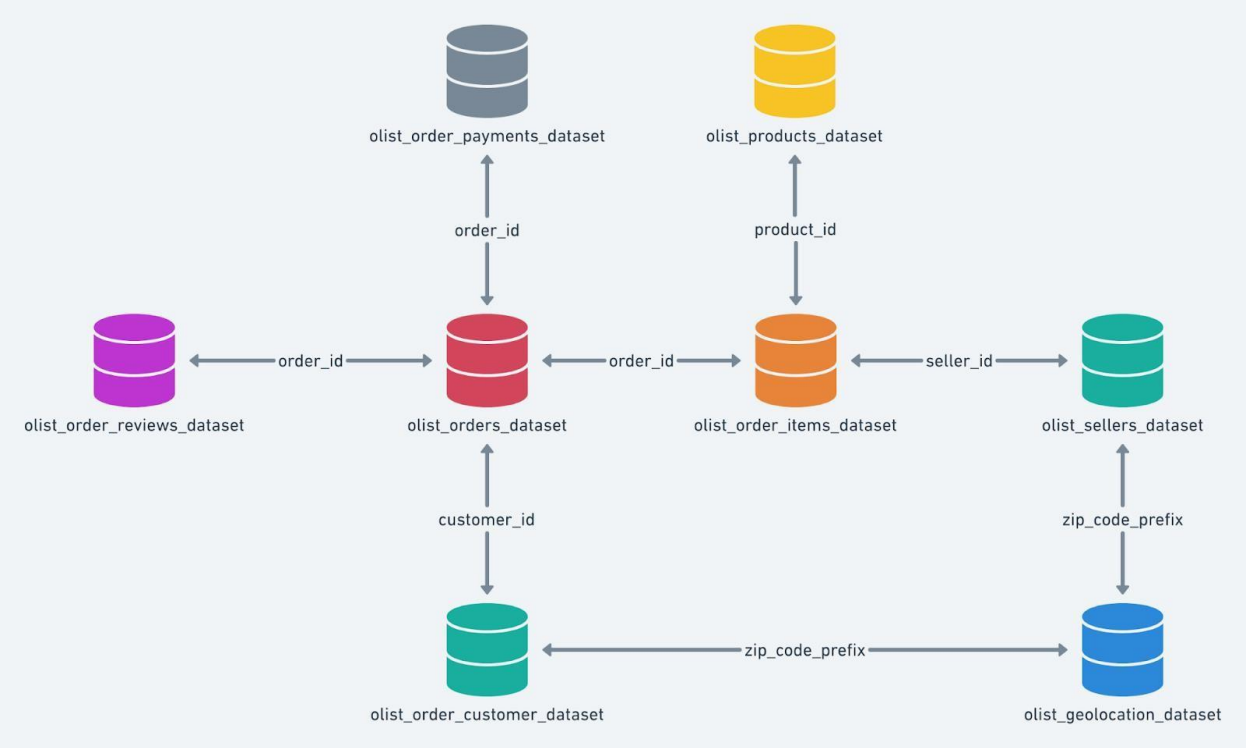
Dataset schema:



What does 'good' look like?

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.

We are retrieving the dataset of all the columns in the **customer table** from the information_schema.columns and we have verified the column data types

Syntax

```
SELECT
   column_name,data_type
 FROM
   neat-axis-409607.Market.INFORMATION_SCHEMA.COLUMNS
 WHERE
table_name='customers';
```

```
SELECT
    column_name,data_type
  FROM
    neat-axis-409607.Market.INFORMATION_SCHEMA.COLUMNS
  WHERE
table_name='customers';
```

Output

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

Graph:

NA

Insights

By understanding the data types of each table.
All columns names store consistency data types through the scheme.

Recommendations:
We ensure analysis and interpretation of the datasets.

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

We need to check in **orders** table **order_purchase_timestamp** column to fetch the output. Need to retrieve the earliest and latest order based on the timestamp.
Below is the scheme of **orders table**
Syntax:

SELECT max(order_purchase_timestamp), min(order_purchase_timestamp) FROM `neat-axis-409607.Market.orders`;

```
SELECT max(order_purchase_timestamp),
min(order_purchase_timestamp) FROM `neat-axis-409607.Market.orders`;
```

Output:

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	first_order	last_order
1	2016-09-04 21:15:00 UTC	2018-10-17 17:30:00 UTC

Graph:

NA

Insights

This allowed us to determine the start and end dates of the data i.e, from Sep 04, 2016 to Oct 17, 2018

Recommendations

- We can analyze the orders based on time stamp along with delivery date too (estimated

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

My understanding for this question is to pull the records from cities and states who ordered during given period and their count.

We need to views the **customer** table (for city and state)
Used count to get the number of cities and states

Syntax:

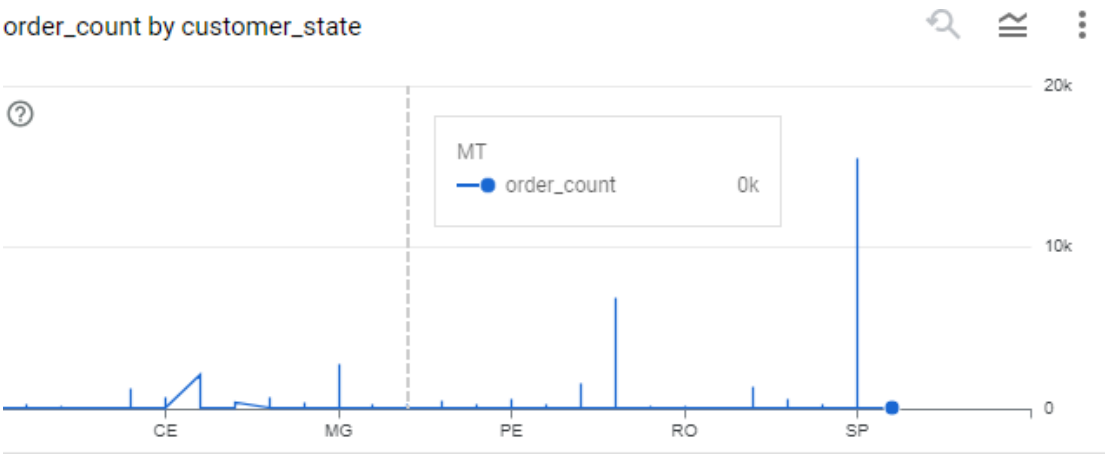
SELECT customer_state,customer_city,count(*) as order_count
FROM neat-axis-409607.Market.customers c
join neat-axis-409607.Market.orders o on c.customer_id=o.customer_id
group by customer_state,customer_city
order by customer_state,customer_city;

```
SELECT customer_state,customer_city,count(*) as order_count
FROM neat-axis-409607.Market.customers c
join neat-axis-409607.Market.orders o on c.customer_id=o.customer_id
group by customer_state,customer_city;
```

