Business Case: Target SQL

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver. This particular 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 order status, price, payment and freight performance, customer location, product attributes, and customer reviews. By analysing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

Problem Statement:

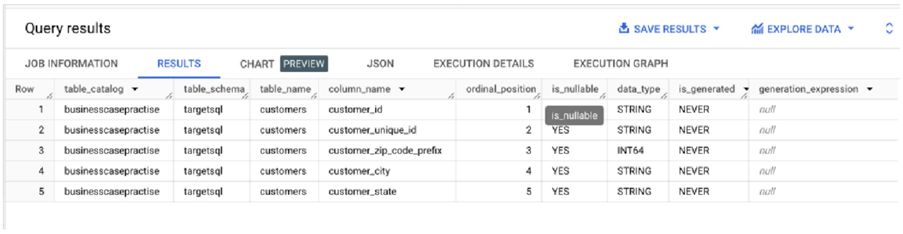
Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

(1)What does 'good' look like?

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

1.1 Data type of all columns in the "customers" table.

```
SELECT *
 FROM  businesscasepractise.targetsql.INFORMATION_SCHEMA.COLUMNS
 WHERE table_name = 'customers'
```

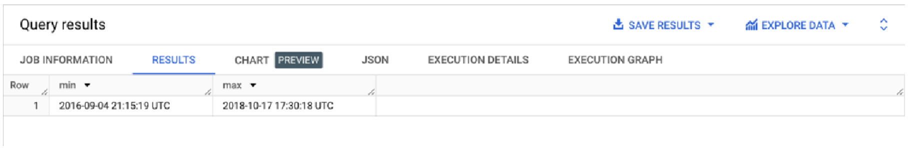


Row	table_catalog	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type	is_generated	generation_expression
1	businesscasepractise	targetsql	customers	customer_id	1		STRING	NEVER	null
2	businesscasepractise	targetsql	customers	customer_unique_id	2	YES	STRING	NEVER	null
3	businesscasepractise	targetsql	customers	customer_zip_code_prefix	3	YES	INT64	NEVER	null
4	businesscasepractise	targetsql	customers	customer_city	4	YES	STRING	NEVER	null
5	businesscasepractise	targetsql	customers	customer_state	5	YES	STRING	NEVER	null

INSIGHTS: N/A
RECOMMENDATIONS: N/A

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

```
select min(order_purchase_timestamp) as min ,max(order_purchase_timestamp) as max
from `targetsql.orders`
```



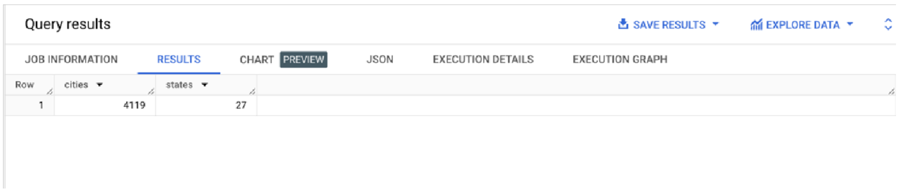
Row	min	max
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

INSIGHTS: The orders were places between 4th September 2016 and 17th October 2018.
It is the list of orders places 3 years

RECOMMENDATIONS: N/A

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

```
    Select count(distinct c.customer_city) cities ,count(distinct c.customer_state) states
from `targetsql.customers` c join `targetsql.orders` o  on c.customer_id = o.customer_id
```



Row	cities	states
1	4119	27

INSIGHTS: Customers made orders from 27 different states and 4119 different cities

RECOMMENDATIONS: N/A

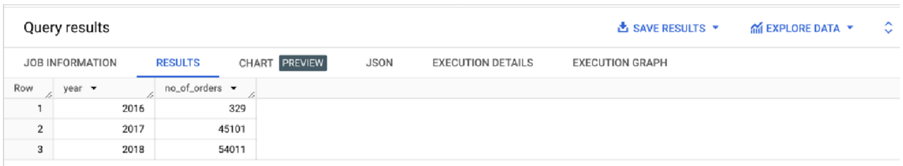
(2)In-depth Exploration:

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

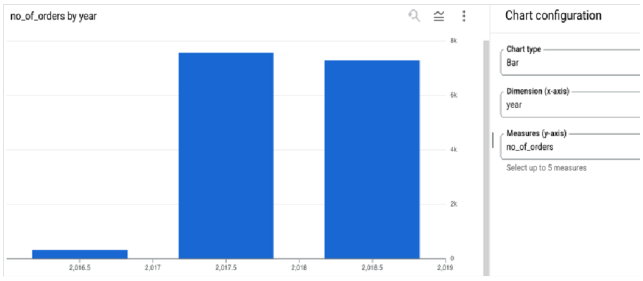
```
 select extract(year from order_purchase_timestamp) year,
         count(order_id) no_of_orders
from `targetsql.orders`

group by 1,2

order by 1,2
```



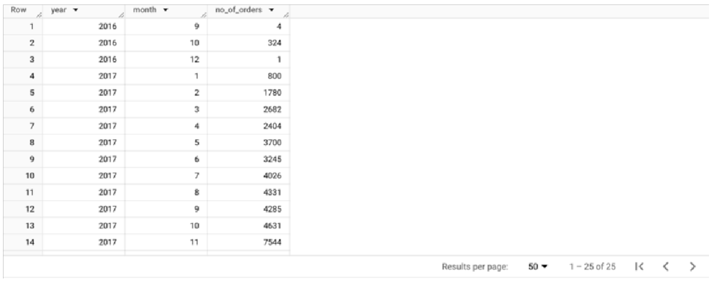
Row	year	no_of_orders
1	2016	329
2	2017	45101
3	2018	54011



```
 select extract(year from order_purchase_timestamp) year,
extract(month from order_purchase_timestamp) month,
         count(order_id) no_of_orders
from `targetsql.orders`

group by 1,2

order by 1,2
```



Row	year	month	no_of_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
14	2017	11	7544

INSIGHTS : (1) The no.of.orders increased yearly.

(2) 2016 has the list of only 3 months where as
2017 has list for whole year and
2018 has list of 10 months

(3). In 2017 the orders were increasing monthly

RECOMMENDATIONS: Consider entering new geographic areas, introducing new products, or targeting additional customer segments.

