

Target - Business Case

Topic: SQL

I. 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.

Ans:

Query:

```
select column_name ,  
       data_type  
from  
       my-1-project-  
408611.Target.INFORMATION_SCHEMA.COLUMNS  
where table_name='customers'  
order by ordinal_position;
```

Output:

SnapShot:

Query results [SAVE RESULTS](#)

	JOB INFORMATION	RESULTS	CHART	PREVIEW
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		

INSIGHTS: Observed that String and Integer data types are used in customers table. It is used to further analysis of dataset.

RECOMMENDATION: NA

B. Get the time range between which the orders were placed.**Ans:****Query:**

```
SELECT
    MIN(order_purchase_timestamp) AS STARTS_DATE,
    MAX(order_purchase_timestamp) AS ENDS_DATE
FROM
    Target.orders;
```

Output:**SnapShot:**

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	STARTS_DATE	ENDS_DATE					
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC					

INSIGHTS: This helps to determine the starting and ending date of the data set.

RECOMMENDATION: NA

C. Count the Cities & States of customers who ordered during the given period.**Ans:****Query:**

```
select distinct c.customer_city,
               c.customer_state,
               count(o.customer_id) as order_count
from
    Target.customers c
join
    Target.orders o
using(customer_id)
group by 1,2
order by 3 desc;
```

Output:
SnapShot:

JOB INFORMATION

RESULTS

CHART

PREVIEW

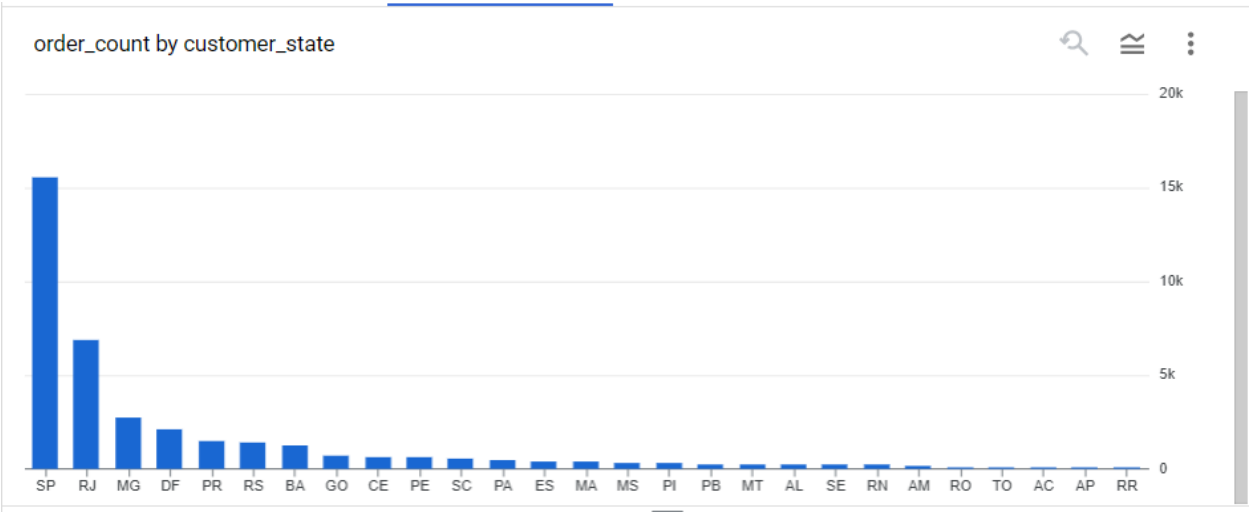
JSON

EXECUTION DETAILS

Row	customer_city	customer_state	order_count	
1	sao paulo	SP	15540	
2	rio de janeiro	RJ	6882	
3	belo horizonte	MG	2773	
4	brasilgia	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	

Results per page: 50 1 – 50 of 4310

Chart:



INSIGHTS: Observed that **Sao Paulo** City from state **SP** has more number of orders. Hence we can say that population is high in that state.

RECOMMENDATION: States like RJ, MG, DF, PR are next to SP in the orders counts. So we can improve the marketing in those states to get more orders from those states. This help to improve overall order counts.

II. In-depth Exploration:

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

Ans:

Query:

```
select extract(year from order_purchase_timestamp) as year,
       count(order_id) as order_count
from   Target.orders
group by 1
order by 1;
```

Output:

SnapShot:



