Target SQL Report

Report by Chahat

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

Ques- 1a. Data type of all columns in the "customers" table

Screenshot :

Filter Enter property name or value

Field name	Type	Mode	Key
customer_id	STRING	NULLABLE	-
customer_unique_id	STRING	NULLABLE	-
customer_zip_code_prefix	INTEGER	NULLABLE	-
customer_city	STRING	NULLABLE	-
customer_state	STRING	NULLABLE	-

Ques-1b. 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 `targetsql.orders`
```

Screenshot:

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

Insights: The first was placed on 4th September 2016 and the last order was placed on 17th October 2018.

Ques-1c. Count the Cities & States of customers who ordered during the given period.

Query:

```
Select count(distinct customer_state) as States,

count(distinct customer_city) as Cities

from `targetsql.customers` c

join `targetsql.orders` o

on c.customer_id = o.customer_id
```

Screenshot :

Row	States	Cities
1	27	4119

Insights:

There are 27 States and 4119 Cities from which customers ordered .

2.**In-depth Exploration** :

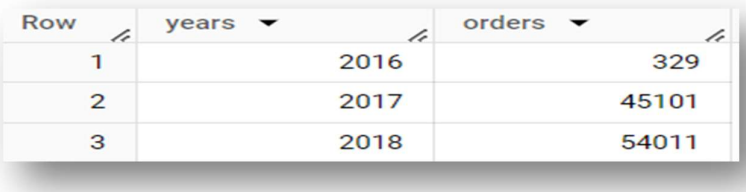
Ques-2a. Is there a growing trend in the no. of orders placed over the past years?

Query:

```
select extract(year from order_purchase_timestamp) as years,

count(order_id) as orders
from `targetsql.orders`
group by years
order by years
```

Screenshot :



Row	years	orders
1	2016	329
2	2017	45101
3	2018	54011

Insights:

The number of orders placed over the years was as follows:

2016: 329 orders (partial year starting from September)

2017: 45,101 orders

2018: 54,011 orders (until October)

1. The company started in **September 2016**, and in that partial year, there were **329 orders** placed, reflecting the early stage of the business.
2. A substantial growth occurred in **2017**, with **45,101 orders**, representing a **136x increase** from 2016. This dramatic rise suggests successful customer acquisition and possibly effective marketing strategies or product demand.
3. In **2018**, orders increased further to **54,011** by October (before the company's closure). This indicates a **19.8% growth** over the previous year, reflecting steady demand and customer retention.

Ques- 2b. 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 years,

extract(month from order_purchase_timestamp) as months,
count(order_id) as orders
from `targetsql.orders`
group by years , months
order by years, months
```

Screenshot:

Row	years	months	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

Insights:

The company missed sales in **November 2016**, but saw a sharp rise in **November 2017**, likely due to the **festive season** and **Thanksgiving**. Sales remained strong through **September 2018** but dropped significantly in **September and October 2018** due to the company's closure.

Recommendations:

1. Focus on **festive season marketing**, especially around **Thanksgiving**, which proved successful in **2017**.
2. Ensure consistent promotions in strong months like **November** to maintain growth.
3. In future ventures, plan a smoother **exit strategy** to avoid abrupt declines like in late **2018**.

Ques-2c. 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:

```
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  day_time,
count(order_id) as order_count
from `targetsql.orders`
group by day_time
order by  order_count desc;
```

Screenshot:

Row	day_time	order_count
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Insights:

1. Maximum orders were placed in the **afternoon (38,135 orders)**, followed by **night (28,331 orders)** and **morning (27,733 orders)**.

2. The afternoon appears to be the peak time for Brazilian customers, likely due to increased online activity during lunch breaks or post-work relaxation.

Recommendations:

1. Focus marketing efforts and promotions during **afternoon and night**, as these periods see the highest order volumes.
2. Introduce limited-time discounts in the **morning** to boost orders and further engage customers.
3. Dawn hours show low activity, so marketing resources may be better allocated to more active time periods.

