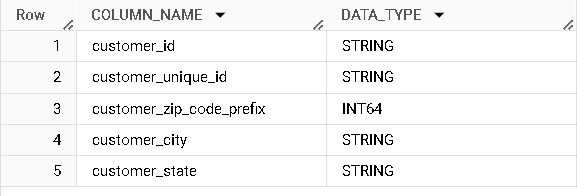
**1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

**1.Data type of all columns in the "customers" table.**

SELECT COLUMN\_NAME, DATA\_TYPE

FROM Target.INFORMATION\_SCHEMA.COLUMNS

where table\_name="customers";



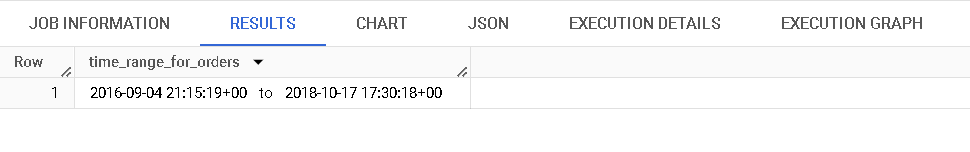
**Insights**: Will get to know about the datatype of each column.

**Recommendation**: As some of the date & time operations are sensitive to datatype, Make sure you are working with the correct datatype.

**2.Get the time range between which the orders were placed.**

select concat(min(order\_purchase\_timestamp),"   to   ",max(order\_purchase\_timestamp)) as time\_range\_for\_orders

from Target.orders;



**Insights:** Will provide us the timeframe of data captured.

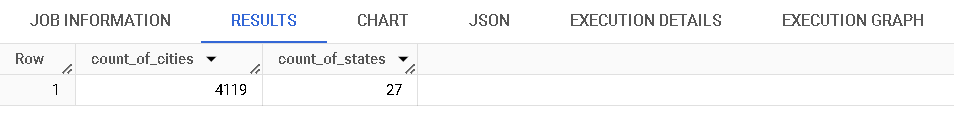
**Recommendation:** Will help us to make the analysis considering the external factors (like lockdown, natural calamities) occurred during that period.

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

select count(distinct customer\_city) as count\_of\_cities,

count(distinct customer\_state) as count\_of\_states

from Target.customers;



**Insights:** Will let us know the magnitude of services the company is offering.

**Recommendations:** Will help in promoting and advertising.

----------------------------------------------------------------------

**2. In-depth Exploration:**

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

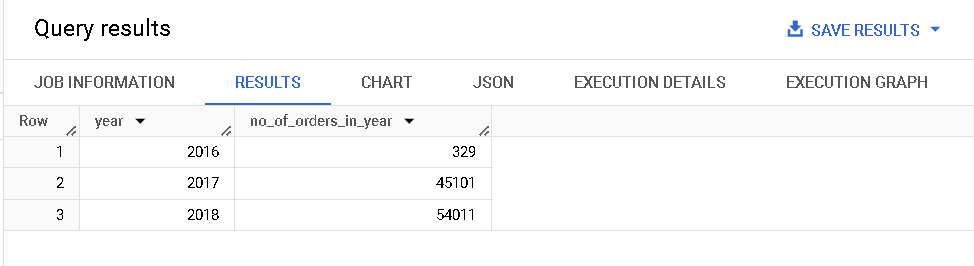
select extract(year from order\_purchase\_timestamp) as year,

count(\*) as no\_of\_orders\_in\_year

from Target.orders

group by extract(year from order\_purchase\_timestamp)

order by year;



**Insights:** Will provide us the number of orders placed in each year

**Recommendations:**

-Orders in the year 2016 started from 9th month which won’t give a correct measure for comparision.

-But when compared between 2017 and 2018, We can see the orders got increased compared to its previous year.

-As there is an increase in sales, there was no need of major changes in company working model currently. Company can focus on improving customer satisfaction and R&D can happen for deploying any changes in working model.