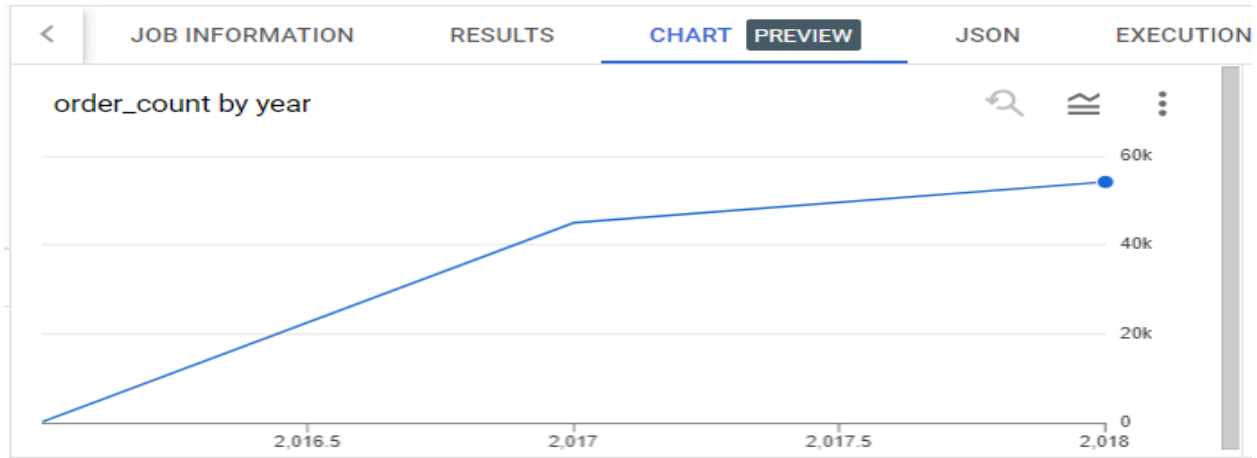
Query results				 SAVE RESULTS
 JOB INFORMATION				RESULTS
				CHART
				PREVIEW
Row	year	order_count		
1	2016	329		
2	2017	45101		
3	2018	54011		

Chart:

INSIGHTS: Observed that there is an increasing Trend (i.e.) No.of.Orders is keeps on increasing from the year 2016 to 2018.

RECOMMENDATION: Implement some referral rewards or offers to satisfy the customers and to repeat purchases from the customers. This helps to increase the trend.

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

Ans:

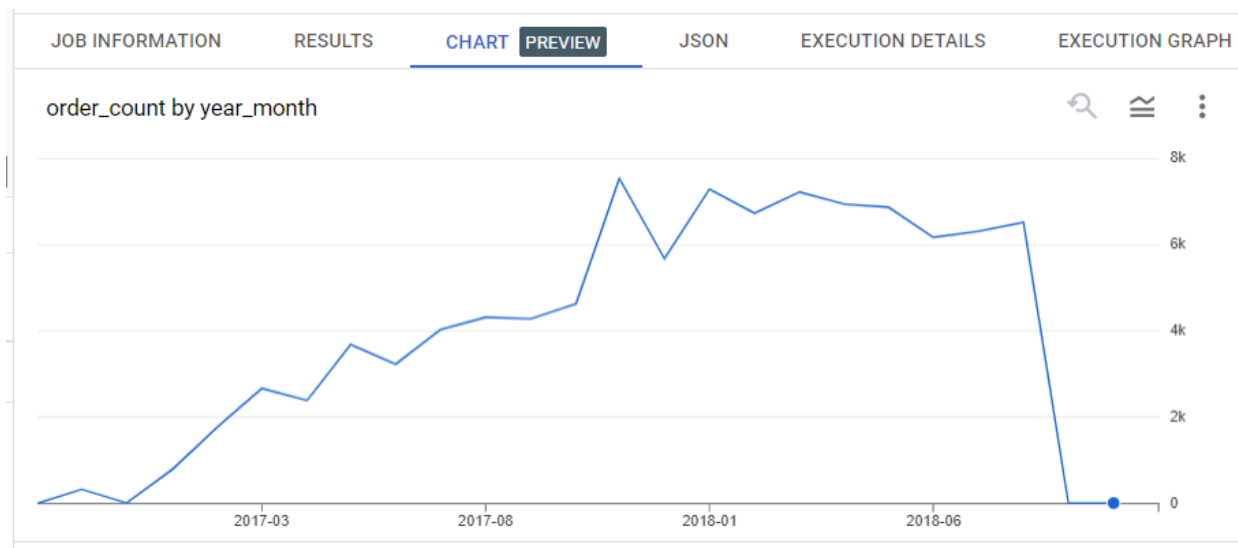
Query:

```
select format_date("%Y-%m",order_purchase_timestamp) as year_month,
       count(order_id) as order_count
from   Target.orders
group by 1
order by 1;
```

Output: SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	E
Row	year_month	order_count					
1	2016-09	4					
2	2016-10	324					
3	2016-12	1					
4	2017-01	800					
5	2017-02	1780					
6	2017-03	2682					
7	2017-04	2404					
8	2017-05	3700					
9	2017-06	3245					
10	2017-07	4026					
Results per page: 50							1 – 25 of 25

Chart:



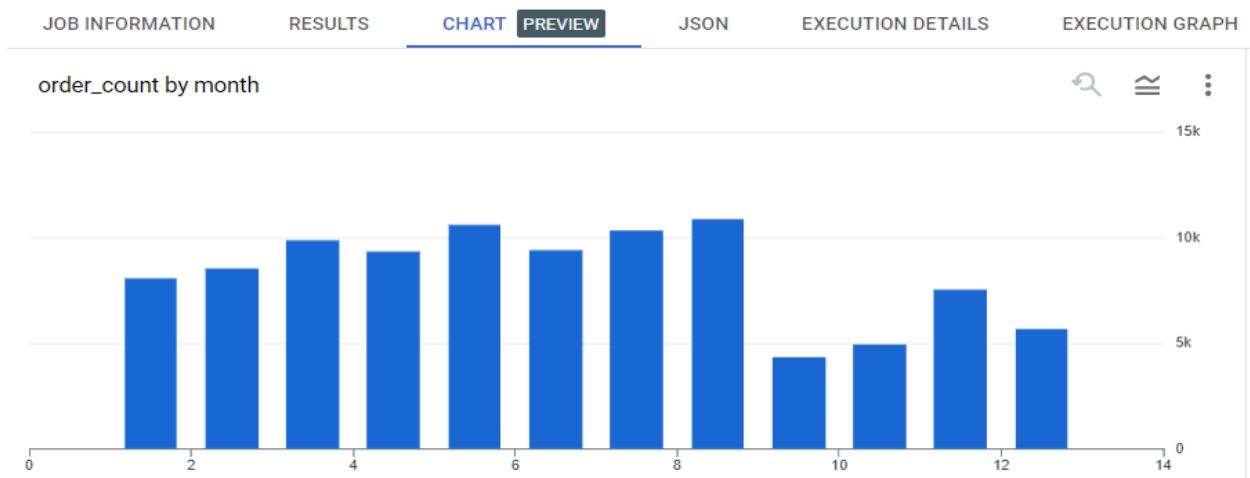
OR

```
select extract(month from order_purchase_timestamp) as month,
       count(order_id) as order_count
from   Target.orders
group by 1
order by 1;
```

**Output:
SnapShot:**

JOB INFORMATION		RESULTS		CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	month	order_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					

Results per page: 50 ▼ 1 – 12 of 12

Chart:

INSIGHTS: Observed high orders on Festival seasons like Carnival (Carnaval) during February to March, Semana Santa during April to May and Festival de Cachaça, Paraty during July to August.

RECOMMENDATION: Implement some rewards or offers for repeated customers and stock clearance sales during the months September to October.

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

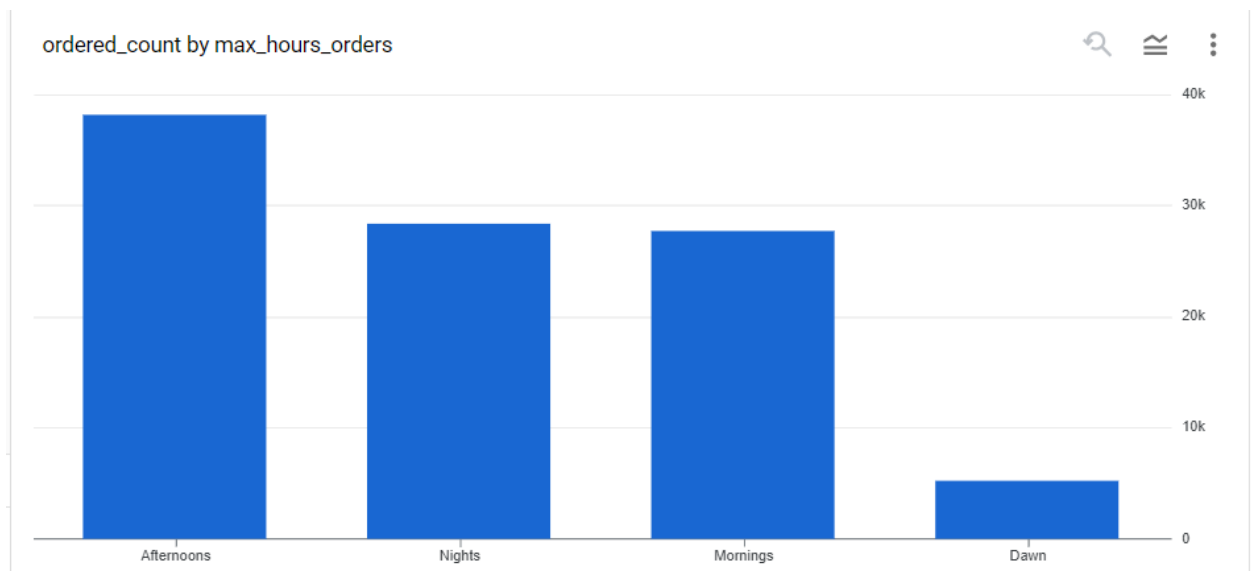
Ans:

Query:

```
select
case
when extract(hour from order_purchase_timestamp) between 0 and 6 then
"Dawn"
when extract(hour from order_purchase_timestamp) between 7 and 12 then
"Mornings"
when extract(hour from order_purchase_timestamp) between 13 and 18 then
"Afternoons"
when extract(hour from order_purchase_timestamp) between 19 and 23 then
"Nights"
end as max_hours_orders,
count(order_id) as ordered_count
from
Target.orders
group by max_hours_orders
order by 2 desc;
```

**Output:
SnapShot:**

Query results			SAVE RESULTS
JOB INFORMATION		RESULTS	CHART
Row	max_hours_orders	ordered_count	PREVIEW
1	Afternoons	38135	
2	Nights	28331	
3	Mornings	27733	
4	Dawn	5242	

Chart:

INSIGHTS: We can observe from the chart that Brazil Customers placing more Orders during day time especially in the Afternoons.

