Business Case Study: Target SQL

Understanding the data

All the 8 data sets were uploaded in Big query and the customer data has been initially analysed.

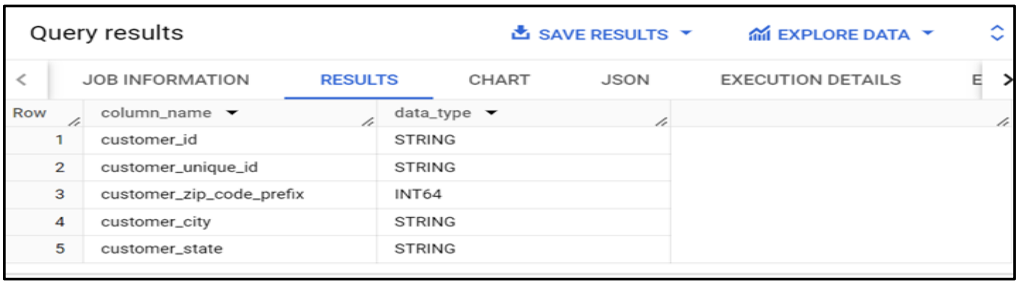
The data is available in 8 csv files:

1. customers.csv
2. sellers.csv
3. order_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

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.

```
    SELECT column_name, data_type
FROM `scaler-dsml-shw-06.Target.INFORMATION_SCHEMA.COLUMNS`
    WHERE table_name = 'customers'
```

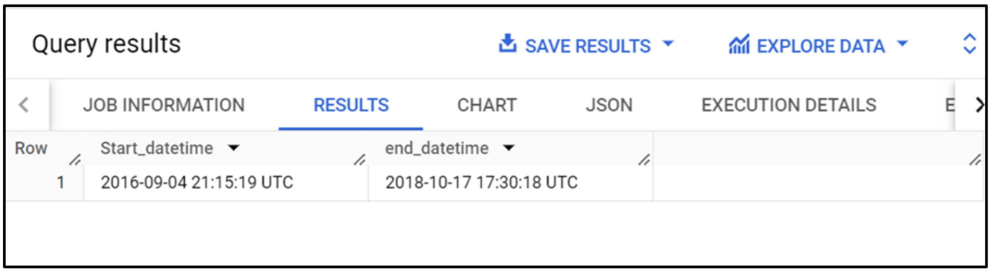


Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

By understanding the accurate data types for each column in a table we can ensure accurate analysis of the data.

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

```sql
SELECT Min(order_purchase_timestamp) as Start_datetime, Max(order_purchase_timestamp) AS end_datetime
FROM `Target.orders`
```

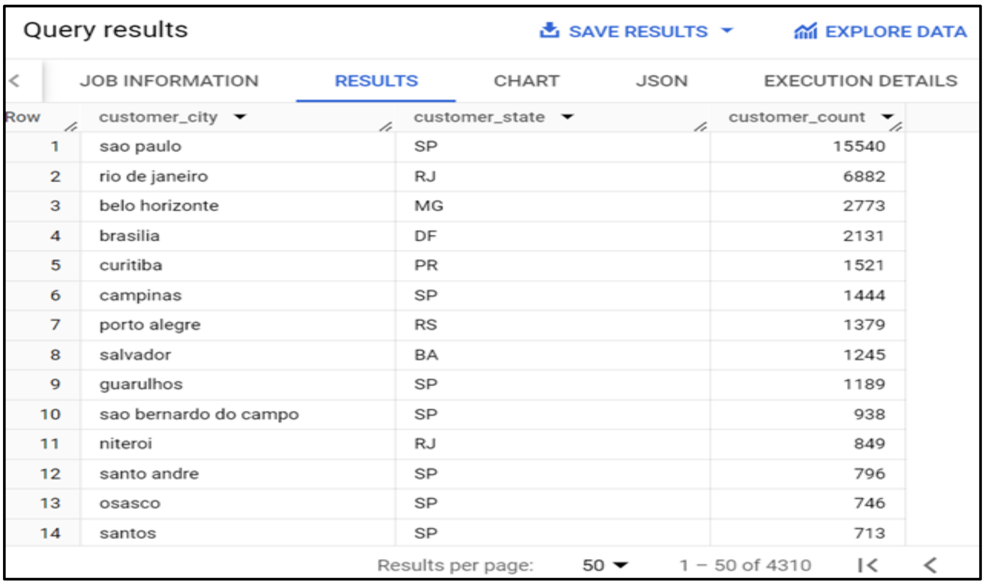


Row	Start_datetime	end_datetime
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

By the results we can understand that the time period in which the orders were placed is between 2016-09-04 21:15:19 UTC to 2018-10-17 17:30:18 UTC which is more than 2 years.

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

```sql
SELECT C.customer_city, C.customer_state, count(C.customer_id) as customer_count

FROM `Target.customers` C

Inner join `Target.orders` O On C.customer_id = O.customer_id

group by 1, 2

order by 3 desc
```



Row	customer_city	customer_state	customer_count
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521
6	campinas	SP	1444
7	porto alegre	RS	1379
8	salvador	BA	1245
9	guarulhos	SP	1189
10	sao bernardo do campo	SP	938
11	niteroi	RJ	849
12	santo andre	SP	796
13	osasco	SP	746
14	santos	SP	713

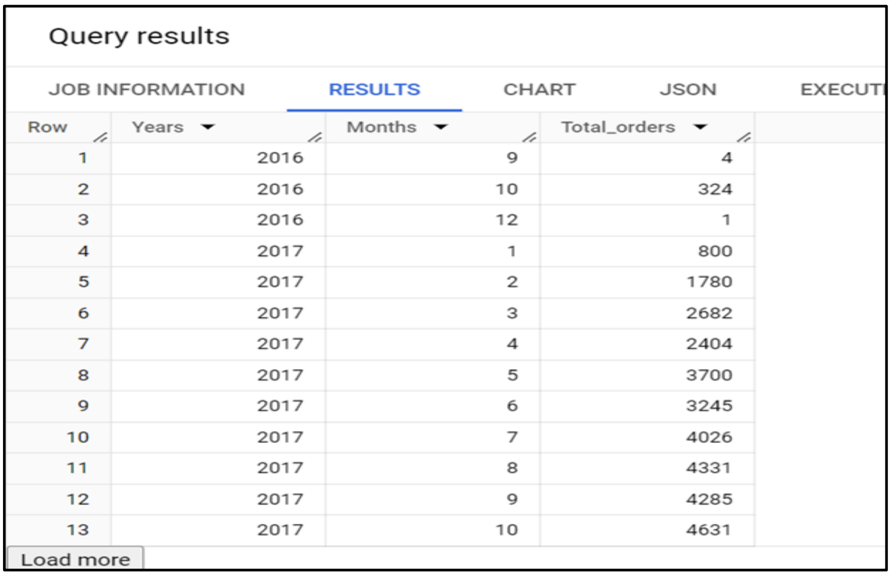
Results per page: 50   1 – 50 of 4310

There are around 4310 cities where the customers ordered, The Total order count is arranged in descending order to represent which city (Saopaulo) has the maximum ordered count.