Output: No of records are 4310

Row	customer_state	customer_city	order_count
1	RN	acu	3
2	CE	ico	8
3	RS	ipe	2
4	CE	ipu	4
5	SC	ita	3
6	SP	itu	136
7	SP	jau	74
8	MG	luz	2
9	SP	poa	85
10	MG	uba	53

Graph



Insights

We can see that SP state alone has more orders in Brazil.

Recommendations

Need to give some offers/discounts so that others country people shows interest to increase to place the orders.

II. In-depth Exploration:

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

We need to extract from month and year from order_purchase_timestamp which is in orders table.

Syntax:

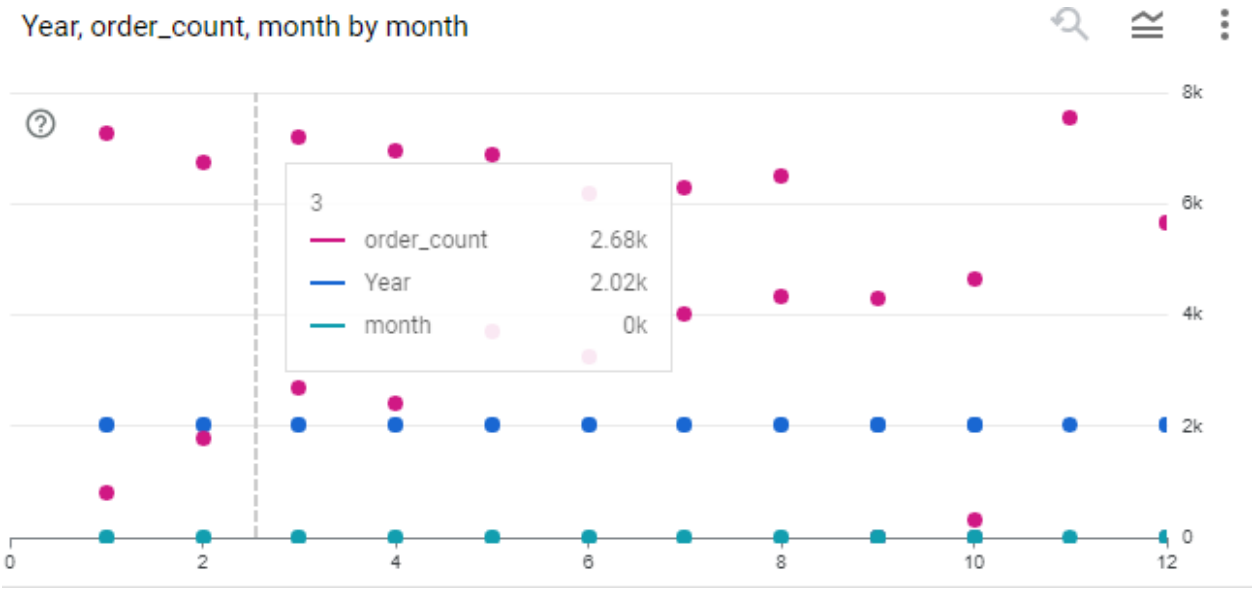
SELECT
Extract (MONTH from order_purchase_timestamp) as month,
Extract (Year from order_purchase_timestamp) as Year,
COUNT(order_id) AS order_count from neat-axis-409607.Market.customers c
left join neat-axis-409607.Market.orders o on c.customer_id=o.customer_id
group by month,year
order by month,year;

```
SELECT
Extract (MONTH from order_purchase_timestamp) as month,
Extract (Year from order_purchase_timestamp) as Year,
COUNT(order_id) AS order_count from neat-axis-409607.Market.customers c
left join neat-axis-409607.Market.orders o on c.customer_id=o.customer_id
group by month,year
order by month,year;
```

Output:

Row	month	Year	order_count
1	1	2017	800
2	1	2018	7269
3	2	2017	1780
4	2	2018	6728
5	3	2017	2682
6	3	2018	7211
7	4	2017	2404
8	4	2018	6939
9	5	2017	3700
10	5	2018	6873

Graph



Insights

There is increase in the orders with year-wise

Recommendations

We need to focus on the orders to increase the business from start up.

