TARGET STUDY

INTRODUCTION

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 September 2016 and October 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.

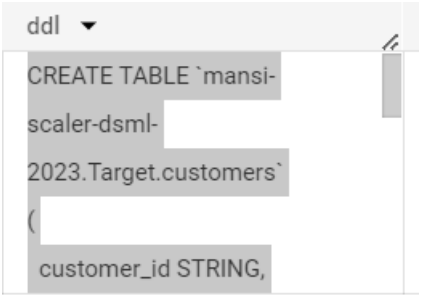
Following is the exploratory analysis done on the dataset to arrive at actionable business insights.

Done By: Mansi Agrawal, MAY 23 Beginner without Python batch

a. Data type of all columns in the "customers" table

The data type of all columns can be witnessed by looking at the INFORMATION_SCHEMA of the table using the following code:

```
SELECT *
FROM Target.INFORMATION_SCHEMA.TABLES
WHERE table_name = 'customers';
```




```
CREATE TABLE `mansi-scaler-dsml-2023.Target.customers`
(
customer_id STRING,
customer_unique_id STRING,
customer_zip_code_prefix INT64,
```

Ddl has all the information regarding the column names and the data types of all the columns in customer table.

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

Since the orders table has the relevant information regarding when the orders were placed, we just do a min and max on the order_purchase_timestamp and get the time range

```
SELECT MAX(order_purchase_timestamp) AS latest_order,
MIN(order_purchase_timestamp) AS earliest_order
FROM `mansi-scaler-dsml-2023.Target.orders`
```

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

This shows that we have the data for orders that were placed between September'2016 and October'2018.

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

To address the above question, we need to understand the cities and states where both the sellers and customers are present.

Relevant SQL queries are as follows:

```
SELECT COUNT(DISTINCT customer_city) AS num_cities, COUNT(DISTINCT
customer_state) AS num_states
FROM `Target.customers`
```

Row	num_cities	num_states
1	4119	27

```
SELECT COUNT(DISTINCT seller_city) AS num_cities, COUNT(DISTINCT seller_state) AS
num_states
FROM `Target.sellers`
```

Row	num_cities	num_states
1	611	23

From the output, we can observe that the customers are quite widespread (spread in 4000+ cities) as compared to the sellers indicating the spread of the online business that Target might be doing.

2. IN-DEPTH EXPLORATION

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

This can be explored by looking at number of orders placed month over month over the total timespan

```
SELECT
EXTRACT (YEAR FROM order_purchase_timestamp) AS Year,
EXTRACT (MONTH FROM order_purchase_timestamp) AS Month,
COUNT(order_id) AS Num_orders_mon_yr
FROM `Target.orders`
GROUP BY Year, Month
ORDER BY Year, Month
```

Row	Year	Month	Num_orders_mon_yr
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
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

The above output shows that the number of orders placed consistently grew from end of 2016 till August 2018 post which they fell to an all-time low. The reason for the same needs to be explored.

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

For checking this we will have to arrive at average orders placed in each month over a 12 month cycle. This can be done using the following code:

```
SELECT t.Month, SUM(t.Num_orders_mon_yr)/COUNT(t.Num_orders_mon_yr) AS Monthly_avg
FROM (
SELECT
EXTRACT (YEAR FROM order_purchase_timestamp) AS Year,
EXTRACT (MONTH FROM order_purchase_timestamp) AS Month,
COUNT(order_id) AS Num_orders_mon_yr
FROM `Target.orders`
GROUP BY Year, Month
ORDER BY Year, Month) AS t
GROUP BY t.Month
ORDER BY t.Month
```

Row	Month	Monthly_avg
1	1	4034.5
2	2	4254.0
3	3	4946.5
4	4	4671.5
5	5	5286.5
6	6	4706.0
7	7	5159.0
8	8	5421.5
9	9	1435.0
10	10	1653.0
11	11	7544.0
12	12	2837.0

Here we observe that, Jan to August, mostly the orders are consistent with a noticeable drop in September and October, again spiking to an all-time high in November and then dropping again in December. November seems to have some sort of annual festival or celebration which calls for more shopping amongst customers.

