

# BUSINESS CASE: TARGET SQL

- ✚ Target is a globally renowned brand and a prominent retailer in the United States.
- ✚ In this particular project, 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 customer data, orders data, order status, price, payment and freight performance, customer location, product attributes, and customer reviews.
- ✚ Here, as an aspiring data analysts and scientists, we are going to analyse this wide range of dataset and gain valuable insights into Targets' operations in Brazil. Using the insights, we can screen various aspects of business in order to improve the efficiency of business, customer satisfaction level, increase in profit of the company.
- ✚ As SQL is the backbone of data analysis and management in the modern era, we are going to work on this case study and manipulate and query the databases and extract meaningful insights from the large volume of the given dataset thus providing recommendations and help the organization to take the best data driven decisions to improve the efficiency and versatility of the business.
- ✚ The following datasets are available in the project.
  - 1. customers.csv
  - 2. geolocation.csv
  - 3. order\_items.csv
  - 4. payments.csv
  - 5. reviews.csv
  - 6. orders.csv
  - 7. products.csv
  - 8. sellers.csv
- ❖ The question, SQL queries along with a screenshot of the first 10 rows from the output are submitted below along with the valuable insights that we drew from our analysis and recommendations regarding the company's growth and profit perspective are submitted below.

# BUSINESS CASE: TARGET SQL

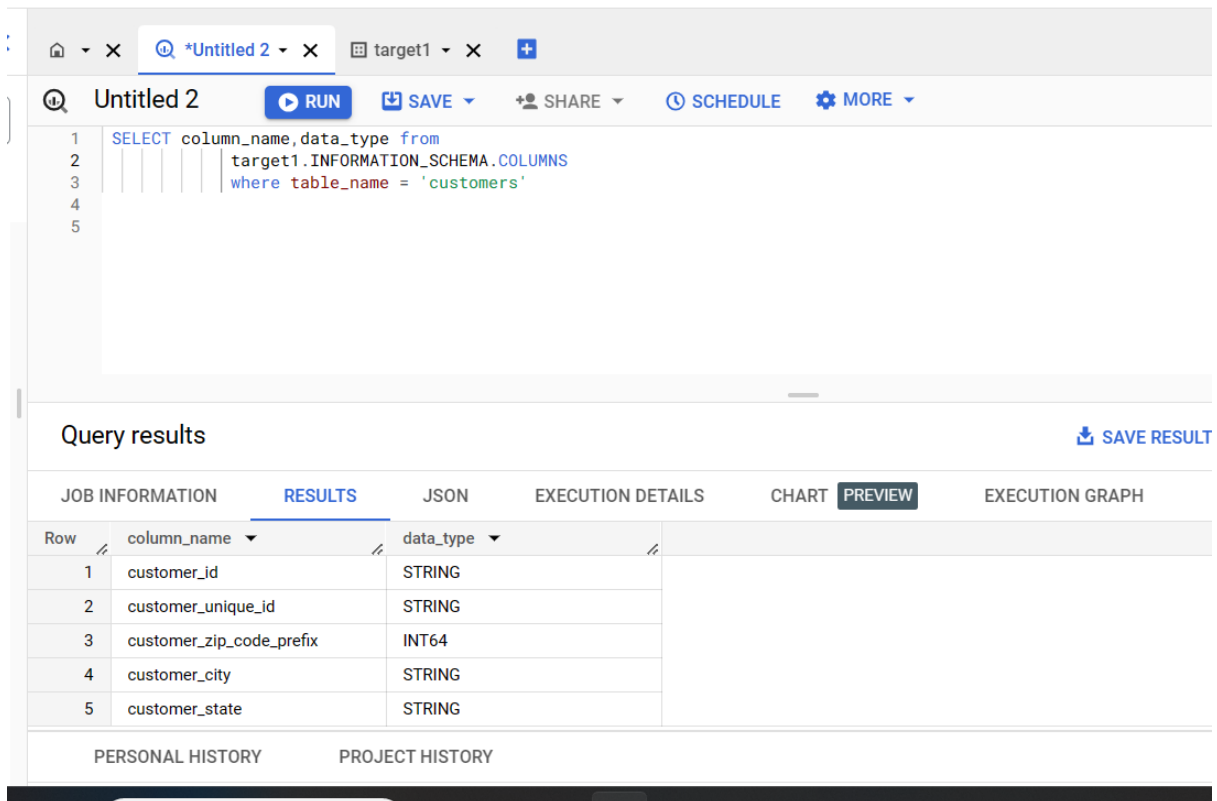
What does 'good' look like?

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

### A. Data type of all columns in the "customers" table.

Query : 

```
select column_name,data_type from
target1.information_schema.columns
where table_name = 'customers'
```



The screenshot shows a SQL query editor interface. The query entered is:

```
SELECT column_name,data_type from
target1.information_schema.columns
where table_name = 'customers'
```

The query results are displayed in a table with the following columns: Row, column\_name, and data\_type. The results show the data types for the first five columns of the 'customers' table.

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

### B. Get the time range between which the orders were placed.

Query : 

```
SELECT MIN(order_purchase_timestamp) as first_order_date,
max(order_purchase_timestamp) as last_order_date,
date_diff(max(order_purchase_timestamp),MIN(order_purchase_timestamp)
,day) as range_of_orders_placed FROM `target1.orders`
```

# BUSINESS CASE: TARGET SQL

payments ×

Untitled ×

orders ×

\*2023-08-04 07:32:36 ×

+

2023-08-04 07:32:36

RUN

SAVE QUERY

SHARE

SCHEDULE

MORE

1

2

3

4

5

```
1 SELECT MIN(order_purchase_timestamp) as first_order_date,
2 max(order_purchase_timestamp) as last_order_date,
3 date_diff(max(order_purchase_timestamp),MIN(order_purchase_timestamp),day) as range_of_orders_placed
4 FROM `target1.orders`
5
```

Query results

SAVE RES

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION GRAPH

row

first\_order\_date

last\_order\_date

range\_of\_orders\_placed

1

2016-09-04 21:15:19 UTC

2018-10-17 17:30:18 UTC

772

PERSONAL HISTORY

PROJECT HISTORY

## C. Count the Cities & States of customers who ordered during the given period.

Query :

```
select count(distinct customer_city) as customer_city_count,
count(distinct customer_state) as customer_state_count
from `target1.customers`
```

Untitled 3

RUN

SAVE

SHARE

SCHEDULE

MORE

1

2

3

4

5

6

7

```
1 SELECT count(distinct customer_city) as customer_city_count,
2 count(distinct customer_state) as customer_state_count
3 FROM `target1.customers`
4
5
6
7
```

Query results

SAV

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION GRAP

row

customer\_city\_count

customer\_state\_count

1

4119

27

# BUSINESS CASE: TARGET SQL

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

The above queries are general queries for gaining knowledge on general structure and characteristics and types of datasets given. In query IA, the datatype of all the datasets is known.

