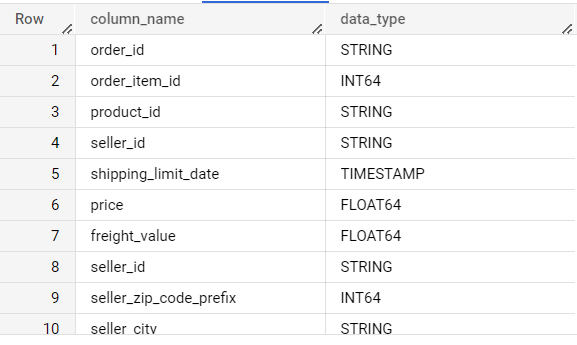
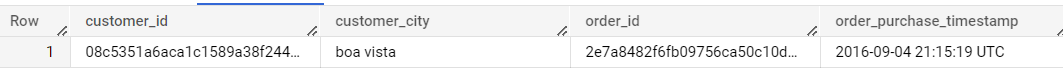
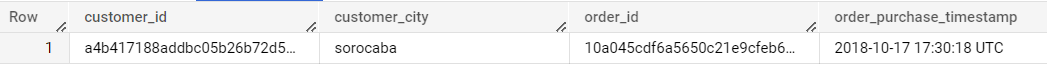
Target SQL

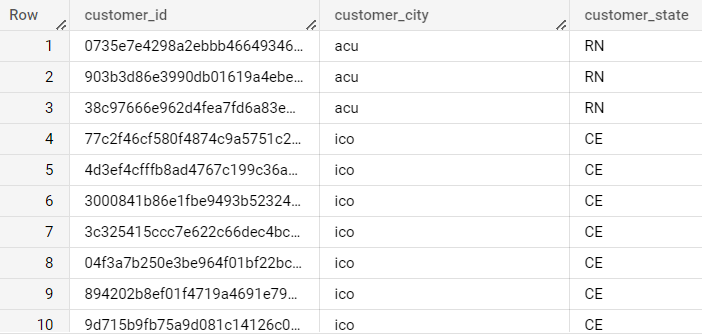
Q1)  
1.1) Data type of columns in a table:  
  
select column\_name, data\_type  
from `target-sql-377207.Target.INFORMATION\_SCHEMA.COLUMNS`  
where table\_name in ('Customers','geolocation','order\_items','order\_reviews','orders','payments','products','sellers')  
  
  


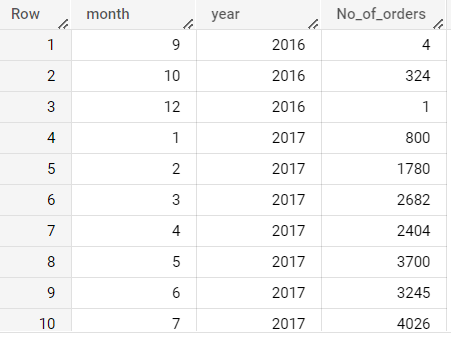
1.2) Time period for which the data is given:  
  
SELECT c.customer\_id,c.customer\_city,o.order\_id,o.order\_purchase\_timestamp  
FROM `Target.orders` as o join `Target.Customers` as c on o.customer\_id=c.customer\_id  
where o.order\_purchase\_timestamp in(select min(order\_purchase\_timestamp) from `Target.orders`)

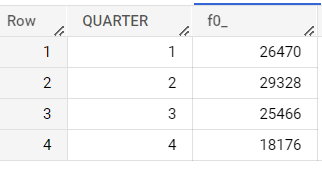
SELECT c.customer\_id,c.customer\_city,o.order\_id,o.order\_purchase\_timestamp

FROM `Target.orders` as o join `Target.Customers` as c on o.customer\_id=c.customer\_id

where o.order\_purchase\_timestamp in(select max(order\_purchase\_timestamp) from `Target.orders`)  
  
  
  
  
  
  
As per the above results,we can see the first and the last order timestamp along with the cutomer id,order id,customer city.  
The data provided to us is from 4th Septemper 2016 to 17th October 2018.

1.3)Cities and States of customers ordered during the given period :-  
  
SELECT c.customer\_id,c.customer\_city,c.customer\_state from `Target.Customers` as c join `Target.orders` as o on o.customer\_id=c.customer\_id  
where o.order\_purchase\_timestamp between '2016-09-04' and '2018-10-17'  
  
  
  
This is the details of all the customers with their customer id,city,states who ordered between 4th Septemper 2016 to 17th October 2018.

