Target Business Case Study

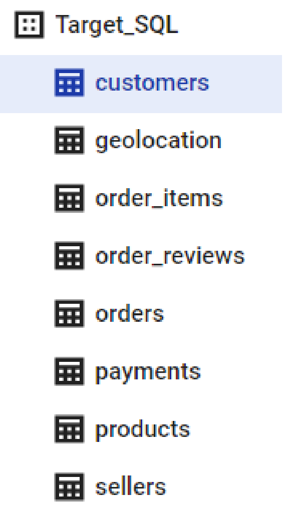
Name: Kaustubh mate

Batch: DSML Beginner July2022

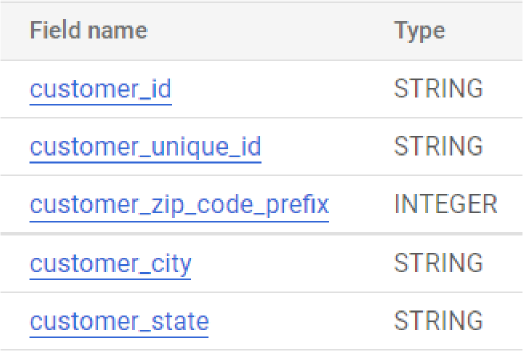
1. Exploratory Analysis

a. Data types of columns in Table

- There are total 8 tables in Database.
- Data type of columns from one table, ‘customers’ is shown in following screenshot.



Target_SQL
- customers
- geolocation
- order_items
- order_reviews
- orders
- payments
- products
- sellers



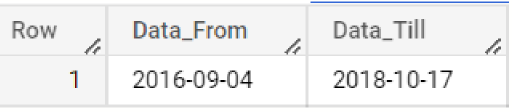
Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

b. Time period for which the data is given

Query:

```
SELECT min(date(order_purchase_timestamp)) as Data_From,
max(date(order_purchase_timestamp))as Data_Till
FROM `targetsql-1.Target_SQL.orders`
```

Output:



Row	Data_From	Data_Till
1	2016-09-04	2018-10-17

- We have data from 4th Sept 2016 to 17th Oct 2018

- **Assumption**: purchase date is taken into consideration for this analysis, order will be shipped, delivered after this purchase date, which is not considered.

c. Cities and States of customers ordered during given time period

We are not sure if customer data which is given is for all customers or for only those who made purchases between 4th Sept 2016 to 17th Oct 2018. So Inner join is used to get data of customers who made purchases between given period

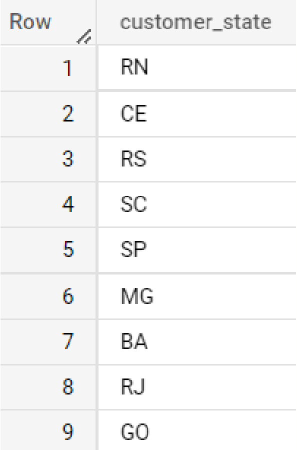
Query1:

```
SELECT distinct c.customer_city FROM targetsql-1.Target_SQL.orders o
inner join targetsql-1.Target_SQL.customers c
on o.customer_id = c.customer_id
```

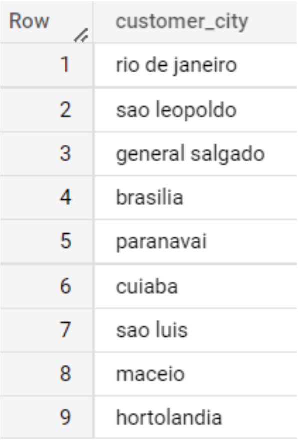
Query2:

```
SELECT distinct c.customer_state FROM targetsql-1.Target_SQL.orders o
inner join targetsql-1.Target_SQL.customers c
on o.customer_id = c.customer_id
```

Output:



Row	customer_state
1	RN
2	CE
3	RS
4	SC
5	SP
6	MG
7	BA
8	RJ
9	GO



Row	customer_city
1	rio de janeiro
2	sao leopoldo
3	general salgado
4	brasilia
5	paranavai
6	cuiaba
7	sao luis
8	maceio
9	hortolandia