In the query IB, the firstorder date, the lastorder date and the time\_range\_of\_orders\_placed is derived.

The query IC gives information about the customer cities and customer states where the business is widespread.

## II. In-depth Exploration:

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

Query :

```
with cte as
(
select count(distinct order_id) as
order_count,max(order_purchase_timestamp) as last_order,
min(order_purchase_timestamp) as first_order,
extract (year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
from `target1.orders`
group by extract (year from order_purchase_timestamp),extract
(month from order_purchase_timestamp)
),
cte1 as

(select concat(cte.year,"-",cte.month)as year_month,
format_date("%B-%Y",cte.first_order)as
name_of_month_year,cte.order_count
from cte)

select cte1.year_month,cte1.name_of_month_year,
cte1.order_count from cte1
group by cte1.year_month,cte1.name_of_month_year,
cte1.order_count
order by cte1.name_of_month_year asc
```

# BUSINESS CASE: TARGET SQL

payments ×

Untitled ×

orders ×

\*2023-08-04 07:32:36 ×

\*2a ×

2a

RUN

SAVE QUERY

SHARE

SCHEDULE

MORE

9

10 cte1 as

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXEC
Row	year_month	name_of_month_year	order_count			
1	2017-4	April-2017	2404			
2	2018-4	April-2018	6939			
3	2017-8	August-2017	4331			
4	2018-8	August-2018	6512			
5	2016-12	December-2016	1			
6	2017-12	December-2017	5673			
7	2017-2	February-2017	1780			
8	2018-2	February-2018	6728			
9	2017-1	January-2017	800			
10	2018-1	January-2018	7269			

Results p

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

**Query :**

```
with cte as
(
select count(distinct order_id) as order_count,
format_date("%B", order_purchase_timestamp) as month_name, extract(month
from order_purchase_timestamp) as month
from `target1.orders`
group by format_date("%B", order_purchase_timestamp), month
)
select
cte.month_name,cte.month,cte.order_count
from cte
group by cte.month_name,cte.order_count,cte.month
order by cte.month
```

# BUSINESS CASE: TARGET SQL

payments × orders × \*5d × 2b month name format ×

2b month name format

RUN

SAVE QUERY

SHARE

SCHEDULE

MORE

5 from `target1.orders`

6 group by format\_date("%B", order\_purchase\_timestamp), month

Press Alt

Query results

SAVE RESULTS

EXI

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
ow	month_name	month	order_count			
1	January	1	8069			
2	February	2	8508			
3	March	3	9893			
4	April	4	9343			
5	May	5	10573			
6	June	6	9412			
7	July	7	10318			
8	August	8	10843			
9	September	9	4305			
10	October	10	4959			

Results per page: 50 1 - 12 of 12

**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**

## Query :

```
with cte as
(
select order_id, count(distinct order_id) as orders_count,
extract(hour from order_purchase_timestamp) as order_time
from `target1.orders`
group by order_time, order_id
order by order_time),
cte2 as
(
select
order_time,
case
when order_time between 0 and 6
then "Dawn"
when order_time between 7 and 12
then "Mornings"
when order_time between 13 and 18
then "Afternoon"
when order_time between 19 and 23
then "Night"
end as time_of_day
from cte
group by order_time, time_of_day)
select cte2.time_of_day,
count(distinct cte.order_id) as order_count from cte
join cte2 on cte.order_time=cte2.order_time
```

# BUSINESS CASE: TARGET SQL

group by cte2.time\_of\_day

20	then "Night"
21	end as time_of_day

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
row	time_of_day	order_count				
1	Mornings	27733				
2	Dawn	5242				
3	Afternoon	38135				
4	Night	28331				

## Insights of In-depth Exploration:

A) The number of orders for the month of October 2016 is average but whereas in November 2016 it was decreased drastically to 0. There were no orders for November 2016. For the months of September 2016 and December 2016 also the number of orders was minimum (4 and 1 respectively). From Jan 2017 to May 2017 the number of orders were considerably increased but there is a slight decrease in June 2017. Again, from July 2017 to March 2018 the number of orders placed were consistently increased to around 7200. From April 2018 to August 2018 the number of orders were around 6000 to 6500 per month. The number of orders drastically decreased in September 2018 and October 2018 to 16 and 4 respectively.

ON observing the above month on month order growing trend, year 2017 and upto August in 2018 were considered to have consistent increase in number of orders. The reasons for inconsistency in 2016 and September and October 2018 are to be detailed out for further progress in business. The graphical representation is shown below.(month-year on x-axis and no. of orders on y-axis)

# BUSINESS CASE: TARGET SQL

Query results

[SAVE](#)



**B)** In terms of monthly seasonality analysis, the number of orders count is considerably high from January to August with noticeably increase in order count for the months of May to August (around 10000 per month). There is a steep decrease in order count in September. The order count is decreased from September to December when compared to first half year data.

Hence in terms of seasonality, the first half year is considered to the most favourable time for the business with May, June, July, August as most constructive months with highest number of orders (Around 10000 per month). The second half year i.e., from September to December the order count is considerably low.