Q2)  
  
2.1) Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?  
  
select extract(month from order\_purchase\_timestamp) as month,extract(year from order\_purchase\_timestamp) as year,count(distinct order\_id)as No\_of\_orders from `Target.orders`  
where order\_purchase\_timestamp between '2016-09-04' and '2018-10-17'  
group by 1,2 order by 2,1  


As we can see from the result there is an increasing trend in the number of orders month by month.  
In November 2017 we have recorded the highest number of orders.Also January and March in the year 2018 have number of orders which are close to the highest number of orders.   
  
select distinct QUARTER,sum(No\_of\_orders) over (partition by Quarter order by QUARTER)from(select  FORMAT\_DATE('%b-%Y',order\_purchase\_timestamp ) AS mon\_year,count(distinct order\_id) as No\_of\_orders,EXTRACT(QUARTER FROM order\_purchase\_timestamp) as Quarter from `Target.orders`  
where order\_purchase\_timestamp between '2016-09-04' and '2018-10-17'  
group by 1,3  
order by 3,1)a  
  


Also we can see the quarter wise sales for the overall time period.  
We can see the highest numbers of orders in the second quarter of every year accumulatively.  
  
  
  
2.2) What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

select extract(time from order\_purchase\_timestamp) as timer,

case when extract(time from order\_purchase\_timestamp) between '00:00:00' and '05:59:59'

then 'Night (12 AM to 6 AM)'

when extract(time from order\_purchase\_timestamp) between '06:00:00' and '11:59:59'

then 'Morning (6 AM to 12 PM)'

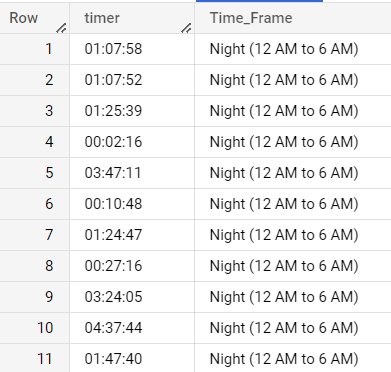
when extract(time from order\_purchase\_timestamp) between '12:00:00' and '17:59:59'

then 'Afternoon (12 PM to 6 PM)'

else 'Evening (6 PM to 12 AM)'

end as Time\_Frame

FROM `target-sql-377207.Target.orders`

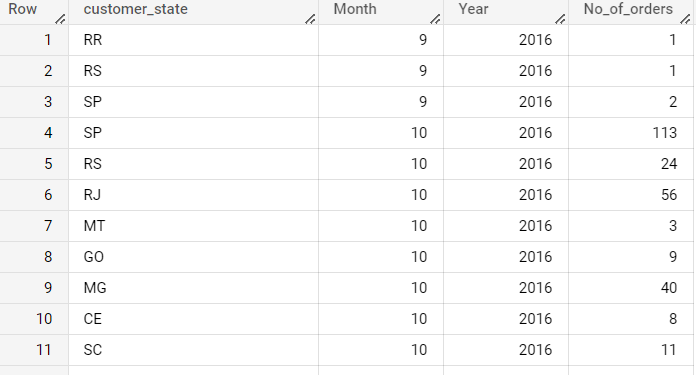


As we can clearly see from the above result,maximum orders are placed in night time i.e. from 12AM to 6PM.  
  
  
  
3.1) Get month on month orders by states

select customer\_state,extract(month from order\_purchase\_timestamp) as Month,extract (year from order\_purchase\_timestamp) as Year,count(distinct order\_id) as No\_of\_orders from `Target.Customers` c join `Target.orders`o on o.customer\_id=c.customer\_id where order\_purchase\_timestamp between '2016-09-04' and '2018-10-17'

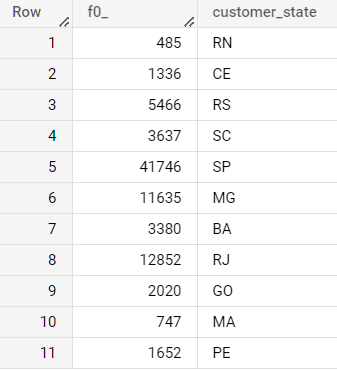
group by 1,2,3

order by 3,2



From the above result,the highest number of orders were from the State of Sau Paulo.  
  