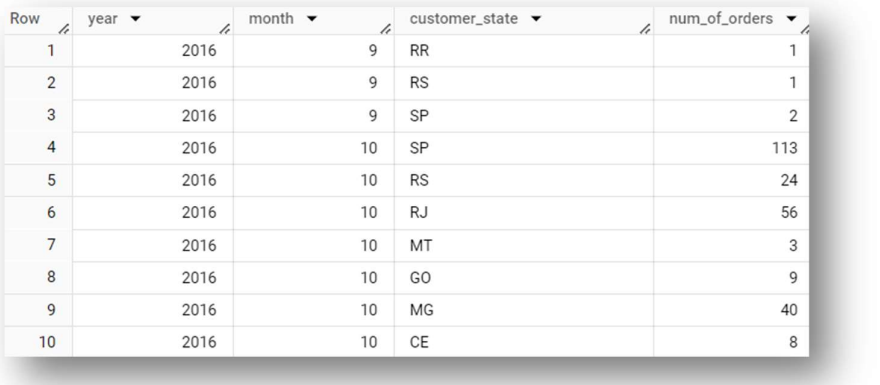
3.Evolution of E-commerce orders in the Brazil region:

Ques-3a Get the month on month no. of orders placed in each state.

Query:

```
select
extract(year from order_purchase_timestamp)as year,
extract(month from order_purchase_timestamp)as month,
customer_state,
count(order_id) as num_of_orders
from `targetsql.orders` o
join `targetsql.customers` c
on o.customer_id = c.customer_id
group by customer_state,month,year
order by year ,month
```

Screenshot:



Row	year	month	customer_state	num_of_orders
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	SP	113
5	2016	10	RS	24
6	2016	10	RJ	56
7	2016	10	MT	3
8	2016	10	GO	9
9	2016	10	MG	40
10	2016	10	CE	8

Insights:

1. (SP) has the largest market, followed by (RJ) and (MG). These three states consistently contributed the largest order volumes.
2. States like (MA), (AL) and (AC) generally had lower order volumes, indicating lower penetration in these regions.
3. November 2017 was a standout month with (SP) achieving an all-time high of over 3,000 orders. This month also saw significant contributions from (RJ) with 1,048 orders, and (MG) with 943 orders.

Recommendations:

1. (SP), (RJ), and (MG) have shown consistent high order volumes, focus future marketing campaigns, product promotions, and delivery optimizations in these states. These markets will likely provide the highest return on investment.
2. The company could look into further expanding in secondary states like (PR), (RS), and (SC), which have shown steady growth potential. Offering incentives like quicker deliveries, discounts, or regional promotions could boost orders in these regions.

Ques-3b. How are the customers distributed across all the states?

Query:

```
select

customer_state,
count(customer_id) as num_of_customers
from `targetsql.customers`
group by customer_state
order by num_of_customers desc
```

Screenshot:

Row	customer_state	num_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

Insights:

1. (SP) has the highest number of customers with 41,746, which is more than triple the number of the second state, (RJ) with 12,852 customers. This shows that the company's customer base is heavily concentrated in SP.
2. The top 3 states—(SP), (RJ), and (MG)—together account for a significant portion of the total customers.
3. States such as (RR) with 46 customers, (AP) with 68, and (AC) with 81, have very few customers. This indicates either low market penetration or demand in these areas.

Recommendations:

1. Since (SP), (RJ), and (MG) hold the majority of customers, the company should continue to invest in these regions with personalized marketing strategies to further strengthen their market share.
2. Use customer loyalty programs and targeted promotions to keep these customers engaged and ensure they remain loyal.
3. Regions like the (AM, RO, AC, AP, RR ,PE, CE, BA, PB) show significant room for growth. Launch targeted marketing campaigns in these regions to increase brand awareness and customer acquisition.
4. Collaborate with local influencers or brands in regions with lower penetration to build trust and gain visibility,
5. Provide region-specific discounts or products to appeal to the cultural and economic preferences of customers in lower-performing regions.