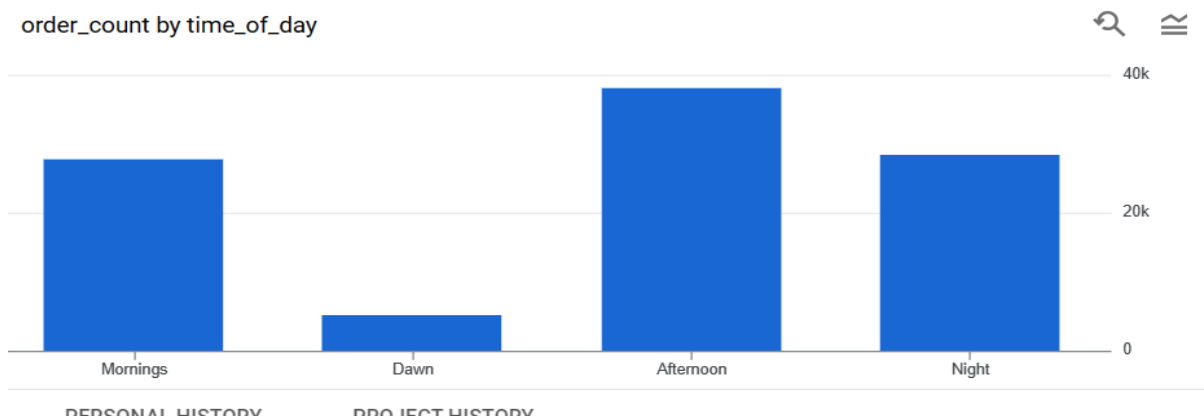
This indicates that the new strategies like discounts, year-end sale, 1+1 offers etc., must be designed to improve the business during the second half of the year. Hence the reasons for seasonality in business is to be studied and new strategies to improve the business must be implemented.

**C)** In terms of time interval of the day, the Brazilian customers mostly placed their orders in the **Afternoon i.e., from 13-18 hrs**. The number of orders placed in Morning is 27733(7-12 hrs), Afternoon (13-18 hrs) 38135, Dawn (0-6 hrs) 5242, Night (19-23 hrs) 28331 respectively. The number of orders place at Dawn is extensively less as this would be the sleeping time of the customers. The number of the orders placed in other intervals of the day like mornings and night may be increased by some specific strategies like hour sale. The other reasons may be enumerated in order to improve the profits in business all around the day. The graphical representation is shown below for clear picture.(order\_count on x-axis and interval of day on y-axis).



# BUSINESS CASE: TARGET SQL

order\_count by time\_of\_day



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

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

**Query:**

```
select c.customer_state,
format_date("%B",o.order_purchase_timestamp) as month,count(distinct
o.order_id) as order_count
from `target1.orders` o join `target1.customers` c
on c.customer_id = o.customer_id
group by month,c.customer_state
order by c.customer_state,order_count desc
```

3a RUN SAVE QUERY SHARE SCHEDULE MORE ✓ This query will pi

```
1 |
2 | select c.customer_state,
```

Press A

Query results

SAVE RESULTS EX

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
row	customer_state	month	order_count				
1	AC	May	10				
2	AC	April	9				
3	AC	July	9				
4	AC	January	8				
5	AC	August	7				
6	AC	June	7				
7	AC	October	6				
8	AC	February	6				
9	AC	November	5				
10	AC	December	5				

Results per page: 50 1 - 50 of 322

PERSONAL HISTORY

PROJECT HISTORY

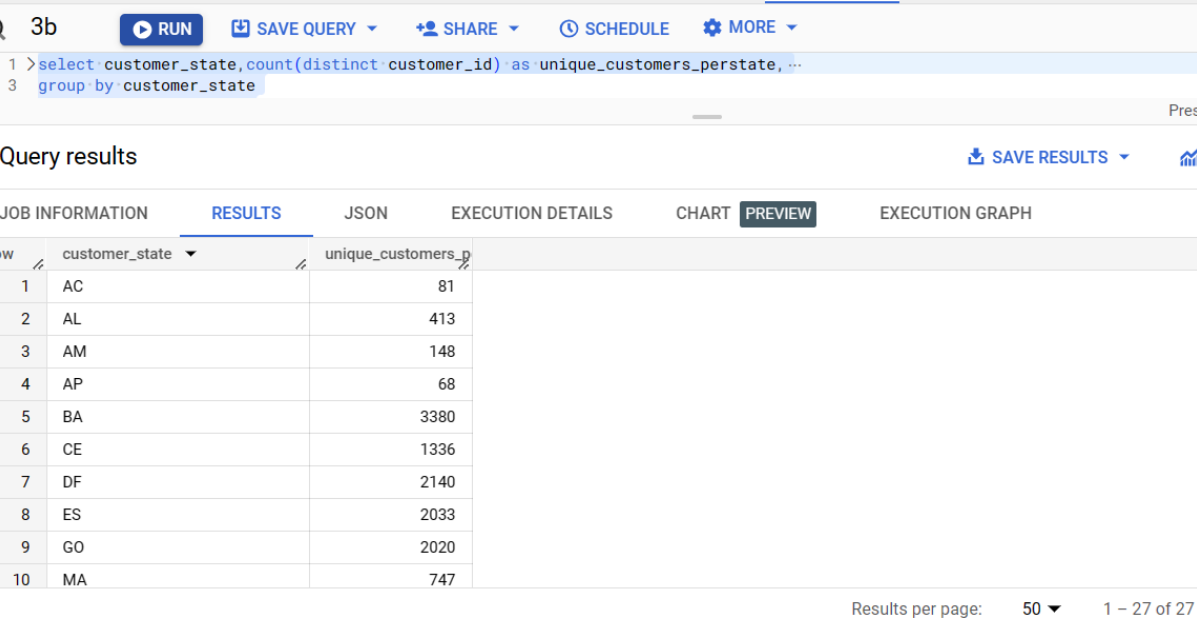
B. How are the customers distributed across all the states?

**Query :**

```
select customer_state,count(distinct customer_id) as
unique_customers_perstate, from `target1.customers`
```

# BUSINESS CASE: TARGET SQL

group by customer\_state  
order by customer\_state



## Insights of Evolution of E-commerce orders in the Brazil region:

A) According to the month on month-on-month number of orders placed in each state, it is visualized that the number of orders placed are maximum during the months of March, April, May, June, July, August across all the states. The maximum number of orders placed are in SP, RJ, MG states. The low business in the remaining states is to be analysed. The reasons for low performance are to be evaluated and the immediate necessary steps are to be taken to improve the business performance in all the states during all the months.The graphical representation is shown below(month on x-axis and order\_count on y-axis)

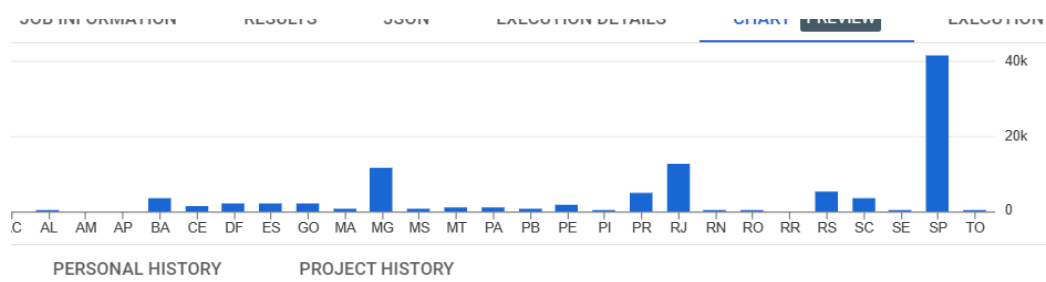
# BUSINESS CASE: TARGET SQL



B) This is regarding the number of customers distributed across all the states.