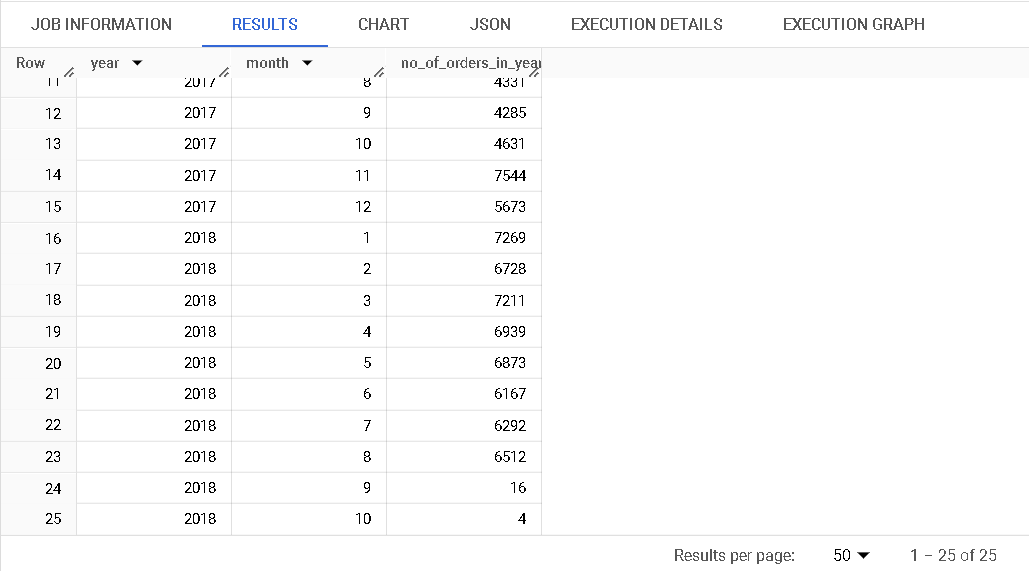
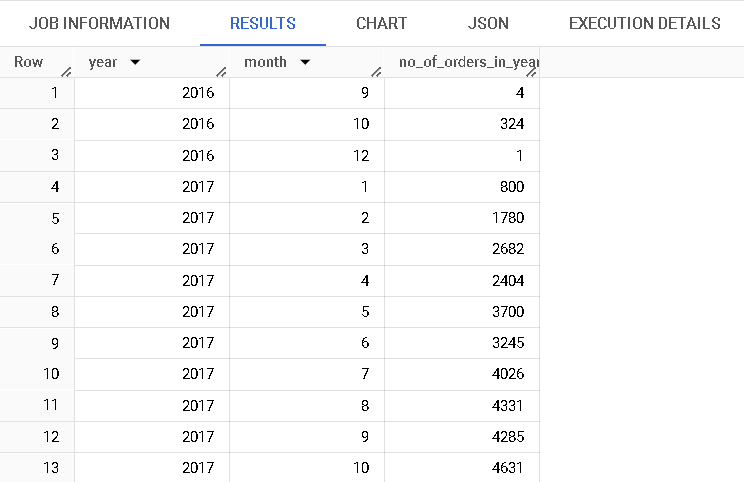
**2.Can we see some kind of monthly seasonality in terms of the no. of orders being placed?**

select extract(year from order\_purchase\_timestamp) as year,extract(month from order\_purchase\_timestamp) as month,count(\*) as no\_of\_orders\_in\_yearm

from Target.orders

group by extract(year from order\_purchase\_timestamp),extract(month from order\_purchase\_timestamp)

order by year,month;



**Insights:** Will provide us the number of orders got placed during each month in a year.

**Recommendations:**

-Unable to withdraw any major conclusions with respect to monthly seasonality as we have only a couple of years to compare. (2017 & 2018)

-Almost an increasing trend has been maintained without any major fluctuations till 8th month of 2018.