RECOMMENDATION: Need to maintain store stock and other resources like network maintainers and customer representative for seamless purchase.

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

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

Ans:

Query:

```
select c.customer_state,
       format_date("% Y-%m", o.order_purchase_timestamp) as year_month,
       count(o.order_id) as order_count
from
  Target.customers c
join
  Target.orders o
using(customer_id)
group by 1,2
order by 1,2;
```

Output:

SnapShot:

Query results

SAVE RESULTS

EX

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

E

Row	customer_state	year_month	order_count	
1	AC	2017-01	2	
2	AC	2017-02	3	
3	AC	2017-03	2	
4	AC	2017-04	5	
5	AC	2017-05	8	
6	AC	2017-06	4	
7	AC	2017-07	5	
8	AC	2017-08	4	
9	AC	2017-09	5	
10	AC	2017-10	6	

Results per page:

50

1 – 50 of 565

INSIGHTS: Observed that **Sao Paulo** City from state **SP** has more number of orders.

RECOMMENDATION: Implement more marketing and sales strategies in the other states.

B. How are the customers distributed across all the states?**Ans:****Query:**

```
select customer_state,  
       count(customer_id) as customer_count_per_state  
from   Target.customers  
group by 1  
order by 2 desc;
```

Output:**SnapShot:**

JOB INFORMATION

RESULTS

CHART

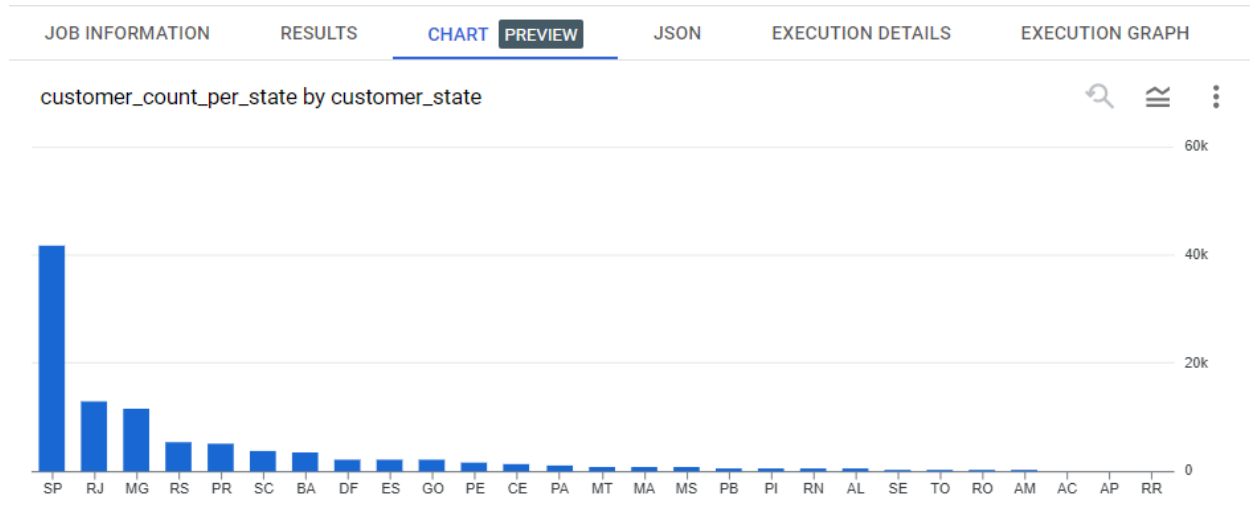
PREVIEW

JSON

EXECUTION DETAILS

Row	customer_state	customer_count_per_state
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

Results per page: 501 – 27 of 27

Chart:

INSIGHTS: Observed that **Sao Paulo** City from state **SP** has more number of customers. Hence we can say that population is high in that state.

RECOMMENDATION: Implement more marketing and sales strategies in the states like RR, AP, AC and AM to increase the customers from those regions.

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). You can use the “payment value” column in the payments table to get the cost of orders.


Ans:

Query:

```
with cte as (  
    select extract(year from order_purchase_timestamp) as years,  
           round(sum(p.payment_value),2) as year_total_value,  
           lag(sum(p.payment_value))over(order by sum(p.payment_value)) as  
previous_year_total  
  
from  
    Target.orders o  
join  
    Target.payments p  
using(order_id)  
where extract(month from order_purchase_timestamp) between 1 and 8  
group by 1  
order by 1)  
select years,  
       year_total_value,  
       round(ifnull(((year_total_value-  
previous_year_total)/previous_year_total)*100,0),2) as percentage_increase  
from cte;
```

**Output:
SnapShot:**

Query results

 SAVE RESULTS

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

Row	years	year_total_value	percentage_increase
1	2017	3669022.12	0.0
2	2018	8694733.84	136.98

Chart:

INSIGHTS: Observed that cost of orders goes on increasing from 2017 to 2018. It is increased by approximately 136.98%

RECOMMENDATION: Implement some sales strategies and marketing to maintain the increasing trend and improve the sales.