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**

To evaluate this, we will construct the time based buckets and then classify the orders in them.

```
SELECT t.time_of_day,COUNT(t.order_purchase_timestamp) As orders_placed
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 "Mornings"
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18
THEN "Afternoon"
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23
THEN "Night"
END AS time_of_day
FROM `Target.orders`) AS t
GROUP BY t.time_of_day
ORDER BY t.time_of_day
```

Row	time_of_day	orders_placed
1	Afternoon	38135
2	Dawn	5242
3	Mornings	27733
4	Night	28331

As is evident from the report, Brazilian customers placed maximum orders in the afternoon followed by night.

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

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

Number of orders placed month on month distributed by state would involve joining the customer and the orders table basis the customer_id key. Orders table has all the details for all the orders while customer table has the details about customers, their city and state.

The relevant query is as follows:

```
SELECT c.customer_state,
EXTRACT (YEAR FROM o.order_purchase_timestamp) AS Year,
EXTRACT (MONTH FROM o.order_purchase_timestamp) AS Month,
COUNT(o.order_id) AS Num_orders_mon_state
FROM `Target.orders` As o
LEFT JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state, Year, Month
ORDER BY c.customer_state, Year, Month
```

The output looks something like this:

Row	customer_state	Year	Month	Num_orders_mon_st
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6

However, additionally it makes sense to look at a more aggregated picture to figure out which states are placing the maximum orders viz. the states which are not to understand Target's customer base concentration.
So, we run the following query:

```
SELECT c.customer_state,
COUNT(o.order_id) AS Num_orders_state,
FROM `Target.orders` As o
LEFT JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Num_orders_state DESC
```

Top 10 states with maximum orders

Row	customer_state	Num_orders_state
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

Bottom 10 states – Ones with minimum orders

Row	customer_state	Num_orders_state
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

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

Analysis of the above question revealed that there is quite a discrepancy between the total no. of orders placed by different states. Let's also look at how is the distribution of customers across states.

Distribution of customers across different states

```
SELECT customer_state, COUNT(customer_id) AS num_customers
FROM `Target.customers`
GROUP BY customer_state
ORDER BY num_customers DESC
```

Row	customer_state	num_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

States with comparatively lesser concentration of customers

Row	customer_state	num_customers
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

Since the numbers are exactly the same between when we count the total order_ids across states and total customer_id's across states…We can conclude that there is one is to one mapping between a customer and his order, which means that there is no case of a customer placing more than one order in the entire period between Sep'2016 to Oct'2018 in a country as big as Brazil. Now that is a bit strange!

We have also run another query in the additional questions section to ascertain the same point. This insight is a striking one….Why would a customer not place multiple orders, is there a lot of dissatisfaction amongst customers? Or is there something else, let's try to figure out as we run more queries.

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.

```
With cte1 AS
(
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS Year,
EXTRACT (MONTH FROM o.order_purchase_timestamp) AS Month,
ROUND(SUM(p.payment_value),2) AS order_2017
FROM `Target.orders` AS o
JOIN `Target.payments` AS p
ON o.order_id = p.order_id
AND EXTRACT (YEAR FROM o.order_purchase_timestamp) = 2017
AND EXTRACT (MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY Year, Month
ORDER BY Year,Month
), cte2 AS
```

```
(SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS Year,
EXTRACT (MONTH FROM o.order_purchase_timestamp) AS Month,
ROUND(SUM(p.payment_value),2) AS order_2018
FROM `Target.orders` AS o
JOIN `Target.payments` AS p
ON o.order_id = p.order_id
AND EXTRACT (YEAR FROM o.order_purchase_timestamp) = 2018
AND EXTRACT (MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY Year, Month
ORDER BY Year,Month
)
SELECT cte1.*,cte2.*, ROUND((cte2.order_2018 - cte1.order_2017)/cte1.order_2017*100,2) AS
percent_increase
FROM cte1
JOIN cte2
ON cte1.Month = cte2.Month
ORDER BY percent_increase DESC
```

Row	Year	Month	order_2017	Year_1	Month_1	order_2018	percent_increase
1	2017	1	138488.04	2018	1	1115004.18	705.13
2	2017	2	291908.01	2018	2	992463.34	239.99
3	2017	4	417788.03	2018	4	1160785.48	177.84
4	2017	3	449863.6	2018	3	1159652.12	157.78
5	2017	6	511276.38	2018	6	1023880.5	100.26
6	2017	5	592918.82	2018	5	1153982.15	94.63
7	2017	7	592382.92	2018	7	1066540.75	80.04
8	2017	8	674396.32	2018	8	1022425.32	51.61

Here, we observe that in January moth itself, the order value has increased 7 folds. However, the trend trails down as the year progresses and by August, the percent increase in sales is only around 52% as the denominator is large. However, in terms of absolute numbers, order value across 2018 is quite consistent throughout the year. So, we can conclude by saying that the order_values kept consistently growing through 2017 and in 2018 beginning, it peaked and then stabilised thereof.