-In 9th and 10th month of 2018 a major drop in sales observed.

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

* + **0-6 hrs : Dawn**
  + **7-12 hrs : Mornings**
  + **13-18 hrs : Afternoon**
  + **19-23 hrs : Night**

select tn,count(\*) as no\_of\_orders

from(

select hour,case

  when hour between 0 and 5  then "Dawn"

  when hour between 6 and 11  then "Mornings"

  when hour between 12 and 17  then "Afternoon"

  when hour between 17 and 23  then "Night"

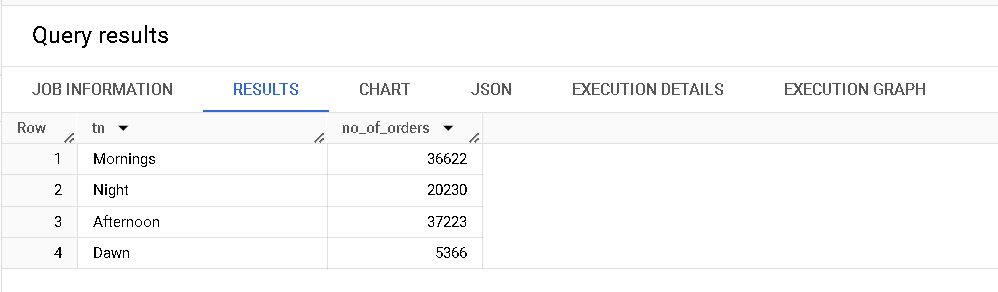
  end as tn

  from(

select extract(hour from TIMESTAMP\_SUB(order\_purchase\_timestamp, INTERVAL 3 HOUR)) as hour

from Target.orders) as t) t1

group by tn ;



**Insights:**

-Most number of orders have been placed on both in Mornings and Afternoon.

-Least number of orders are placed in Dawn.

**Recommendations:**

-Can try leveraging the delivery charges and providing extra discounts & coupons during the night & dawn period to increase the order count.

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

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

select customer\_state,extract(month from order\_purchase\_timestamp) as month,count(\*) as no\_of\_orders

