Target SQL Business Case Study

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

1.1 Data type of columns in a table

Query:

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

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

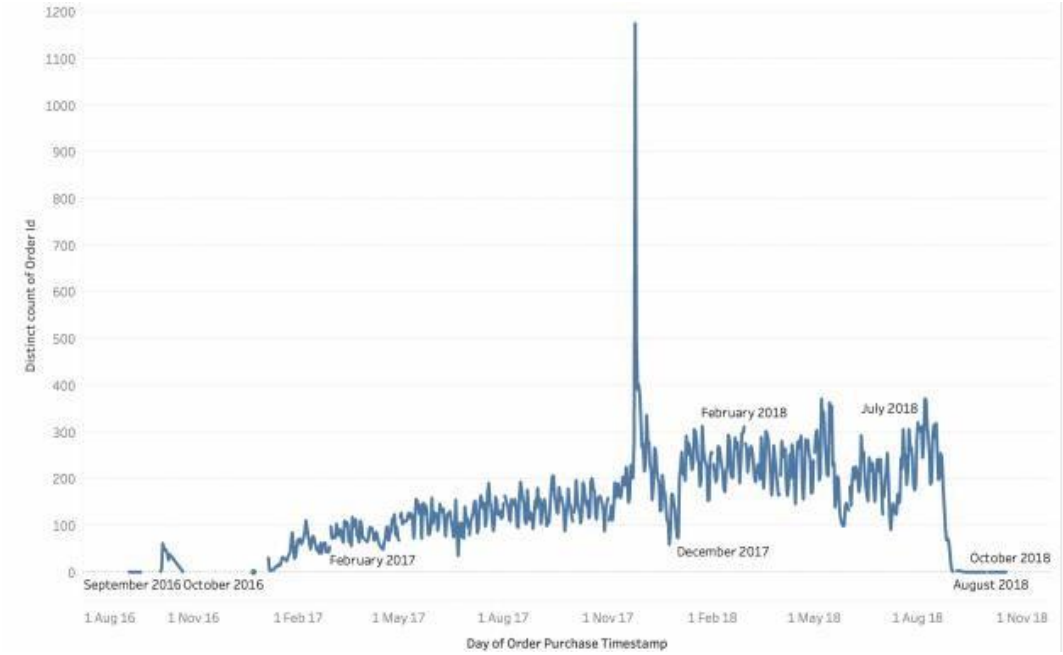
1.2 Time period for which the data is given

Query:

```
select      min(order_purchase_timestamp)      as
first_transaction_date,
max(order_purchase_timestamp)                  as
last_transaction_date
from `Target.orders`
```

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

Insights: E-commerce is growing at a really fast. The detailed analysis of the datasets are done to reflect the trend in the Brazilian market. After digging into the datasets for the given period of 2016-2018 , it could be seen they had first transaction on 4th Sept, 2016 (based on the given data).



1.3 Cities and States of customers ordered during the given period

Query:

```
select count(distinct customer_city) as
total_cities,
       count(distinct customer_state) as total_states
from `Target.customers`c
JOIN
`Target.ord
ers`o
on c.customer_id = o.customer_id
```