2. In depth exploration

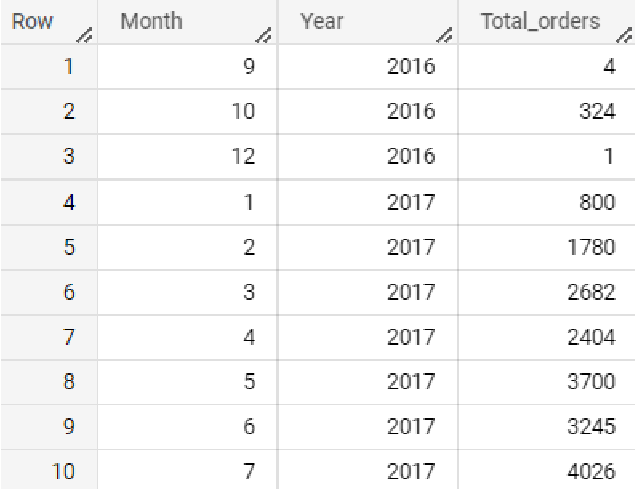
a. Monthly Trend:

To see trend on ecommerce in Brazil, we should look no. of orders vs Time. Here I have retrieve no. of orders month wise.

Query :

```
SELECT extract (month from order_purchase_timestamp) as Month,
extract (year from order_purchase_timestamp) as Year,
count(*) as Total_orders
FROM `targetsql-1.Target_SQL.orders`
group by Year,Month order by Year,Month
```

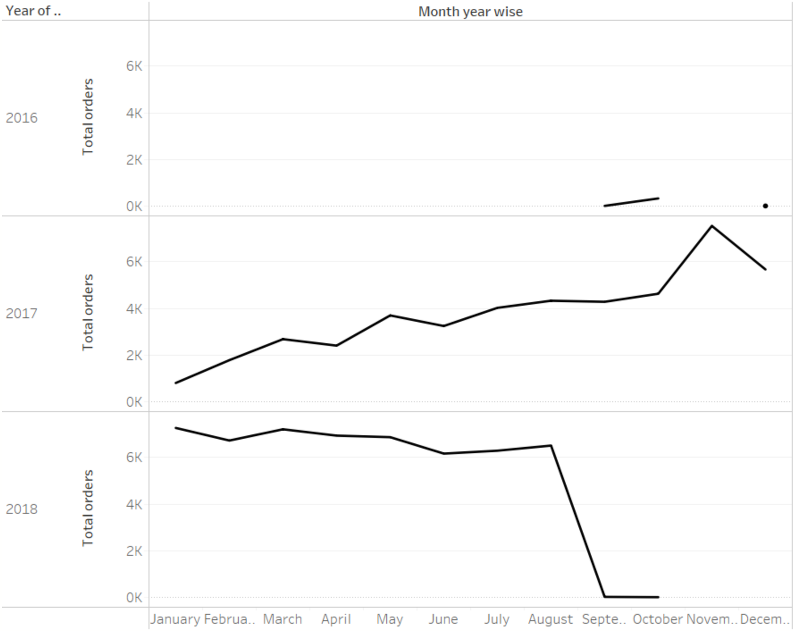
Output:



Row	Month	Year	Total_orders
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245
10	7	2017	4026

Data Visualisation:

Data for Nov2016 is missing, Data is exported as .xlsx file and imported in tableau to check trend



- From graph we can see that data doesn’t look proper for Year 2016, sept 2018 and Oct 2018.
- We see growing trend in order in year 2017
- Trend is almost constant for year 2018 (data up to august is considered)
- Can’t say Seasonal trend month wise, It’s better to have data for more number of years

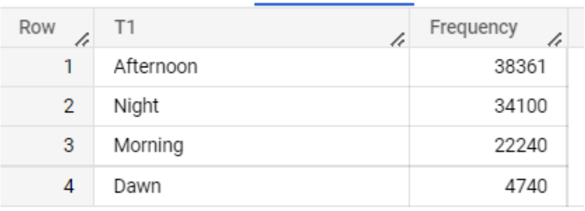
b. Daily trend:

To see at what time Brazilian customers tend to go for shopping, Timespan is divided into 4 Spans

Query :

```
select T1, count(*) as Frequency from (
SELECT time(order_purchase_timestamp) as T,
case
when time(order_purchase_timestamp) < "6:00:00" then "Dawn"
when time(order_purchase_timestamp) < "12:00:00" then "Morning"
when time(order_purchase_timestamp) < "18:00:00" then "Afternoon"
else "Night"
end as T1
FROM `targetsql-1.Target_SQL.orders`)
group by T1
order by Frequency desc
```

Output:



Row	T1	Frequency
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740

Assumptions: New day starts at 12 am, up to 5:59:59 am is dawn, from 6am-11:59:59 am is morning, 12pm -5:59:59 pm is afternoon and after 6 pm its night.