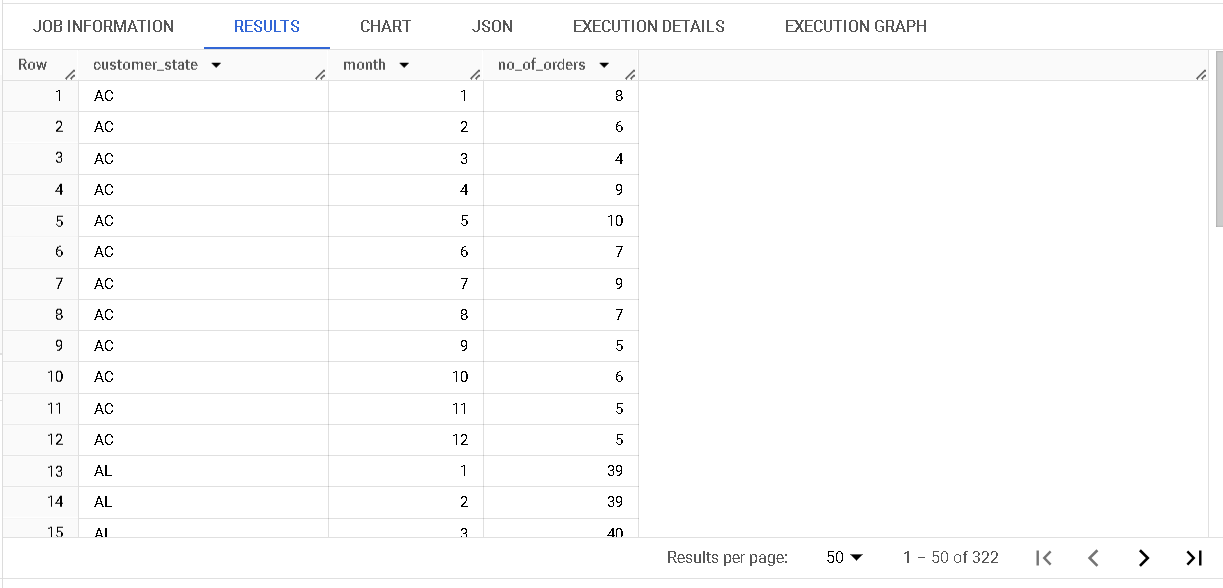
from Target.customers

left join Target.orders

using(customer\_id)

group by customer\_state,extract(month from order\_purchase\_timestamp)

order by customer\_state,month;



**Insights:** Provide us the month wise orders count for each state.

**Recommendations:**

* Will help us in analyzing what was working in states with highest orders count compared to other states.