Maximum number of customers are distributed among the following states: **SP(41764),RJ(12852),MG(11635),RS(5466),PR(5045).**

All the remaining states are having a smaller number of customers. A few states like AC, AP, RR are having no. of unique customers in two digit figure(81,68,46 respectively).The whys and wherefores for such a small number of customers in different states are to be primarily and firstly evaluated on immediate basis using the state managers and the strategies to improve the business platform according the customer needs is to be prioritized.The graphical representation is shown below(customer\_state on x-axis,count on y-axis)



# BUSINESS CASE: TARGET SQL

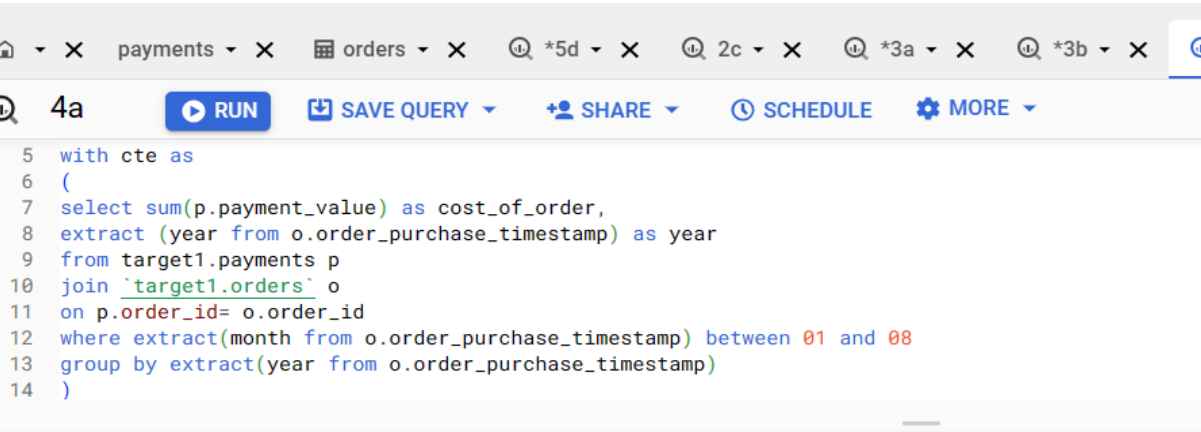
## IV. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

**Query :**

```
with cte as
(
select sum(p.payment_value) as cost_of_order,
extract (year from o.order_purchase_timestamp) as year
from target1.payments p
join `target1.orders` o
on p.order_id= o.order_id
where extract(month from o.order_purchase_timestamp) between 01 and 08
group by extract(year from o.order_purchase_timestamp)
)
select cte.year,round(cost_of_order,2) as cost_of_orders,
round(((cost_of_order)-lag(cte.cost_of_order) over (order by
year)))/lag(cte.cost_of_order) over (order by year)*100) as
percentage_increase
from cte

order by year
```



### Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION
row	year	cost_of_orders	percentage_increase				
1	2017	3669022.12	null				
2	2018	8694733.84	137.0				

# BUSINESS CASE: TARGET SQL

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

Query:

```
select c.customer_state, round(sum(price),2) as
total_price, round(avg(price),2) as avg_price,
from `target1.orderitems` oitems
join target1.orders o
on oitems.order_id=o.order_id
join `target1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by avg_price, total_price desc
```

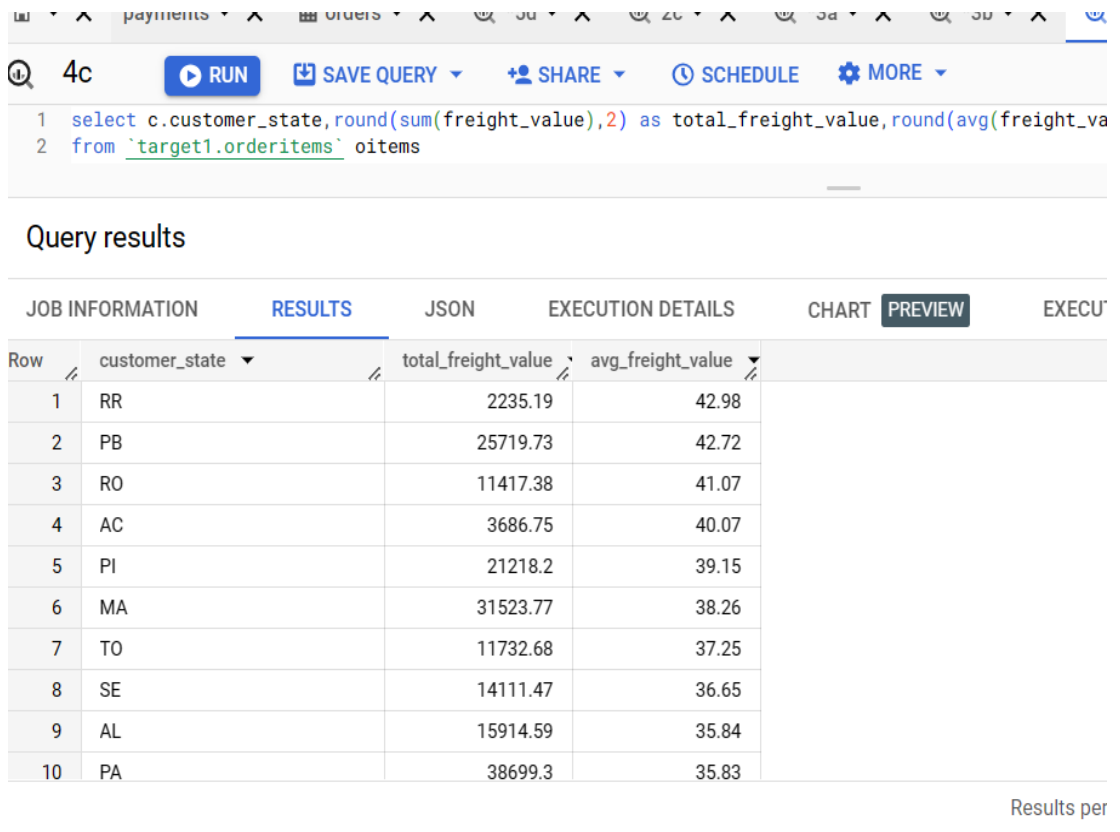
4b	RUN	SAVE QUERY	SHARE	SCHEDULE	MORE
1 select c.customer_state, round(sum(price),2) as total_price, round(avg(price),2) as avg_price, 2 from `target1.orderitems` oitems					
Query results					
JOB INFORMATION RESULTS JSON EXECUTION DETAILS CHART PREVIEW EXECUTION GRAPH					
Row	customer_state	total_price	avg_price		
1	SP	5202955.05	109.65		
2	PR	683083.76	119.0		
3	RS	750304.02	120.34		
4	MG	1585308.03	120.75		
5	ES	275037.31	121.91		
6	SC	520553.34	124.65		
7	RJ	1824092.67	125.12		
8	DF	302603.94	125.77		
9	GO	294591.95	126.27		
10	BA	511349.99	134.6		
Results per page: 50 1 - 27 of 27					