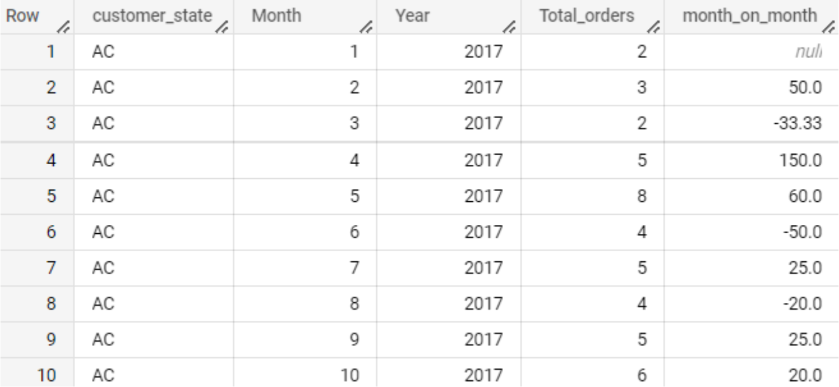
3. Evolution of E commerce orders in Brazil region
   a. Month on month orders by state

For each state, no of orders are compared with previous month and percentage change is calculated

Query:

```
select *,
round(((total_orders-
lag(total_orders,1)over(order by customer_state,Year,Month))*100/lag(
total_orders,1)over(order by customer_state,Year,Month)),2) as month_
on_month
from (
SELECT customer_state,extract (month from order_purchase_timestamp) a
s Month,
extract (year from order_purchase_timestamp) as Year,
count(*) as Total_orders
FROM `targetsql-1.Target_SQL.orders` o inner join targetsql-
1.Target_SQL.customers c on c.customer_id = o.customer_id
group by customer_state,Year,Month
order by Year,Month)
order by customer_state,Year,Month
```

Output:



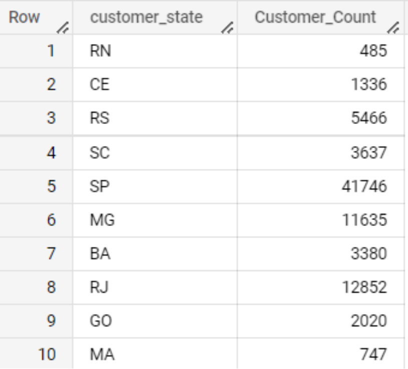
Row	customer_state	Month	Year	Total_orders	month_on_month
1	AC	1	2017	2	*null*
2	AC	2	2017	3	50.0
3	AC	3	2017	2	-33.33
4	AC	4	2017	5	150.0
5	AC	5	2017	8	60.0
6	AC	6	2017	4	-50.0
7	AC	7	2017	5	25.0
8	AC	8	2017	4	-20.0
9	AC	9	2017	5	25.0
10	AC	10	2017	6	20.0

b. Distribution of customers across states in Brazil

Query:

```
SELECT customer_state,count(customer_unique_id) as Customer_Count
FROM `targetsql-1.Target_SQL.customers` group by customer_state
```

Output:



Row	customer_state	Customer_Count
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747

4. Impact on economy

a. % increase in cost of orders from 2017 to 2018

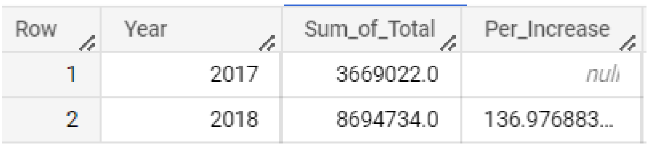
To get insight about how economy has impacted, we can see money movement between year 2017 and 2018