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

Since the payment_value denotes the actual price paid for the purchase order and we want to access the order price distribution by state, it is sensible to consider the payment value column from payments table as order price against each order id. For coming across the state information, we will have to look into the customers table. Hence, we will join the payments table with orders table and then orders table with customers table and then calculate the total and average order price basis the code given below:

```
SELECT c.customer_state,ROUND(SUM(p.payment_value),2) AS Total_order_price,
ROUND(AVG(p.payment_value),2) AS avg_order_price
FROm `Target.orders` AS o
JOIN `Target.payments` AS p
ON o.order_id = p.order_id
JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
```

```
GROUP BY c.customer_state
ORDER BY total_order_price DESC, avg_order_price DESC
```

Row	customer_state	Total_order_price	avg_order_price
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76
10	ES	325967.55	154.71
11	PE	324850.44	187.99

The above table starts with the states having the highest total order price but the average order price is in the range of 140-160. Comparing other values of average prices across the table, it denotes that a high number of customers are placing the order with a small ticket size of each order. Also, let's compare this output with the output we got in 3b, we find that all of these states are amongst the top 10 states which place maximum orders.

If Target is able to just increase the average spend by just 10% also, it will have huge benefits in terms of its total sales. Target can then reap the benefits of customer penetration.

Let's turn the table a bit and order it by Avg_order_price DESC, total_order_price DESC and evaluate what we get

Row	customer_state	Total_order_price	avg_order_price
1	PB	141545.72	248.33
2	AC	19680.62	234.29
3	RO	60866.2	233.2
4	AP	16262.8	232.33
5	AL	96962.06	227.08
6	RR	10064.62	218.8
7	PA	218295.85	215.92
8	SE	75246.25	208.44
9	PI	108523.97	207.11
10	TO	61485.33	204.27
11	CE	279464.03	199.9

In states like PB and AL, Target has a higher average order price, which means one single customer is contributing relatively more towards Target's revenue. If we compare this table with the table we got in 3b, we can observe that all of these states are not the largest contributors towards the total orders that they place but there average ticket size is relatively higher.

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

For looking at order freight, we need to evaluate the order items table. That will be joined via orders table with customers table as in the previous scenario.

```
SELECT c.customer_state, ROUND(SUM(oi.freight_value),2) AS total_freight, ROUND(AVG(oi.freight_value),2)
AS avg_freight
FROM `Target.order_items` As oi
JOIN `Target.orders` AS o
ON oi.order_id = o.order_id
JOIN `Target.customers` As c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY total_freight DESC, avg_freight DESC
```

Row	customer_state	total_freight	avg_freight
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04
11	ES	49764.6	22.06

Here, in the states where Target is quite popular, the avg freight value per order is close to 20-25. This indicates that there is economic freight pricing for Target. However, just by ordering the table by avg_freight DESC, total_freight DESC, we might get a different insight.

Row	customer_state	total_freight	avg_freight
1	RR	2235.19	42.98
2	PB	25719.73	42.72
3	RO	11417.38	41.07
4	AC	3686.75	40.07
5	PI	21218.2	39.15
6	MA	31523.77	38.26
7	TO	11732.68	37.25
8	SE	14111.47	36.65

Above table is implying that Target is spending double the freight per order In States where it isn't so popular as compared to places where it is popular. This shows that there are economies of scale that can be reaped further. Target must definitely look into shrinking its freight costs in the above regions.

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_estimated_delivery_date - order_delivered_customer_date

The above calculation can be done using the orders table itself using the TIMESTAMP_DIFF() function of bigquery AS shown below:

```
 SELECT order_id, order_delivered_customer_date,order_purchase_timestamp, order_estimated_delivery_date,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS
time_to_deliver,TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY) AS
diff_estimated_delivery

FROM `Target.orders`
WHERE order_delivered_customer_date IS NOT NULL
ORDER BY order_id
```

Below, we are showing the output basis the ascending order of the order_id just to demonstrate how varied the time_to_deliver and diff_estimated_delivery fields are. However, there are cases where the delivery is superfast and orders where it has taken almost half a year in delivery. Similarly, there are instances where the

diff_estimated_delivery is positive, which indicates faster delivery as compared to what had been promised to the customer, cases where the diff_estimated delivery is 0, indicating no discrepancy between order expected date and actual delivery date, Also, there are cases when the metric is negative indicating that more time was taken in the delivery than the expectation. We have encountered all such cases while looking at the results of the above query.