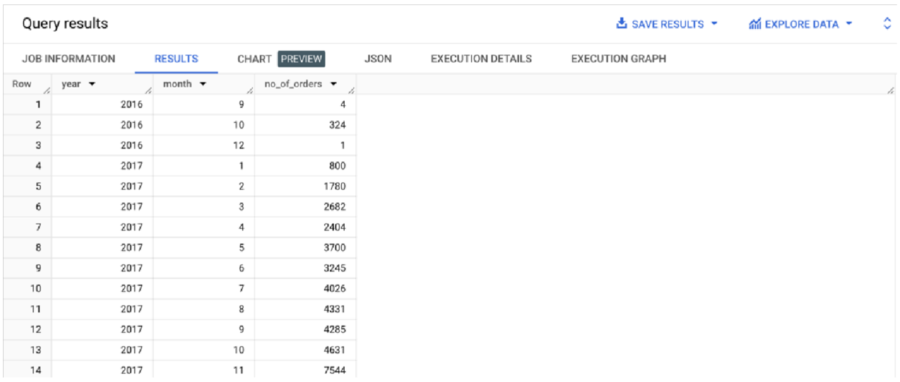
Explore collaborations or partnerships with suppliers, logistics providers, or other businesses in the industry to enhance efficiency and meet the growing demand effectively

Implement efficient tracking systems, reorder points, and supplier relationships to maintain adequate stock levels.

Implement loyalty programs, personalized communication, and excellent customer service to ensure that existing customers continue to choose your business.

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

```
    select extract(year from order_purchase_timestamp) year,
extract(month from order_purchase_timestamp) month,
         count(order_id) no_of_orders
from `targetsql.orders`
group by 1,2
 order by 1,2
```



Row	year	month	no_of_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
14	2017	11	7544

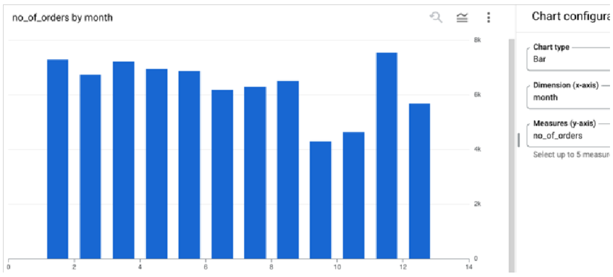


Chart type
Bar

Dimension (x-axis)
month

Measures (y-axis)
no_of_orders

Select up to 5 measur

INSIGHTS: Lots of people placed orders in November 2017, especially during the holiday season. This might happen every year.

There were fewer orders in September and October 2018. We need to figure out why this happened.

From May 2017 to November 2017, orders kept growing. After that, there were some ups and downs.

RECOMMENDATIONS: Since November is super popular for orders, make special offers and do more advertising during the holidays to get even more orders.

Find out why fewer people ordered in September and October 2018. Do special promotions or offers to get more orders during those months.

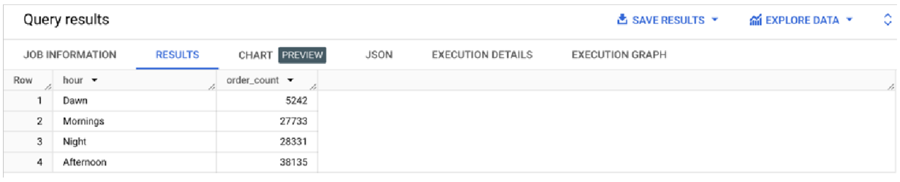
Make sure to have enough stuff in stock for November and December because that's when lots of people want to buy things.

Ask customers what they like and don't like. Use their feedback to make the ads and promotions better

2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

1. **0-6 hrs : Dawn**
2. **7-12 hrs : Mornings**
3. **13-18 hrs : Afternoon**
4. **19-23 hrs : Night**

```
SELECT
   CASE
     WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) <= 6 THEN 'Dawn'
     WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 7 AND 12 THEN 'Mornings'
     WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
     WHEN EXTRACT(HOUR FROM o.order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
END AS hour,
   COUNT(o.order_id) AS order_count
 FROM `targetsql.orders` o JOIN `targetsql.customers` c
     ON o.customer_id = c.customer_id
 GROUP BY hour
 ORDER BY order_count
```



Row	hour	order_count
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

INSIGHTS: Most people order stuff in the Afternoon—busy time.

People also order things during Mornings and Nights, with Nights being a bit busier.

Early in the morning (Dawn) is when fewer people order things.

RECOMMENDATIONS: Be super ready to handle lots of orders in the Afternoon.

Let people know about great deals in the Morning and Night to get more orders.

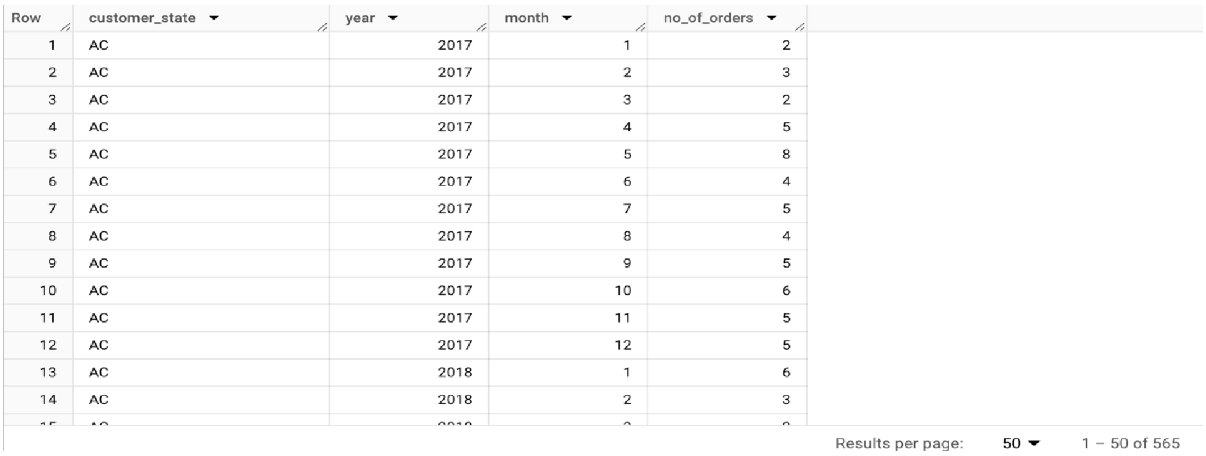
Talk to customers in a way that makes them want to order, especially during the busy times.

Make sure you have enough things in stock, especially during the busiest hours.