**2.How are the customers distributed across all the states?**

select customer\_state,count(customer\_id) as no\_of\_customers\_in\_state

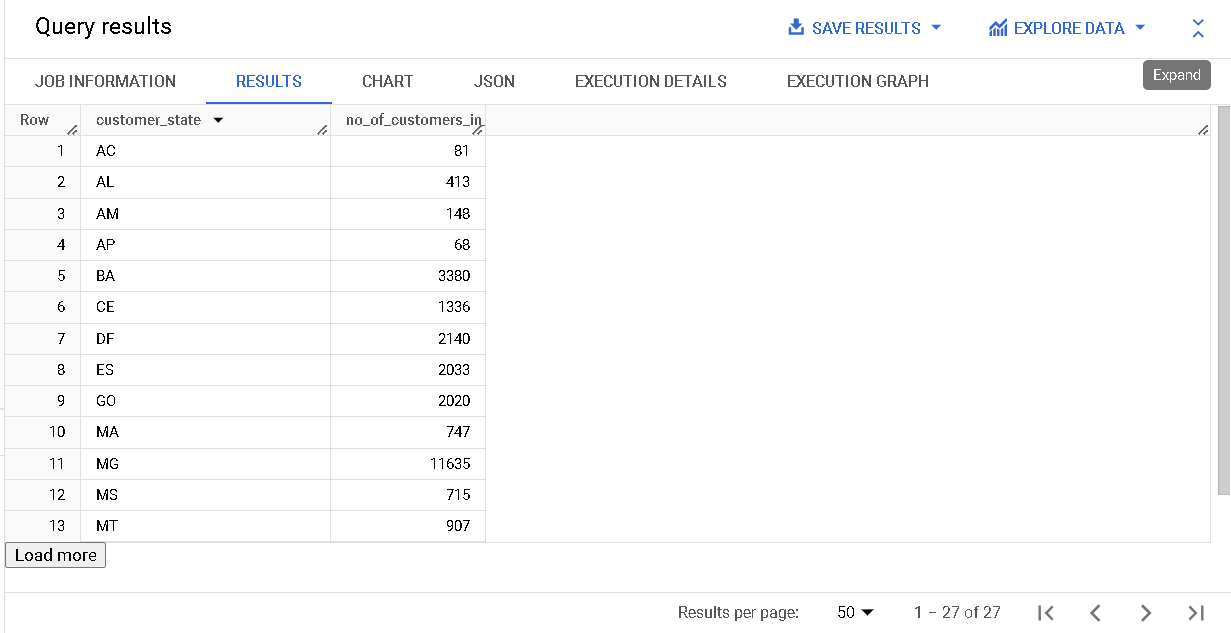
from Target.customers

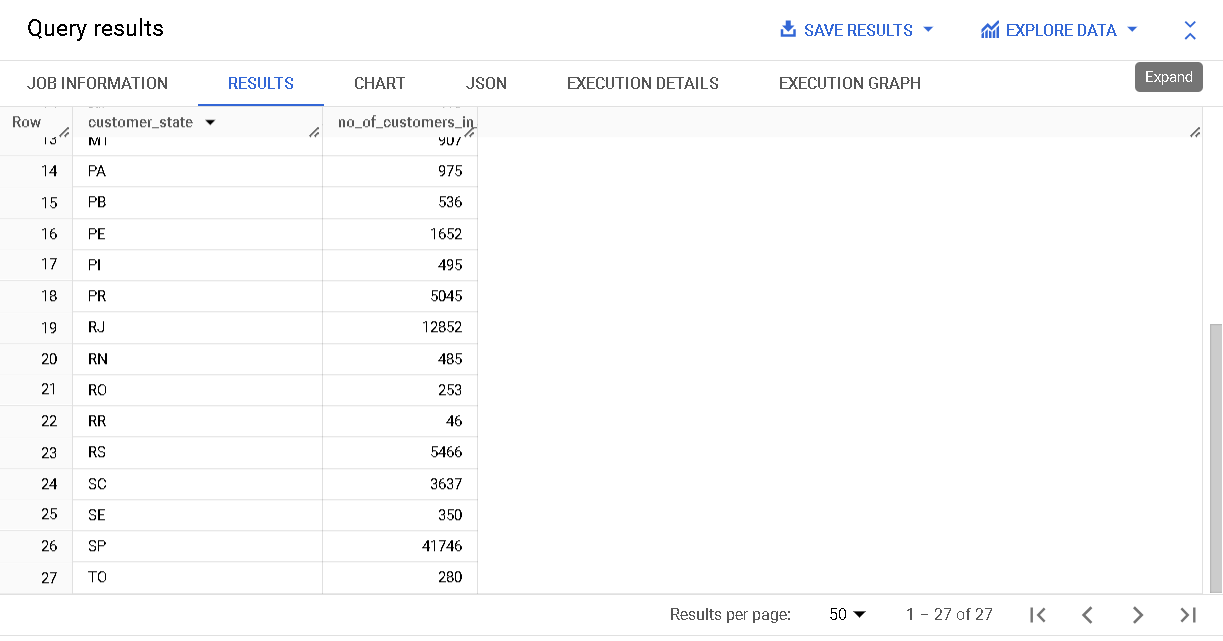
left join Target.orders

using(customer\_id)

group by customer\_state

order by customer\_state;





**Insights:** Provide us the number of customers in each state who ordered at least once.

**Recommendations:**

-Helps in promoting and advertisement.

--------------------------------------------------------------------------------------------------------------------------

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

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.**

select sum(cost\_in\_2017) as Total\_cost\_in\_2017,sum(cost\_in\_2018) as Total\_cost\_in\_2018,Round(((sum(cost\_in\_2018)-sum(cost\_in\_2017))/sum(cost\_in\_2017))\*100,2) as percentage\_increase

