

Target SQL Business Case

- By Tanmoy Jash

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.

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

1.1 Data type of all columns in the “customers” table.

Query:

```
SELECT column_name,data_type
FROM `target`.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers';
```

Result:

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

1.2 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 FROM `target.orders`;
```

Result:

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

1.3 Count the number of Cities and States in our dataset

Query:

```
SELECT count(DISTINCT geolocation_city) as No_of_Cities ,count(DISTINCT geolocation_state) as No_of_States FROM `target.geolocation`
```

Result:

Row	No_of_Cities	No_of_States
1	8011	27

2. In-depth Exploration

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

Query:

```
SELECT year, month, COUNT(month) AS num_of_orders FROM (
SELECT order_purchase_timestamp, EXTRACT(year
FROM order_purchase_timestamp) AS year, EXTRACT(month
FROM order_purchase_timestamp) AS month, FROM `target.orders`) AS orders_table
GROUP BY year, month
ORDER BY year, month;
```

Result

Row	year	month	num_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

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

Query:

```
SELECT EXTRACT(year FROM order_purchase_timestamp) AS year, EXTRACT(month FROM
order_purchase_timestamp) AS month, COUNT(order_id) AS num_of_orders FROM
`target.orders`
GROUP BY year, month
ORDER BY year, num_of_orders DESC;
```

Result:

Row	year	month	num_of_orders
1	2016	10	324
2	2016	9	4
3	2016	12	1
4	2017	11	7544
5	2017	12	5673
6	2017	10	4631
7	2017	8	4331
8	2017	9	4285
9	2017	7	4026
10	2017	5	3700

2.3 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:

```
WITH cte AS (SELECT CASE
WHEN hour BETWEEN 0 AND 6 THEN 'Dawn'
WHEN hour BETWEEN 7 AND 12 THEN 'Morning'
WHEN hour BETWEEN 13 AND 18 THEN 'Afternoon'
WHEN hour BETWEEN 19 AND 23 THEN 'Night'
END AS interval_of_day FROM `target.orders`) SELECT interval_of_day, COUNT(*) AS
num_of_orders FROM cte
GROUP BY interval_of_day;
```

Result:

Row	interval_of_day ▼	num_of_orders ▼
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

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

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

Query

```
WITH cte AS ( SELECT O.order_id, C.customer_state, O.month
FROM `target.orders_info` AS O INNER JOIN `target.customers` C ON O.customer_id =
C.customer_id) SELECT customer_state, month, COUNT(order_id) AS num_of_orders FROM cte
GROUP BY customer_state, month ORDER BY customer_state, month;
```

Result

Row	customer_state ▼	month ▼	num_of_orders ▼
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7

3.2 How are the customers distributed across all the states?

Query

```
SELECT customer_state, COUNT(DISTINCT customer_unique_id) AS num_of_customers FROM
`target.customers`
GROUP BY customer_state
ORDER BY num_of_customers DESC;
```

Result:

Row	customer_state ▼	num_of_customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

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

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

Query

```
WITH cte AS (
SELECT EXTRACT(year FROM O.order_purchase_timestamp) as year,
ROUND(SUM(P.payment_value), 2) AS total_payment_value FROM `target.orders` AS O
INNER JOIN `target.payments` AS P USING (order_id)
WHERE EXTRACT(year FROM O.order_purchase_timestamp) BETWEEN 2017 AND 2018
AND EXTRACT(month FROM O.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY year) SELECT
T1.year AS Year1, T1.total_payment_value AS Value1, T2.year AS Year2,
T2.total_payment_value AS Value2, ROUND(((T2.total_payment_value -
T1.total_payment_value)/T1.total_payment_value)*100,2) AS percent_increase FROM
cte AS T1, cte AS T2 WHERE T1.year < T2.year;
```

Result:

Row	Year1	Value1 ▼	Year2	Value2 ▼	percent_increase ▼
1	2017	3669022.12	2018	8694733.84	136.98

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

Query:

```
SELECT customer_state,
```

```

ROUND(SUM(P.payment_value), 0) AS total_price, ROUND(AVG(P.payment_value), 0) AS
avg_price
FROM `target.orders` AS O INNER JOIN `target.payments` AS P ON O.order_id = P.order_id
INNER JOIN `target.customers` AS C ON O.customer_id = C.customer_id
GROUP BY C.customer_state
ORDER BY C.customer_state

```

Result

Row	customer_state	total_price	avg_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

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

Query