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

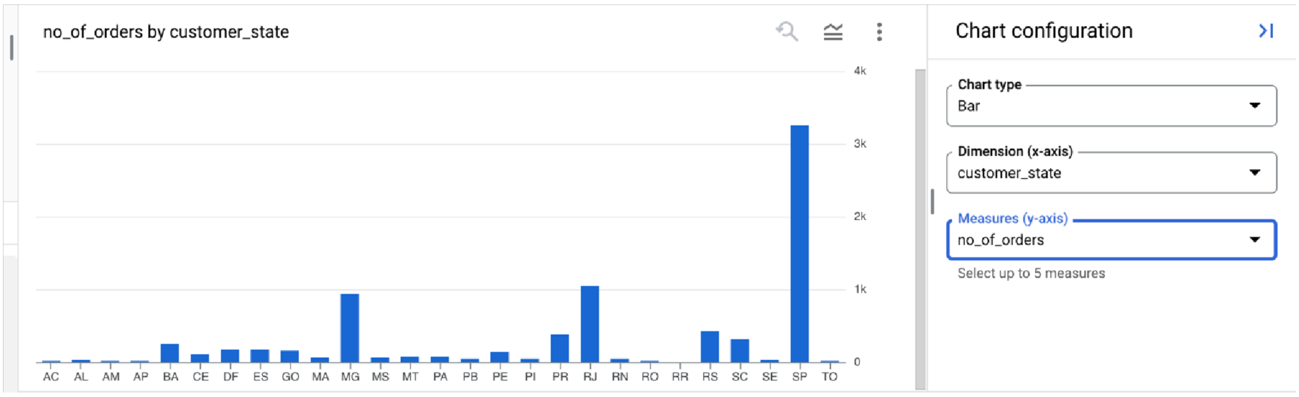
3.1 Get the month on month no. of orders placed in each state.

```
select
        distinct customer_state,        extract(year from order_purchase_timestamp ) as
year,       extract(month from order_purchase_timestamp ) as month,
count(order_id) as no_of_orders
from `targetsql.orders` o join `targetsql.customers` c on o.customer_id = c.customer_id
group by 1,2,3
order by 1,2,3
```



Row	customer_state	year	month	no_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5
12	AC	2017	12	5
13	AC	2018	1	6
14	AC	2018	2	3

Results per page: 50 1 – 50 of 565



no_of_orders by customer_state

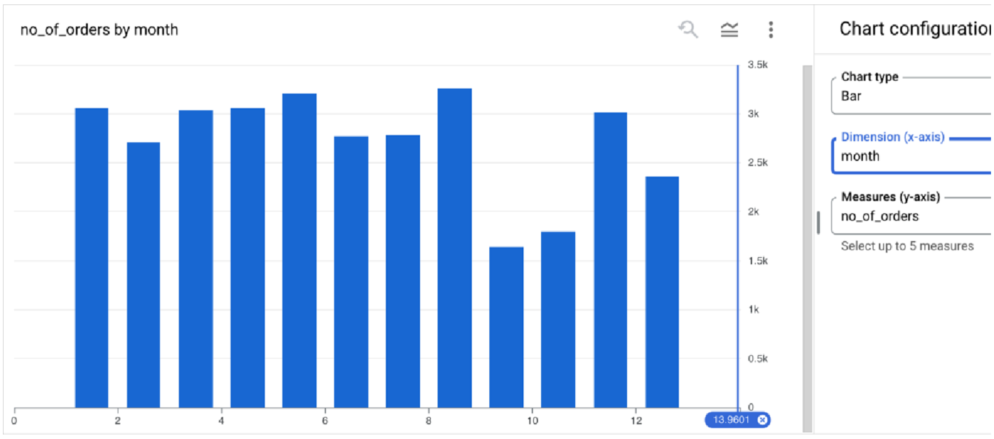
Chart configuration

Chart type: Bar

Dimension (x-axis): customer_state

Measures (y-axis): no_of_orders

Select up to 5 measures



no_of_orders by month

Chart configuratio

Chart type: Bar

Dimension (x-axis): month

Measures (y-axis): no_of_orders

Select up to 5 measures

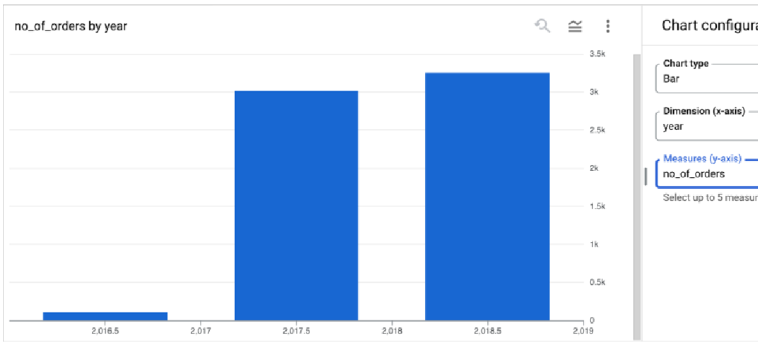


Chart type
Bar

Dimension (x-axis)
year

Measures (y-axis)
no_of_orders

Select up to 5 measur

INSIGHTS: More no.of orders were placed in the state SP, probably because it's a busy place with many people ordering online

Overall month wise it can be seen that during May and August more no.of orders were places

The no.of orders kept increasing yearly

RECOMMENDATIONS: Since SP is a hotspot for orders, make sure to pay extra attention there. Maybe offer some exclusive deals or partner with local businesses.

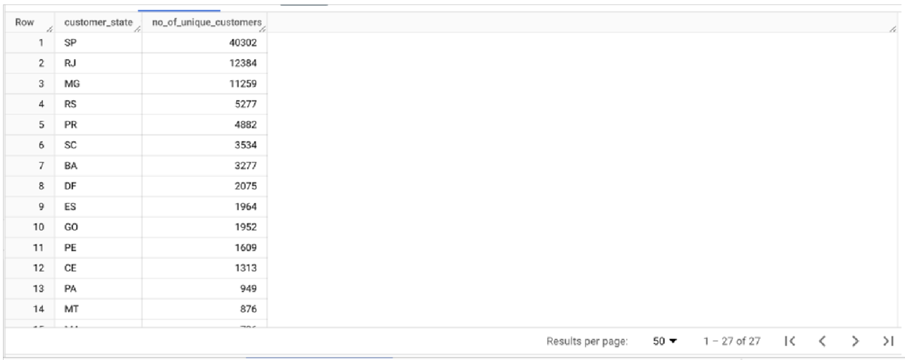
Plan exciting things for May and August, consider planning special deals or promotions during these months.

As the business is growing each year, make sure everything can handle the extra load – delivery, customer service, everything.