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

Query:

```
select c.customer_state, round(sum(freight_value),2) as
total_freight_value, round(avg(freight_value),2) as avg_freight_value,
from `target1.orderitems` oitems
join target1.orders o
on oitems.order_id=o.order_id
join `target1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by avg_freight_value desc
```

# BUSINESS CASE: TARGET SQL



The screenshot shows a SQL query interface with a search bar containing '4c'. The query is: `1 select c.customer_state,round(sum(freight_value),2) as total_freight_value,round(avg(freight_value),2) as avg_freight_value  
2 from `target1.orderitems` oitems`. The results are displayed in a table with columns: Row, customer\_state, total\_freight\_value, and avg\_freight\_value. The results are sorted by total\_freight\_value in descending order.

Row	customer_state	total_freight_value	avg_freight_value
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
9	AL	15914.59	35.84
10	PA	38699.3	35.83

Results per

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

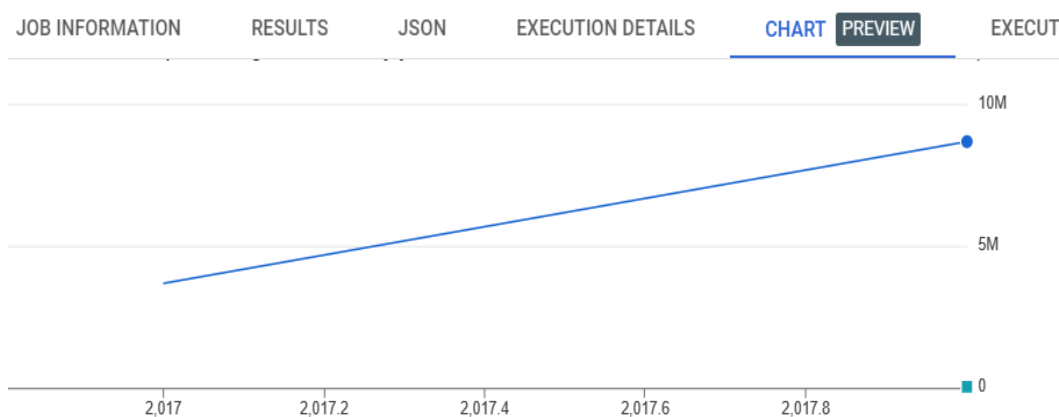
A) According to the data from % increase in cost of orders from the year 2017 to 2018(data included from January to August months only), there is a significant increase in number of orders placed from the years **2017 to 2018 i.e.,137%**. According to the months considered there was a significant increase in number of orders as maximum no of orders were placed from June to August only. However, techniques may be deduced to improve the business performance to more than 200%.

Nevertheless on the other hand considering the overall performance there is no significant increase in number of order(considering January to December months) even though there is an increase in %.

The number of orders are still to be increased by formulating new methodology in sales and ecommerce business.

A graphical representation is placed below.

# BUSINESS CASE: TARGET SQL



**B)** The states SP, RJ, MJ had shown highest total price of the items ordered (total\_price - 5202955.05, 1824092.67, 1585308.03) and the highest average\_price of the items ordered is shown by PB,AL,AC states respectively(191.48,180.89,173.73).

Almost remaining all the states showed the lowest performance in gaining the highest total price which is an indicator of poor business performance where necessary caution should be taken to prevent the business from shut down. With regard to the average price of the items ordered almost all the states showed the equal result. The strategies followed by the top states may be implemented by the poor performing states to improve the business. May be the price of the product may be formulated according to the cities of the state and needs of the customers.

**C)** The total value of the order freight is highest in SP,RJ,MG,RS,PR. The average freight value is highest among the following states - RR,PB,RO,AC,PI. The high freight value may indicate the increase in ecommerce business i.e., online orders and deliveries. But high freight value does not necessarily indicate more profits. It depends on the demand and supply of shipping services, the cost of capital, and the efficiency of operations.The transportation and logistics companies need to create more value by improving their performance and resilience and customer satisfaction levels.

# BUSINESS CASE: TARGET SQL

The avg\_freight\_value is lowest among SP,PR,MG,RJ states which is between 15-20. The total\_freight\_value is lowest among RR,AP,AC,AM which is below 6000/-.

The avg\_freight\_value is almost distributed same all over the states.

The field level scenario should be studied and necessary formulations to formulated and implemented at field level to grow the business.