2. In-depth Exploration:

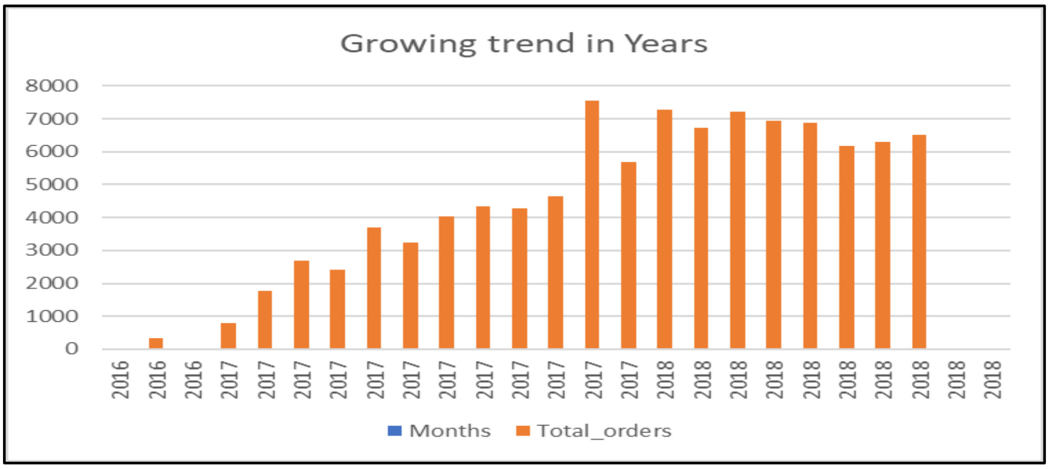
A. Is there a growing trend in the no. of orders placed over the past years?

```
SELECT EXTRACT(year from order_purchase_timestamp ) as Years, Extract(Month from
order_purchase_timestamp ) as Months, count(distinct order_id) as Total_orders

FROM `Target.orders`

Group by 1,2

order by 1,2
```



Query results

Row	Years	Months	Total_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
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631

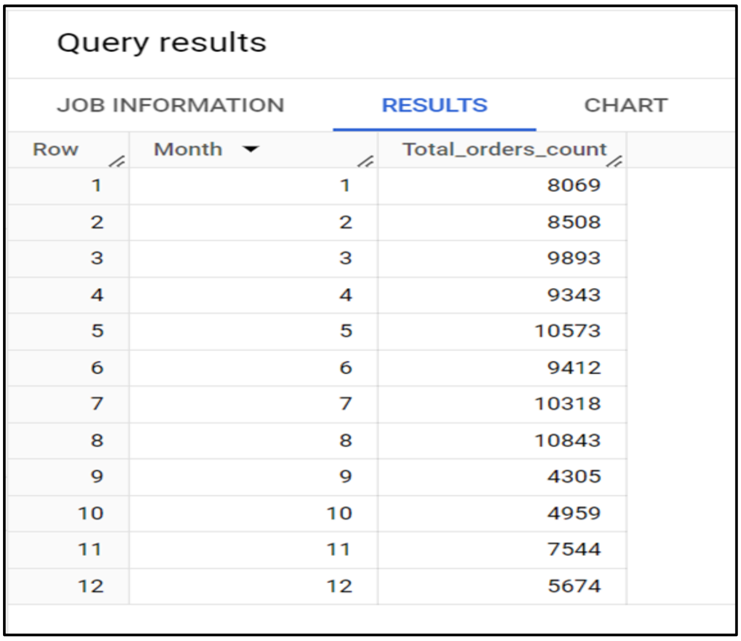


Yes, Initially the order count was very less in 2016 but it increased rapidly by the end of 2017 and kept increasing/fluctuating until 2018.

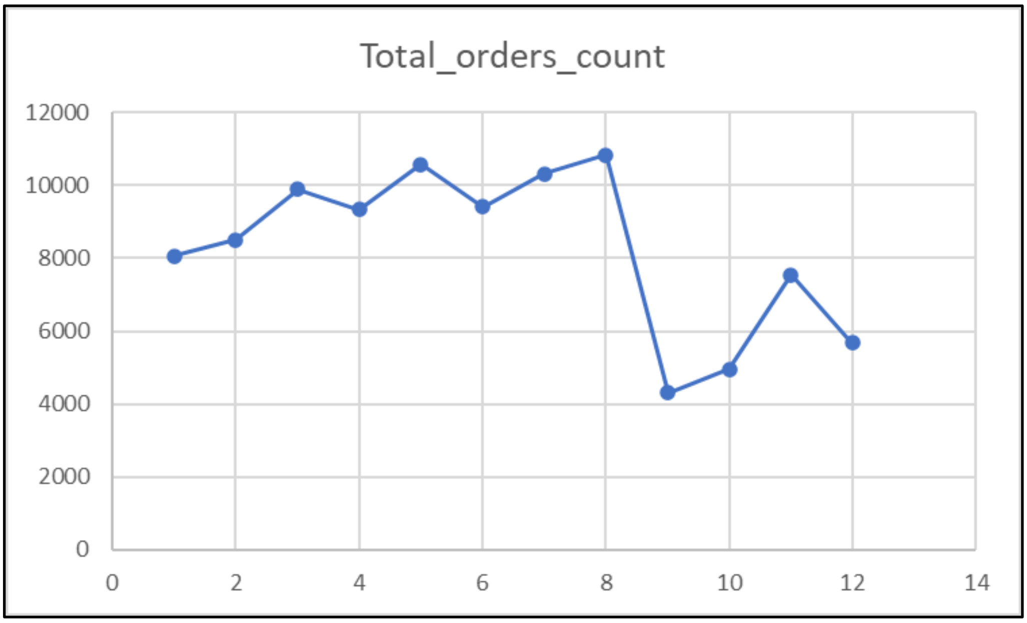
However, it was less after Sept 2018 due to the fact that there was no sufficient data from 2018. But overall the sales trend is upward.