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

Ans:

Query:

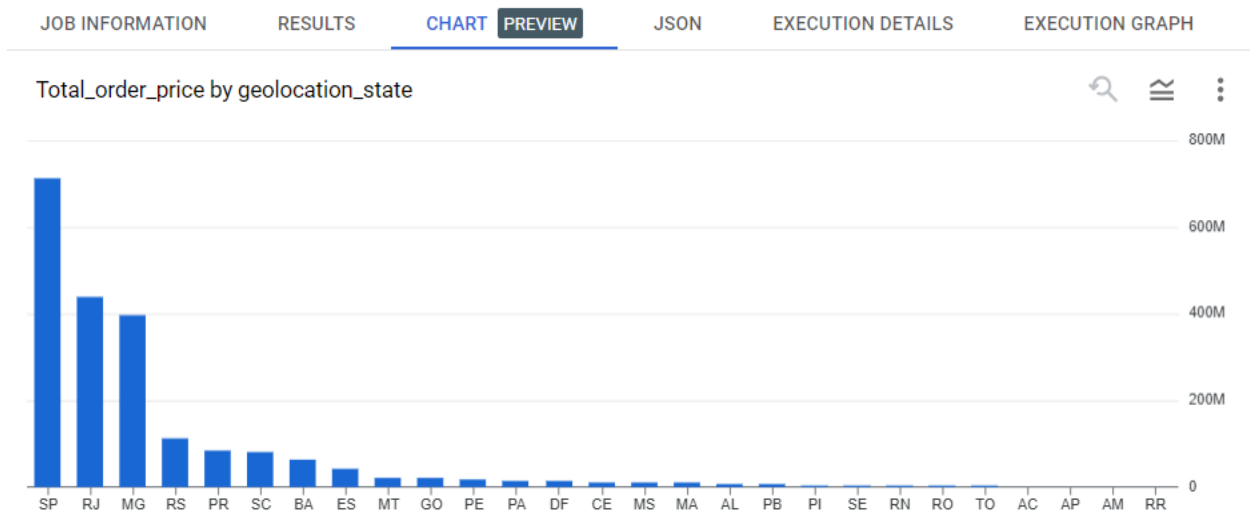
```
select g.geolocation_state,
       round(sum(oi.price),2) as Total_order_price,
       round(Avg(oi.price),2) as Average_order_price
from
  Target.geolocation g
left join
  Target.customers c
on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
left join
  Target.orders o using(customer_id)
left join
  Target.order_items oi
on o.order_id=oi.order_id
group by g.geolocation_state
order by g.geolocation_state;
```

Output:

SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	geolocation_state ▼	Total_order_price ▼	Average_order_price			
1	AC	1494037.73	179.31			
2	AL	7191886.1	196.64			
3	AM	825147.21	131.67			
4	AP	988578.63	177.1			
5	BA	62377311.67	149.64			
6	CE	10819201.81	151.32			
7	DF	13141649.62	124.66			
8	ES	43634878.56	123.36			
9	GO	20860945.92	134.62			
10	MA	9020091.01	150.95			

Results per page: 50 ▼ 1 – 27 of 27

Chart:**Chart:**

INSIGHTS: Observed that the state **SP** has the highest order price and the state **PB** has the highest Average order price among all the states.

RECOMMENDATION: Implement some awareness programs, marketing and offers on the products to increase the order rate in the lower order price areas.

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

Ans:

Query:

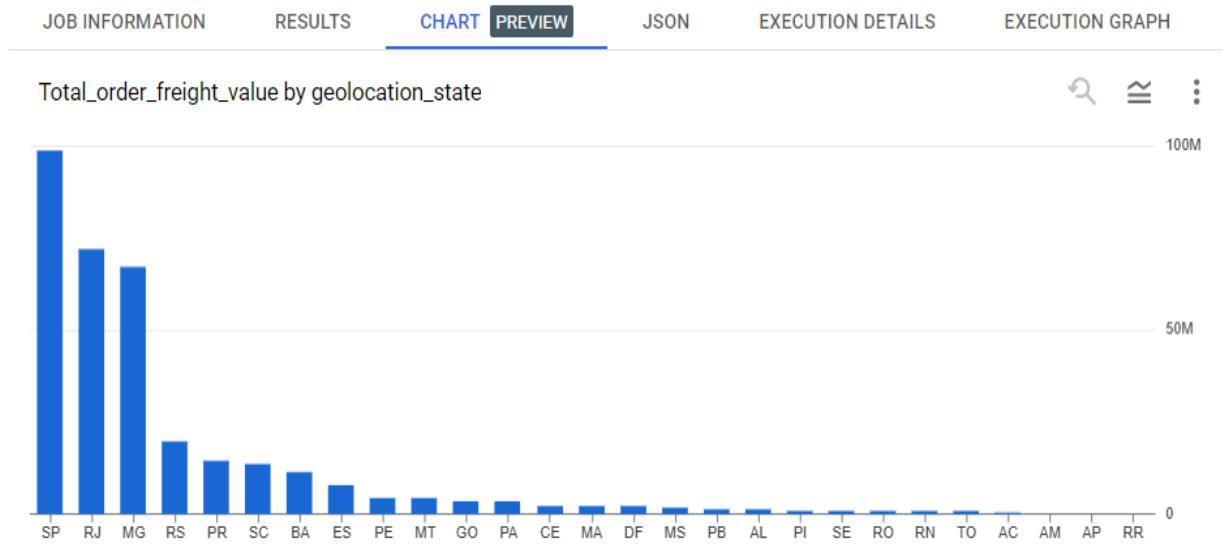
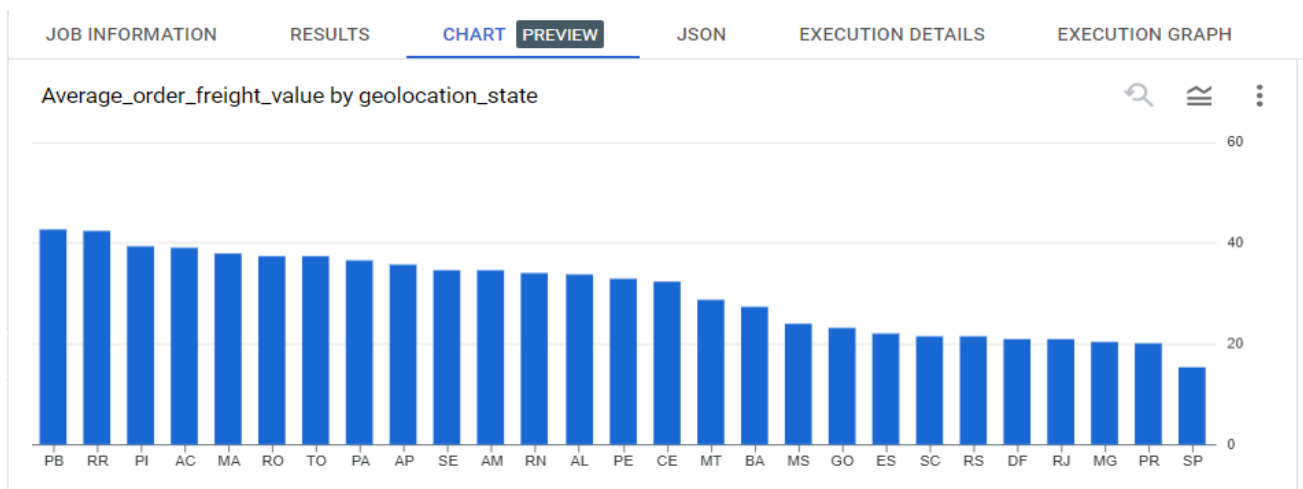
```
select g.geolocation_state,
       round(sum(oi.freight_value),2) as Total_order_freight_value,
       round(Avg(oi.freight_value),2) as Average_order_freight_value
from
  Target.geolocation g
left join
  Target.customers c
on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
left join
  Target.orders o using(customer_id)
left join
  Target.order_items oi
on o.order_id=oi.order_id
group by g.geolocation_state
order by g.geolocation_state;
```

Output:

SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	geolocation_state ▼	Total_order_freight_value ▼	Average_order_freight_value ▼			
1	AC	325767.64	39.1			
2	AL	1237356.22	33.83			
3	AM	216974.1	34.62			
4	AP	199028.01	35.66			
5	BA	11345094.0	27.22			
6	CE	2306600.06	32.26			
7	DF	2214955.55	21.01			
8	ES	7799979.09	22.05			
9	GO	3590268.56	23.17			
10	MA	2275191.86	38.08			

Results per page: 50 ▼ 1 – 27 of 27

Chart:**Chart:**

INSIGHTS: Observed that the state **SP** has the highest Freight Value and the state **PB** has the highest Average Freight Value among all the states.

RECOMMENDATION: Increase the efficiency of their Logistics.

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. You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

Ans:

Query:

```
select order_id,
       date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY
) as time_to_deliver,
       date_diff(order_estimated_delivery_date,order_delivered_customer_date,D
AY) as diff_estimated_delivery
from
Target.orders
where order_status="delivered";
```

Output:

SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	order_id	time_to_deliver	diff_estimated_delivery			
1	635c894d068ac37e6e03dc54eccb6189	30	1			
2	3b97562c3aee8bdedcb5c2e45a50d5e1	32	0			
3	68f47f50f04c4cb6774570cfde3a9aa7	29	1			
4	276e9ec344d3bf029ff83a161c6b3ce9	43	-4			
5	54e1a3c2b97fb0809da548a59f64c813	40	-4			
6	fd04fa4105ee8045f6a0139ca5b49f27	37	-1			
7	302bb8109d097a9fc6e9cefc5917d1f3	33	-5			
8	66057d37308e787052a32828cd007e58	38	-6			
9	19135c945c554eebfd7576c733d5ebdd	36	-2			
10	4493e45e7ca1084efcd38ddebf174dda	34	0			

Results per page: 50 1 – 50 of 96478

INSIGHTS: Observed that the time taken to deliver each order.