## V. 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.

**Query :**

```
select order_id,extract(date from order_purchase_timestamp) as
purchase_date,
extract(date from order_delivered_customer_date) as delivery_date,
extract(date from order_estimated_delivery_date) as estimated_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
as
time_to_deliver,date_diff(order_estimated_delivery_date,order_deliver
ed_customer_date,day) as diff_estimated_delivery
from `target1.orders`
where order_delivered_customer_date is not null
order by purchase_date
```



# BUSINESS CASE: TARGET SQL

5a RUN SAVE QUERY SHARE SCHEDULE MORE

12  
13 `select order_id,extract(date from order_purchase_timestamp) as purchase_date,`

Query results

[SAVE RES](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAF
w	order_id	purchase_date	delivery_date	estimated_date	time_to_deliver	diff_estimated_delive	
1	bfb0f9bdef...	2016-09-15	2016-11-09	2016-10-04	54	-36	
2	65d1e226df...	2016-10-03	2016-11-08	2016-11-25	35	16	
3	be5bc2f0da...	2016-10-03	2016-10-27	2016-11-07	24	10	
4	ae8a60e4b0...	2016-10-03	2016-11-03	2016-12-01	30	27	
5	cd3b8574c8...	2016-10-03	2016-10-14	2016-11-23	10	39	
6	d207cc2726...	2016-10-03	2016-10-31	2016-11-23	27	22	
7	a41c8759fb...	2016-10-03	2016-11-03	2016-11-29	30	25	
8	ef1b29b591...	2016-10-03	2016-11-01	2016-11-25	28	23	
9	3b697a20d9...	2016-10-03	2016-10-26	2016-10-27	23	0	
10	35d3a51724...	2016-10-04	2016-10-26	2016-12-20	22	54	

Results per page: 50 ▼

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

Query:

```
with cte as
( select c.customer_state as highest_freight_states,round(avg(freight_value),2) as
highest_avg_freight_value,dense_rank() over (order by round(avg(freight_value),2)
desc) as highest_freight_values
from `target1.orderitems` oitems
join target1.orders o
on oitems.order_id=o.order_id
join `target1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by round(avg(freight_value),2) desc
limit 5),
cte2 as
(select c.customer_state as lowest_freight_states,round(avg(freight_value),2) as
lowest_avg_freight_value,dense_rank() over (order by round(avg(freight_value),2)
asc) as lowest_freight_values
from `target1.orderitems` oitems
join target1.orders o
on oitems.order_id=o.order_id
join `target1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by round(avg(freight_value),2) asc
limit 5
```

# BUSINESS CASE: TARGET SQL

```
)  
select  
cte.highest_freight_states,cte.highest_avg_freight_value,cte2.lowest_freight_states  
,cte2.lowest_avg_freight_value from cte2  
join cte  
on cte.highest_freight_values=cte2.lowest_freight_values
```

```
5b  RUN SAVE QUERY SHARE SCHEDULE MORE  
4 from `target1.orderitems` oitems  
5 join target1.orders o  
6 on oitems.order_id=o.order_id  
7 join `target1.customers` c  
8 on o.customer_id=c.customer_id  
9 group by c.customer_state  
10 order by round(avg(freight_value),2) desc  
11 limit 5
```

## Query results

[SAVE RE](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRA
low	highest_freight_states	highest_avg_freight	lowest_freight_states	lowest_avg_freight_y			
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			

## C. Find out the top 5 states with the highest & lowest average delivery time.

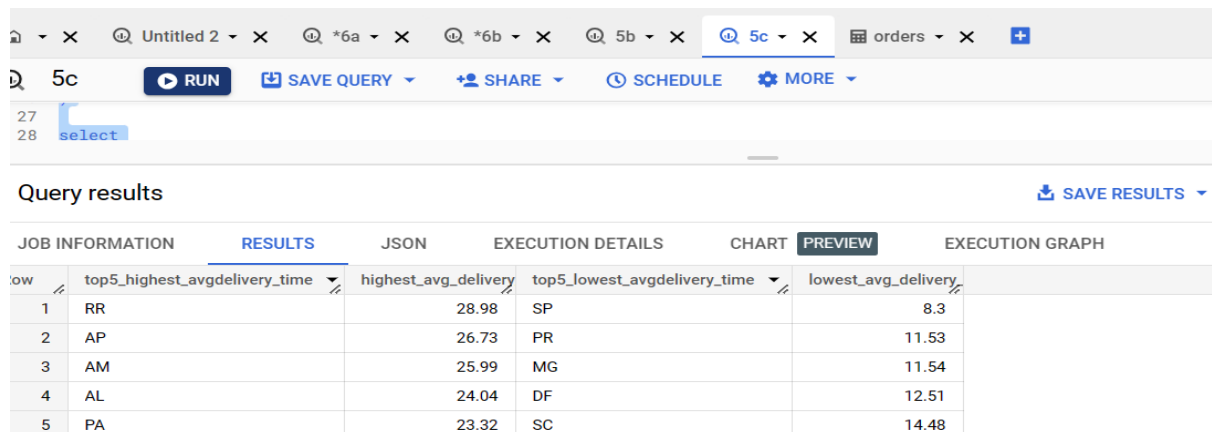
### Query :

```
with cte as  
(  
select c.customer_state,  
avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as  
avg_delivery_time,  
dense_rank() over (order by  
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)  
,2) asc) as lowest_avg_delivery_time,  
row_number() over (order by  
(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day))) asc  
) as row_no  
from `target1.orders` o  
join `target1.customers` c  
on o.customer_id=c.customer_id  
group by c.customer_state  
)  
,  
cte1 as  
(  
select c.customer_state,  
avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as  
avg_delivery_time,
```

# BUSINESS CASE: TARGET SQL

```
dense_rank() over (order by
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp, day)
),2) desc) as highest_avg_delivery_time,
row_number() over (order by
(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp, day)))des
c) as row_no
from `target1.orders` o
join `target1.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
)
```

```
select
cte1.customer_state as top5_highest_avgdelivery_time,
round(cte1.avg_delivery_time,2) as highest_avg_delivery_time,
cte.customer_state as top5_lowest_avgdelivery_time,
round(cte.avg_delivery_time,2) as lowest_avg_delivery_time
from cte join cte1
on cte.row_no=cte1.row_no
where(cte1.highest_avg_delivery_time between 1 and 5) and
(cte.lowest_avg_delivery_time between 1 and 5)
order by cte.row_no
```



The screenshot shows a SQL query editor with a toolbar containing buttons for RUN, SAVE QUERY, SHARE, SCHEDULE, and MORE. The query is displayed in the editor, and the results are shown in a table below. The table has columns for job information, results, JSON, execution details, chart, preview, and execution graph. The results column shows the top 5 states with their highest and lowest average delivery times.

row	top5_highest_avgdelivery_time	highest_avg_delivery	top5_lowest_avgdelivery_time	lowest_avg_delivery
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