Row	total_cities	total_states
1	4119	27

Insights: The brand ‘Target’ has covered 27 states in Brazil

2. In-depth Exploration

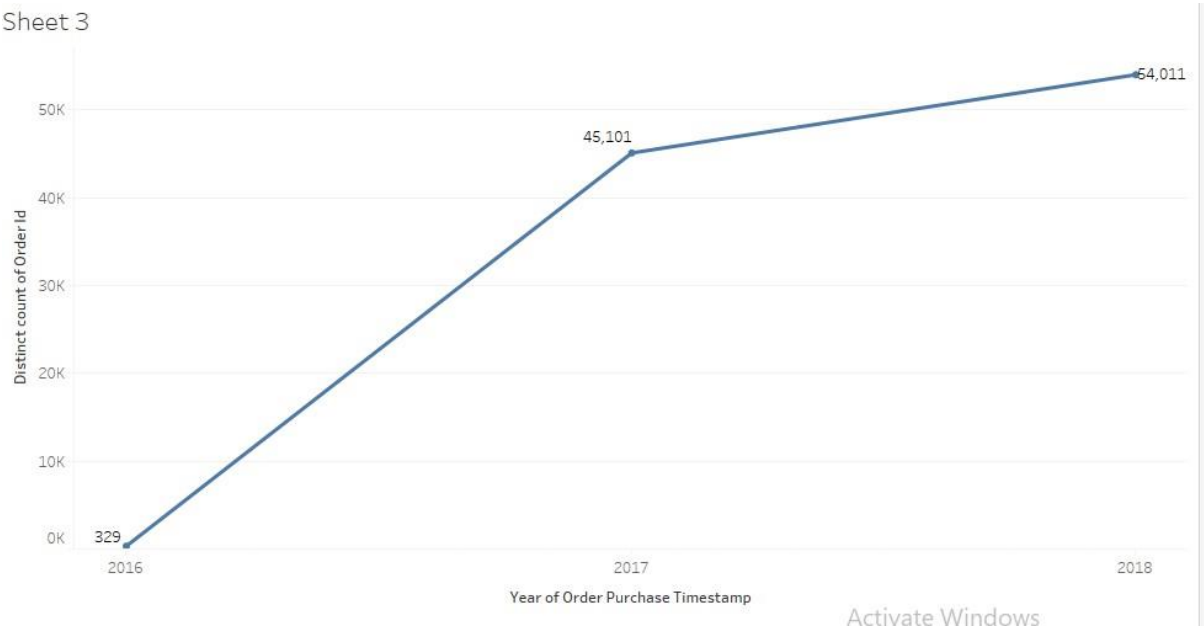
2.1 Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

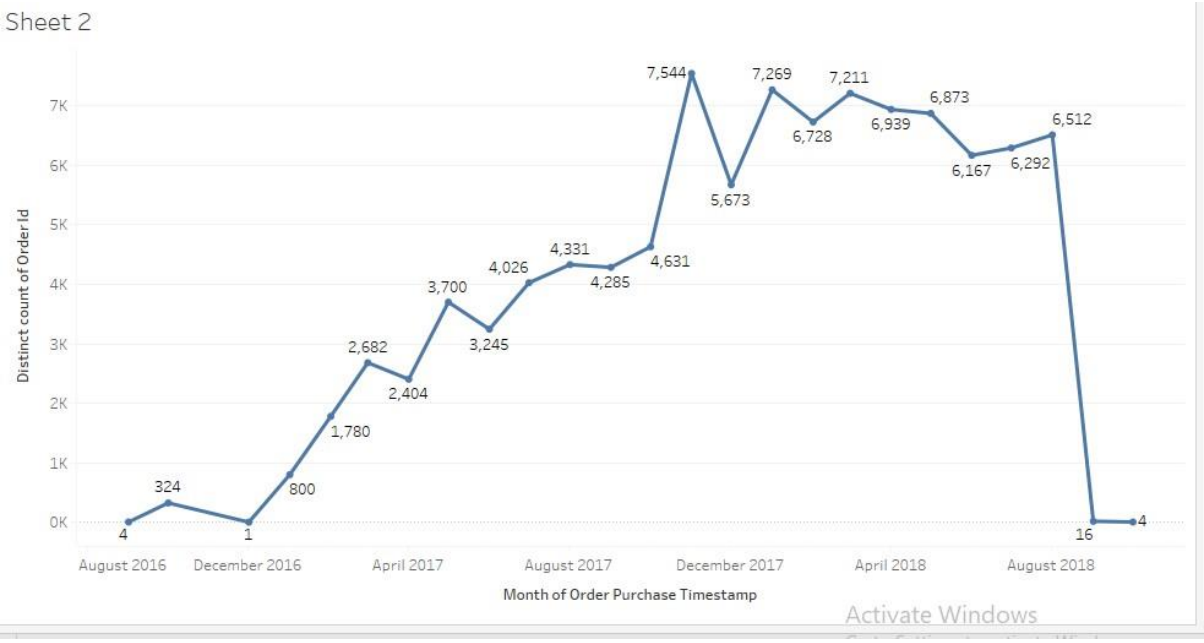
Query:

```
select extract(year from
order_purchase_timestamp)as
yearoforder,
count (distinct(order_id))as
no_of_orders  from `Target.orders`
group by yearoforder order by
yearoforder
```

Row	yearoforder	no_of_orders
1	2016	329
2	2017	45101
3	2018	54011

Insights: This evaluation clarifies the growing trend of e-commerce along the time in Brazil . The demand of the orders increases at a very fast pace from 2016 to 2018. The below mentioned graph gives a deep insight for the monthly count of orders allowing you to observe growing trend or seasonality at the specific months in e-commerce in Brazil. Majority of the customers prefer to buy the online stuff during the late hours of the day, that is at night.