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

We need to extract from month and year from order_purchase_timestamp which is in orders table

Syntax

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
```

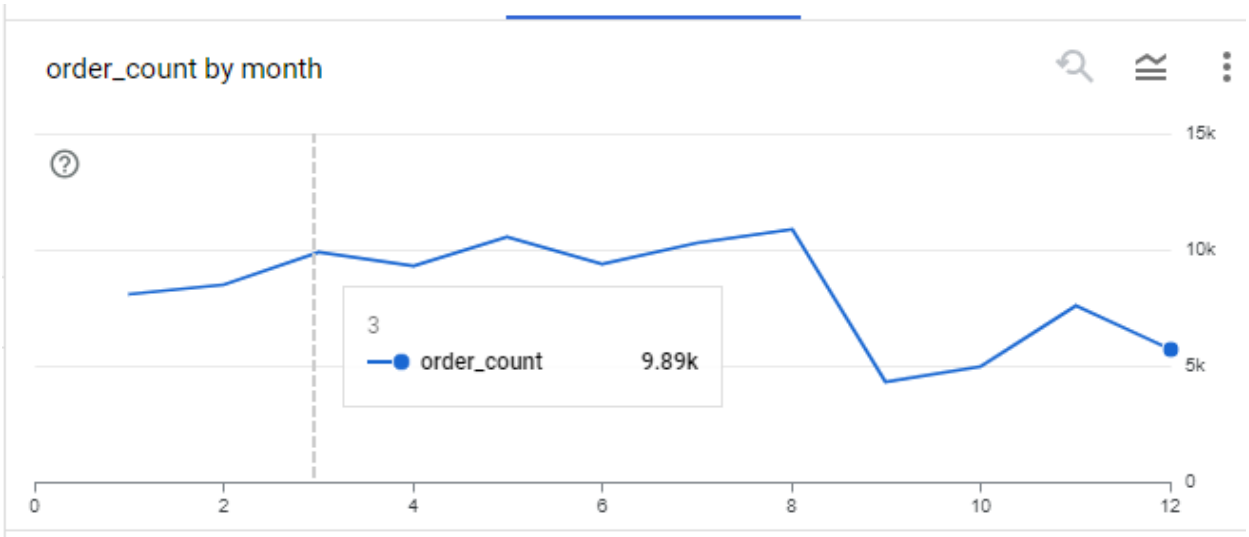
```
  COUNT(DISTINCT order_id) AS order_count
FROM
  neat-axis-409607.Market.orders
GROUP BY
  month
ORDER BY
  month;
```

```sql
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  COUNT(DISTINCT order_id) AS order_count
FROM
  neat-axis-409607.Market.orders
GROUP BY
  month
ORDER BY
  month;
```

Output

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

Graph



Insights

Seasonal variations in sales are observed, with increased sales during festive periods.

Recommendations

Businesses should plan their marketing and sales strategies in the peak seasons.

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**

We need to find the orders placed in during different times.

Syntax

```
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 'Morning'
      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_of_day,
   COUNT(*) AS order_count
FROM `neat-axis-409607.Market.orders`

GROUP BY
   time_of_day
ORDER BY
   time_of_day;
```

```sql
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 'Morning'
        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_of_day,
    COUNT(*) AS order_count
FROM `neat-axis-409607.Market.orders`

GROUP BY
    time_of_day
ORDER BY
    time_of_day;
```

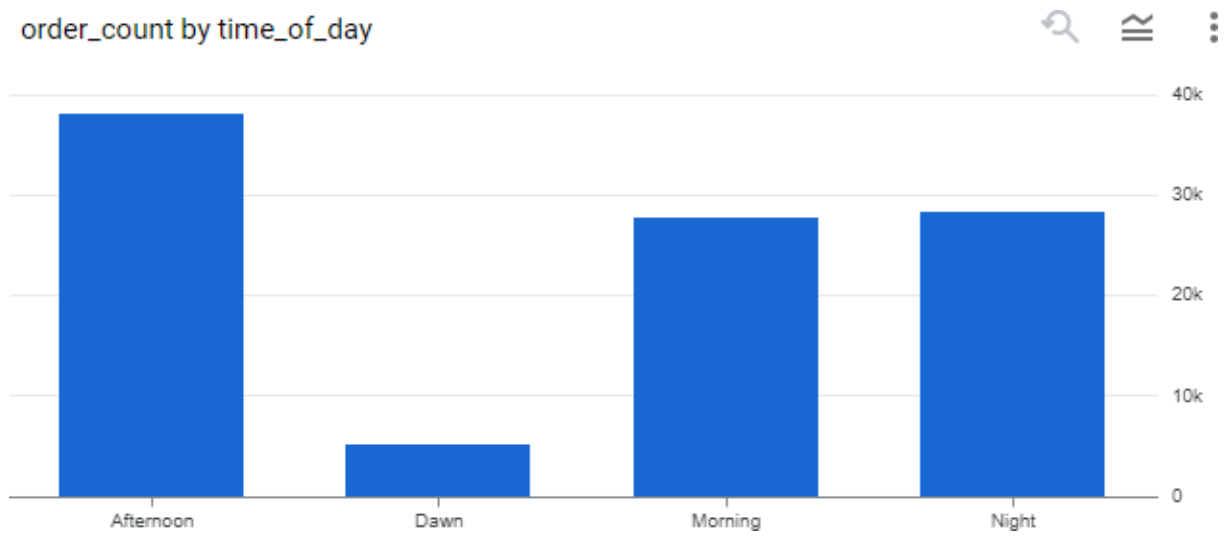
Output

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	time_of_day	order_count
1	Afternoon	38135
2	Dawn	5242
3	Morning	27733
4	Night	28331

Graph



Insights

Orders are more during afternoon and low in Dawn.

Recommendations

We need to increase the orders by marketing campaigns, loyalty programs, and exceptional customer service experiences.

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

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

We need get the customer state, order purchase timestamp from customer table and orders.

Syntax