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

**Query :**

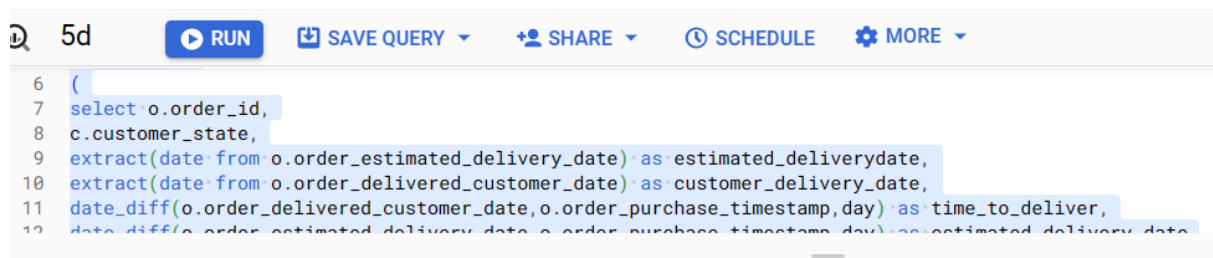
```
with cte as
(
select o.order_id,
c.customer_state,
extract(date from o.order_estimated_delivery_date) as estimated_deliverydate,
extract(date from o.order_delivered_customer_date) as customer_delivery_date,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp, day) as
time_to_deliver,
```

# BUSINESS CASE: TARGET SQL

```
date_diff(o.order_estimated_delivery_date,o.order_purchase_timestamp,day) as
estimated_delivery_date
from `target1.orders` o
join `target1.customers` c
on o.customer_id=c.customer_id
where extract(date from o.order_delivered_customer_date) is not null
),

cte1 as
(select cte.customer_state as customer_state,
round(avg(cte.time_to_deliver),2) as avg_delivery_time,
round(avg(cte.estimated_delivery_date),2) as avg_estimated_deliverytime
from cte
group by customer_state)

select cte1.customer_state as
top_5_states,cte1.avg_delivery_time,cte1.avg_estimated_deliverytime,
round((cte1.avg_estimated_deliverytime-cte1.avg_delivery_time),2) as
fast_delivery_speed
from cte1
order by fast_delivery_speed desc
limit 5
```



The screenshot shows a SQL query editor with a toolbar at the top containing buttons for 'RUN', 'SAVE QUERY', 'SHARE', 'SCHEDULE', and 'MORE'. Below the toolbar, a SQL query is displayed, which is a copy of the one provided in the previous block. The query is highlighted in blue. Below the query, there is a section titled 'Query results' with a 'SAVE RESULTS' button. The results are displayed in a table with 6 columns: 'JOB INFORMATION', 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'CHART', 'PREVIEW', and 'EXECUTION GRA'. The table has 5 rows of data, corresponding to the top 5 states based on fast delivery speed.

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRA
row	top_5_states	avg_delivery_time	avg_estimated_delivery_time	fast_delivery_speed			
1	AC	20.64	40.72	20.08			
2	RO	18.91	38.39	19.48			
3	AP	26.73	45.87	19.14			
4	AM	25.99	44.92	18.93			
5	RR	28.98	45.63	16.65			

## Insights of Analysis based on sales, freight and delivery time:

- A) Based on the data, most of the orders are delivered within the estimated delivery date except a few. The span of the delivery can still be reduced if possible at the field level to improve the customer

# BUSINESS CASE: TARGET SQL

satisfaction level. But the orders are delivered within the estimated delivery period.

- B)** The top five highest and bottom five lowest freight values along with avg\_freight values were displayed. The highest freight states are RR,PB,R0,AC,PI and lowest freight states are SP,PR,MG,RJ,DF respectively.
- C)** The top five highest and bottom five lowest average delivery time states are displayed. The top five include RR,AP,AM,AL,PA and bottom five include SP,PR,MG,DF,SC. The lowest average delivery states indicate the process of delivery is fast in those states. Hence more concentration should be taken and parameters are to be evaluated for highest average time taken for delivery of the order. The average should be brought down by decreasing the number of days taken for the delivery of the order.
- D)** The top five state where delivery of the order is very fast when compared to the estimated delivery date are AC,R0,AP,AM,RR which is a good sign of growth of the business. All the orders must be bought under this category of fastest delivery than estimated delivery date. Suitable strategies like improving the freight services should be looked upon.

## VI. Analysis based on the payments:

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

#### Query :

```
select format_date("%Y-%m",o.order_purchase_timestamp) as  
year_month,p.payment_type,count(distinct o.order_id) as no_of_orders  
from `target1.orders` o  
join `target1.payments` p  
on o.order_id = p.order_id  
group by p.payment_type,year_month  
order by year_month
```

# BUSINESS CASE: TARGET SQL

6a

RUN

SAVE QUERY

SHARE

SCHEDULE

MORE

1

2

```
select format_date("%Y-%m", o.order_purchase_timestamp) as year_month, p.payment_type, count(distinct o.order_id)
```

Query results

SAVE RESL

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

CHART

PREVIEW

EXECUTION GRAPH

row	year_month	payment_type	no_of_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	253
3	2016-10	voucher	11
4	2016-10	debit_card	2
5	2016-10	UPI	63
6	2016-12	credit_card	1
7	2017-01	voucher	33
8	2017-01	UPI	197
9	2017-01	credit_card	582
10	2017-01	debit_card	9

Results per page: 50

PERSONAL HISTORY

PROJECT HISTORY

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

Query :

```
select payment_installments, count(distinct order_id) as no_of_orders from
`target1.payments`
where payment_installments > 0
group by payment_installments
order by payment_installments
```

# BUSINESS CASE: TARGET SQL

6b

RUN

SAVE QUERY

SHARE

SCHEDULE

group by

payment\_installments

order by

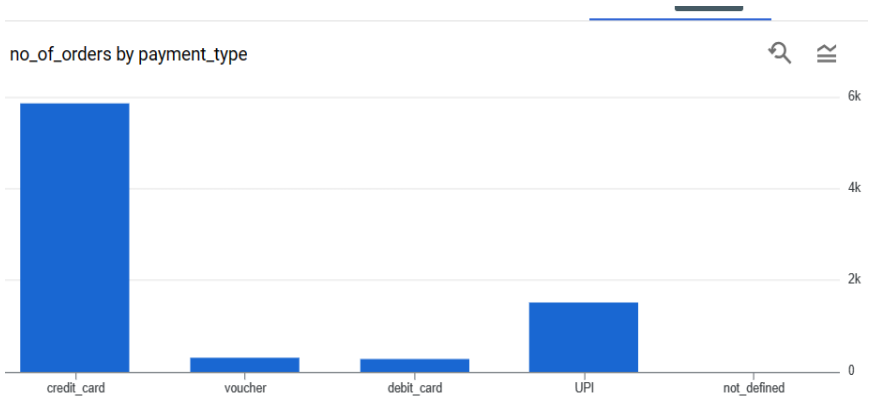
payment\_installments

## Query results

B INFORMATION	RESULTS	JSON	EXECUTION DETAILS
	payment_installment	no_of_orders	
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	