Row	order_id	order_delivered_customer_date	order_purchase_timestamp	order_estimated_delivery_date	time_to_deliver	diff_estimated_deliv
1	00010242fe8c5a6d1ba2dd792...	2017-09-20 23:43:48 UTC	2017-09-13 08:59:02 UTC	2017-09-29 00:00:00 UTC	7	8
2	00018f77f2f0320c557190d7a1...	2017-05-12 16:04:24 UTC	2017-04-26 10:53:06 UTC	2017-05-15 00:00:00 UTC	16	2
3	000229ec398224ef6ca0657da...	2018-01-22 13:19:16 UTC	2018-01-14 14:33:31 UTC	2018-02-05 00:00:00 UTC	7	13
4	00024acbcdf0a6daa1e931b03...	2018-08-14 13:32:39 UTC	2018-08-08 10:00:35 UTC	2018-08-20 00:00:00 UTC	6	5
5	00042b26cf59d7ce69dfabb4e...	2017-03-01 16:42:31 UTC	2017-02-04 13:57:51 UTC	2017-03-17 00:00:00 UTC	25	15
6	00048cc3ae777c65dbb7d2a06...	2017-05-22 13:44:35 UTC	2017-05-15 21:42:34 UTC	2017-06-06 00:00:00 UTC	6	14
7	00054e8431b9d7675808bcb8...	2017-12-18 22:03:38 UTC	2017-12-10 11:53:48 UTC	2018-01-04 00:00:00 UTC	8	16
8	000576fe39319847cbb9d288c...	2018-07-09 14:04:07 UTC	2018-07-04 12:08:27 UTC	2018-07-25 00:00:00 UTC	5	15
9	0005a1a1728c9d785b8e2b08...	2018-03-29 18:17:31 UTC	2018-03-19 18:40:33 UTC	2018-03-29 00:00:00 UTC	9	0
10	0005f50442cb953dcd1d21e1f...	2018-07-04 17:28:31 UTC	2018-07-02 13:59:39 UTC	2018-07-23 00:00:00 UTC	2	18

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

Similar to the previous Q4 part c, the average freight value can be assessed using the following code:

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) AS avg_freight
FROM `Target.order_items` As oi
JOIN `Target.orders` AS o
ON oi.order_id = o.order_id
JOIN `Target.customers` As c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY avg_freight DESC
LIMIT 5
```

This will give the top 5 states with the highest freight value as follows:

Row	customer_state	avg_freight
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

Just ordering the above code in ascending order, we will get the top 5 states with lowest freight value

```
SELECT c.customer_state, ROUND(AVG(oi.freight_value),2) AS avg_freight
FROM `Target.order_items` As oi
JOIN `Target.orders` AS o
ON oi.order_id = o.order_id
JOIN `Target.customers` As c
ON o.customer_id = c.customer_id
```

```
GROUP BY c.customer_state
ORDER BY avg_freight
LIMIT 5
```

Row	customer_state	avg_freight
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 & lowest average delivery time.

For this, we will join the orders table with the customers table

```
SELECT c.customer_state,
ROUND(AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY)),2) AS
avg_delivery_time
FROM `Target.orders` AS o
JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_time
```

The result is as follows:

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	RS	14.82
7	RJ	14.85
8	GO	15.15
9	MS	15.19
10	ES	15.33

This means that the smallest avg delivery time is 8.3 days in the most popular state. This is not a very good average to be on, even if extreme cases are considered, and this is an area where Target can definitely work on.

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.