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

Ques-4a-- Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

Query:

```
with cte1 as
        (select extract(year from order_purchase_timestamp) as year ,
round(sum(payment_value),2) as total_cost
from `targetsql.payments` p
join `targetsql.orders` o
on p.order_id = o.order_id
where extract(month from order_purchase_timestamp)between 1 and 8 and
extract(year from order_purchase_timestamp)in (2017 , 2018)
group by year),
cte2 as
(select
max(case when year = 2017 then total_cost end) as total_2017,
max(case when year = 2018 then total_cost end) as total_2018
from cte1)
select total_2017 , total_2018 , round((total_2018 - total_2017) / total_2017 *
100,2) as percent_increase
from cte2
```

Screenshot:

Row	total_2017	total_2018	percent_increase
1	3669022.12	8694733.84	136.98

Insights:

From January to August, the total cost of orders increased from **3,669,022.12** in 2017 to **8,694,733.84** in 2018. This represents a **136.98% increase** in total cost year-over-year. Both the total values and the percentage increase are **rounded to 2 decimal places**.

Recommendations:

1. Investigate which factors contributed most to this 136.98% increase. Look at variables such as marketing campaigns, changes in pricing strategy, or the addition of new product lines to replicate similar growth in future years.
2. Since the revenue has increased significantly, ensure that the business operations (like logistics, customer service, and delivery) are scaling appropriately because poor service can impact on customer interests towards company.

Ques-4b-- Calculate the Total & Average value of order price for each state.

Query:

```
select
    c.customer_state,
round(sum(oi.price),2) as total_price,
round(avg(oi.price),2) as avg_price
from `targetsql.order_items` oi
join  `targetsql.orders` o
on o.order_id = oi.order_id
join `targetsql.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by total_price desc, avg_price desc
```

Screenshot:

Row	customer_state	total_price	avg_price
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

Insights:

1. The state of (SP) has the highest total order price, contributing around 5.2 million, with an average order price of 109.65, indicating a high volume of orders but relatively lower average price per order.
2. (RJ) and (MG) also show significant order activity, with total prices of 1.82 million and 1.58 million, and average order prices of 125.12 and 120.75, respectively.
3. (PB) and (AL) have the highest average order prices, with 191.48 and 180.89 respectively, despite having lower total sales, indicating fewer but more expensive orders.
4. All values, both total and average order prices, have been rounded to 2 decimal places for clarity.

Recommendations:

1. Focus marketing efforts in states with higher average order prices (PB, AL, PA) to boost high-value transactions.
2. Investigate why some states like SP have high total sales but relatively lower average order prices.

Ques-4c-- Calculate the Total & Average value of order freight for each state.

Query:

```
select
c.customer_state,
round(sum(oi.freight_value),2) as total_freight_value,
round(avg(oi.freight_value),2) as avg_freight_value
from `targetsql.order_items` oi
join  `targetsql.orders` o
on o.order_id = oi.order_id
join `targetsql.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by total_freight_value desc, avg_freight_value desc
```

Screenshot:

Row	customer_state	total_freight_value	avg_freight_value
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

Insights:

1. (SP) has the highest total freight value at 718.7K, with a relatively low average freight value of 15.15, indicating a large number of smaller shipments.
2. States like (RJ), (MG), and (RS) also have significant total freight values, but their average freight costs are higher, ranging from 20.53 to 21.74.
3. States such as (PB), (RR), and (AL) have some of the highest average freight costs, with values exceeding 42, indicating either higher shipping costs or lower shipment volumes in these regions.
4. All values, both total and average freight values, are rounded to 2 decimal places for clarity.

Recommendations:

1. Consider negotiating better shipping rates or implementing local distribution centres in states with high average freight costs, such as PB, RR, and AL, to reduce shipping cost for customers.
2. Investigate the reasons behind higher freight costs in specific states and start offering free shipping promotions in these states to promote more purchases.

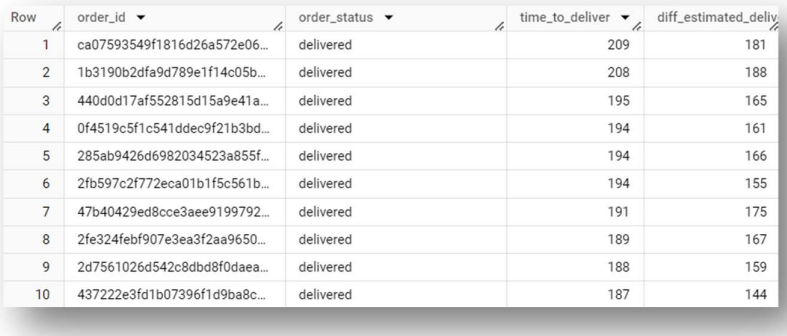
5.Analysis based on sales, freight and delivery time.