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

```
SELECT EXTRACT(month from order_purchase_timestamp ) as Month,count(distinct order_id) as
Total_orders_count

FROM `Target.orders`

group by 1

order by 1
```



Row	Month	Total_orders_count
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674



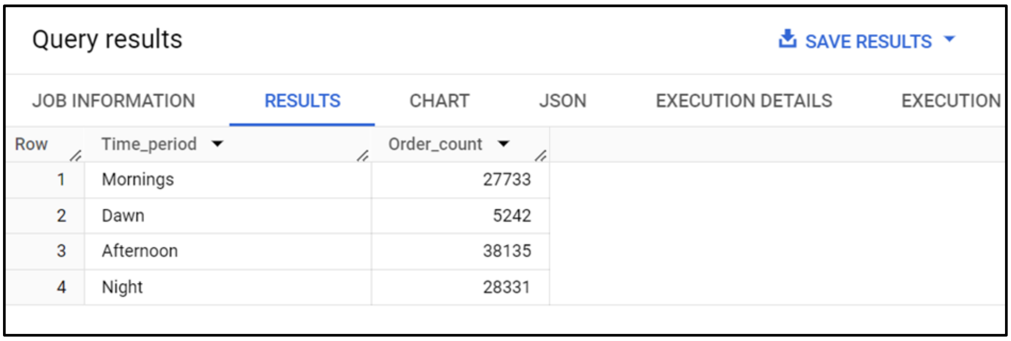
We see that the sales are most high in the months of May, July and August due to festivals in Brazil like Semana Santa in May. And overall the sales were high from January to September but went low after september.

Since the data is concluded till September 2018 we can see less order count after September.

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

```
SELECT

CASE WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 0 and 6 then 'Dawn'

WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 7 and 12 then 'Mornings'

WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 13 and 18 then 'Afternoon'

WHEN EXTRACT(HOUR FROM order_purchase_timestamp) between 19 and 23 then 'Night'

END  AS Time_period, COUNT(DISTINCT order_id) as Order_count

FROM `Target.orders`

GROUP BY 1
```



Row	Time_period	Order_count
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

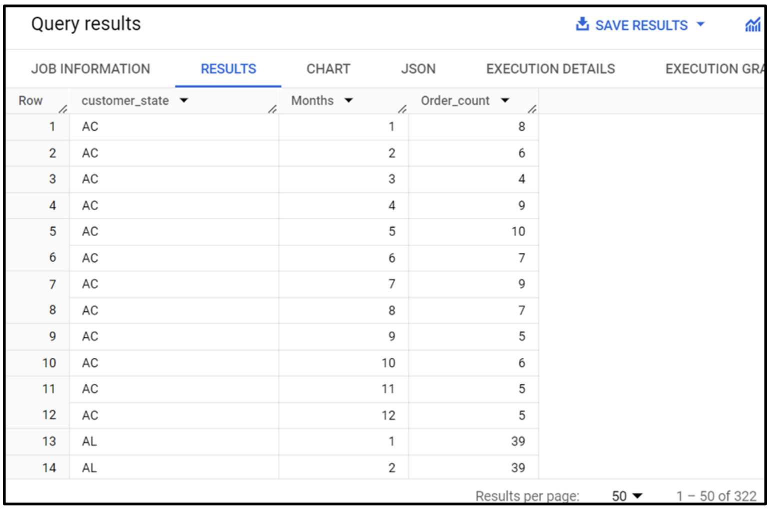
By Extracting the time from order_purchase_timestamp we divide time in 4 different periods Morning, Afternoon, Dawn, Night which helps us to analyse at what time the sales are more or less.

Brazilian customers tend to buy more in the afternoons and Mornings. Also it is understandable that the count is less in Dawn between 12 AM to 6 AM.

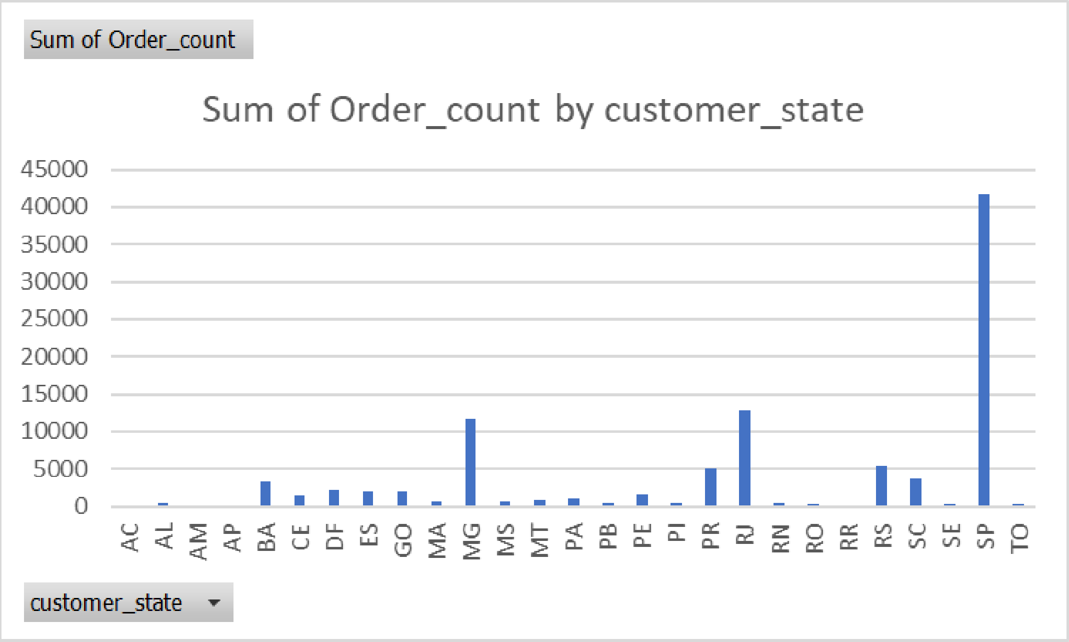
3. Evolution of E-commerce orders in the Brazil region:

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

```
SELECT C.customer_state,EXTRACT(Month from O.order_purchase_timestamp) as Months,count(*) Order_count

FROM `Target.orders` O

Inner join `Target.customers` C ON O.customer_id = C.customer_id

group by 1,2

order by 1,2
```



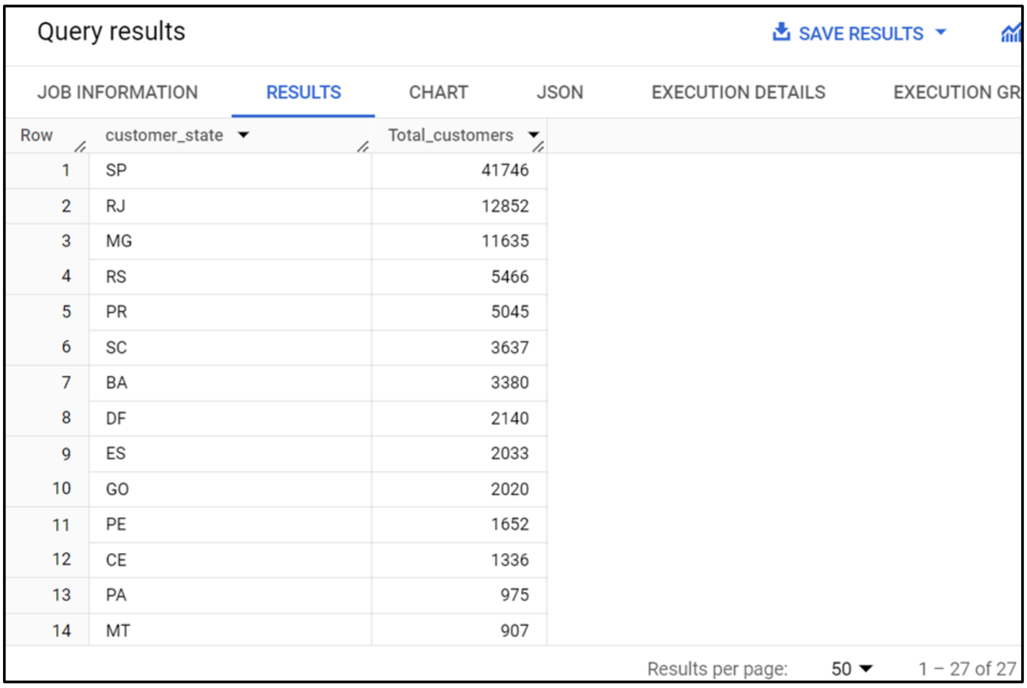
Row	customer_state	Months	Order_count
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5
12	AC	12	5
13	AL	1	39
14	AL	2	39



Here we can see that SP has the highest number of order count overall as compared to other states.

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

```
SELECT customer_state, count(customer_id) as Total_customers
    FROM `Target.customers`
    group by 1
    order by 2 desc
```



Row	customer_state	Total_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
11	PE	1652
12	CE	1336
13	PA	975
14	MT	907

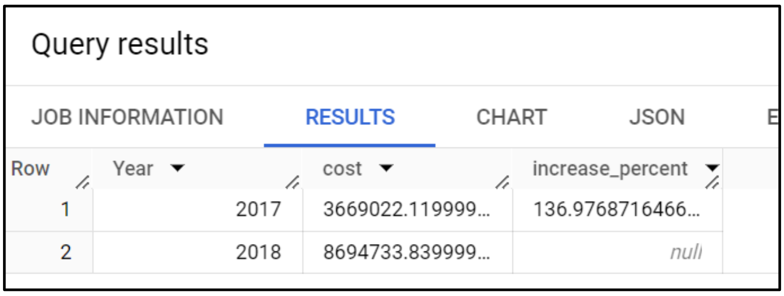
Results per page: 50 1 – 27 of 27

We can observe that Sao Paulo has the highest number of customers which is acceptable as Sao Paulo constitutes the heart of the Southeast, Brazil's most developed and populous region and is more economically productive.

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

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

```
with CTE AS(select*
          from `Target.orders` O
          join `Target.payments` P On O.order_id = P.order_id
       where extract(year from O.order_purchase_timestamp) between 2017        and 2018 AND
       extract(month from O.order_purchase_timestamp) between 0 and 8 ),

       Temp AS( select extract(year from order_purchase_timestamp) as Year, sum(payment_value)
       as cost
       from CTE
   group by 1
   order by 1)
   select*, ((lead(cost,1) over (order by Year) - cost)/cost)*100  as
increase_percent
       from Temp
```

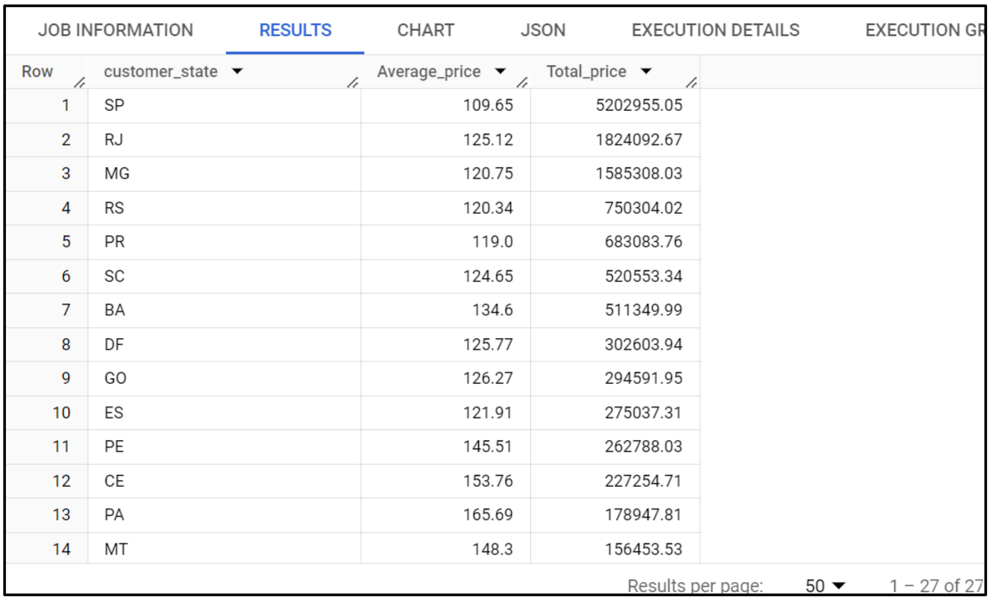


Row	Year	cost	increase_percent
1	2017	3669022.119999...	136.9768716466...
2	2018	8694733.839999...	null

The percent increase in the cost of orders from 2017 to 2018 is 136.97% henceforth we can conclude that the cost increase is reasonable and beneficial for Target, this data is calculated only for month January to August.