```
SELECT c.customer_state,
ROUND(Avg(TIMESTAMP_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)),2)
AS avg_diff_estimated_delivery
FROM `Target.orders` AS o
JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY avg_diff_estimated_delivery DESC
LIMIT 5
```

Superfast delivery happens in the following states

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

This means that in AC the ordered product reached the customer 20 days in advance of the estimated date of delivery. However, the fact that the estimated date of delivery is more than 20 days long in itself is not a very efficient way of delivering.

6. Analysis based on the payments

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

```
SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) As Year, EXTRACT(MONTH FROM
o.order_purchase_timestamp) AS Month, p.payment_type, COUNT(o.order_id) AS num_orders
FROM `Target.orders` AS o
JOIN `Target.payments` As p
ON o.order_id = p.order_id
GROUP BY Year, Month, p.payment_type
ORDER BY Year, Month, p.payment_type
```

Row	Year	Month	payment_type	num_orders
31	2017	7	UPI	845
32	2017	7	credit_card	3086
33	2017	7	debit_card	22
34	2017	7	voucher	364
35	2017	8	UPI	938
36	2017	8	credit_card	3284
37	2017	8	debit_card	34
38	2017	8	voucher	294
39	2017	9	UPI	903
40	2017	9	credit_card	3283

Here we observe that, customers are making most of their purchases using credit cards followed by using UPI method for payment.

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

The question seeks to explore the number of orders against the no. of instalments that have been paid.

This can be done via using the following code:

```
SELECT payment_installments, COUNT(DISTINCT order_id)
from `Target.payments`
GROUP BY payment_installments
ORDER BY payment_installments
```

This means that the customers are opting for lesser no. of instalments for the product they are purchasing as the numbers are trailing down as the numbers of instalments are increasing. This also means that majority customers don't want to take up liabilities, they want to pay off as soon as possible.

7. Additional Questions

a. To check the number of orders against each customer, in order to look for repeat business

```
SELECT customer_id, COUNT(order_id) AS num_of_orders_per_customer
FROM `Target.orders`
GROUP BY customer_id
HAVING COUNT(order_id) > 1
ORDER BY customer_id
```

There is no data to display.

As had been discussed in question 3b, we again reconfirm that no customers have placed any repeat orders since there is no data to display where the COUNT(order_id)>2 for a single customer.

b. To inspect the customer review comments that are present in the database to hint at the customer satisfaction levels

To give an indication of the customer satisfaction levels, we look at all the different review comments that customers have given

```
SELECT DISTINCT review_comment_title
FROM `Target.order_reviews`
```

Row	review_comment_title
1	*null*
2	The product came wrong
3	Wrong product
4	I hated it sa's at loss
5	Lack of respect
6	The cÃ
7	NON-RECEIPT
8	I didn't receive my order
9	I received the product
10	Delay
11	I don't want to evaluate
12	Delivery delay
13	delivery delay
14	Dissatisfied
15	Bad product
16	Late delivery
17	Productless
18	Do not entertain the product
19	Product not delivered
20	Pessimal
21	Product NÃ £ o came 250 ml

Row	review_comment_title
22	Delay DMS
23	PÃ © SSIMA
24	PÃ © ssimo
25	Product does not work
26	Caveat
27	Delay delivery
28	My product came wrong
29	Absence of follow -up
30	I didn't receive
31	I recommend it
32	Product NÃ £ o Received

Row	review_comment_title
40	Delivered wrong product
41	I recommend
42	Delay - SD Cart
43	I didn't receive the product
44	Product not received
45	PÃ © ssimo
46	5
47	Product still has not arrived
48	Upset
49	not delivered
50	PÃ © this I recommend

It is noteworthy that there is not a single positive comment about Target in all the diverse comments as shown and repeatedly the customers are grieving about the delay in delivery and have expressed their dissatisfaction for the same. This is definitely a Red Flag for Target, as on one hand they have been able to do some really good customer penetration but if they do not improve on the product delivery, very soon they might see a dip in sales and a negative word of mouth publicity.

c. To analyse the count of each review_score in the data

Since there is a one on one relationship in the data between each customer and each order, it is fair to count the total number of orders against each review score to assess the customer satisfaction level of Target as a brand

```
SELECT review_score,COUNT(DISTINCT order_id) As num_orders
FROM `Target.order_reviews`
GROUP BY review_score
ORDER BY review_score
```

Row	review_score	num_orders
1	1	11393
2	2	3148
3	3	8160
4	4	19098
5	5	57076

Here, we observe that around 75% of the customers have given 4 or 5 as a review score to Target, which is a positive for Target. However, the remaining 25% still don't think target as the best. This can be easily worked upon if there is better management in the Target supply chain.

d. To explore state wise seller distribution

This can be done using the sellers table