```
select customer_state,EXTRACT(MONTH FROM order_purchase_timestamp) AS MONTH, count(1) as orders
from neat-axis-409607.Market.customers c inner join neat-axis-409607.Market.orders o
ON  c.customer_id = o.customer_id
group by customer_state,MONTH
order by orders desc;
```

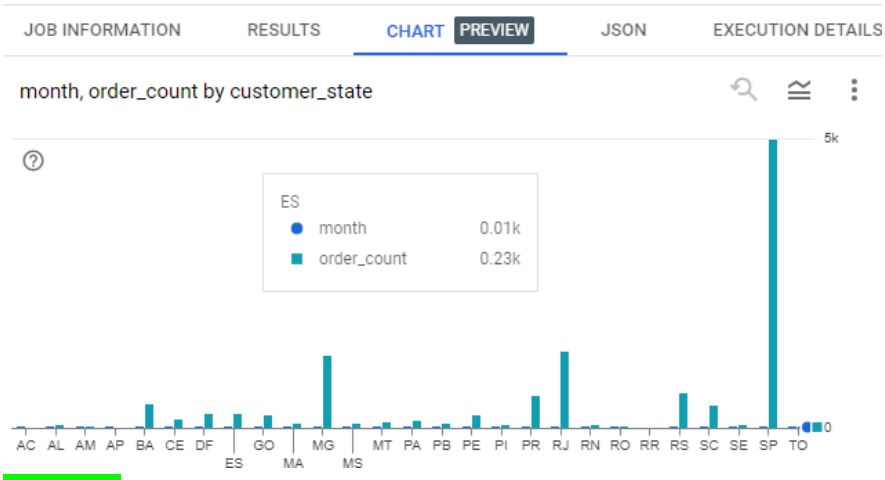
```sql
select customer_state,EXTRACT(MONTH FROM order_purchase_timestamp) AS MONTH, count(1) as orders
from neat-axis-409607.Market.customers c inner join neat-axis-409607.Market.orders o
ON  c.customer_id = o.customer_id
group by customer_state,MONTH
order by orders desc;
```

Output

No of records are 322

Row	customer_state	MONTH	orders
1	SP	8	4982
2	SP	5	4632
3	SP	7	4381
4	SP	6	4104
5	SP	3	4047
6	SP	4	3967
7	SP	2	3357
8	SP	1	3351
9	SP	11	3012
10	SP	12	2357

Graph



Insights

We have observed that SP state has been placed highest and AP has lowest orders

Recommendations

Company has give exciting offer to the customer to buy more products to rapid evolving in the market.

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

We need to count the states from customer table

Syntax

select customer_state,
count(*) as customer_count
from neat-axis-409607.Market.customers
group by customer_state
order by customer_state;

```
SELECT
  c.customer_state,
  COUNT(c.customer_id) AS no_of_customers
from neat-axis-409607.Market.customers c
  GROUP BY
  c.customer_state
ORDER BY
  no_of_customers DESC;
```

Output

No of states are 27

Row	customer_state	no_of_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

Graph

Insights

SP and RJ states has highest orders.

Recommendations

Need to implement customer retention strategies to encourage repeat purchases.

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.

We need calculate the % increase by using time stamp which is in order table, purchase from payments table.

Syntax

```
select month,
(((year_2018-year_2017)/year_2017)*100) as increase_percentage from
(SELECT
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
sum
(CASE
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp)=2017 and
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value end) as year_2017,
sum
(case
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp)=2018 and
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value end) as year_2018
from
`neat-axis-409607.Market.orders` o
JOIN
 neat-axis-409607.Market.payments p ON o.order_id = p.order_id

 WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
group by 1
```

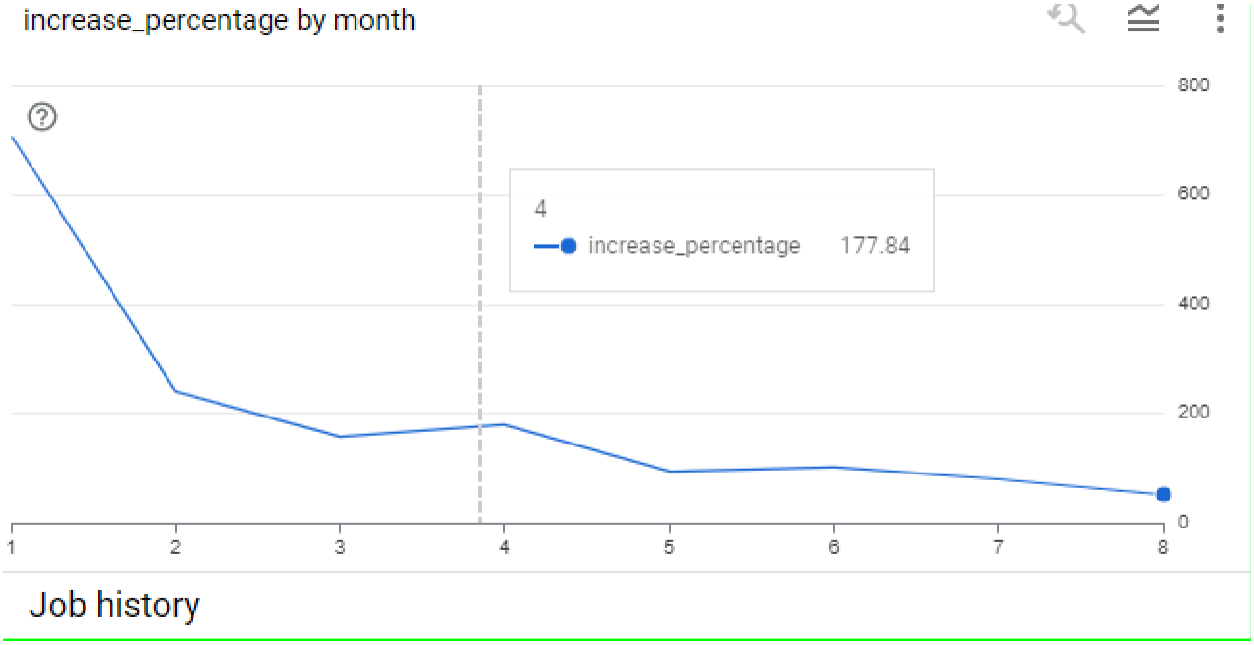
order by 1);

```
select month,
(((year_2018-year_2017)/year_2017)*100) as increase_percentage from
(SELECT
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
sum
(CASE
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp)=2017 and
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value end) as year_2017,
sum
(case
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp)=2018 and
EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value end) as year_2018
from
`neat-axis-409607.Market.orders` o
JOIN
 neat-axis-409607.Market.payments p ON o.order_id = p.order_id

 WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND
  EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
group by 1
order by 1);
```

Output

Row	month	increase_percentage
1	1	705.1266954171...
2	2	239.9918145445...
3	3	157.7786066709...
4	4	177.8407701149...
5	5	94.62734375677...
6	6	100.2596912456...
7	7	80.04245463390...
8	8	51.60600520477...

Graph:



Insights

Orders has been increased due to New Year eve.

Recommendations

Need to provide offers, discounts or promotions during off-peak seasons.

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

We need to fetch the columns customer state and order id from orders and order items tables

Syntax

```
SELECT c.customer_state,
round(sum(price),2) as sum,
round(avg(price),2) as Average
 FROM `neat-axis-409607.Market.order_items` o2
join neat-axis-409607.Market.orders o1 on
o1.order_id=o2.order_id
join neat-axis-409607.Market.customers c on
```

c.customer_id=o1.customer_id

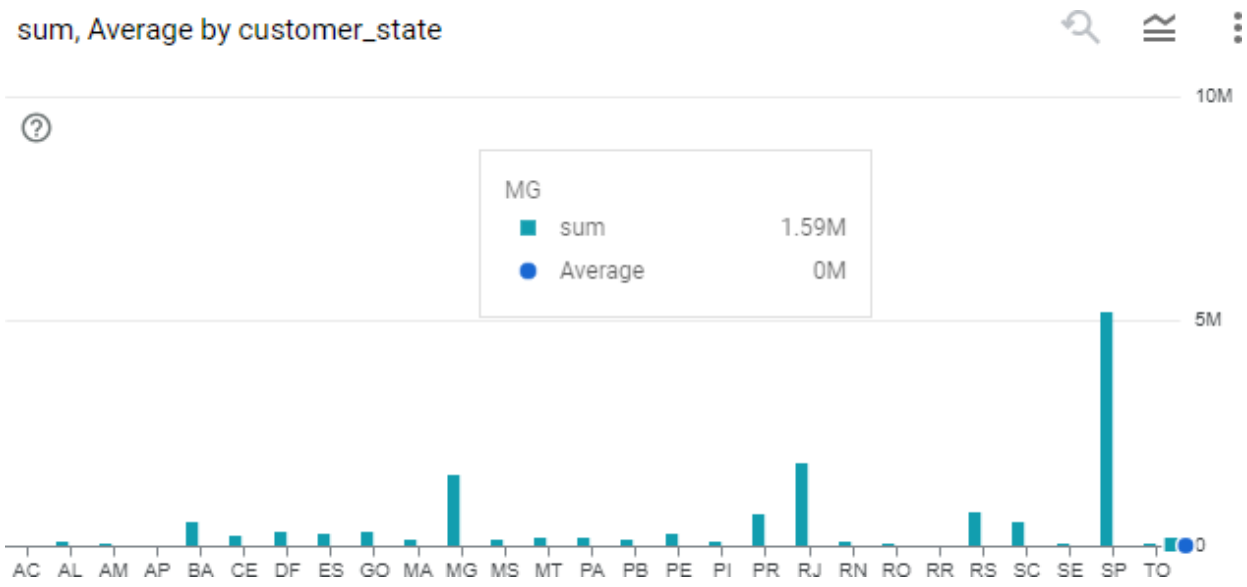
GROUP BY
c.customer_state
ORDER BY
c.customer_state;

```sql
SELECT c.customer_state,
COUNT(*) AS customer_count,
round(sum(price),2) as sum,
avg(price) as Average
 FROM `neat-axis-409607.Market.order_items` o2
join neat-axis-409607.Market.orders o1 on
 o1.order_id=o2.order_id
join neat-axis-409607.Market.customers c on
 c.customer_id=o1.customer_id

GROUP BY
c.customer_state
ORDER BY
c.customer_state;
```

Output

Row	customer_state	sum	Average
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

Graph



Insights

The data reveals that the state of SP has the highest number of customers.

Recommendation

Company has to focus on customer interest to get the orders placed in other states by providing the personalized offers.

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

We need to fetch the columns customer state , fright value and order id from orders and order items.

Syntax:

```
SELECT c.customer_state,
round(sum(freight_value),2) as freight_sum,
round(avg(freight_value),2) as freight_Average
FROM `neat-axis-409607.Market.order_items` o2
join neat-axis-409607.Market.orders o1 on
o1.order_id=o2.order_id
join neat-axis-409607.Market.customers c on
 c.customer_id=o1.customer_id
```

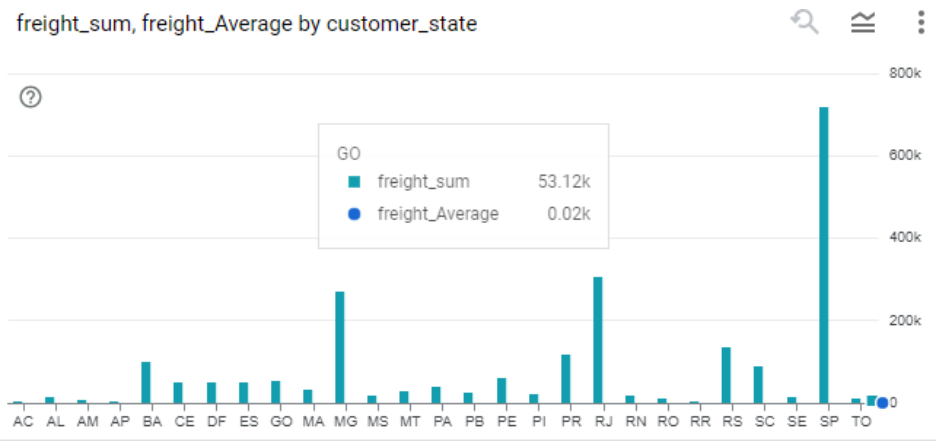
GROUP BY
c.customer_state
ORDER BY
c.customer_state;