```

SELECT C.customer_state, round(sum(i.freight_value),2) as Total_freight_value,
round(avg(i.freight_value),2) as avg_freight_value FROM `target.customers` as C
inner join `target.orders` as O using(customer_id)
inner join `target.order_items` as i Using(order_id)
GROUP BY C.customer_state
ORDER BY C.customer_state;

```

Result

Row	customer_state	Total_freight_value	avg_freight_value
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71

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

5.1 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_status = 'delivered' ORDER BY order_id;
```

Result:

Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcd0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

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

Query

```
With CTE1 as(  
SELECT C.customer_state, round(avg(i.freight_value),2) as avg_freight_value, 'Bottom 5' as  
Remarks FROM `target.customers` as C inner join `target.orders` as O using(customer_id)  
inner join `target.order_items` as i Using(order_id)  
GROUP BY C.customer_state  
ORDER BY avg_freight_value  
LIMIT 5 ),  
CTE2 AS (  
SELECT C.customer_state, round(avg(i.freight_value),2) as avg_freight_value, 'Top 5' as Remarks  
FROM `target.customers` as C inner join `target.orders` as O using(customer_id)  
inner join `target.order_items` as i Using(order_id)  
GROUP BY C.customer_state  
ORDER BY avg_freight_value desc
```

LIMIT 5)

SELECT customer_state, avg_freight_value, Remarks FROM cte1

UNION ALL

SELECT customer_state, avg_freight_value, Remarks FROM cte2

Result

Row	customer_state ▼	avg_freight_value ▼	Remarks ▼
1	SP	15.15	Bottom 5
2	PR	20.53	Bottom 5
3	MG	20.63	Bottom 5
4	RJ	20.96	Bottom 5
5	DF	21.04	Bottom 5
6	RR	42.98	Top 5
7	PB	42.72	Top 5
8	RO	41.07	Top 5
9	AC	40.07	Top 5
10	PI	39.15	Top 5

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

Query

With CTE1 as(

```
SELECT C.customer_state, round(avg(DATE_DIFF(O.order_delivered_customer_date,
O.order_purchase_timestamp, day)),2) as avg_delivery_time, 'Bottom 5' as Remarks FROM
`target.customers` as C inner join `target.orders` as O using(customer_id)
inner join `target.order_items` as i Using(order_id)
```

GROUP BY C.customer_state

ORDER BY avg_delivery_time

LIMIT 5),

CTE2 AS (

```
SELECT C.customer_state, round(avg(DATE_DIFF(O.order_delivered_customer_date,
O.order_purchase_timestamp, day)),2) as avg_delivery_time, 'Top 5' as Remarks FROM
`target.customers` as C inner join `target.orders` as O using(customer_id)
inner join `target.order_items` as i Using(order_id)
```

GROUP BY C.customer_state

ORDER BY avg_delivery_time desc

LIMIT 5)

SELECT customer_state, avg_delivery_time, Remarks FROM cte1

UNION ALL

SELECT customer_state, avg_delivery_time, Remarks FROM cte2

Result

Row	customer_state ▼	avg_delivery_time ▼	Remarks ▼
1	SP	8.26	Bottom 5
2	PR	11.48	Bottom 5
3	MG	11.52	Bottom 5
4	DF	12.5	Bottom 5
5	SC	14.52	Bottom 5
6	RR	27.83	Top 5
7	AP	27.75	Top 5
8	AM	25.96	Top 5
9	AL	23.99	Top 5
10	PA	23.3	Top 5

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

Query:

*With CTE as (SELECT C.customer_state,
DATE_DIFF(O.order_estimated_delivery_date, O.order_delivered_customer_date, day) AS
diff_delivery FROM `target.customers` as C inner join `target.orders` as O using(customer_id))
SELECT customer_state, avg(diff_delivery) as avg_diff from CTE
GROUP BY customer_state
ORDER BY avg_diff desc*

Result:

Row	customer_state ▼	avg_diff ▼
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

6 Analysis based on the payments:

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

Query

With CTE as (

SELECT

Extract(Year from O.order_purchase_timestamp) as Year,

CASE

WHEN Extract(Month from O.order_purchase_timestamp) = 1 THEN 'January'

WHEN Extract(Month from O.order_purchase_timestamp) = 2 THEN 'February'

WHEN Extract(Month from O.order_purchase_timestamp) = 3 THEN 'March'