Offer rewards, special discounts, or just be really awesome at customer service for the customers to keep coming back

3.2 How are the customers distributed across all the states?

```
select customer_state,
      count(distinct customer_unique_id) no_of_unique_customers
from `targetsql.customers` group by 1 order by 2 desc
```



Row	customer_state	no_of_unique_customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952
11	PE	1609
12	CE	1313
13	PA	949
14	MT	876

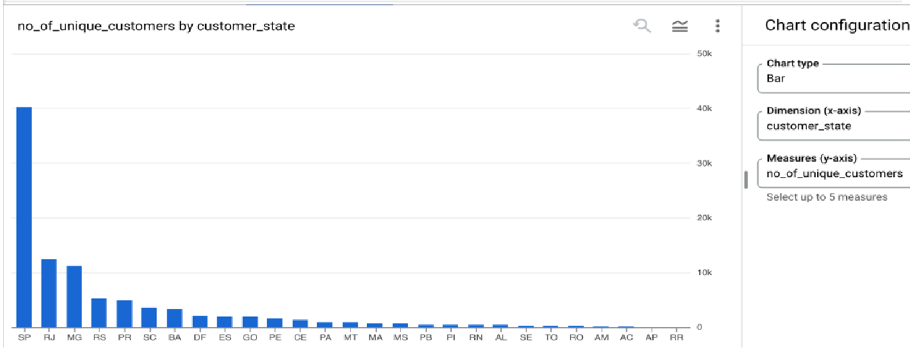


Chart type
Bar

Dimension (x-axis)
customer_state

Measures (y-axis)
no_of_unique_customers

Select up to 5 measures

INSIGHTS: The number of unique customers varies across states, with SP having the highest

RECOMMENDATIONS: Since SP has the most customers, consider focusing marketing efforts and promotions there to attract even more customers.

For states with fewer customers like AC ,AP or RR, explore strategies to increase brand awareness and attract more customers.

Do marketing campaigns based on the preferences and behaviours of customers in each state to maximize effectiveness.

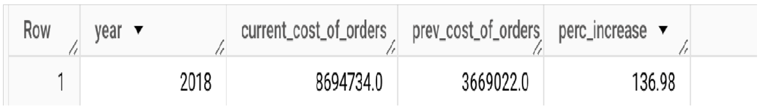
Keep an eye on states with potential growth, like PE or CE, and consider expanding marketing efforts to tap into emerging markets.

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

4.1 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.

```
WITH cte AS (
                          SELECT
                                    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
                                    ROUND(SUM(p.payment_value)) AS cost_of_orders
                          FROM `targetsql.payments` p JOIN `targetsql.orders` o ON p.order_id = o.order_id
                         WHERE EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
                         GROUP BY 1
                          ORDER BY 1
                    )

   SELECT
            cte.year,
            ROUND(cte.cost_of_orders) AS current_cost_of_orders,
            LAG(cte.cost_of_orders) OVER (ORDER BY cte.year) AS prev_cost_of_orders,
            ROUND(100 * (cte.cost_of_orders - LAG(cte.cost_of_orders) OVER (ORDER BY cte.year)) / LAG(cte.cost_of_orders) OVER
           (ORDER BY cte.year), 2) AS perc_increase
FROM cte      ORDER BY 1
    limit 1
    offset 1;
```



Row	year	current_cost_of_orders	prev_cost_of_orders	perc_increase
1	2018	8694734.0	3669022.0	136.98

INSIGHTS: The cost of orders in 2018 (from January to August) is 136.98% higher than the same period in 2017. This indicates a significant increase in expenses.

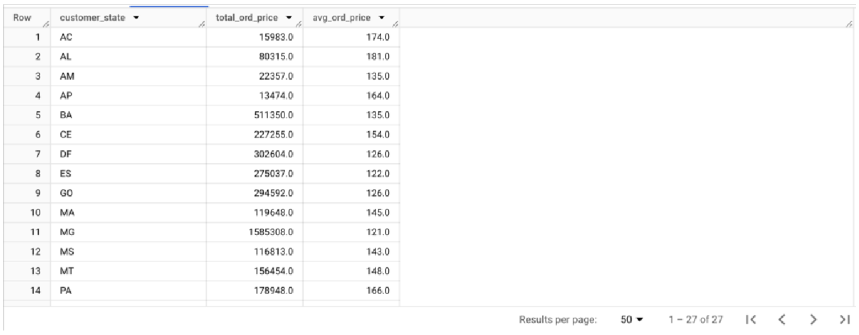
RECOMMENDATIONS: Investigate why the costs increased so much. Look at specific months or types of orders to understand what caused the big jump.

Take steps to control spending. This might involve negotiating better deals with suppliers or finding ways to do things more efficiently.

Regularly check how much money you're spending. If you see any big changes, figure out why and decide if there's something you can do about it.

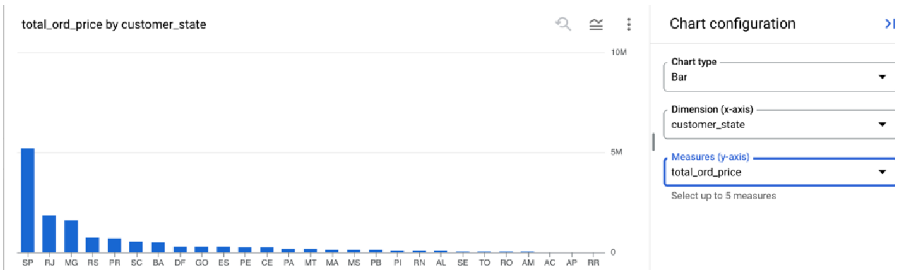
4.2 Calculate the Total & Average value of order price for each state

```
 select distinct customer_state, round(sum(price)) total_ord_price,
round(avg(price)) avg_ord_price
from `targetsql.orders`o join `targetsql.customers` c on o.customer_id= c.customer_id
join `targetsql.order_items` oi on o.order_id =oi.order_id   group by 1
order by 1
```



Row	customer_state	total_ord_price	avg_ord_price
1	AC	15983.0	174.0
2	AL	80315.0	181.0
3	AM	22357.0	135.0
4	AP	13474.0	164.0
5	BA	511350.0	135.0
6	CE	227255.0	154.0
7	DF	302604.0	126.0
8	ES	275037.0	122.0
9	GO	294592.0	126.0
10	MA	119648.0	145.0
11	MG	1585308.0	121.0
12	MS	116813.0	143.0
13	MT	156454.0	148.0
14	PA	178948.0	166.0