```sql
SELECT c.customer_state,
round(sum(freight_value),2) as freight_sum,
round(avg(freight_value),2) as freight_Average
 FROM `neat-axis-409607.Market.order_items` o2
join neat-axis-409607.Market.orders o1 on
 o1.order_id=o2.order_id
join neat-axis-409607.Market.customers c on
 c.customer_id=o1.customer_id

GROUP BY
c.customer_state
ORDER BY
c.customer_state;
```

Output

Row	customer_state	freight_sum	freight_Average
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

Graph



Insights

SP has the highest total price value and total freight value.

Recommendations

Company has to focus on enhance the logistics to provide smooth services.

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

We need to fetch no of days by using orders table by using customer delivery date and timestamp

Syntax:

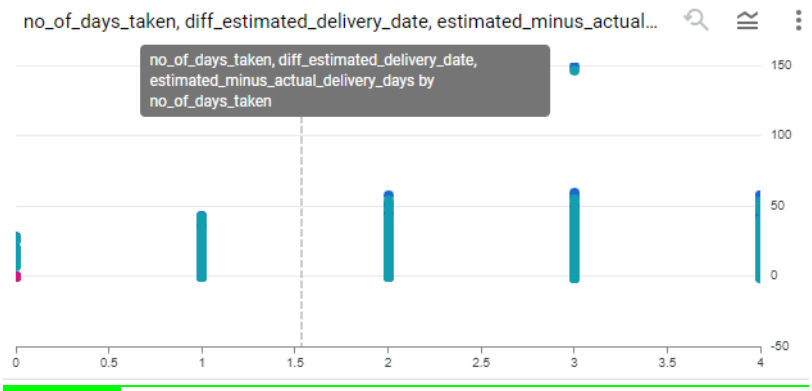
SELECT
order_id,
TIMEstamp_diff(order_delivered_customer_date,order_purchase_timestamp,day
) as no_of_days_taken,
TIMEstamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day)
as diff_estimated_delivery_date,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
DAY)
AS estimated_minus_actual_delivery_days
FROM neat-axis-409607.Market.orders
WHERE
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS
NOT NULL
order by no_of_days_taken;

```
SELECT
order_id,
TIMEstamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as no_of_days_taken,
TIMEstamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day)as diff_estimated_delivery_date,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
  AS estimated_minus_actual_delivery_days
FROM neat-axis-409607.Market.orders
WHERE
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) IS NOT NULL
order by no_of_days_taken;
```

Output

Row	order_id	no_of_days_taken	diff_estimated_delive	estimated_minus_ac
1	e65f1eeee1f52024ad1dcd034...	0	10	9
2	bb5a519e352b45b714192a02f...	0	26	25
3	434cecee7d1a65fc65358a632...	0	20	19
4	d3ca7b82c922817b06e5ca211...	0	12	11
5	1d893dd7ca5f77ebf5f59f0d20...	0	10	10
6	d5fbeedc85190ba88580d6f82...	0	8	7
7	79e324907160caea526fd8b94...	0	9	8
8	38c1e3d4ed6a13cd0cf612d4c...	0	17	16
9	8339b608be0d84fca9d8da68b...	0	28	27
10	f349cdb62f69c3fae5c4d7d3f3...	0	13	12

Graph



Insights

Few are taking less time or within 1 day the orders are delivered.

Recommendations

We need to focus on delivery partner such as Fedex, where the orders has been delivery through mode of transport.

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

We need to fetch the freight value on customers and orders tables.

Syntax

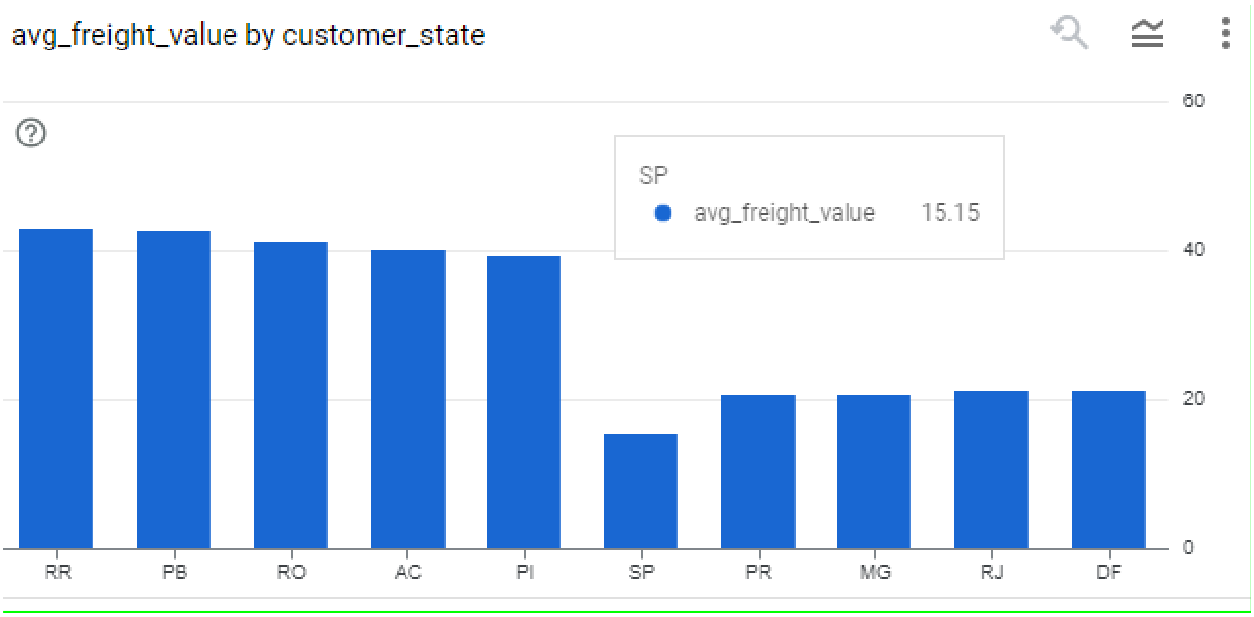
```
(
SELECT customer_state,
ROUND(Avg(freight_value),2) AS avg_freight_value
FROM neat-axis-409607.Market.customers c
join `neat-axis-409607.Market.orders` o1
using (customer_id)
join `neat-axis-409607.Market.order_items` o2
using (order_id)
GROUP BY customer_state
order by avg_freight_value DESC
LIMIT 5
)

