are available and almost everyone is being used in good numbers.

Red Flags and Suggestions/Strategy:

- Although company have reach over all the 27 states, but most of its customers are limited to few states and there is no sign of good growth rate in those states.
- Good discount offers, faster delivery time, more lenient EMI options, advertisement through local popular stars may help in improving the demand.
- Last 4 month of the year need special attention because in no time the trend can impact other months. More leniency and stock pile of demanded goods during those months can improve. And in long term more advertisement is needed so that people are well aware of the company.