Results per page: 50 1 – 27 of 27



total_ord_price by customer_state

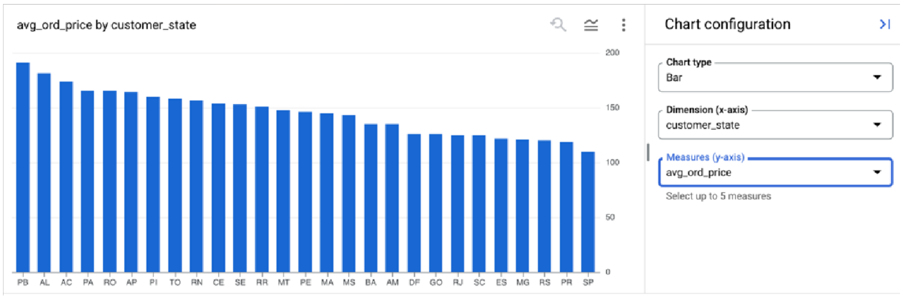
Chart configuration

Chart type: Bar

Dimension (x-axis): customer_state

Measures (y-axis): total_ord_price

Select up to 5 measures



avg_ord_price by customer_state

Chart configuration

Chart type: Bar

Dimension (x-axis): customer_state

Measures (y-axis): avg_ord_price

Select up to 5 measures

INSIGHTS: Every state spends money differently. SP spends a lot overall so highest total order price, but people there don't spend much in each order so lowest avg order price .

RECOMMENDATIONS: In SP make customers happy even if they don't spend a lot in one order. Maybe give them deals to encourage buying more things.

For states where people don't spend much per order, try to get them to buy more things at once. Maybe offer them discounts for bigger orders.

Change marketing a bit for each state based on what people there like. Show them products or deals that match their preferences.

Look for states where there's a chance to sell more. Do special things to get more people to buy there

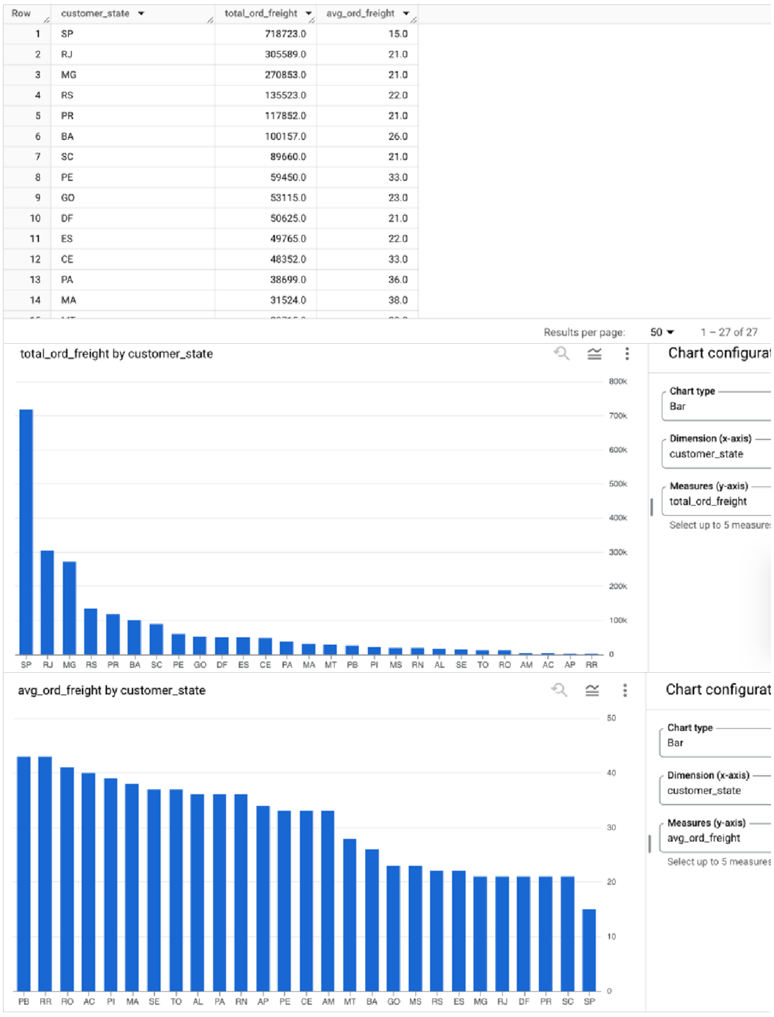
4.3 Calculate the Total & Average value of order freight for each state.

```
select distinct customer_state,  round(sum(freight_value)) total_ord_freight,
round(avg(freight_value)) avg_ord_freight

from `targetsql.orders`o join `targetsql.customers` c on o.customer_id= c.customer_id
join `targetsql.order_items` oi on o.order_id =oi.order_id

group by 1

order by 2 desc ,3 desc
```



Row	customer_state	total_ord_freight	avg_ord_freight
1	SP	718723.0	15.0
2	RJ	305589.0	21.0
3	MG	270853.0	21.0
4	RS	135523.0	22.0
5	PR	117852.0	21.0
6	BA	100157.0	26.0
7	SC	89660.0	21.0
8	PE	59450.0	33.0
9	GO	53115.0	23.0
10	DF	50625.0	21.0
11	ES	49765.0	22.0
12	CE	48352.0	33.0
13	PA	38699.0	36.0
14	MA	31524.0	38.0

Results per page: 50 1 – 27 of 27

total_ord_freight by customer_state

Chart configurat

Chart type: Bar

Dimension (x-axis): customer_state

Measures (y-axis): total_ord_freight

Select up to 5 measure

800k
700k
600k
500k
400k
300k
200k
100k
0
SP RJ MG RS PR BA SC PE GO DF ES CE PA MA MT PB PI MS RN AL SE TO RO AM AC AP RR

avg_ord_freight by customer_state

Chart configurat

Chart type: Bar

Dimension (x-axis): customer_state

Measures (y-axis): avg_ord_freight

Select up to 5 measures

50
40
30
20
10
0
PB RR RO AC PI MA SE TO AL PA RN AP PE CE AM MT BA GO MS RS ES MG RJ DF PR SC SP

INSIGHTS: SP has a lot of orders, but the cost of shipping for each order is not very high.

RECOMMENDATIONS: Since SP has many orders but lower shipping costs, figure out smart ways to ship things efficiently. Maybe find better deals with shipping companies.

Change how shipping works for each state based on how much they usually spend. Some states might be willing to pay more for shipping, so offer them good services.