UNION ALL

(
SELECT customer_state,
ROUND(Avg(freight_value),2) AS avg_freight_value
FROM neat-axis-409607.Market.customers c
join `neat-axis-409607.Market.orders` o1
using (customer_id)
join `neat-axis-409607.Market.order_items` o2
using (order_id)
GROUP BY customer_state
order by avg_freight_value
LIMIT 5
);
```

```sql
(SELECT customer_state,
ROUND(Avg(freight_value),2) AS avg_freight_value
FROM neat-axis-409607.Market.customers c
join `neat-axis-409607.Market.orders` o1
using (customer_id)
join `neat-axis-409607.Market.order_items` o2
using (order_id)
GROUP BY customer_state
order by avg_freight_value DESC
LIMIT 5
)
UNION ALL
(
SELECT customer_state,
ROUND(Avg(freight_value),2) AS avg_freight_value
FROM neat-axis-409607.Market.customers c
join `neat-axis-409607.Market.orders` o1  using (customer_id)
join `neat-axis-409607.Market.order_items` o2 using (order_id)
GROUP BY customer_state
order by avg_freight_value
LIMIT 5
);
```

Output

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

Graph



Insights

RR has highest freight value and SP state has lowest freight value.

Recommendations

Evaluate freight fees should be done frequently to focus on increase in orders.

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

We need to fetch the average delivery time on customers and orders tables.

Sytnax

```
(
SELECT customer_state,
round(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS avg_delivery_time

FROM  `neat-axis-409607.Market.orders`o
join neat-axis-409607.Market.customers c
using (customer_id)
GROUP BY customer_state
order by avg_delivery_time DESC
LIMIT 5
)
```

UNION ALL

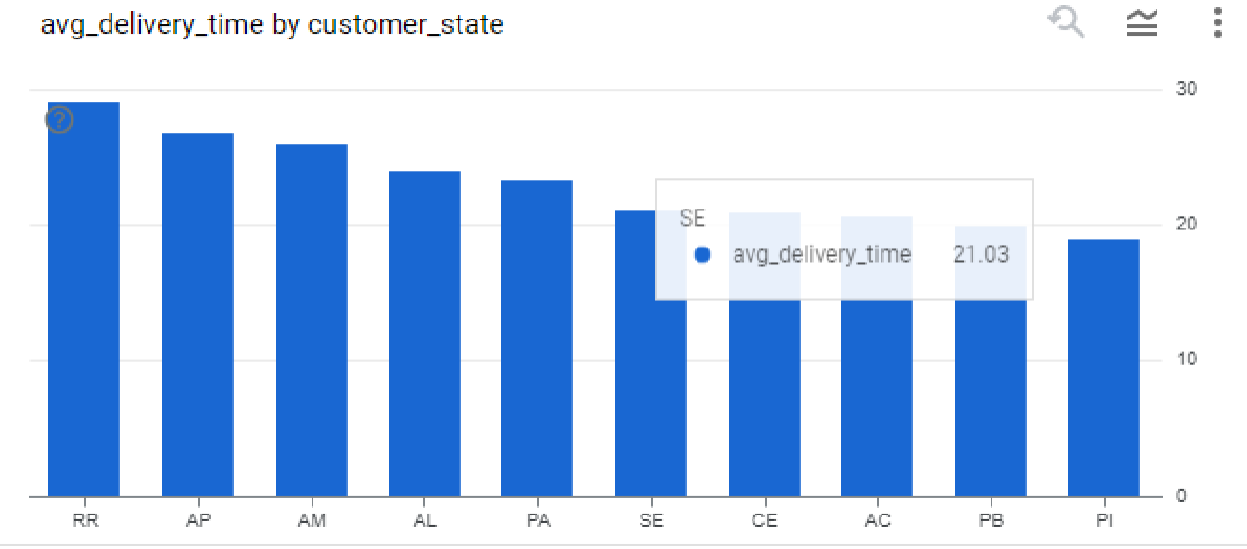
(
SELECT customer_state,
round(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS avg_delivery_time
FROM `neat-axis-409607.Market.orders`o
join neat-axis-409607.Market.customers c
using (customer_id)
GROUP BY customer_state
order by avg_delivery_time DESC
LIMit 5 offset 6);

```
(
SELECT customer_state,
round(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)),2) AS avg_delivery_time

FROM `neat-axis-409607.Market.orders`o
join neat-axis-409607.Market.customers c
using (customer_id)
GROUP BY customer_state
order by avg_delivery_time DESC
LIMIT 5
)

UNION ALL

(
SELECT customer_state,
round(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)),2) AS avg_delivery_time
FROM `neat-axis-409607.Market.orders`o
join neat-axis-409607.Market.customers c
using (customer_id)
GROUP BY customer_state
order by avg_delivery_time DESC
LIMit 5 offset 6);
```

Output

Row	customer_state	avg_delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32
6	SE	21.03
7	CE	20.82
8	AC	20.64
9	PB	19.95
10	PI	18.99

Graph



Insights

RR has highest delivery time and SE has lowest delivery time.

Recommendations

Need focus on delivery agencies to improve the turnaround time on delivery time.

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.

> We need to fetch the data by using customers and orders table to showcase the 5 states on order delivery very fast.

Syntax