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

```
SELECT C.customer_state,Round(AVG(I.price),2) as Average_price,

Round(SUM(I.price),2) as Total_price

FROM `Target.orders` O

JOIN `Target.order_items` I ON O.order_id = I.order_id
JOIN `Target.customers` C ON O.customer_id = C.customer_id

GROUP BY 1

Order by 3 desc
```

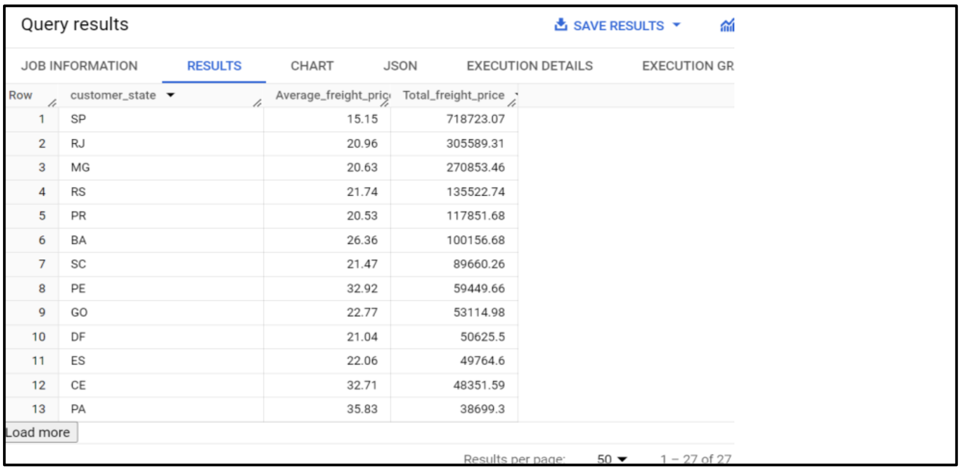


Row	customer_state	Average_price	Total_price
1	SP	109.65	5202955.05
2	RJ	125.12	1824092.67
3	MG	120.75	1585308.03
4	RS	120.34	750304.02
5	PR	119.0	683083.76
6	SC	124.65	520553.34
7	BA	134.6	511349.99
8	DF	125.77	302603.94
9	GO	126.27	294591.95
10	ES	121.91	275037.31
11	PE	145.51	262788.03
12	CE	153.76	227254.71
13	PA	165.69	178947.81
14	MT	148.3	156453.53

We see that Sao Paulo has the highest value of total _price but has the lowest Average_price.

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

```
SELECT C.customer_state, Round(AVG(I.freight_value),2) as Average_freight_price,

 Round(SUM(I.freight_value),2) as Total_freight_price

FROM `Target.orders` O

JOIN `Target.order_items` I ON O.order_id = I.order_id

JOIN `Target.customers` C ON O.customer_id = C.customer_id

GROUP BY 1 order by 3 desc
```



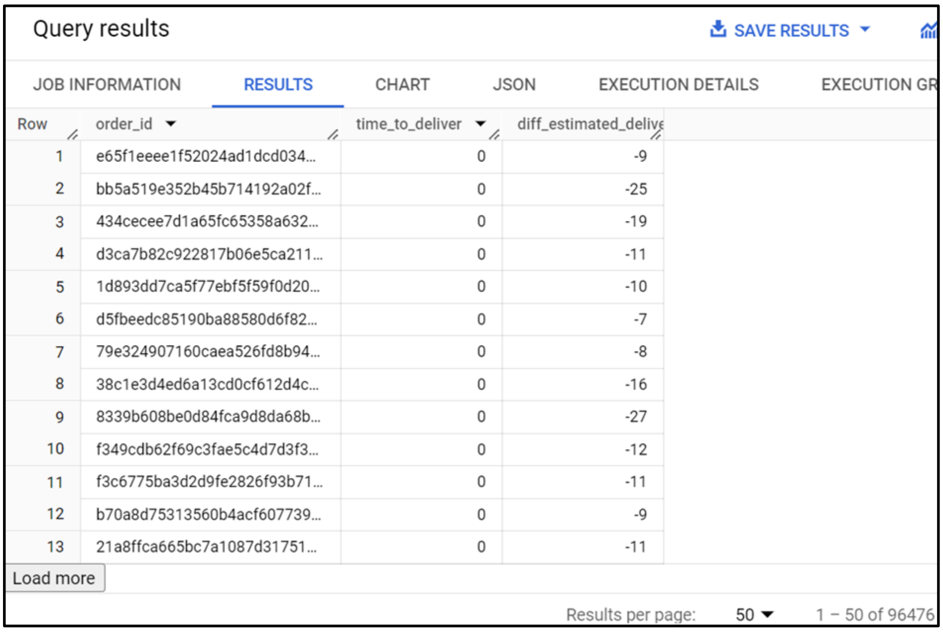
Row	customer_state	Average_freight_pric	Total_freight_price
1	SP	15.15	718723.07
2	RJ	20.96	305589.31
3	MG	20.63	270853.46
4	RS	21.74	135522.74
5	PR	20.53	117851.68
6	BA	26.36	100156.68
7	SC	21.47	89660.26
8	PE	32.92	59449.66
9	GO	22.77	53114.98
10	DF	21.04	50625.5
11	ES	22.06	49764.6
12	CE	32.71	48351.59
13	PA	35.83	38699.3

Similarly, We see that Sao Paulo has the highest value of Total_freight_value but has the lowest Average_freight_value.