2.2 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query:

```
SELECT  COUNT(*) as purchase_count,
CASE
WHEN order_purchase between '03:00:00'
and '06:00:00' THEN 'Dawn'
WHEN order_purchase between '06:00:00'
and '12:00:00'
THEN 'Morning'
WHEN  order_purchase between '12:00:00'
and '17:00:00'
THEN 'Afternoon'
ELS
E
'Ni
ght
'
END
AS
pur
cha
se_
```

```
tim
e
from (select*,extract(time from
order_purchase_timestamp)as order_purchase
from `Target.orders`) time_stamp GROUP BY
purchase_time order by purchase_count
```

Row	purchase_count	purchase_time
1	666	Dawn
2	22240	Morning
3	32212	Afternoon
4	44323	Night

Insight: In the given dataset, we have distributed the whole day into four parts Dawn, Morning, Afternoon, and Night as per the analysis we can see more orders were made in the night as compared to morning and afternoon we should focus on the mid-day buyers by providing some offers and sale.

Recommendations: As we can see night has the highest rate of buyers as compared to afternoon and morning customers. The company should focus on the categorized buyers with some strategies.

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

3.1 Get month-on-month orders by states

Query:

```
select extract(month from
o.order_purchase_timestamp)as month,
count(o.order_id)as order_id, c.customer_state,
from
`Target.orders`a
s      o      Join
`Target.customer
s`as      c      on
o.customer_id=c.
customer_id
group  by  month,
c.customer_state
order by month
```

Row	month	order_id	customer_state
1	1	990	RJ
2	1	3351	SP
3	1	151	DF
4	1	427	RS
5	1	99	CE
6	1	113	PE
7	1	443	PR
8	1	264	BA
9	1	971	MG
10	1	51	RN

Insight: company receives the more of the sale from 'SP' state

3.2 Distribution of customers across the states in Brazil

Query:

```
select distinct c.customer_state, count(distinct
c.customer_id)as number_of_customer
from `Target.customers`c
right join `Target.orders`o
on
c.customer_id=o.cust
omer_id
group by 1
```

Row	customer_state	number_of_custome
1	RJ	12852
2	RS	5466
3	SP	41746
4	DF	2140
5	PR	5045
6	MT	907
7	MA	747
8	AL	413
9	MG	11635
10	PE	1652

Insight: Brazil has the highest customer base from the state SP with the count of 41746, as compare to other states.

4. **Impact on Economy**

4.1 Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

Query:

```
with cte as ( select *
from (select round(sum(payment_value), 2) as
payment_sum_2017
from
`Target.orders` ord
INNER JOIN
`Target.payments`
pay on ord.order_id
= pay.order_id
where extract(year from order_purchase_timestamp) = 2017
and extract(month from
order_purchase_timestamp) between 1 and 8
) x
Cross Join
(
select round(sum(payment_value), 2) as
payment_sum_2018 from `Target.orders` ord
INNER JOIN
```

```
`Target.payments`
pay on ord.order_id
= pay.order_id
where extract(year from order_purchase_timestamp) = 2018
and extract(month from
order_purchase_timestamp) between 1 and 8
)y
group by payment_sum_2017, payment_sum_2018
)
select*,      round(((      payment_sum_2018-
payment_sum_2017)/   payment_sum_2017   ),4)
*100 as percent_change from cte
```

Row	payment_sum_2017	payment_sum_2018	percent_change
1	3669022.12	8694733.84	136.98

Insight: Here, we have analyzed the increase in the cost of orders for 2017 and 2018 respectively.

4.2 Mean & Sum of price and freight value by customer state Query:

```
select
c.customer_state,round(sum(oi.price+oi.freight_value),4
)as total_price,
round(avg(oi.price+oi.freight_value),4)as avg_price
from `Target.order_items` oi join `Target.orders` o
on o.order_id=oi.order_id join `Target.customers` c
on c.customer_id=o.customer_id
group by 1
```

Row	customer_state	total_price	avg_price
1	SP	5921678.12	124.8009
2	RJ	2129681.98	146.0787
3	PR	800935.44	139.5358
4	SC	610213.6	146.1239
5	DF	353229.44	146.8119
6	MG	1856161.49	141.3787
7	PA	217647.11	201.5251
8	BA	611506.67	160.9652

Insights: We have calculated the total no. of orders placed from each state and the total no. of orders from state “SP” is greater than among all states with a value of 5921678.12 and the highest average value for a month by state “PB” with 191.48.