RECOMMENDATION: Collaborate with logistics and plan to deliver the products before the estimated delivery time of that product.

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

Ans:

Query:

```
with cte as (select g.geolocation_state as state,
    round(avg(oi.freight_value),2) as avg_freight_value,
    dense_rank()over(order by round(avg(oi.freight_value),2) desc) as
desc_freight_rank,
    dense_rank()over(order by round(avg(oi.freight_value),2) asc) as
asc_freight_rank
from
    Target.geolocation g
join
    Target.customers c
on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
join
    Target.orders o using(customer_id)
join
    Target.order_items oi
on o.order_id=oi.order_id
group by g.geolocation_state
order by 2 desc)
select state,
    "Highest" as highest_lowest_avg_freight_value,
    avg_freight_value
from
cte
where desc_freight_rank <=5
union all
select state,
    "Lowest" as highest_lowest_avg_freight_value,
    avg_freight_value
from
cte
where asc_freight_rank <=5;
```

Output: SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	state ▼	highest_lowest_avg_freight_value	avg_freight_value ▼			
1	SP	Lowest	15.41			
2	PR	Lowest	20.15			
3	RJ	Lowest	20.9			
4	DF	Lowest	21.01			
5	MG	Lowest	20.46			
6	PB	Highest	42.77			
7	RR	Highest	42.47			
8	PI	Highest	39.48			
9	AC	Highest	39.1			
10	MA	Highest	38.08			

INSIGHTS: Observed that the state **PB** has the highest Average freight value and the state **SP** lowest has the Average freight value.

RECOMMENDATION: Optimize their Logistics.

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

Ans:

Query:

```

with cte as (select g.geolocation_state as state,
    round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase
_timestamp,DAY)),2) as avg_delivery_time,
    dense_rank()over(order by
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_tim
estamp,DAY)),2) desc) as desc_avg_delivery_time,
    dense_rank()over(order by
round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_tim
estamp,DAY)),2) asc) as asc_avg_delivery_time
from
    Target.geolocation g
join
    Target.customers c

```

```

on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
join
  Target.orders o using(customer_id)
join
  Target.order_items oi
on o.order_id=oi.order_id
group by g.geolocation_state
order by 2 desc)
select state,
       "Highest" as highest_lowest_avg_delivery_time,
       avg_delivery_time
from
cte
where desc_avg_delivery_time <=5
union all
select state,
       "Lowest" as highest_lowest_avg_freight_value,
       avg_delivery_time
from
cte
where asc_avg_delivery_time <=5;

```

Output:
SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	
Row	state ▼	highest_lowest_avg_delivery_time ▼	avg_delivery_time ▼				
1	PR	Lowest				11.0	
2	DF	Lowest				12.44	
3	SP	Lowest				8.44	
4	MG	Lowest				11.36	
5	RJ	Lowest				14.39	
6	AP	Highest				30.4	
7	AM	Highest				24.38	
8	RR	Highest				23.98	
9	AL	Highest				22.87	
10	PA	Highest				22.73	

INSIGHTS: Observed that the state **AP** has the highest Average Delivery Time and the state **SP** lowest has the Average Delivery Time.

RECOMMENDATION: Optimize their Logistics.

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.

Ans:

Query:

```
with cte as (select g.geolocation_state as state,
    round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)),2) as avg_diff_delivery_days,
    dense_rank()over(order by
round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)),2) desc) as desc_avg_diff_delivery_days
from
    Target.geolocation g
join
    Target.customers c
on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix
join
    Target.orders o using(customer_id)
join
    Target.order_items oi
on o.order_id=oi.order_id
where o.order_status="delivered"
group by g.geolocation_state
order by 2 desc)
select state,
    "Highest" as highest_lowest_avg_delivery_time,
    avg_diff_delivery_days
from
cte
where desc_avg_diff_delivery_days <=5
```


**Output:
SnapShot:**

Query results [SAVE RESULTS](#) ▼

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	state ▼	highest_lowest_avg_delivery_time	avg_diff_delivery_days ▼			
1	RR	Highest				20.88
2	AM	Highest				20.56
3	RO	Highest				19.1
4	AC	Highest				18.56
5	AP	Highest				15.65

INSIGHTS: Observed that the state RR has the highest Order Delivery Rate.

RECOMMENDATION: Optimize their Logistics.