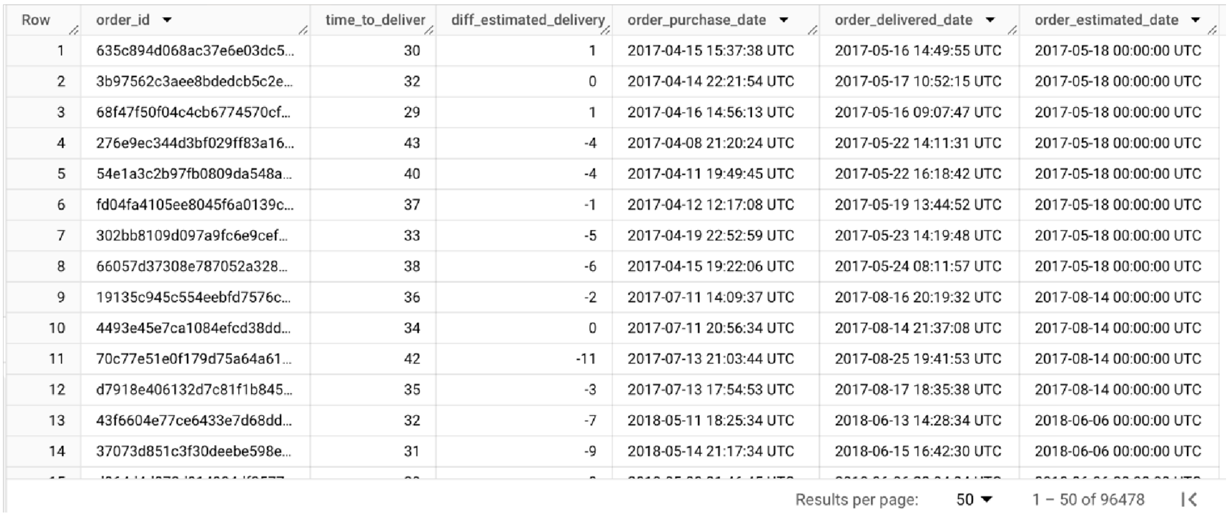
If the cost of shipping is causing issues, think about changing product prices or offering special deals with lower or free shipping.

(5)Analysis based on sales, freight and delivery time.

5.1 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. Do this in a single query.

```
select
order_id,
     date_diff(timestamp(`order_delivered_customer_date`),timestamp(`order_purchase_timestamp`),day) time_to_deliver,
date_diff(timestamp(`order_estimated_delivery_date`),timestamp(`order_delivered_customer_date`),day) diff_estimated_delivery,
order_purchase_timestamp as order_purchase_date,       order_delivered_customer_date as order_delivered_date,
order_estimated_delivery_date as order_estimated_date

from `targetsql.orders` where order_status ="delivered" and
order_status is not null
```



Row	order_id	time_to_deliver	diff_estimated_delivery	order_purchase_date	order_delivered_date	order_estimated_date
1	635c894d068ac37e6e03dc5...	30	1	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC
2	3b97562c3aee8bdedcb5c2e...	32	0	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC
3	68f47f50f04c4cb6774570cf...	29	1	2017-04-16 14:56:13 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC
4	276e9ec344d3bf029ff83a16...	43	-4	2017-04-08 21:20:24 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC
5	54e1a3c2b97fb0809da548a...	40	-4	2017-04-11 19:49:45 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC
6	fd04fa4105ee8045f6a0139c...	37	-1	2017-04-12 12:17:08 UTC	2017-05-19 13:44:52 UTC	2017-05-18 00:00:00 UTC
7	302bb8109d097a9fc6e9cef...	33	-5	2017-04-19 22:52:59 UTC	2017-05-23 14:19:48 UTC	2017-05-18 00:00:00 UTC
8	66057d37308e787052a328...	38	-6	2017-04-15 19:22:06 UTC	2017-05-24 08:11:57 UTC	2017-05-18 00:00:00 UTC
9	19135c945c554eebfd7576c...	36	-2	2017-07-11 14:09:37 UTC	2017-08-16 20:19:32 UTC	2017-08-14 00:00:00 UTC
10	4493e45e7ca1084efcd38dd...	34	0	2017-07-11 20:56:34 UTC	2017-08-14 21:37:08 UTC	2017-08-14 00:00:00 UTC
11	70c77e51e0f179d75a64a61...	42	-11	2017-07-13 21:03:44 UTC	2017-08-25 19:41:53 UTC	2017-08-14 00:00:00 UTC
12	d7918e406132d7c81f1b845...	35	-3	2017-07-13 17:54:53 UTC	2017-08-17 18:35:38 UTC	2017-08-14 00:00:00 UTC
13	43f6604e77ce6433e7d68dd...	32	-7	2018-05-11 18:25:34 UTC	2018-06-13 14:28:34 UTC	2018-06-06 00:00:00 UTC
14	37073d851c3f30deebe598e...	31	-9	2018-05-14 21:17:34 UTC	2018-06-15 16:42:30 UTC	2018-06-06 00:00:00 UTC

INSIGHTS: Time to deliver tells the number of days taken to deliver each order from the order's purchase date

Diff estimated delivery tells the difference in days between the estimated and actual delivery date of an order
And some orders with the negative value means that they were delivered earlier than estimated

RECOMMENDATIONS: Implement corrective actions to prevent delays

Work on optimizing delivery processes to ensure more accurate estimates. This can help in reducing variations in delivery times

5.2

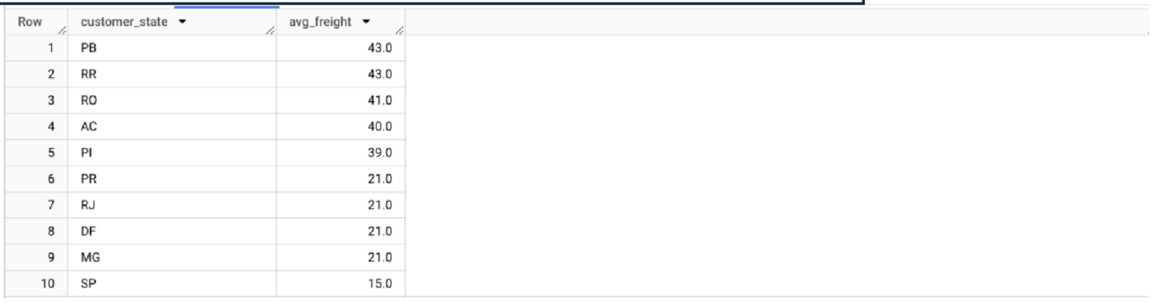
Find out the top 5 states with the highest & lowest average freight value.

```
with top as
        (select distinct customer_state,
              round(avg(freight_value)) avg_freight,
         from `targetsql.orders` o join `targetsql.customers` c on o.customer_id= c.customer_id
join `targetsql.order_items` oi on o.order_id =oi.order_id
group by 1
order by 2 desc
 limit 5)

    ,bottom as
        (select distinct customer_state,
              round(avg(freight_value)) avg_freight,
              from `targetsql.orders` o join `targetsql.customers` c on
              o.customer_id=c.customer_id   join `targetsql.order_items` oi on o.order_id
              =oi.order_id

         group by 1
         order by 2
         limit 5)

select customer_state,avg_freight
from top  union all
select customer_state,avg_freight from
bottom
order by avg_freight desc ;
```