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

5.1 Calculate days between purchasing, delivering and estimated delivery Query:

```
select
distinct order_id, extract(date from
order_purchase_timestamp) as Purchase_date,
extract(date from order_delivered_carrier_date) as
Delivery_Carrier_Date,
extract(date from order_estimated_delivery_date) as
Estimated_Delivery_Date,
(extract(day from order_delivered_carrier_date) -
extract(day from order_purchase_timestamp)) as
pur_delivery_days,
(extract(day from order_delivered_carrier_date) -
extract(day from order_estimated_delivery_date)) as
diff_est_delivery from `Target.orders` where
order_status = 'delivered'
```

Row	order_id	Purchase_date	Delivery_Carrier_Dat	Estimated_Delivery_[	pur_delivery_days	diff_est_delivery
1	44879a8f19c5e8a5e9278477b...	2018-08-23	2018-08-27	2018-10-04	4	23
2	0562291f2b37f55cc259053d2...	2018-04-30	2018-04-30	2018-06-06	0	24
3	a01f50d51f398895df76f09531...	2018-05-12	2018-05-28	2018-06-06	16	22
4	628923e74a955e432c826a2e0...	2018-05-11	2018-05-30	2018-06-06	19	24
5	ab879558e02a4aec8e7aa5941...	2018-04-21	2018-04-28	2018-06-06	7	22
6	b10350ed7f626af6db79354ad...	2018-05-23	2018-05-28	2018-06-06	5	22
7	656ada45a719393ba3e97d8d...	2018-04-27	2018-04-30	2018-06-06	3	24
8	d1594d3b636b86cf4ce22e2ca...	2018-04-29	2018-05-29	2018-06-06	0	23
9	a8214c9e03a43c85448925684...	2018-05-22	2018-05-30	2018-06-06	8	24
10	7a74611af770d37eb88aa4c25...	2018-05-22	2018-05-29	2018-06-06	7	23
11	0ed4bf3a5a970c8b076ce4cce...	2018-05-22	2018-05-28	2018-06-06	6	22
12	3407bfcbaa0cb49c244ededbc...	2018-04-21	2018-04-30	2018-06-06	9	24

5.2 Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

Query:

```
(SELECT
      concat("Highest_avg_delivery_time:"," ",c.customer_state) as
STATE,
      AVG(o.order_delivered_customer_date -
o.order_purchase_timestamp) AS avg_delivery_time
  FROM `Target_sql.orders` o
  INNER JOIN `Target_sql.customers` c ON o.customer_id =
c.customer_id
  GROUP BY c.customer_state
  ORDER BY avg_delivery_time desc
  LIMIT 5)
union all

(SELECT
      concat("Lowest_avg_delivery_time:"," ",c.customer_state) AS
STATE,
      AVG(o.order_delivered_customer_date -
o.order_purchase_timestamp) AS avg_delivery_time
  FROM `Target_sql.orders` o
  INNER JOIN `Target_sql.customers` c ON o.customer_id =
c.customer_id
  GROUP BY c.customer_state
  ORDER BY avg_delivery_time ASC
  LIMIT 5)
```

Row	STATE	avg_freight
1	Highest_avg_freight_value: RR	42.98
2	Highest_avg_freight_value: PB	42.72
3	Highest_avg_freight_value: RO	41.07
4	Highest_avg_freight_value: AC	40.07
5	Highest_avg_freight_value: PI	39.15
6	Lowest_avg_freight_value: SP	15.15
7	Lowest_avg_freight_value: PR	20.53
8	Lowest_avg_freight_value: MG	20.63
9	Lowest_avg_freight_value: RJ	20.96
10	Lowest_avg_freight_value: DF	21.04

Insights: We have calculated the top 5 highest and lowest average freight and we can observe that the top highest average freight is for the state “RR” of 42.98 and lowest for the state “SP” of 15.15. We calculated by using union all operator

5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Query:

```
select distinct customer_state, avg(freight_value) as
meanfreight, avg(extract(datetime from
order_delivered_customer_date) - extract(datetime
from order_purchase_timestamp)) as time_to_delivery,
avg(extract(datetime from order_estimated_delivery_date) -
extract(datetime
from order_delivered_customer_date)) as
diff_estimated_delivery from `Target.customers` cust
Inner join
`Target.orders`
ord
on cust.customer_id = ord.customer_id
Inner join
`Target.order_items`
ord_items                on
ord_items.order_id        =
ord.order_id          where
order_status = 'delivered'
group                    by
cust.customer_state
```