VI. Analysis based on the payments:

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

Ans:

Query:

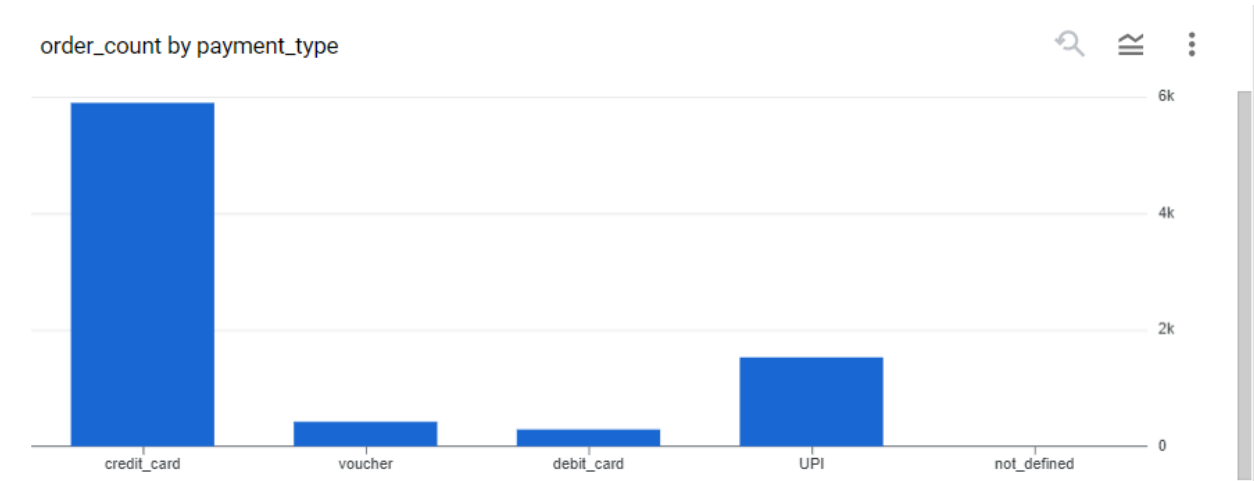
```
select format_date("%Y-%m",o.order_purchase_timestamp) as year_month,
       p.payment_type,
       count(o.order_id) as order_count
from
  Target.orders o
join
  Target.payments p
using(order_id)
group by 1,2;
```

Output:

SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	year_month	payment_type	order_count			
1	2016-09	credit_card	3			
2	2016-10	credit_card	254			
3	2016-10	voucher	23			
4	2016-10	debit_card	2			
5	2016-10	UPI	63			
6	2016-12	credit_card	1			
7	2017-01	voucher	61			
8	2017-01	UPI	197			
9	2017-01	credit_card	583			
10	2017-01	debit_card	9			

Results per page: 50 ▼ 1 – 50 of 90

Chart:

INSIGHTS: Observed that most of the customer using **Credit Card** as payment type followed by **UPI**.

RECOMMENDATION: Maintain Point of sale terminal and Network for seamless shopping. Enhance the customer service in case of any issue facing the customers.

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

Ans:

Query:

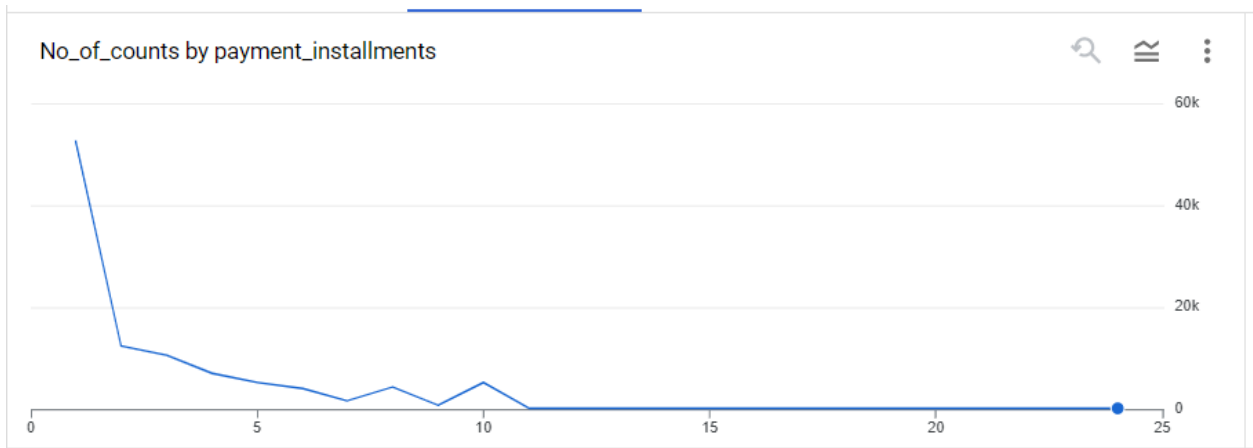
```
select payment_installments,
       count(*) as No_of_counts
from   Target.payments
where  payment_installments >= 1
group by payment_installments;
```

Output: SnapShot:

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS
Row	payment_installment	No_of_counts				
1	1	52546				
2	2	12413				
3	3	10461				
4	4	7098				
5	5	5239				
6	6	3920				
7	7	1626				
8	8	4268				
9	9	644				
10	10	5328				

Results per page: 50 ▼ 1 – 23 of 23

Chart:



INSIGHTS: Observed that payment installments 1 have maximum counts whereas highest payment installment 24 has 18 counts. It shows us the increasing trend of one month installment and credit card usage.

RECOMMENDATION: Implement some payment strategies and improve customer satisfaction.