from(

select extract(year from order\_purchase\_timestamp) as year,case

 when extract(year from order\_purchase\_timestamp) = 2017 then price

 else 0 end as cost\_in\_2017,case

 when extract(year from order\_purchase\_timestamp) = 2018 then price

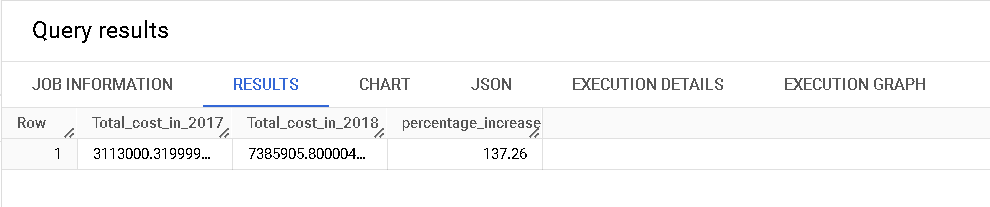
 else 0 end as cost\_in\_2018

from Target.order\_items

join Target.orders

using(order\_id)

where (extract(year from order\_purchase\_timestamp)=2017 or extract(year from order\_purchase\_timestamp) =2018) and extract(month from order\_purchase\_timestamp)!=9 and extract(month from order\_purchase\_timestamp)!=10 and extract(month from order\_purchase\_timestamp)!=11 and extract(month from order\_purchase\_timestamp)!=12) as t;



**Insights:** Will give us the percentage increase in cost of orders from 2017 to 2018

**Recommendations:**

Helps in planning for expanding the company market.