Row	customer_state	meanfreight	time_to_delivery	diff_estimated_delivery
1	GO	22.56286780851...	0-0 14 33:40:48.375933245	0-0 11 14:18:7.996486605
2	SP	15.11518235446...	0-0 8 17:22:20.194547921	0-0 10 12:17:14.311513533
3	RS	21.61319204434...	0-0 14 28:31:32.063916517	0-0 13 10:22:7.941790314
4	BA	26.48755633994...	0-0 18 29:56:34.191963073	0-0 10 6:55:31.734727124
5	MG	20.62634252090...	0-0 11 23:34:48.313564571	0-0 12 15:23:35.678460823
6	MT	27.99691417550...	0-0 17 23:4:49.308582449	0-0 13 21:30:17.274831243
7	RJ	20.91143604610...	0-0 14 27:32:58.179594145	0-0 11 7:17:3.752386339
8	SC	21.50735904320...	0-0 14 23:56:36.372711740	0-0 10 20:50:21.306809860
9	SE	36.57317333333...	0-0 20 35:12:59.317333333	0-0 9 7:49:3.408
10	PE	32.69333333333...	0-0 17 30:42:8.667239404	0-0 12 18:20:47.015463917
11	TO	37.43503225806...	0-0 17 10:45:40.490322580	0-0 11 15:36:27.777419354
12	CE	32.73449509116...	0-0 20 23:41:56.934081346	0-0 10 9:51:29.995792426
13	PR	20.47181625066...	0-0 11 22:43:59.571074526	0-0 12 19:0:31.719242343
14	PA	35.62901328273...	0-0 23 18:5:50.211574952	0-0 13 13:39:52.679316888

Sort the data to get the following:

5.4 Top 5 states with highest/lowest average freight value - sort in desc/asc

limit 5

Query:

: Top 5 states with lowest average freight value

```
select
c.customer_state,round(avg(oi.freight_value),4
) as lowest_avg_freight_value from
`Target.customers`c join `Target.orders`o on
c.customer_id=o.customer_id join
`Target.order_items`oi on
o.order_id=oi.order_id group by 1 order by 2
limit 5
```

Row	customer_state	lowest_avg_freight_v
1	SP	15.1473
2	PR	20.5317
3	MG	20.6302
4	RJ	20.9609
5	DF	21.0414

Top 5 states with highest average freight value :

Query:

```
select
c.customer_state,round(avg(oi.freight_value),4
)as highest_avg_freight_value
from `Target.customers`c join `Target.orders`o
on c.customer_id=o.customer_id
join `Target.order_items`oi
on o.order_id=oi.order_id
group by 1
order by 2 desc
limit 5
```

Row	customer_state	highest_avg_freight
1	RR	42.9844
2	PB	42.7238
3	RO	41.0697
4	AC	40.0734
5	PI	39.148

5.5 Top 5 states with highest/lowest average time to delivery

lowest average time to delivery:

Query:

```
selectc.customer_state,
round(avg(date_diff(o.order_delivered_customer_date
,o.order_purchase_timestamp,day)),4)as time_to_delivery,
from `Target.orders`o join `Target.customers`c
on c.customer_id=o.customer_id
group by 1
order by 2
limit 5
```

Row	customer_state	time_to_delivery
1	SP	8.2981
2	PR	11.5267
3	MG	11.5438
4	DF	12.5091
5	SC	14.4796

highest average time to delivery : Query:

```
select
c.customer_state,round(avg(date_diff(o.order_delivered_cus
tomer_date ,o.order_purchase_timestamp,day)),4)as
highest_time_to_delivery, from `Target.orders`o join
`Target.customers`c on c.customer_id=o.customer_id group
by 1 order by 2 desc limit 5
```

Row	customer_state	highest_time_to_delivery
1	RR	28.9756
2	AP	26.7313
3	AM	25.9862
4	AL	24.0403
5	PA	23.3161

5.6 Top 5 states where delivery is really fast/ not so fast compared to estimated date :

Top 5 states where delivery is really fast

Query:

```
select distinct cust.customer_state,
avg(extract(datetime from order_estimated_delivery_date) -
extract(datetime from
order_delivered_customer_date)) as
delivery_diff from `Target.orders` ord
left join
`Target.customers` cust on
ord.customer_id =
cust.customer_id group by
cust.customer_state order
by delivery_diff limit 5
```