```
select c.customer_state,
round(avg(date_diff(order_estimated_delivery_date,order_purchase_timestamp,
day)),2) as order_delivery
from neat-axis-409607.Market.orders o
```

join neat-axis-409607.Market.customers c on c.customer_id=o.customer_id
GROUP BY
c.customer_state
order by
customer_state,order_delivery asc
limit 5;

```
select c.customer_state,
round(avg(date_diff(order_estimated_delivery_date,order_purchase_timestamp,day)),2) as order_delivery
from neat-axis-409607.Market.orders o
join neat-axis-409607.Market.customers c on c.customer_id=o.customer_id
GROUP BY
c.customer_state
order by
order_delivery asc
limit 5;
```

Output

Row	customer_state	order_delivery
1	SP	18.81
2	DF	24.06
3	MG	24.22
4	PR	24.25
5	ES	25.27

customer_state

Graph

Insights

SP state has the fast delivery the orders.

Recommendations

We can consider the distance and geographic location may be orders got delayed, we need to focus on best delivery agencies such as DHL,Fedex etc to reduce the turnaround time on deliveries.

VI. Analysis based on the payments:

A. Find the month on month no. of orders placed using different payment types.
We need to use payments tables for timestamp and payment columns join with orders table.

Syntax

```
SELECT
extract(Month from order_purchase_timestamp) as month,
payment_type,
count(*) as no_of_orders
FROM `neat-axis-409607.Market.payments` p
left join neat-axis-409607.Market.orders o on p.order_id=o.order_id
```

group by month,payment_type
order by month,payment_type;

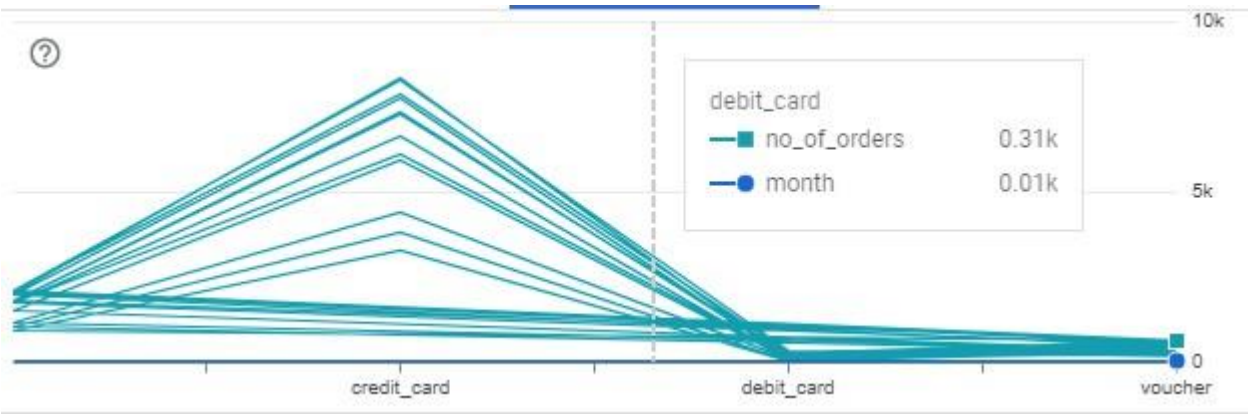
```
SELECT
extract(Month from order_purchase_timestamp) as month,
payment_type,
count(*) as no_of_orders
FROM `neat-axis-409607.Market.payments` p
left join neat-axis-409607.Market.orders o on p.order_id=o.order_id
group by month,payment_type
order by month,payment_type;
```

Output

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	month	payment_type	no_of_orders
1	1	UPI	1715
2	1	credit_card	6103
3	1	debit_card	118
4	1	voucher	477
5	2	UPI	1723
6	2	credit_card	6609
7	2	debit_card	82
8	2	voucher	424
9	3	UPI	1942
10	3	credit_card	7707

Graph



Insights

Credit card transactions are the most popular payment method.

Recommendations

Need to provide the offers based out payments with debit card or cash on delivery.

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

We need fetch the records from payments and orders table to show the payment installments along with month.

Syntax:

```
SELECT
extract(Month from order_purchase_timestamp) as month,
payment_installments,
count(*) as no_of_orders
FROM `neat-axis-409607.Market.payments` p
left join neat-axis-409607.Market.orders o on p.order_id=o.order_id
group by month,payment_installments
```

order by month,payment_installments,month;

```
SELECT
extract(Month from order_purchase_timestamp) as month,
payment_installments,
count(*) as no_of_orders
FROM `neat-axis-409607.Market.payments` p
left join neat-axis-409607.Market.orders o on p.order_id=o.order_id
group by month,payment_installments
order by month,payment_installments;
```

Output

Row	month	payment_installment	no_of_orders
1	1	1	4545
2	1	2	964
3	1	3	834
4	1	4	552
5	1	5	394
6	1	6	289
7	1	7	113
8	1	8	320
9	1	9	34
10	1	10	346

Graph

Insights

The highest number of installments is 24, which is associated with 18 orders.

Recommendation

Need to encourage the customers by providing the offers on payment gateway to improve customer satisfaction.

In conclusion

By taking the above all analysis into consideration, company has to focus on strategies on growth of the business in many aspects.

Evaluation Criteria (100 points)

1. Initial exploration like checking the structure & characteristics of the data (15 points)
2. In-depth Exploration (15 points)
3. Evolution of E-commerce orders in the Brazil region (10 points)
4. Impact on Economy (20 points)
5. Analysis on sales, freight and delivery time (20 points)
6. Analysis based on the payments (10 points)
7. Actionable Insights & Recommendations (10 points)

Submission Process

Once you're done with the case study...

- Use a Word document to paste your SQL queries along with a screenshot of the first 10 rows from the output.
- List down any valuable insights that you find during the analysis and provide some action items from the company's perspective in order to improve the current situation.
- Convert your solutions doc into a PDF, and upload the same on the platform.
- Please note that after submitting once, you will not be allowed to edit your submission.