  
3.2) Distribution of customers across the states in Brazil  
  
select count(distinct customer\_id),customer\_state from `Target.Customers`

group by 2

  
  
Target has major customers coming in from 3 states with Sau Paulo being the highest followed by Rio de Janeiro and Minas Gerais.

4.1) Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment\_value” column in payments table

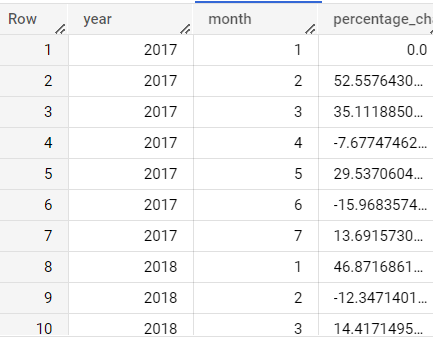
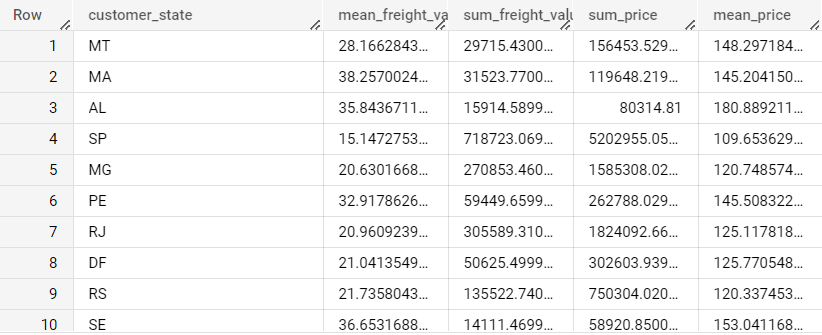
select year,month,ifnull((Total\_Value-lag(Total\_Value) over (order by year,month)),0)/Total\_Value \* 100 as percentage\_change

from(select sum(payment\_value)as Total\_Value,extract(month from order\_purchase\_timestamp) as Month,extract(year from order\_purchase\_timestamp) as year from `Target.payments` p join `Target.orders` o on o.order\_id=p.order\_id

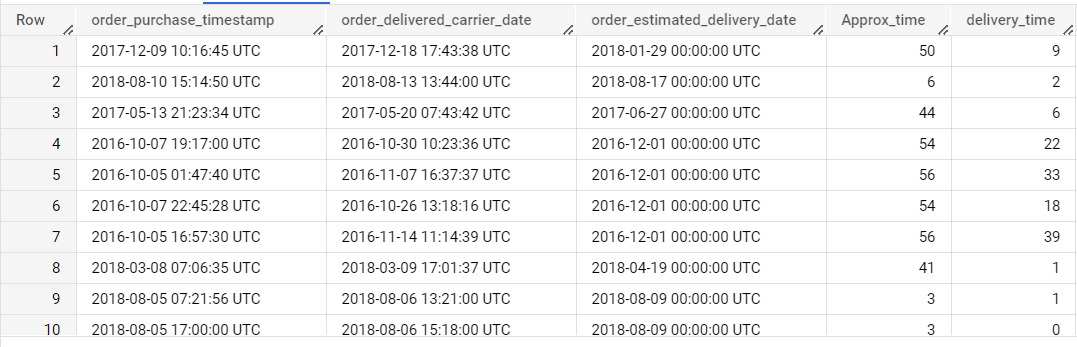
group by year,month order by 2,1)a

where year in (2017,2018) and month in(1,2,3,4,5,6,7)

order by year,month

  
  
  
From the above result we can observe that there is a increase in payment collected by Target from its customers in majority of the months.  
  
  
4.2) Mean & Sum of price and freight value by customer state :-  
  
select customer\_state,avg(freight\_value) as mean\_freight\_value,sum(freight\_value)as sum\_freight\_value,sum(price)as sum\_price,avg(price)as mean\_price from `Target.order\_items`oi join `Target.orders`o on o.order\_id=oi.order\_id join `Target.Customers`c on c.customer\_id=o.customer\_id  
group by 1  
  