Query:

```
with cost_of_order as (
select * from (
SELECT extract (year from order_purchase_timestamp) as Year,extract (
month from order_purchase_timestamp) as Month,sum(payment_value) as T
otal
FROM `targetsql-1.Target_SQL.orders` o inner join targetsql-
1.Target_SQL.payments p on  o.order_id=p.order_id
group by Year,Month order by Year,Month) temp
where temp.Month in (1,2,3,4,5,6,7,8) and temp.Year in (2017,2018)),

summation as
(select Year,round(sum(Total)) as Sum_of_Total from cost_of_order gro
up by Year order by Year)
```

```
select *,(Sum_of_Total-
lag(Sum_of_Total) over(order by Year))/lag(Sum_of_Total) over(order b
y Year)*100 as Per_Increase from summation
```

Output:



Row	Year	Sum_of_Total	Per_Increase
1	2017	3669022.0	null
2	2018	8694734.0	136.976883...

Condition of economy looks good as there is increase in money movement by 137%

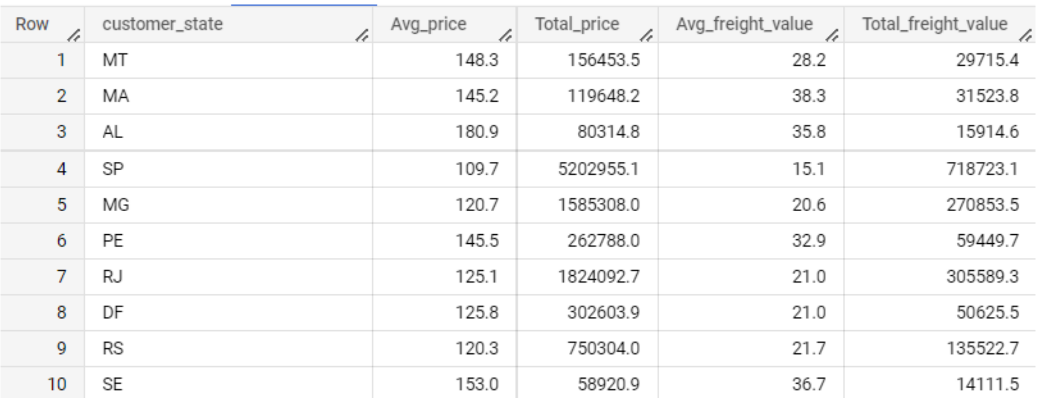
b. Mean & Sum of price and freight value by customer state

This analysis will tell state of economy state wise for overall period

Query:

```
with temp as (
SELECT * FROM `targetsql-
1.Target_SQL.customers` c inner join targetsql-
1.Target_SQL.orders o on c.customer_id = o.customer_id
inner join targetsql-
1.Target_SQL.order_items oid on oid.order_id = o.order_id)

select customer_state,round(avg(price),1) as Avg_price,round(sum(pric
e),1) as Total_price,
round(avg(freight_value),1) as Avg_freight_value,round(sum(freight_va
lue),1) as Total_freight_value
from temp group by customer_state
```

Output:



Row	customer_state	Avg_price	Total_price	Avg_freight_value	Total_freight_value
1	MT	148.3	156453.5	28.2	29715.4
2	MA	145.2	119648.2	38.3	31523.8
3	AL	180.9	80314.8	35.8	15914.6
4	SP	109.7	5202955.1	15.1	718723.1
5	MG	120.7	1585308.0	20.6	270853.5
6	PE	145.5	262788.0	32.9	59449.7
7	RJ	125.1	1824092.7	21.0	305589.3
8	DF	125.8	302603.9	21.0	50625.5
9	RS	120.3	750304.0	21.7	135522.7
10	SE	153.0	58920.9	36.7	14111.5

5. Analysis on sales, freight and delivery time

I have created CTEs which can be used to get required data

Query:

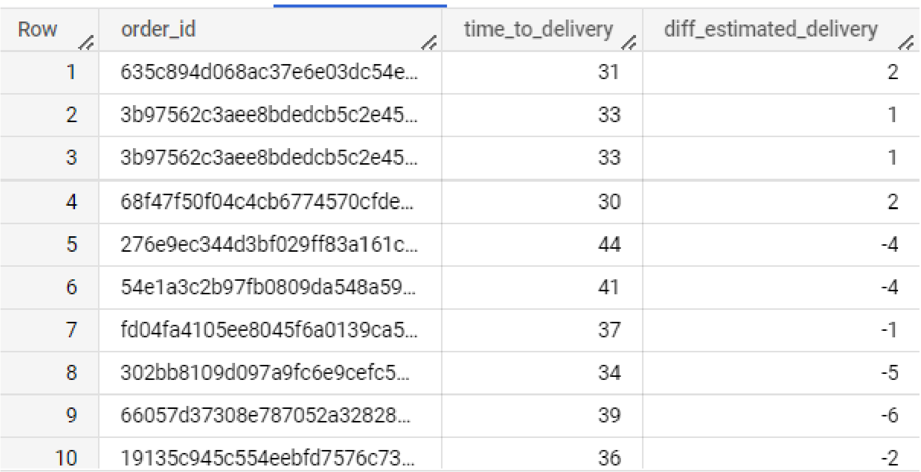
```
with table1 as (
SELECT o.order_id, date(order_purchase_timestamp) as purchase_date, date(ord
er_delivered_customer_date) as delivery_date, date(order_estimated_delivery_
date) as estimated,
customer_state ,freight_value
FROM `targetsql-1.Target_SQL.orders` o
inner join targetsql-
1.Target_SQL.customers c on o.customer_id = c.customer_id
inner join targetsql-
1.Target_SQL.order_items oid on oid.order_id = o.order_id
where  o.order_status = "delivered" and order_delivered_customer_date is not
 null),

table2 as (
select order_id, customer_state,freight_value, date_diff(delivery_date,purch
ase_date,day) as time_to_delivery,
date_diff(estimated,delivery_date,day) as diff_estimated_delivery from table
1),

table3 as (
select customer_state, round(avg(time_to_delivery),2) as average_delivery_ti
me,round(avg(freight_value),2) as average_freight_value,
round(avg(diff_estimated_delivery),2) as average_diff_estimated_delivery fro
m table2 group by customer_state)
```

a. Calculate days between purchasing, delivering and estimated delivery

Query:

```
        select order_id,time_to_delivery,diff_estimated_delivery from table2
```

Output:



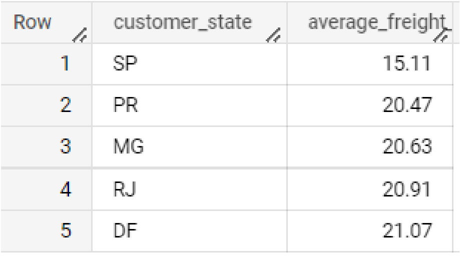
Row	order_id	time_to_delivery	diff_estimated_delivery
1	635c894d068ac37e6e03dc54e...	31	2
2	3b97562c3aee8bdedcb5c2e45...	33	1
3	3b97562c3aee8bdedcb5c2e45...	33	1
4	68f47f50f04c4cb6774570cfde...	30	2
5	276e9ec344d3bf029ff83a161c...	44	-4
6	54e1a3c2b97fb0809da548a59...	41	-4
7	fd04fa4105ee8045f6a0139ca5...	37	-1
8	302bb8109d097a9fc6e9cefc5...	34	-5
9	66057d37308e787052a32828...	39	-6
10	19135c945c554eebfd7576c73...	36	-2

b. Top 5 states with highest/lowest average freight value

Query:

Bottom 5 states with lowest average freight value

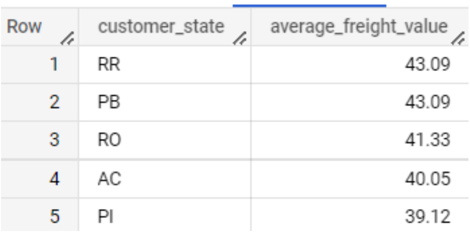
```
select customer_state,average_freight_value from table3 order by average_freight_value limit 5
```



Row	customer_state	average_freight
1	SP	15.11
2	PR	20.47
3	MG	20.63
4	RJ	20.91
5	DF	21.07