5. Analysis based on sales, freight and delivery time

1. 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.

```
 SELECT order_id,

 Date_diff (order_delivered_customer_date, order_purchase_timestamp, Day) as time_to_deliver,

 Date_diff(order_delivered_customer_date , order_estimated_delivery_date, Day) as diff_estimated_delivery

FROM `Target.orders`

WHERE DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS NOT NULL

order by 2
```



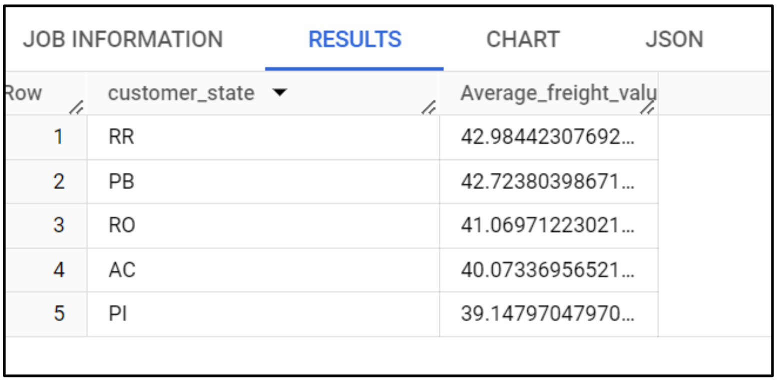
Row	order_id	time_to_deliver	diff_estimated_delive
1	e65f1eeee1f52024ad1dcd034...	0	-9
2	bb5a519e352b45b714192a02f...	0	-25
3	434cecee7d1a65fc65358a632...	0	-19
4	d3ca7b82c922817b06e5ca211...	0	-11
5	1d893dd7ca5f77ebf5f59f0d20...	0	-10
6	d5fbeedc85190ba88580d6f82...	0	-7
7	79e324907160caea526fd8b94...	0	-8
8	38c1e3d4ed6a13cd0cf612d4c...	0	-16
9	8339b608be0d84fca9d8da68b...	0	-27
10	f349cdb62f69c3fae5c4d7d3f3...	0	-12
11	f3c6775ba3d2d9fe2826f93b71...	0	-11
12	b70a8d75313560b4acf607739...	0	-9
13	21a8ffca665bc7a1087d31751...	0	-11

When the difference of order_delivered_customer_date and order_estimated_delivery_date results in negative that means the order was delivered to the customer before estimated delivery date.(before time)

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

## Top 5 states with highest average freight value

```
with CTE AS (SELECT C.customer_state, AVG(I.Freight_value) as Average_freight_value, SUM(I.Freight_value)

 FROM `Target.customers` C

 JOIN `Target.orders` O ON O.customer_id = C.customer_id

 JOIN `Target.order_items` I ON O.order_id = I.order_id

 GROUP BY 1 )

 SELECT customer_state, Average_freight_value

 from CTE

 ORDER BY 2 DESC

 LIMIT 5
```

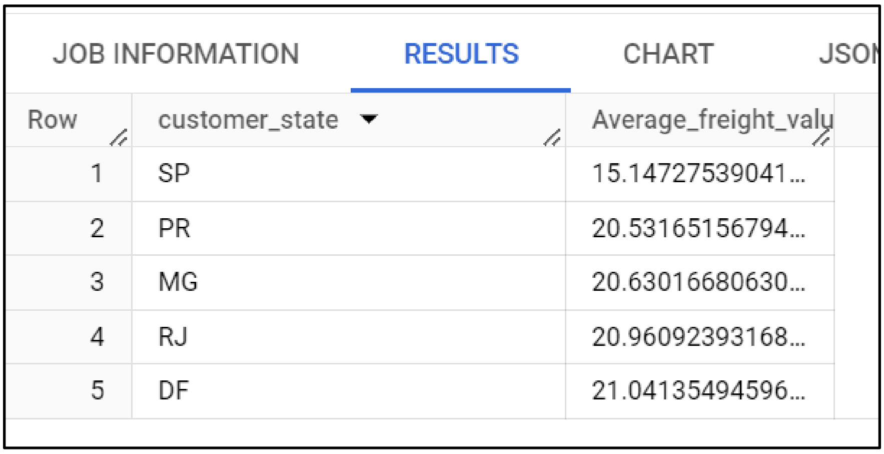


Row	customer_state	Average_freight_valu
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

Target should improve management & logistics in these states so as to bring down the average freight value.

## Top 5 states with lowest average freight value

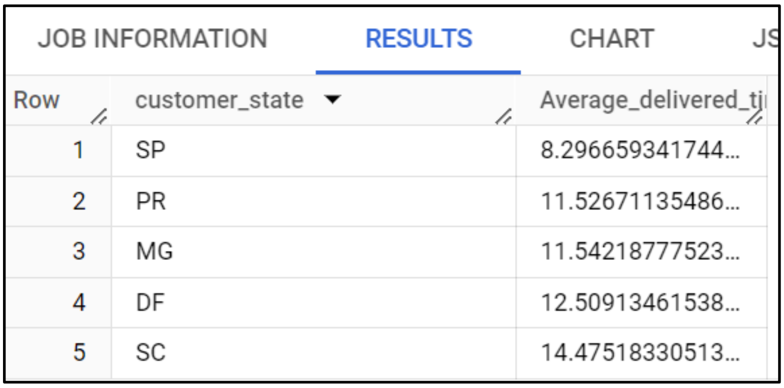
```
with CTE AS (
  SELECT C.customer_state, AVG(I.Freight_value) as Average_freight_value, SUM(I.Freight_value)
   FROM `Target.customers` C
  JOIN `Target.orders` O ON O.customer_id = C.customer_id
  JOIN `Target.order_items` I ON O.order_id = I.order_id
  GROUP BY 1 )
SELECT customer_state, Average_freight_value
from CTE
ORDER BY 2
LIMIT 5
```



Row	customer_state	Average_freight_valu
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

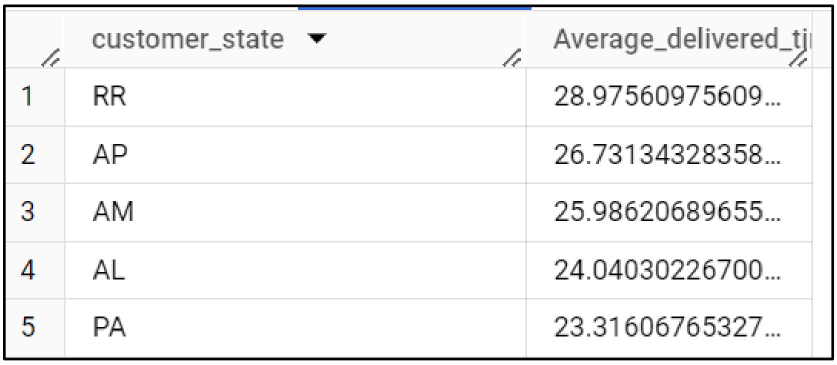
c. Find out the top 5 states with the highest & lowest average delivery time

```
##  Top 5 states with lowest average delivery time

SELECT C.customer_state, sum(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day))/count(order_id) as Average_delivered_time

from `Target.orders` O

JOIN `Target.customers` C ON O.customer_id = C.customer_id

WHERE order_status = 'delivered'

group by 1

order by 2

limit 5
```