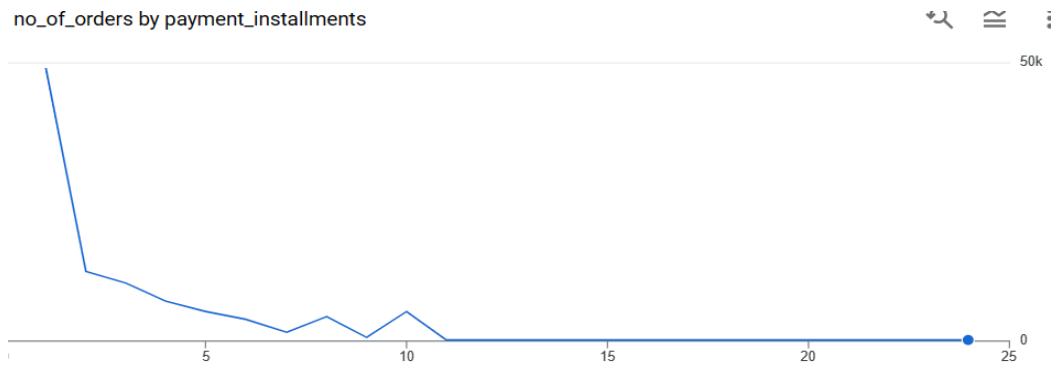
## Insights of Analysis based on the payments:

- A) Based on month on month no. of orders placed using different payment types, the payments made with credit card are maximum, then comes UPI payments when compared to other methods of payment as seen from the graph below.(payment mode on X-axis and number of orders on Y-axis. Hence the customers may be provided the facility to pay with credit card in all the cities in every state to gain customer satisfaction and to improve overall business.



# BUSINESS CASE: TARGET SQL

B) Based on the analysis made on no. of orders placed on the basis of the payment instalments that have been paid large number of customers are interested in paying the order amount in payment emi instalment type as seen from the graph below.(payment instalments on x-axis and no.of orders on y-axis)



In order to improve the business, the customers may be offered emi option for order payment method as most of the orders can be seen in 1 to 10 instalments. Moreover the emi's must be auto deducted from bank account so that they cannot skip the emi payments or the monthly emi's should be recovered from the customers properly by the sales force to improve the business.



# BUSINESS CASE: TARGET SQL

## ACTIONABLE INSIGHTS AND RECOMMENDATIONS :

The primary motto is improvement in the efficiency of business performance and to achieve more profits in the business. In spite of the above factors customer satisfaction is also a key indicator for the business growth and development.

The business efficiency can be increased by formulating new strategies, ideologies, methodologies, formulations and implemented at field level without any abstractions.

Firstly, there is some inconsistency in the data so the insights may not be cent percent accurate.

The months from April to September have shown maximum growth in business but the year ending performance was dull. On observing the month on month order growing trend, year 2017 and upto August in 2018 were considered to have consistent increase in number of orders. Some sort of discounts may be introduced to improve the business in all the months of the year. First half of the year is showing more business but the second half is dull, so some sort of offers may be introduced to increase the sales like discount, half price, 1+1, sale ect.,

A few states are having maximum number of customers but most of the states are having a little number of customers. The whys and wherefores for such a small number of customers in different states are to be primarily and firstly evaluated on immediate basis using the state managers and the strategies to improve the business platform according the customer needs is to be prioritized. Wide range of advertising may be taken up in the states where there are less number of customers.

There is an increase in percentage cost of orders from 2017 to 2018 to 137%.

Nevertheless on the other hand considering the overall performance there is no significant increase in number of

# BUSINESS CASE: TARGET SQL

order(considering January to December months) even though there is an increase in %.

The number of orders are still to be increased by formulating new methodology in sales and ecommerce business. The ecommerce business may be improved by wide publicity and elucidating the ecommerce business to the existing offline customers as and when approach the store and make them convenient with ecommerce which will save time and hardwork.

There a few states which are top in freight value and average delivery value but most of them are on low side. With regard to the average price of the items ordered almost all the states showed the equal result.

The strategies followed by the top states may be implemented by the poor performing states to improve the business. May be the price of the product may be formulated according to the cities of the state and needs of the customers.

The avg\_freight\_value is almost distributed same all over the states.

The field level scenario should be studied and necessary formulations to formulated and implemented at field level to grow the business.

Based on the data available, most of the orders are delivered within the estimated delivery date except a few. The span of the delivery can still be reduced if possible at the field level to improve the customer satisfaction level. But the orders are delivered within the estimated delivery period. If the concept of one day delivery or two day delivery is introduced the number of customers may increase and the business may be improved.

The lowest average delivery states indicate the process of delivery is fast in those states. Hence more concentration should be taken and parameters are to be evaluated for highest average time taken for delivery of the order. The average should be brought down by decreasing the number of days taken for the delivery of the order.

# BUSINESS CASE: TARGET SQL

All the orders must be bought under the category of fastest delivery than estimated delivery date. Suitable strategies like improving the freight services should be looked upon.

Based on month on month, no. of orders placed using different payment types, the payments made with credit card are maximum, then comes UPI payments.

Hence the customers may be provided the facility to pay with credit card in all the cities in every state to gain customer satisfaction and to improve overall business.

Based on the analysis made on no. of orders placed on the basis of the payment instalments that have been paid large number of customers are interested in paying the order amount in emis' instalment type.

In order to improve the business, the customers may be offered emi option for order payment method as most of the orders can be seen in 1 to 10 instalments. Moreover the emi's must be auto deducted from bank account so that they cannot skip the emi payments or the monthly emi's should be recovered from the customers properly by the sales force to improve the business.

Hence implementing small strategies to attract and impress customers may grow the business in large scale.

On the overall the growth of the business is good on half of the year and dull on the half of the year depicting seasonality in the business trend. Extensive research should be made on low business in most of the states and necessary strategies and appropriate deeds should be taken for consistent growth in the business in Brazil.