Top 5 states with highest average freight value

```
select customer_state,average_freight_value from table3 order by average_freight_value desc limit 5
```



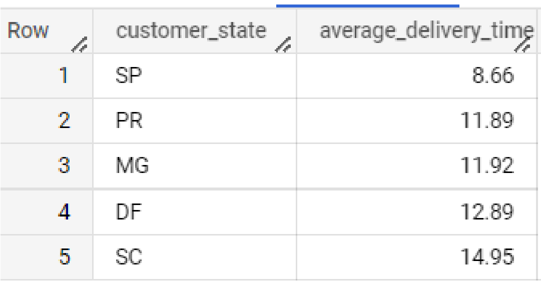
Row	customer_state	average_freight_value
1	RR	43.09
2	PB	43.09
3	RO	41.33
4	AC	40.05
5	PI	39.12

c. Top 5 states with highest/lowest average **time to delivery**

Query:

Top 5 performing states with lowest delivery time

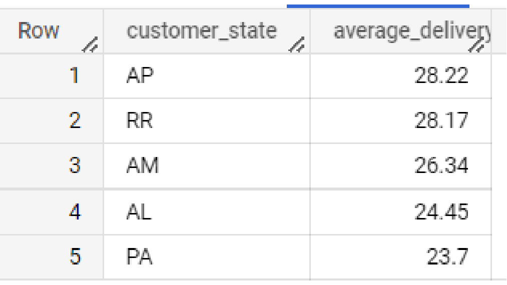
```
select customer_state, average_delivery_time from table3 order by average_delivery_time limit 5
```



Row	customer_state	average_delivery_time
1	SP	8.66
2	PR	11.89
3	MG	11.92
4	DF	12.89
5	SC	14.95

Bottom 5 performing states with highest delivery time

```
select customer_state, average_delivery_time from table3 order by average_de
livery_time desc limit 5
```



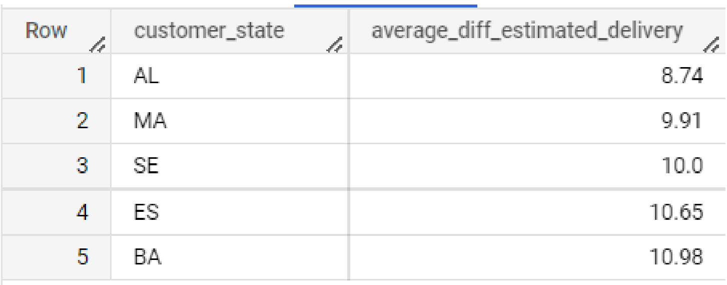
Row	customer_state	average_delivery
1	AP	28.22
2	RR	28.17
3	AM	26.34
4	AL	24.45
5	PA	23.7

d. Top 5 states **where delivery is really fast/ not so fast compared to estimated date**

Query:

Bottom 5 performing states with delivery not so fast

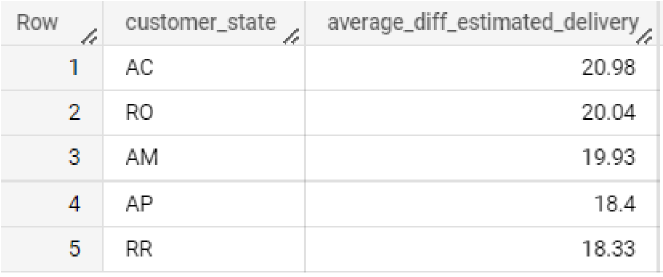
```
select customer_state, average_diff_estimated_delivery from table3 order by
average_diff_estimated_delivery limit 5
```



Row	customer_state	average_diff_estimated_delivery
1	AL	8.74
2	MA	9.91
3	SE	10.0
4	ES	10.65
5	BA	10.98

Top 5 performing states with delivery is really fast

```
select customer_state, average_diff_estimated_delivery from table3 order by
average_diff_estimated_delivery desc limit 5
```



Row	customer_state	average_diff_estimated_delivery
1	AC	20.98
2	RO	20.04
3	AM	19.93
4	AP	18.4
5	RR	18.33