Ques-5a-- 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 ,
order_status,
date_diff(order_delivered_customer_date , order_purchase_timestamp , day) as
time_to_deliver_days,
date_diff(order_delivered_customer_date , order_estimated_delivery_date , day) as
diff_estimated_delivery_days
from `targetsql.orders`
where order_status = 'delivered' and
order_delivered_customer_date is not null and
order_estimated_delivery_date is not null
order by time_to_deliver_days desc
```

Screenshot:



Row	order_id	order_status	time_to_deliver	diff_estimated_deliv
1	ca07593549f1816d26a572e06...	delivered	209	181
2	1b3190b2dfa9d789e1f14c05b...	delivered	208	188
3	440d0d17af552815d15a9e41a...	delivered	195	165
4	0f4519c5f1c541ddec9f21b3bd...	delivered	194	161
5	285ab9426d6982034523a855f...	delivered	194	166
6	2fb597c2f772eca01b1f5c561b...	delivered	194	155
7	47b40429ed8cce3aee9199792...	delivered	191	175
8	2fe324febf907e3ea3f2aa9650...	delivered	189	167
9	2d7561026d542c8dbd8f0daea...	delivered	188	159
10	437222e3fd1b07396f1d9ba8c...	delivered	187	144

Insights:

1. The first 10 orders in this data shows extremely long delivery times, ranging from 187 to 209 days, which is highly unusual and suggests significant inefficiencies in the delivery process.
2. The difference between the actual delivery date and the estimated delivery date ranges from 144 to 188 days, highlighting that the orders were delivered much later than initially promised, indicating either logistical issues or external delays affecting the delivery timeline.
3. Some of the orders were delivered much earlier than the estimated delivery date, with differences ranging from 108 to 146 days early.
4. While early deliveries are positive, the large gap between the estimated and actual delivery suggests irregular behaviour, which might confuse or mislead customers about when they should expect their orders.

Recommendations:

1. The company should investigate and optimize its delivery process to ensure more accurate estimated delivery times. Implementing real-time tracking .
2. Consider notifying customers early about any potential delays so they can plan accordingly.
3. Early deliveries are great for customer satisfaction, but it's important to set accurate expectations to avoid customer confusion .
4. If the system shows an unusually late estimated delivery date (e.g., over 100 days) when the order can be delivered in 7-10 days customers looses interest so fix that issue as well.

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

Query:

```
with cte1 as
(select
c.customer_state,
round(avg(oi.freight_value),2) as avg_freight_value
from `targetsql.order_items` oi
join  `targetsql.orders` o
on o.order_id = oi.order_id
join `targetsql.customers` c
on o.customer_id = c.customer_id
group by c.customer_state),


highest_freight as (select
customer_state as highest_state,
avg_freight_value as highest_avg_freight,
row_number() over (order by avg_freight_value desc) as top_ranks
from cte1
order by  avg_freight_value desc
limit 5),


lowest_freight as (
select
customer_state as lowest_state,
avg_freight_value as lowest_avg_freight,
row_number() over (order by avg_freight_value asc) as last_ranks
from cte1
order by  avg_freight_value asc
limit 5)

select
h.highest_state,
h.highest_avg_freight,
l.lowest_state,
l.lowest_avg_freight
from highest_freight h
join lowest_freight l
on h.top_ranks = l.last_ranks
```

Screenshot:

Row	highest_state	highest_avg_freight	lowest_state	lowest_avg_freight
1	RR	42.98	SP	15.15
2	PB	42.72	PR	20.53
3	RO	41.07	MG	20.63
4	AC	40.07	RJ	20.96
5	PI	39.15	DF	21.04

Insights: **(RR)** has the highest average freight value of 42.98 and **(SP)** has the lowest of 15.15 this shows a massive difference in freight cost.

All values has been rounded to 2 decimals for clarity.

Ques-5c 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(order_delivered_customer_date ,
order_purchase_timestamp,day)),2) as time_to_deliver
from `targetsql.customers` c
join `targetsql.orders` o
on c.customer_id = o.customer_id
where order_status = 'delivered'
group by c.customer_state),


highest_delivery as
(select
customer_state as highest_time,
time_to_deliver,
row_number() over(order by time_to_deliver desc) as ranks
from cte1
order by time_to_deliver desc
limit 5),

lowest_delivery as
(select
customer_state as lowest_time,
time_to_deliver,
row_number() over(order by time_to_deliver asc) as ranks
from cte1
order by time_to_deliver asc
limit 5)

select
highest_time,
hd.time_to_deliver,
lowest_time,
ld.time_to_deliver
from highest_delivery hd
join lowest_delivery ld
on hd.ranks = ld.ranks
```