Row	customer_state	delivery_diff
1	AL	0-0 7 24:46:9.886649874
2	MA	0-0 8 21:18:29.762900976
3	SE	0-0 9 7:53:14.623880597
4	ES	0-0 9 19:7:50.208521303
5	BA	0-0 9 26:22:39.872542997

Top 5 states where delivery is not so fast: Query:

```
select distinct cust.customer_state,
avg(extract(datetime from order_estimated_delivery_date)-
extract(datetime from
order_delivered_customer_date)) as
delivery_diff from `Target.orders` ord
left join
`Target.customers` cust on
ord.customer_id =
cust.customer_id group by
cust.customer_state order
by delivery_diff desc limit
5
```

Row	customer_state	delivery_diff
1	AC	0-0 19 25:50:53.400
2	RO	0-0 19 9:31:25.802469135
3	AP	0-0 18 25:25:34.119402985
4	AM	0-0 18 20:26:36.986206896
5	RR	0-0 16 14:16:13.243902439

6. Payment type analysis:

6.1 Month over Month count of orders for different payment types

Query:

```
select distinct payment_type,
extract(month from
order_purchase_timestamp) as month,
count(*) as order_count from
`Target.payments` pay inner join
`Target.orders` ord on pay.order_id =
ord.order_id group by month,
payment_type order by month
```

Row	payment_type	month	order_count
1	voucher	1	477
2	credit_card	1	6103
3	debit_card	1	118
4	UPI	1	1715
5	credit_card	2	6609
6	voucher	2	424
7	UPI	2	1723
8	debit_card	2	82
9	voucher	3	591
10	credit_card	3	7707
11	UPI	3	1942
12	debit_card	3	109
13	credit_card	4	7301
14	voucher	4	572