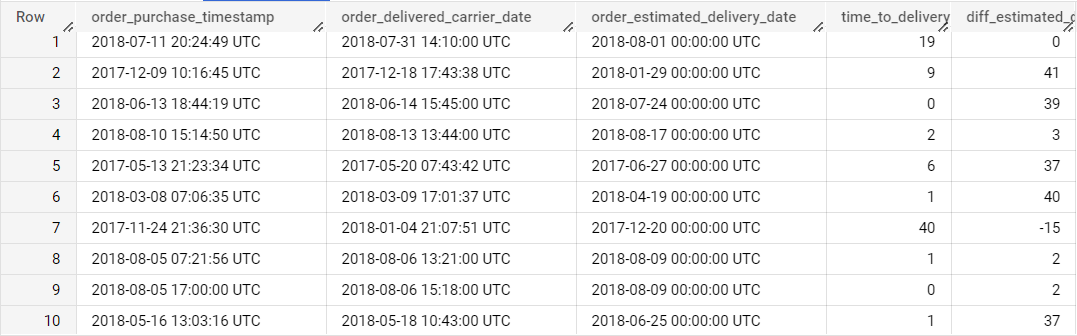
This is the mean and sum of freight value and price for each state.  
  
  
5.1) Calculate days between purchasing, delivering and estimated delivery

select order\_purchase\_timestamp,order\_delivered\_carrier\_date,order\_estimated\_delivery\_date, ifnull(Datetime\_diff(order\_estimated\_delivery\_date,order\_purchase\_timestamp,day),0)Approx\_time,ifnull(Datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day),0)as delivery\_time  from `Target.orders`  
  


The above results tell us that the entire logistics of Target is very efficient as we can see a good amount of difference in days between the estimated and actual delivery.  
  
  
5.2) Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:

* + time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date
  + diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

select order\_purchase\_timestamp,order\_delivered\_carrier\_date,order\_estimated\_delivery\_date,datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)as time\_to\_delivery,datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)as diff\_estimated\_delivery

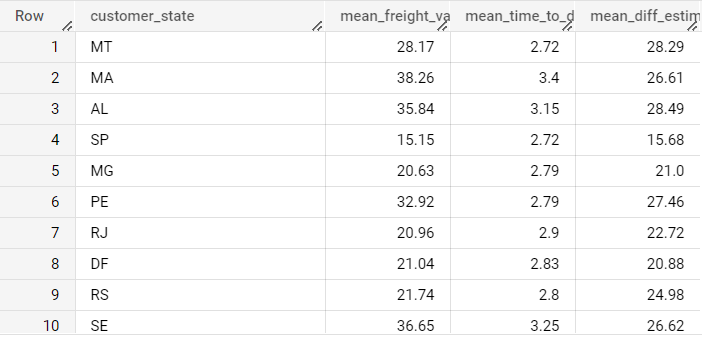
from `Target.orders`  
  


This result shows the time between order and actual delivery.It also depicts the difference in time between estimated and actual delivery which suggest the strong logistics of Target.  
  
  
5.3) Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1



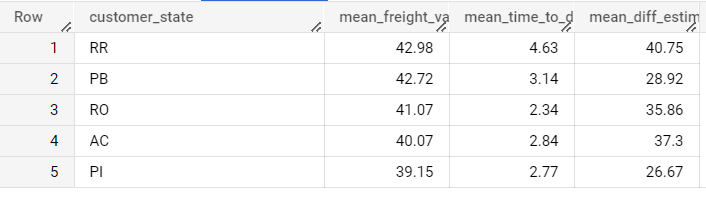
5.5) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5  
  
select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1

order by 2 desc

Limit 5

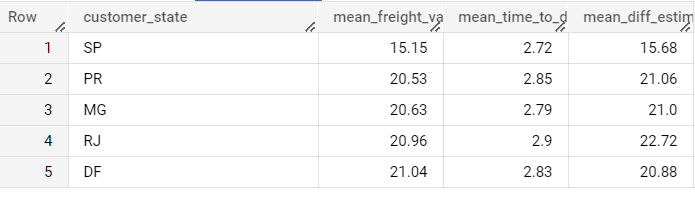


These are the top 5 states with highest mean freight value.  
  
select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1

order by 2

Limit 5  
  


Top 5 States with lowest mean freight value.  
  
  
5.6) Top 5 states with highest/lowest average time to delivery

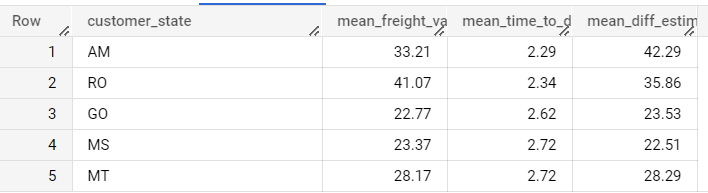
select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1

order by 3

Limit 5



Top 5 states with lowest mean time to delivery  
  
select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1

order by 3 desc

Limit 5



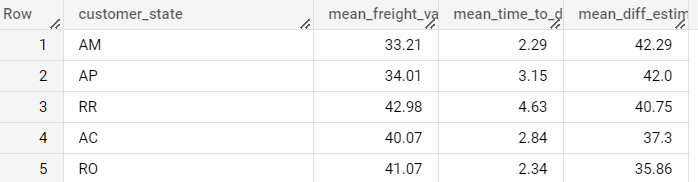
Top 5 States with highest mean time to delivery.  
  