WHEN Extract(Month from O.order_purchase_timestamp) = 4 THEN 'April'

WHEN Extract(Month from O.order_purchase_timestamp) = 5 THEN 'May'

WHEN Extract(Month from O.order_purchase_timestamp) = 6 THEN 'June'

WHEN Extract(Month from O.order_purchase_timestamp) = 7 THEN 'July'

WHEN Extract(Month from O.order_purchase_timestamp) = 8 THEN 'August'

WHEN Extract(Month from O.order_purchase_timestamp) = 9 THEN 'September'

WHEN Extract(Month from O.order_purchase_timestamp) = 10 THEN 'October'

WHEN Extract(Month from O.order_purchase_timestamp) = 11 THEN 'November'

ELSE 'December'

END as month, O.Order_id,P.Payment_type

FROM `target.orders` as O INNER JOIN `target.payments` AS P USING (order_id))

Select Year,month,Payment_type,count(Order_id) as No_of_orders from CTE

GROUP BY Year,month,Payment_type

ORDER BY Year,month

Result

Row	Year	month	Payment_type	No_of_orders
1	2016	December	credit_card	1
2	2016	October	voucher	23
3	2016	October	credit_card	254
4	2016	October	UPI	63
5	2016	October	debit_card	2
6	2016	September	credit_card	3
7	2017	April	credit_card	1846
8	2017	April	UPI	496
9	2017	April	debit_card	27
10	2017	April	voucher	202

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

Query

```
SELECT payment_installments, COUNT(order_id) AS num_of_orders
FROM `target.payments` WHERE payment_installments >= 1
GROUP BY payment_installments
ORDER BY payment_installments
```

Result

Row	payment_installment	num_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
10	10	5328

7. Insights and Recommendations

7.1 Insights

- With just 3 months data available in 2016, the number of orders peak in October. In 2017, the top 3 months are November, December and October. In 2018, January, March and April are the top 3 months. With the limited data available I don't see any monthly seasonality in terms of number of orders being placed.
- The customers order the most during afternoon and least during dawn with mornings and night, being almost same, in the middle
- São Paulo has the largest number of customers followed not so closely by Rio de Janeiro and Minas Gerais. Roraima has the least number of customers.
- The cost of orders has increased by around 137% from year 2017 to 2018, which is a huge increase, more than double.
- The total price is highest for São Paulo and this is expected as the number of customers are also highest in São Paulo. The average price is around 200
- The total freight, following the similar pattern as total price, is highest for São Paulo. The average is around 30.
- The top state has order delivery 20 days faster than the estimate date.

- Majority of the orders are placed using credit card followed by UPI
- Around 50% of the orders placed have at least one installment paid.
- Company received low rating for maximum orders in highlighted states; need to study further about the reasons for customer dissatisfaction to such great extent in these states. This is the query for counting the number of rating in each state.

```

SELECT
*
FROM (
  SELECT
    c.customer_state,
    orv.review_score
  FROM
    `target.order_reviews` orv
  JOIN
    `target.orders` o
  USING
    (order_id)
  JOIN
    `target.customers` c
  USING
    (customer_id)) PIVOT(COUNT(*) FOR review_score IN (1, 2, 3, 4, 5));

```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
Row	customer_state	_1	_2	_3	_4	_5	
1	RS	560	172	449	1098	3204	
2	RJ	2183	464	1050	2137	6931	
3	PR	473	156	381	1009	3019	
4	SC	413	119	321	712	2058	
5	SP	4054	1211	3299	7991	25135	
6	BA	504	130	337	744	1642	
7	GO	233	67	191	423	1110	
8	MG	1207	339	969	2259	6851	
9	PE	221	53	131	322	919	
10	RO	24	15	27	44	142	
11	RN	54	14	39	95	280	
12	SE	63	13	29	67	177	
13	MA	131	31	74	157	353	
14	PA	155	35	96	197	485	
15	CE	210	54	131	263	671	
16	ES	243	57	182	425	1109	

8. Recommendations

- As Brazilian customers usually tend to buy in afternoon and night, we can increase staff in during this time frame in order to manage the customers' requests, and services better during this time by reducing workforce of morning and dawn.
- Target should try to replicate the same which worked in getting increasing orders from Jan 2017 to Nov 2017
- Avg delivery time is quite high for most of those states from where company is receiving quite less volume of orders, detailed study is needed further for checking the other reasons behind such low volume of orders from majority of states. Huge delivery time can be the one of the reason and need to work on it.