Screenshot:

Row	highest_time	time_to_deliver	lowest_time	time_to_deliver_1
1	RR	28.98	SP	8.3
2	AP	26.73	PR	11.53
3	AM	25.99	MG	11.54
4	AL	24.04	DF	12.51
5	PA	23.32	SC	14.48

Insights:

The states with the highest average time to deliver include **RR (28.98 days)**, **AP (26.73 days)**, and **AM (25.99 days)**, indicating significant delays in these regions. On the other hand, the states with the lowest average delivery times, such as **SP (8.3 days)**, **PR (11.53 days)**, and **MG (11.54 days)**, demonstrate much faster delivery speeds.

All values has been rounded to 2 decimals for clarity.

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

Query: SELECT

```
c.customer_state,
round(avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,
DAY)),2) as avg_delivery_time,
round(avg(date_diff(order_estimated_delivery_date, order_purchase_timestamp,
DAY)),2) as avg_estimated_delivery_time,
round(avg(date_diff(order_delivered_customer_date, order_estimated_delivery_date,
DAY)),2) as avg_difference
FROM `targetsql.orders` o
join `targetsql.customers` c
on o.customer_id = c.customer_id
where order_status = 'delivered'
group by c.customer_state
order by avg_difference asc
limit 5;
```

Screenshot:

Row	customer_state	avg_delivery_time	avg_estimated_deliv	avg_difference
1	AC	20.64	40.72	-19.76
2	RO	18.91	38.39	-19.13
3	AP	26.73	45.87	-18.73
4	AM	25.99	44.92	-18.61
5	RR	28.98	45.63	-16.41

Insights: The states AC, RO, AP, AM, and RR have consistently faster actual delivery times compared to their estimated delivery.

All values has been rounded to 2 decimals for clarity.

6.Analysis based on the payments:

Ques-6a -- Find the month on month no. of orders placed using different payment types.

Query: `select`

```
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
p.payment_type,
count(o.order_id) as orders,
from `targetsql.orders` o
join `targetsql.payments` p
on o.order_id = p.order_id
group by year , month, p.payment_type
order by year , month
```

Screenshot:

Row	year	month	payment_type	orders
1	2016	9	credit_card	3
2	2016	10	credit_card	254
3	2016	10	UPI	63
4	2016	10	voucher	23
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	583
8	2017	1	UPI	197
9	2017	1	voucher	61
10	2017	1	debit_card	9
11	2017	2	credit_card	1356

Insights:

1. **Dominance of Credit Card Payments**: Across all years and months, credit card payments are highest number in orders, with maximum at 5,691 in March 2018. This suggests that credit cards are the preferred payment method among customers.
2. **Growth of UPI Payments**: UPI payments have shown steady growth, from just 63 orders in October 2016 to over 1,500 orders in January 2018. This reflects the increasing adoption of digital payments in the market.
3. **Vouchers**: vouchers are relatively underutilized, with the number of orders remaining consistently low compared to credit cards and UPI.

Recommendation:

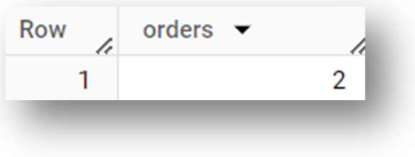
1. **Reevaluate Voucher Programs**: As by current program they are not appealing so can design in a way that customers get attracted.

Ques-6b-- Find the no. of orders placed on the basis of the payment installments that have been paid.

Query:

```
select
count(o.order_id) as orders
from `targetsql.orders` o
join `targetsql.payments` p
on o.order_id = p.order_id
where p.payment_installments <= 0
```

Screenshot:



Row	orders
1	2

Insights: There are only 2 orders where installments have been paid.