Row	customer_state	avg_freight
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0
6	PR	21.0
7	RJ	21.0
8	DF	21.0
9	MG	21.0
10	SP	15.0

INSIGHTS: The top five have highest shipping value and bottom five have lowest shipping value

RECOMMENDATION : For highest average freight value communicate value and reliability in shipping

For lowest average freight value consider offering promotions to boost order volume

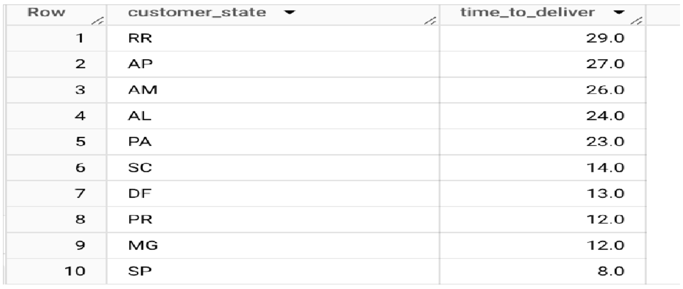
5.3

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

```
    with top as
(select
            customer_state,
round(avg(timestamp_diff(`order_estimated_delivery_date`,`order_delivered_customer_date`,day)))time_to_deliver
from `targetsql.orders` o join `targetsql.customers` c on o.customer_id= c.customer_id          where order_status ="delivered"
and order_status is not null
         GROUP BY customer_state
        order by 2 desc
        limit 5)

 ,bottom as
        (select
           distinct customer_state,
        round(avg(timestamp_diff(`order_estimated_delivery_date`,`order_delivered_customer_date`,day))) time_to_deliver
from `targetsql.orders` o join `targetsql.customers` c on o.customer_id= c.customer_id          where order_status
="delivered" and order_status is not null
        GROUP BY customer_state
order by 2          limit 5)

select customer_state,time_to_deliver
from top   union all
 select customer_state,time_to_deliver
from bottom order
by 2 desc
```



Row	customer_state	time_to_deliver
1	RR	29.0
2	AP	27.0
3	AM	26.0
4	AL	24.0
5	PA	23.0
6	SC	14.0
7	DF	13.0
8	PR	12.0
9	MG	12.0
10	SP	8.0

INSIGHTS: The top 5 states experience longer average delivery times.

The bottom 5 states experience shorter average delivery times.

RECOMMENDATIONS: Focus on improving logistics and delivery processes in states with longer delivery times. Explore ways to enhance efficiency and reduce delays.

Maintain the efficiency in delivery processes for states with shorter delivery times. Continue providing quick and reliable delivery services.

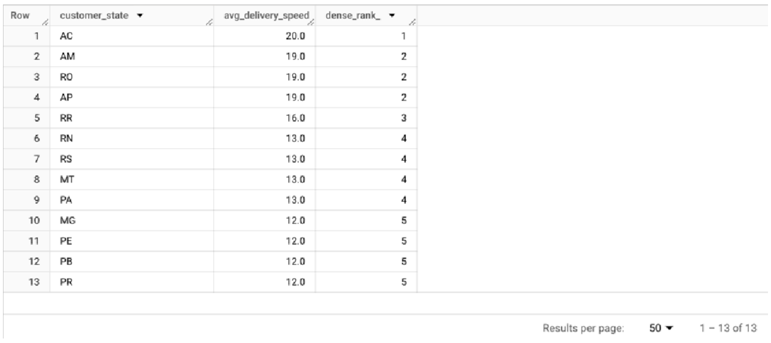
Consider partnerships with reliable carriers to enhance delivery speed.

5.4

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.

```
   with cte as
                  (SELECT
                       customer_state,
                       ROUND(AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date, DAY))) AS
avg_delivery_speed,            dense_rank() over (order
by(ROUND(AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,                    DAY)))) desc) as
dense_rank_
                  FROM `targetsql.orders` o JOIN `targetsql.customers` c ON o.customer_id = c.customer_id
                  WHERE order_status = "delivered" AND order_status IS NOT NULL
GROUP BY customer_state
                  ORDER BY  avg_delivery_speed desc)

 select *  from cte
where dense_rank_ <=5
```



Row	customer_state	avg_delivery_speed	dense_rank_
1	AC	20.0	1
2	AM	19.0	2
3	RO	19.0	2
4	AP	19.0	2
5	RR	16.0	3
6	RN	13.0	4
7	RS	13.0	4
8	MT	13.0	4
9	PA	13.0	4
10	MG	12.0	5
11	PE	12.0	5
12	PB	12.0	5
13	PR	12.0	5

INSIGHTS: The top 5 states would be AC,AM,RO,AP,RR in case it was given rank but in case of dense rank the top 5 are 13 states

The average delivery speed ranges from 12 to 20 days ahead of the estimated date in these states.

RECOMMENDATIONS: Recognize and acknowledge the efficiency in these states with faster delivery speeds. Consider celebrating this achievement to motivate the delivery teams.

Highlighting fast deliveries as it can positively impact the brand image.

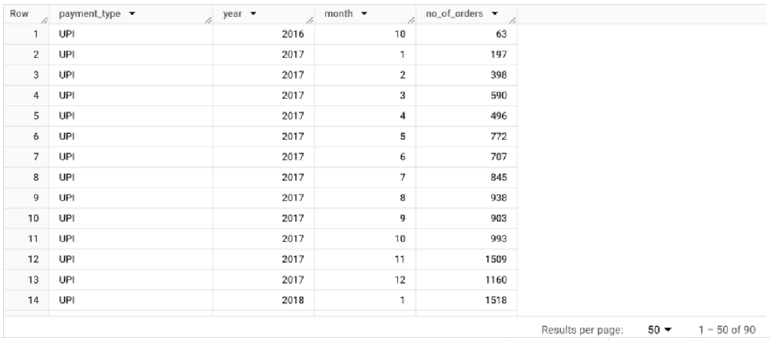
Understand the practices that contribute to fast deliveries in these states. Apply these lessons to improve delivery speed in other areas.