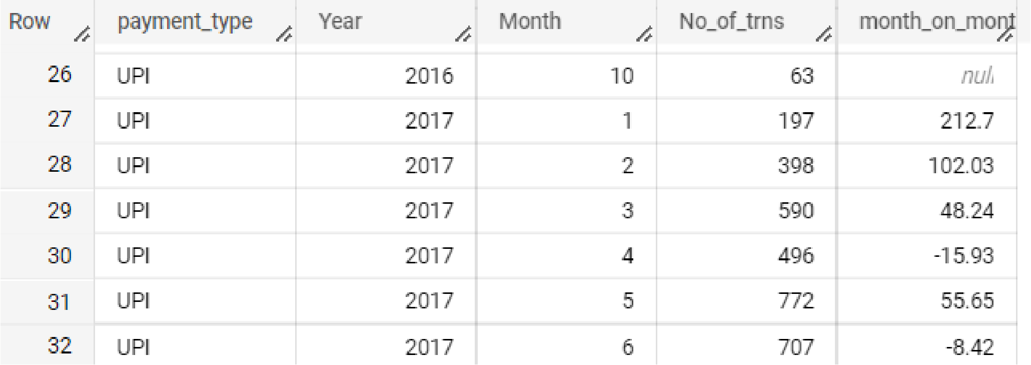
6. **Payment type analysis**

a. Month over Month count of orders for different payment types

Query:

```
with table1 as (

SELECT o.order_id,extract (month from order_purchase_timestamp) as Month,
extract (year from order_purchase_timestamp) as Year,payment_type FROM `targ
etsql-1.Target_SQL.payments` p
inner join targetsql-1.Target_SQL.orders o on p.order_id = o.order_id),

table2 as(
select payment_type,Year,Month, count(*) as No_of_trns from table1 group by
payment_type,Year,Month order by payment_type,Year,Month)

select *,
round((No_of_trns-
lag(No_of_trns,1) over(partition by payment_type order by Year,Month))*100/l
ag(No_of_trns,1) over(partition by payment_type order by Year,Month),2) as m
onth_on_month from table2
```

Output:



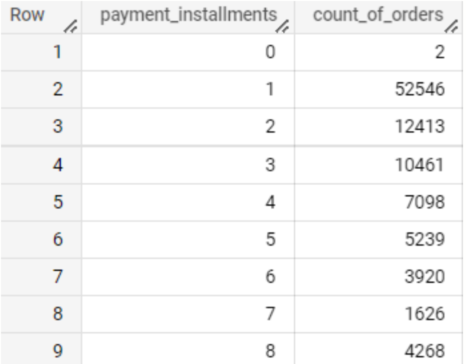
Row	payment_type	Year	Month	No_of_trns	month_on_mont
26	UPI	2016	10	63	nul
27	UPI	2017	1	197	212.7
28	UPI	2017	2	398	102.03
29	UPI	2017	3	590	48.24
30	UPI	2017	4	496	-15.93
31	UPI	2017	5	772	55.65
32	UPI	2017	6	707	-8.42

b. Count of orders based on the no. of payment instalments

Query:

```
SELECT  payment_installments, count(*) as count_of_orders FROM `targetsql-
1.Target_SQL.payments` group by payment_installments
```

Output:



Row	payment_installments	count_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268

7. **8. Insights and recommendations**

- For Target, in 2017 trend for order was growing. In 2018 it is constant but maintained above 6000 till august. sudden spike in Nov2017.
  Target should be ready with inventories for such sudden spikes

- Most customer tend to buy in afternoon (12am-6pm) and Night (6pm-12am), so target should make sure that queue time is less for customers by arranging more number of active counters open during that period, night time they can reduce no of active counters to save money on manpower

- States such as SP, RJ, MG has more than 10000 customer base, so more stores and warehouses should be opened to reduce transportation cost and manage inventory effectively, currently SP and MG in top 5 when it comes to lowest freight value and lowest time to deliver, but RJ is not in list for fast delivery so it should be focused

- Most customers tend to purchase in single, twice and thrice instalments, for most products, this option should be made available to increase sales. Banks who give such credits should be approached so it will be win-win situation for banks as well as target. Target can earn from these banks by making proper strategies as it will help such banks to increase account holders

- States like RR, AP, AC has low customer base. Proper marketing and discount strategies can improve customer base in such states