  
5.7) Top 5 states where delivery is really fast/ not so fast compared to estimated date  
  
select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1

order by 4 desc

Limit 5

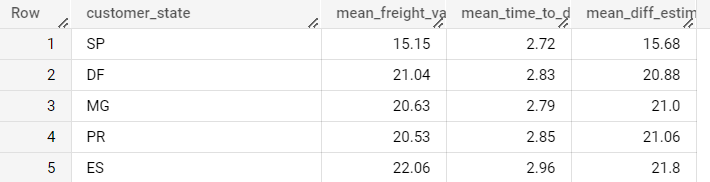
  
Top 5 States with highest mean difference between estimated and actual delivery i.e. the delivery is much faster than expected.  
  
select customer\_state,round(avg(freight\_value),2)as mean\_freight\_value,round(avg(datetime\_diff(order\_delivered\_carrier\_date,order\_purchase\_timestamp,day)),2)as mean\_time\_to\_delivery,round(avg(datetime\_diff(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,day)),2)as mean\_diff\_estimated\_delivery

from `Target.orders` o join `Target.Customers`c on o.customer\_id=c.customer\_id join `Target.order\_items`oi on o.order\_id=oi.order\_id

group by 1

order by 4

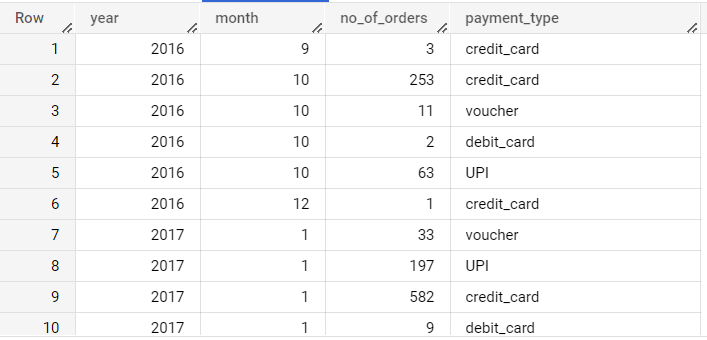
Limit 5



Top 5 states with lowest mean difference between estimated and original delivery i.e. the delivery is still faster than expected but not that fast as compared to the above result.  
  
  
6.1) Month over Month count of orders for different payment types

select extract(year from order\_purchase\_timestamp)as year,extract(month from order\_purchase\_timestamp)as month,count(distinct o.order\_id)as no\_of\_orders,payment\_type from `Target.payments`p join `Target.orders`o on o.order\_id=p.order\_id

group by 1,2,4

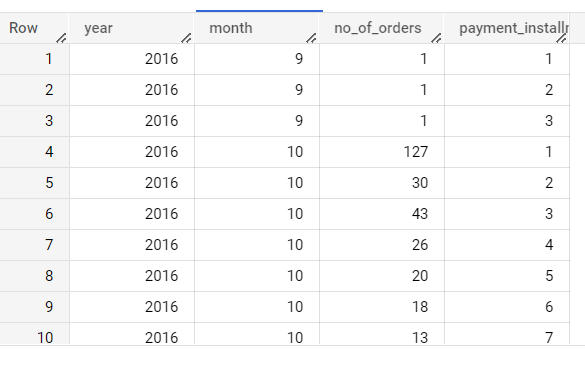
order by 1,2  
  


Customers use credit cards mostly.They are also utilising the vouchers and debit cards efficiently.In 1 out of 5 payments customer use UPI as well.

6.2) Count of orders based on the no. of payment installments  
  
select extract(year from order\_purchase\_timestamp)as year,extract(month from order\_purchase\_timestamp)as month,count(distinct o.order\_id)as no\_of\_orders,payment\_installments from `Target.payments`p join `Target.orders`o on o.order\_id=p.order\_id

group by 1,2,4

order by 1,2

  
This is the count of orders based on the number of payment installments month over month.