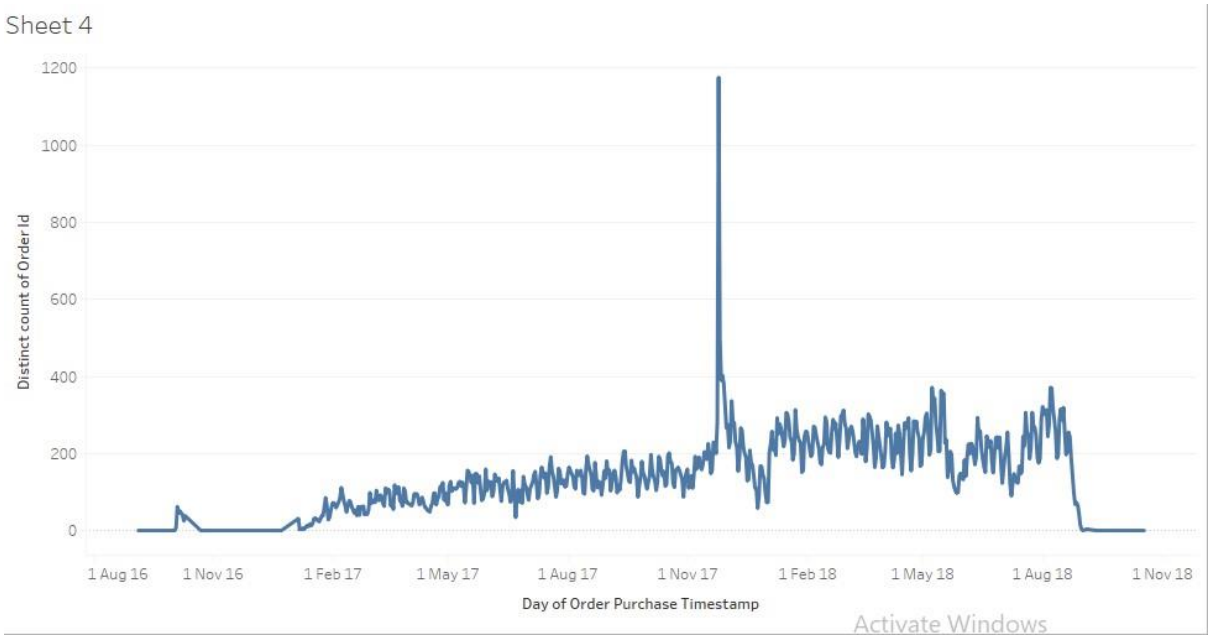
Load more

6.2 Count of orders based on the no. of payment installments Query:

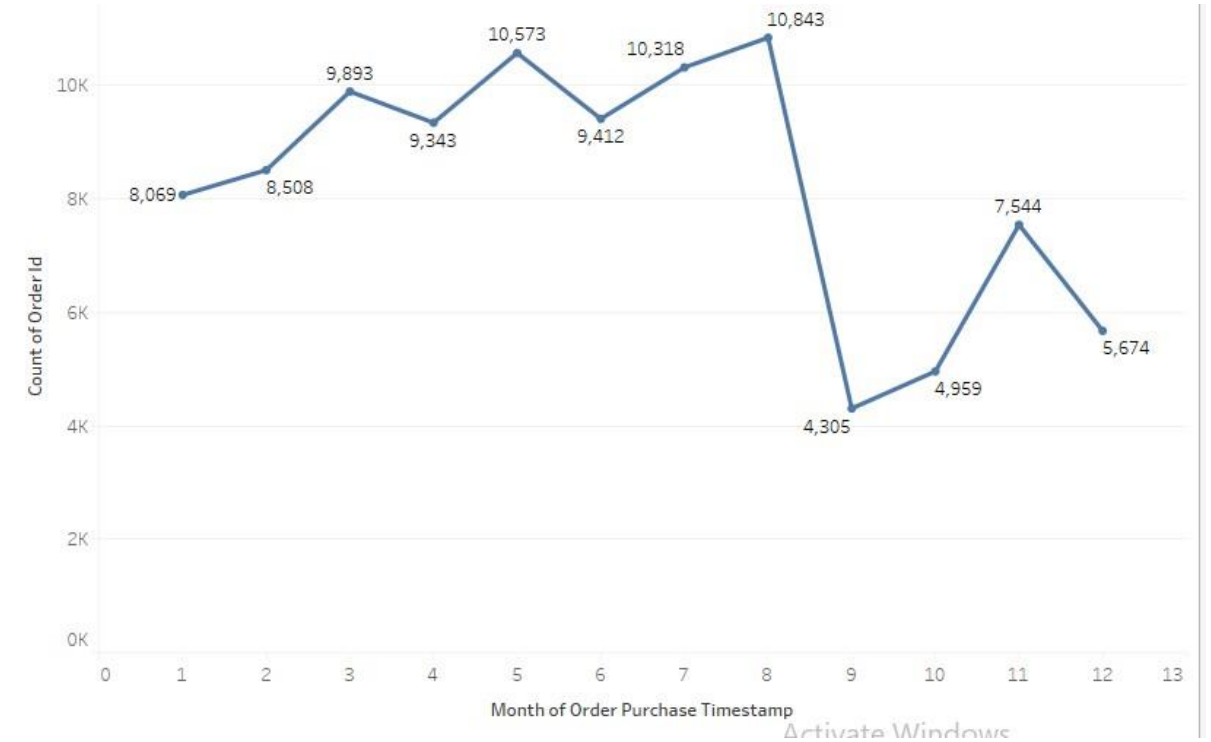
```
select p.payment_installments,count(distinct o.order_id) as
orders_count
from `Target_sql.payments` p
inner join `Target_sql.orders` o
on p.order_id = o.order_id
where payment_installments >= 1
group by payment_installments
order by payment_installments
```

Row	payment_installment	orders_count
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315
11	11	23