Row	customer_state	Average_delivered_ti
1	SP	8.296659341744...
2	PR	11.52671135486...
3	MG	11.54218777523...
4	DF	12.50913461538...
5	SC	14.47518330513...

```
## Top 5 states with highest average delivery time

SELECT C.customer_state, sum(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day))/count(order_id) as Average_delivered_time

from `Target.orders` O

JOIN `Target.customers` C ON O.customer_id = C.customer_id

WHERE order_status = 'delivered'

group by 1

order by 2 desc

limit 5
```



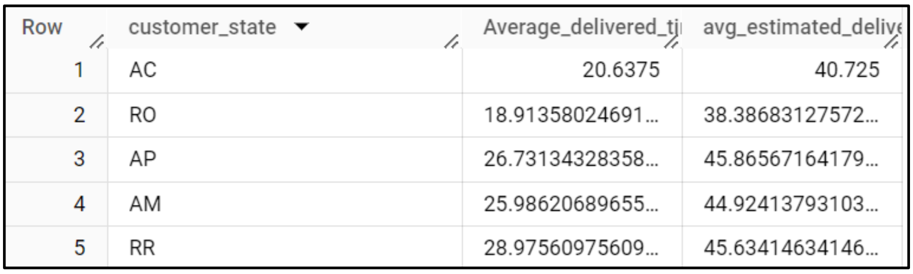
	customer_state	Average_delivered_ti
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...

Target should improve logistics in these states to bring down the average time of delivery. We notice that states that have high average freight cost also have high average time of delivery.

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

You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state

```
SELECT C.customer_state, sum(timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day))/count(order_id) as Average_delivered_time,

sum(timestamp_diff(order_estimated_delivery_date, order_purchase_timestamp, day))/count(order_id) as avg_estimated_delivery_date

from `Target.orders` O

JOIN `Target.customers` C ON O.customer_id = C.customer_id

WHERE order_status = 'delivered'

group by 1

order by (Average_delivered_time - avg_estimated_delivery_date)

limit 5
```



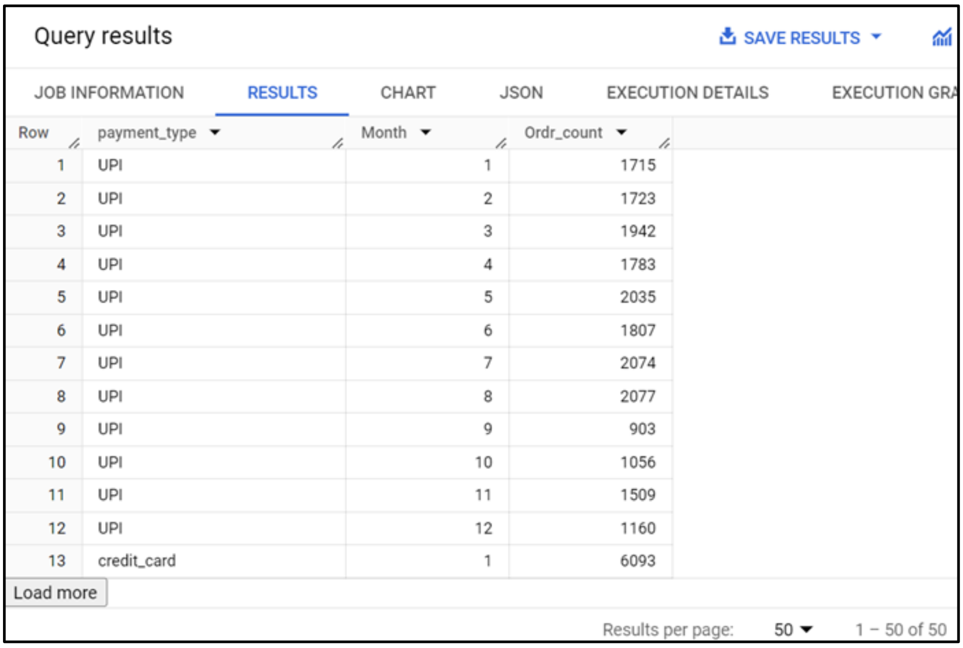
Row	customer_state	Average_delivered_ti	avg_estimated_delive
1	AC	20.6375	40.725
2	RO	18.91358024691...	38.38683127572...
3	AP	26.73134328358...	45.86567164179...
4	AM	25.98620689655...	44.92413793103...
5	RR	28.97560975609...	45.63414634146...

Top 5 states where the average estimated time is very fast as compared to average delivered time. Almost there is a gap of 20 days, Target should improve the accuracy of their estimated time algorithm.

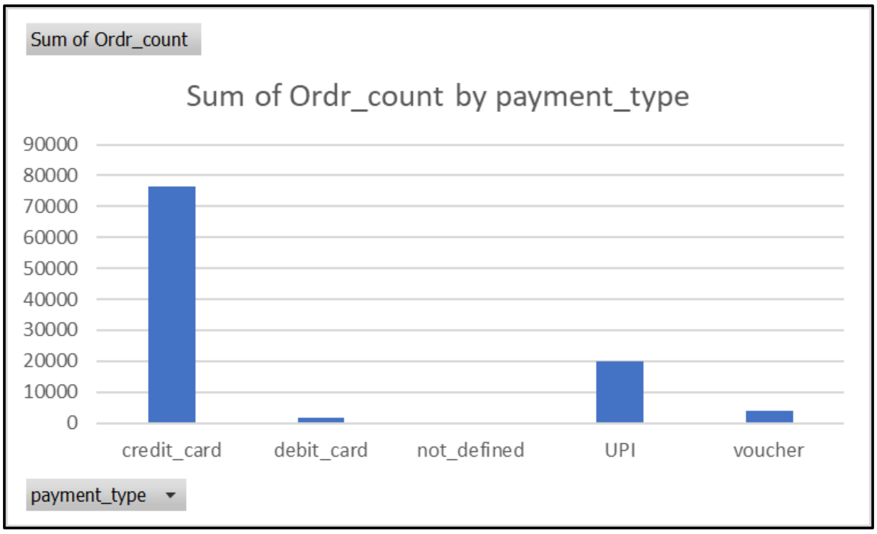
6. Analysis based on the payments-

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

```
SELECT P.payment_type, extract(month from O.order_purchase_timestamp) as Month, count(distinct
O.order_id) as Ordr_count

FROM `Target.orders` O

JOIN `Target.payments` P ON O.order_id = P.order_id

group by 1, 2

order by 1, 2
```