(6) Analysis based on the payments:

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

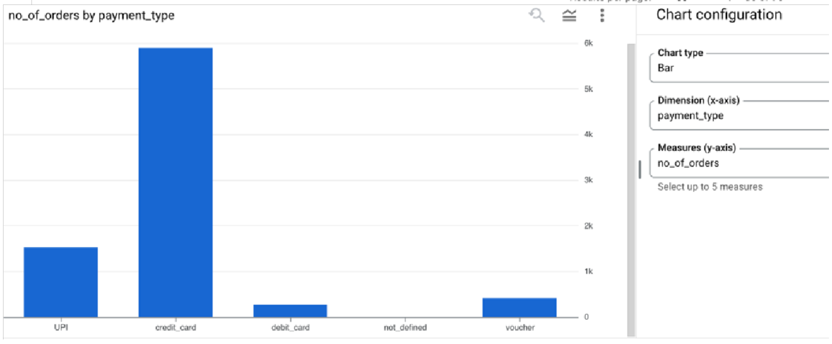
```
       select
payment_type,
        extract(year from o.order_purchase_timestamp) as year,
extract(month from o.order_purchase_timestamp) as month,
        count(*) no_of_orders   from `targetsql.payments` p join `targetsql.orders`
o on p.order_id = o.order_id   group by 1,2,3


 order by 1,2,3
```



Row	payment_type	year	month	no_of_orders
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903
11	UPI	2017	10	993
12	UPI	2017	11	1509
13	UPI	2017	12	1160
14	UPI	2018	1	1518

Results per page: 50 ▾ 1 – 50 of 90



no_of_orders by payment_type

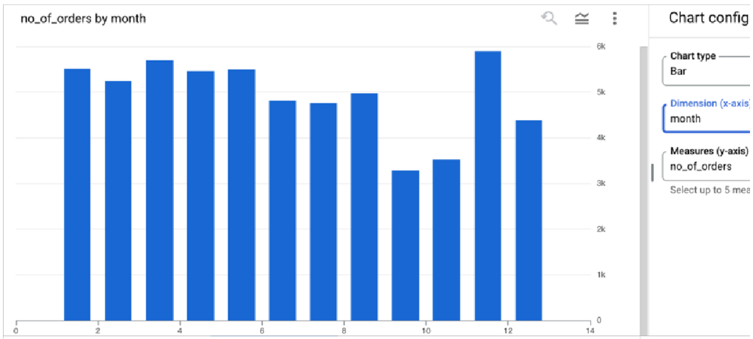
Chart configuration

Chart type: Bar

Dimension (x-axis): payment_type

Measures (y-axis): no_of_orders

Select up to 5 measures



no_of_orders by month

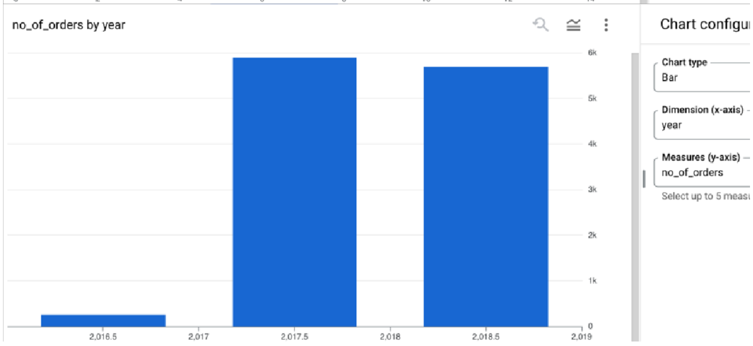
Chart config

Chart type: Bar

Dimension (x-axis): month

Measures (y-axis): no_of_orders

Select up to 5 mea



no_of_orders by year

Chart configu

Chart type: Bar

Dimension (x-axis): year

Measures (y-axis): no_of_orders

Select up to 5 measu

INSIGHTS: Most of the orders were placed using credit_card

It has peak no.of orders in the month of march and November might be because of the holiday time

The orders placed were more in the year 2017

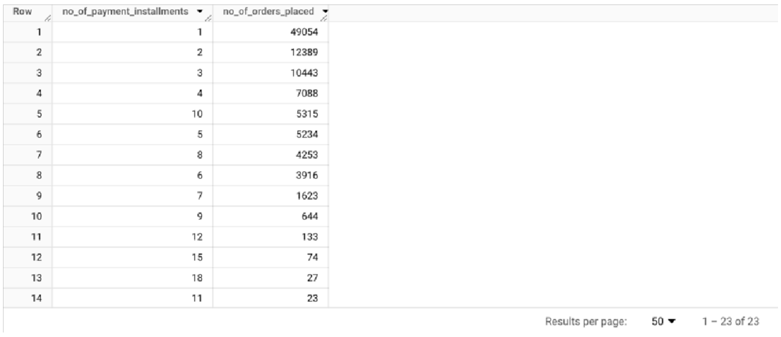
RECOMMENDATIONS: Make sure paying with a credit card is super easy because that's what people like.

During March and November, plan something special – discounts, offers, or cool stuff to attract more orders.

Let's keep growing and maybe beat the number of orders we had in 2017 by
(1) Advertising more
(2) Create cool discounts or special deals
(3) Bring in new and exciting products
(4) Connect with customers through social media or special events

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

```
    select payment_installments as no_of_payment_installments,
      count(distinct o.order_id) no_of_orders_placed   from
`targetsql.payments` p join `targetsql.orders` o on p.order_id = o.order_id
where payment_value  >=1 and payment_installments >=1    group by 1
 order by 1,2
```



Row	no_of_payment_installments	no_of_orders_placed
1	1	49054
2	2	12389
3	3	10443
4	4	7088
5	10	5315
6	5	5234
7	8	4253
8	6	3916
9	7	1623
10	9	644
11	12	133
12	15	74
13	18	27
14	11	23

INSIGHTS: Customers frequently choose 1, 2, and 3 installment plans.

The majority of orders fall into lower installment plans, indicating a preference for shorter payment periods.

RECOMMENDATIONS: Focus on streamlining and promoting 1, 2, and 3 installment plans, as they are the most popular choices among customers.

Encourage customers to explore longer-term installment plans by offering incentives or benefits. Highlight the advantages of these plans to attract more users.

Implement no cost EMI for long term installment plans