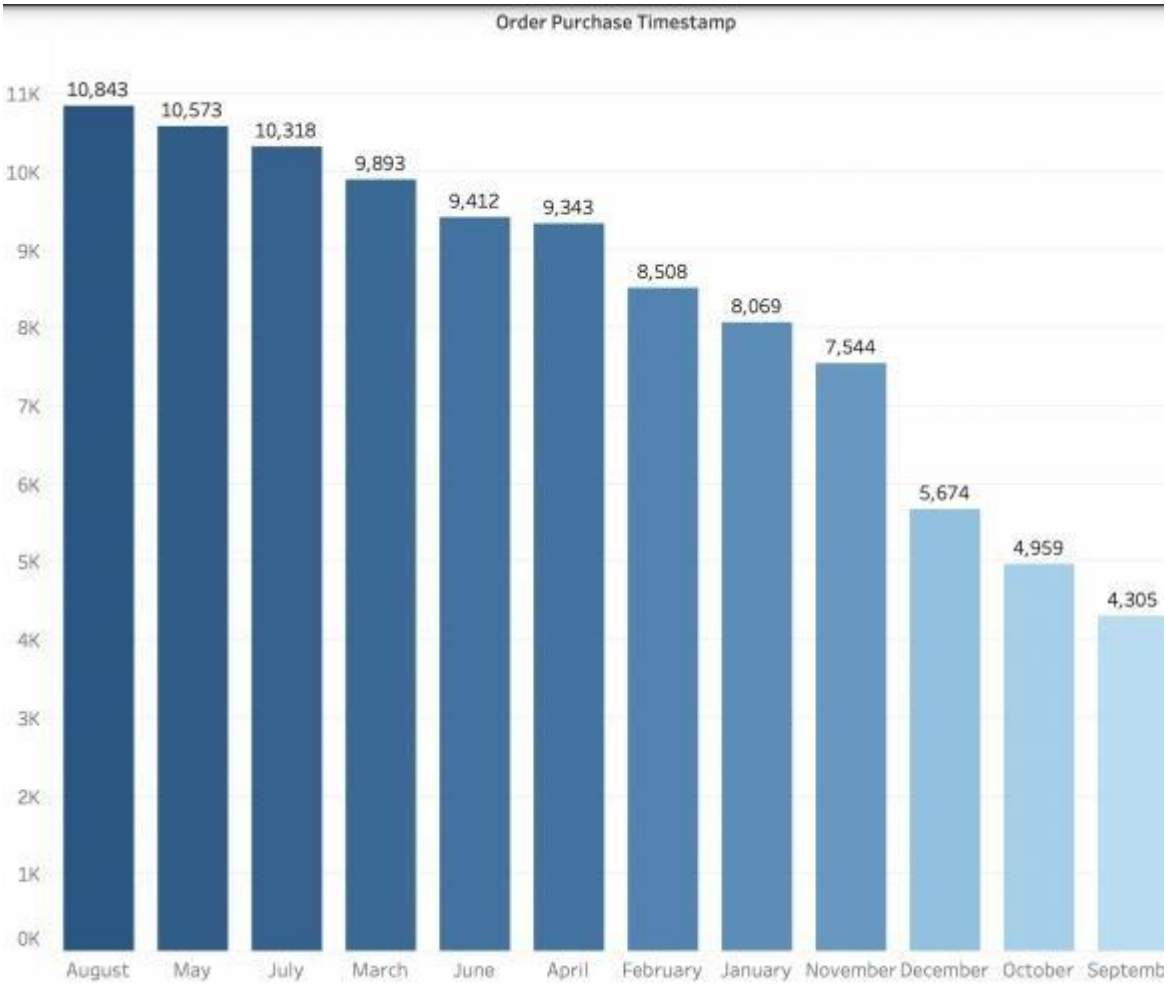
Insights: We have calculated the no. of orders placed on the basis of the payment instalments that have been paid. And we can conclude that the most no. of orders are ordered on the first 3 instalments.



The company has the least sale in the month of September but has the highest sale in August month. In the below mentioned graph, we can have a detailed insight about the trend of orders throughout the year and according we can focus on sales.



Insights: The graph shows peak order counts in June and November, with a significant drop in September. There is a general trend of increasing orders from January to June, followed by a decline through September, and a recovery towards the end of the year. This indicates seasonal fluctuations, suggesting targeted promotions in low months like September and strategic boosts during high months like June and November.



Insights: August has the highest order count, while September has the lowest. Orders drop from June to September, then recover slightly in October and November before declining again in December. Targeted strategies are

needed to boost sales in low-performing months and capitalize on high-performing ones.

Recommendation:

1. Focus on Night Buyers:

- As most orders are made at night compared to morning and afternoon, the company should focus on categorized buyers with specific strategies.

2. State-Specific Insights:

- The state 'SP' has the highest number of customers, so marketing efforts should prioritize this region.

3. Increase Sales in August:

- Since August shows the highest sales, plan more ideas to boost sales during this month. This could include offering good deals, mid-sale discounts, vouchers, and gift cards for medium-range buyers.

4. Boost Sales in Low-Sale Months:

- Address the low sales in September by focusing on sales strategies during this month to balance out the yearly trend.

5. Customer Engagement:

- Send additional small gifts, like freebies, to boost customer buying interest and loyalty.

6. Offer Promotions Based on Purchase Times:

- Implement strategies such as special promotions or discounts during midday hours to increase sales in times with lower order volumes compared to the night.