Row	payment_type	Month	Ordr_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6093

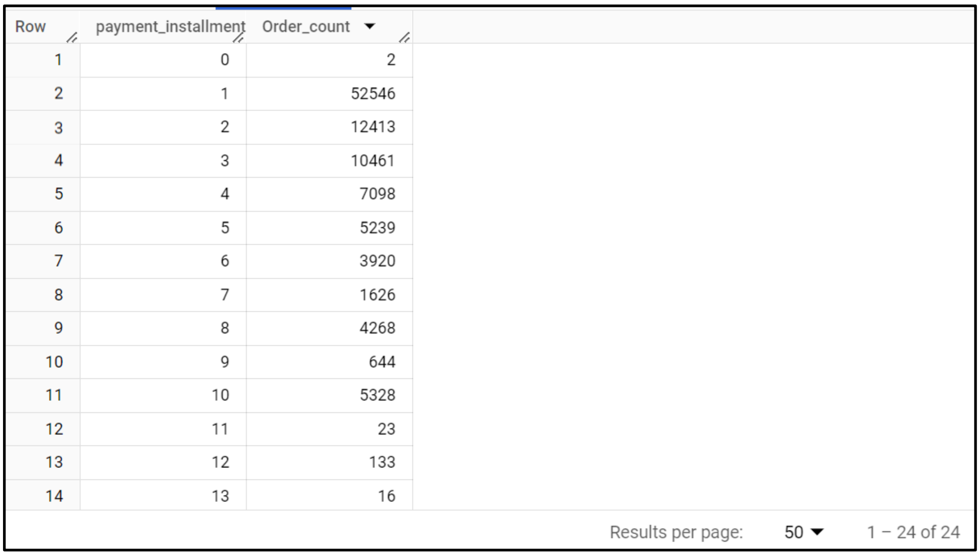


We see that the number of orders are steadily increasing month by month for all the payment types, However the payments by Debit cards/Vouchers are very less as compared to other payment methods. They should plan to offer more discounts on Debit cards and provide more vouchers on previous purchases.

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

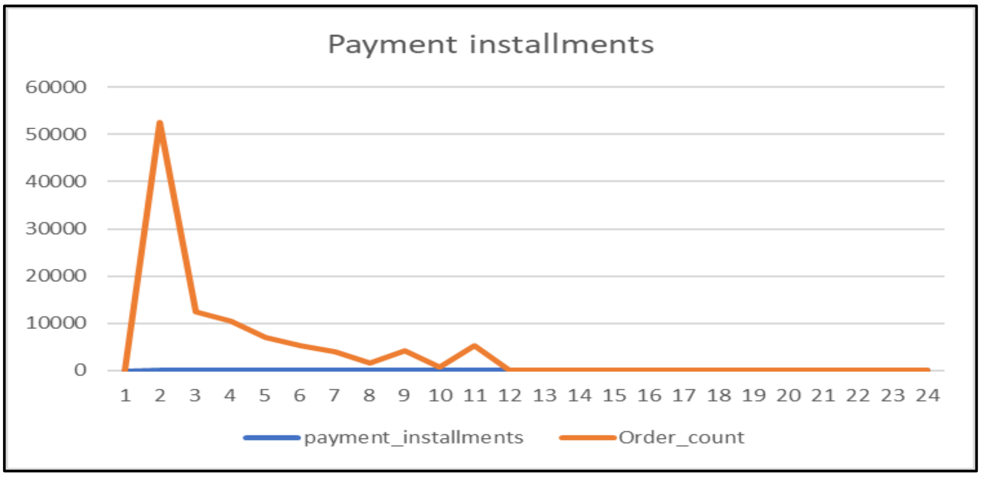
```
SELECT P.payment_installments, count(O.order_id) as Order_count

FROM `Target.orders` O

JOIN `Target.payments` P ON O.order_id = P.order_id

group by 1
```

```
order by 1, 2 desc
```



Row	payment_installment	Order_count
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
10	9	644
11	10	5328
12	11	23
13	12	133
14	13	16

Results per page: 50 ▾ 1 – 24 of 24



By the Result we observe that the most of order were purchased in single instalments, That is One time purchases are highest.

Observations and recommendations(Insights)

- There are around 4310 cities where the customers ordered, The Total order count is arranged in descending order to represent which city (Saopaulo) has the maximum ordered count.

- We see that the sales are most high in the months of May, July and August due to festivals in Brazil like Semana Santa in May. And overall the sales were high from January to September but went low after September.

- Since the data is concluded till September 2018, we can see less order count after September.

- By Extracting the time from order_purchase_timestamp we divide time in 4 different periods Morning, Afternoon, Dawn, Night which helps us to analyse at what time the sales are more or less.
    ○

- Brazilian customers tend to buy more in the afternoons and Mornings. Also, it is understandable that the count is less in Dawn between 12 AM to 6 AM.

- We can observe that Sao Paulo has the highest number of customers which is acceptable as Sao Paulo constitutes the heart of the Southeast, Brazil's most developed and populous region and is more economically productive.

- The percent increase in the cost of orders from 2017 to 2018 is 136.97% henceforth we can conclude that the cost increase is reasonable , this data is calculated only for month January to August.

- When the difference of order_delivered_customer_date and order_estimated_delivery_date results in negative that means the order was delivered to the customer before estimated delivery date.(before time)

- Target should improve logistics in these states to bring down the average time of delivery. We notice that states that have high average freight cost also have high average time of delivery.

- Top 5 states where the average estimated time is very fast as compared to average delivered time. Almost there is a gap of 20 days, Target should improve the accuracy of their estimated time algorithm.

- We see that the number of orders is steadily increasing month by month for all the payment types, However the payments by Debit cards/Vouchers are very less as compared to other payment methods. They should plan to offer more discounts on Debit cards and provide more vouchers on previous purchases.