```
SELECT seller_state ,COUNT(seller_id) As num_sellers_state
FROm `Target.sellers`
GROUP BY seller_state
ORDER BY num_sellers_state DESC
```

Row	seller_state	num_sellers_state
1	SP	1849
2	PR	349
3	MG	244
4	SC	190
5	RJ	171
6	RS	129
7	GO	40
8	DF	30
9	ES	23
10	BA	19
11	CE	13
12	PE	9

It is noticeable that only SP has the majority seller concentration. If we look even more closely and try to correlate the delivery time in these states, we will observe that there is inverse relationship between the number of sellers and the delivery time in that State. Lesser the sellers, more is the delivery time in that region.

Let's reconfirm this by running the following code:

```
WITH del_time AS (
```

```
SELECT c.customer_state,
ROUND(AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY)),2) AS
avg_delivery_time
FROM `Target.orders` AS o
JOIN `Target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_time
), seller_state AS (
SELECT seller_state ,COUNT(seller_id) As num_sellers_state
FROm `Target.sellers`
GROUP BY seller_state
ORDER BY num_sellers_state DESC)

SELECT dt.customer_state, dt.avg_delivery_time,ss.num_sellers_state
FROM del_time AS dt
LEFT JOIN seller_state AS ss
ON dt.customer_state = ss.seller_state
ORDER BY dt.avg_delivery_time
```

Row	customer_state	avg_delivery_time	num_sellers_state
1	SP	8.3	1849
2	PR	11.53	349
3	MG	11.54	244
4	DF	12.51	30
5	SC	14.48	190
6	RS	14.82	129
7	RJ	14.85	171
8	GO	15.15	40
9	MS	15.19	5
10	ES	15.33	23
11	TO	17.23	null
12	MT	17.59	4

This strongly suggests that there is a lacuna in the supply side and sourcing of the products which is causing the delay in the delivery and if rectified, target can reap great benefits.

KEY OBSERVATIONS

1. Between Sep'2016 to Aug'2018, there has been an increasing trend noticed in the total orders placed by the customers. Notably, there has been a sharp 700% increase noticed between the orders placed between Jan'2017 and Jan'2018.
2. There is a one to one mapping between the number of customers and the number of orders placed which further adds up to saying that there is not even a single customer in the database who has placed the order twice or more.

3. Putting both the above observations together, we can safely assume that whatever growth in orders is observed has been because of new customers. Hence, Target has been successful in acquiring more and more customers.
4. In states where Target is popular (SP, RJ, MG etc), the average spend is relatively lesser than in those areas where it is not as popular (PB, AC, RO etc).
5. The minimum average delivery time across all the states is 8.3 days in the most popular state SP. In the other states, the average delivery time increases further ranging between 10 – 15 days or greater.
6. Notably, this long delivery time appears to be one of the chief reasons of customer dissatisfaction.

RECOMMENDATIONS AND ACTION ITEMS FOR TARGET

1. The increase in the number of orders placed between 2016 to 2018 is indicative of the customer penetration that Target has been trying to achieve in the Brazilian Market. This drive has to be continued in areas where it is still not as popular.

2. Target should aim at building strategies around increasing the average order size of each customer. They can do so by leveraging on certain customer behavioural patterns that we have observed and tie them up with some attractive discounts to facilitate the same.

Some ideas can be:

a. A typical Brazilian customer pays via credit card, Target can roll out schemes in partnership with credit card companies to offer cashbacks or special discounts on minimum spending.
b. Similarly, customers usually place their orders during afternoons between 1 to 6 pm. Special discounts can be rolled out during that period which may excite the customers to buy more.
c. Various discounts can be given and cross selling of other products can be encouraged to increase the average order size from each customer, which will increase the overall sales.

3. On an urgent basis, Target needs to work on reducing the delivery time of each product. So that it continues to retain its existing customers and encourages them to place repeat orders with Target. Also, along with minimising the delivery time, there has to be proper publicity of the same where in people get to know that Target is providing speed delivery.

Few ideas around this can be:
a. Supply chain management is key to achieving optimised delivery time of products. Target can consider setting up of warehouses in and around states that have a large customer base to reduce lead time of each product.
b. In areas where this is not possible or is costly, tie ups with more and more local sellers have to be made and incentives need to be given to them, so that Target can save on freight costs and freight time.

4. Additionally, a customer loyalty program needs to be in place where a customer is rewarded for making repeat purchases to encourage repeat sales.