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

select customer\_state,sum(price) Total\_sum\_for\_each\_state,avg(price) Avg\_price\_for\_each\_state

from Target.order\_items

join Target.orders

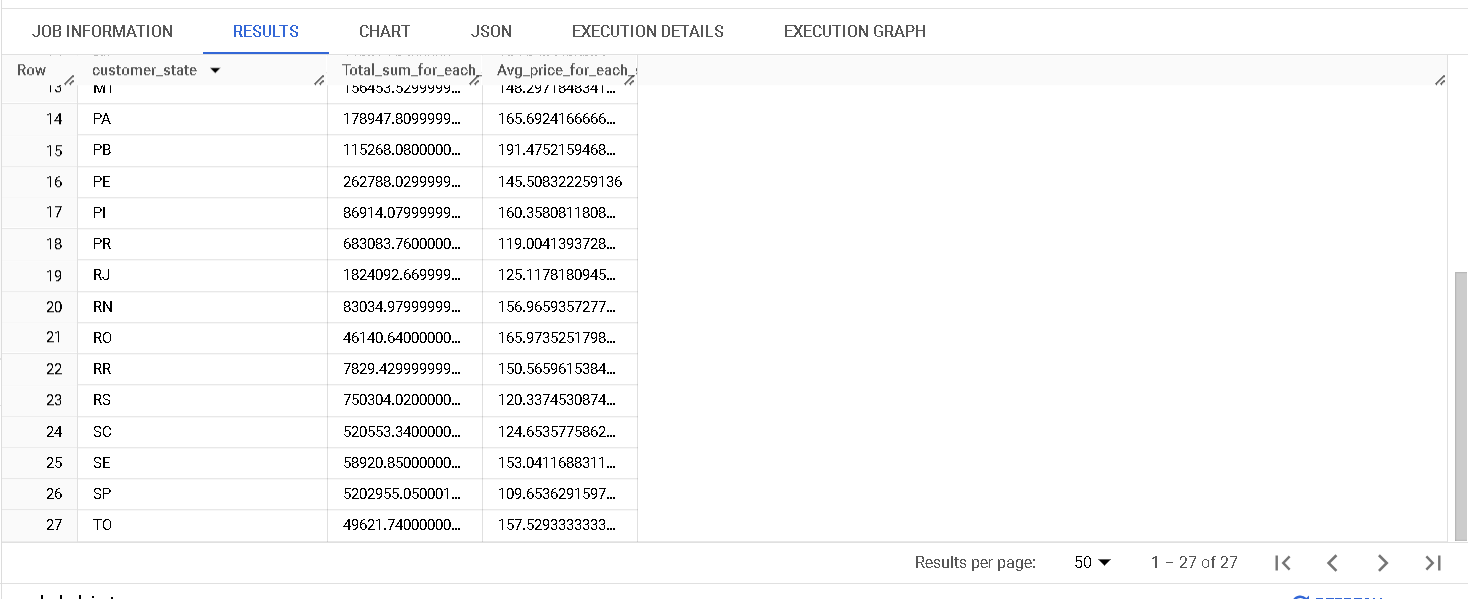
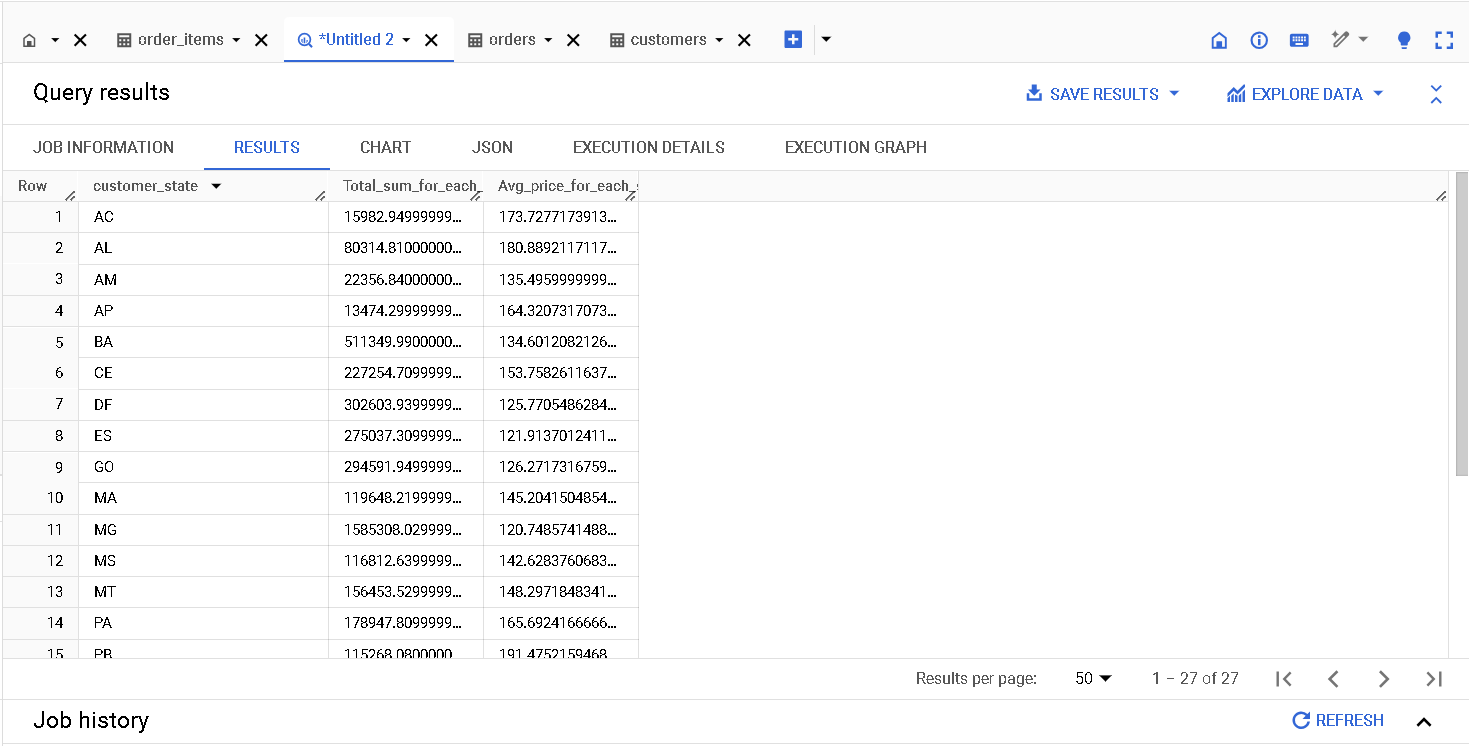
using(order\_id)

right join Target.customers

using(customer\_id)

group by customer\_state

order by customer\_state ;



**Insights:** Provide us the total sum & Average order price in each state

**Recommendations:**

* Helps us in targeting which state is spending more amount on orders and from which state the average order price is high.

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

select customer\_state,sum(freight\_value) Total\_sum\_for\_each\_state,avg(freight\_value) Avg\_freight\_value\_for\_each\_state

from Target.order\_items

join Target.orders

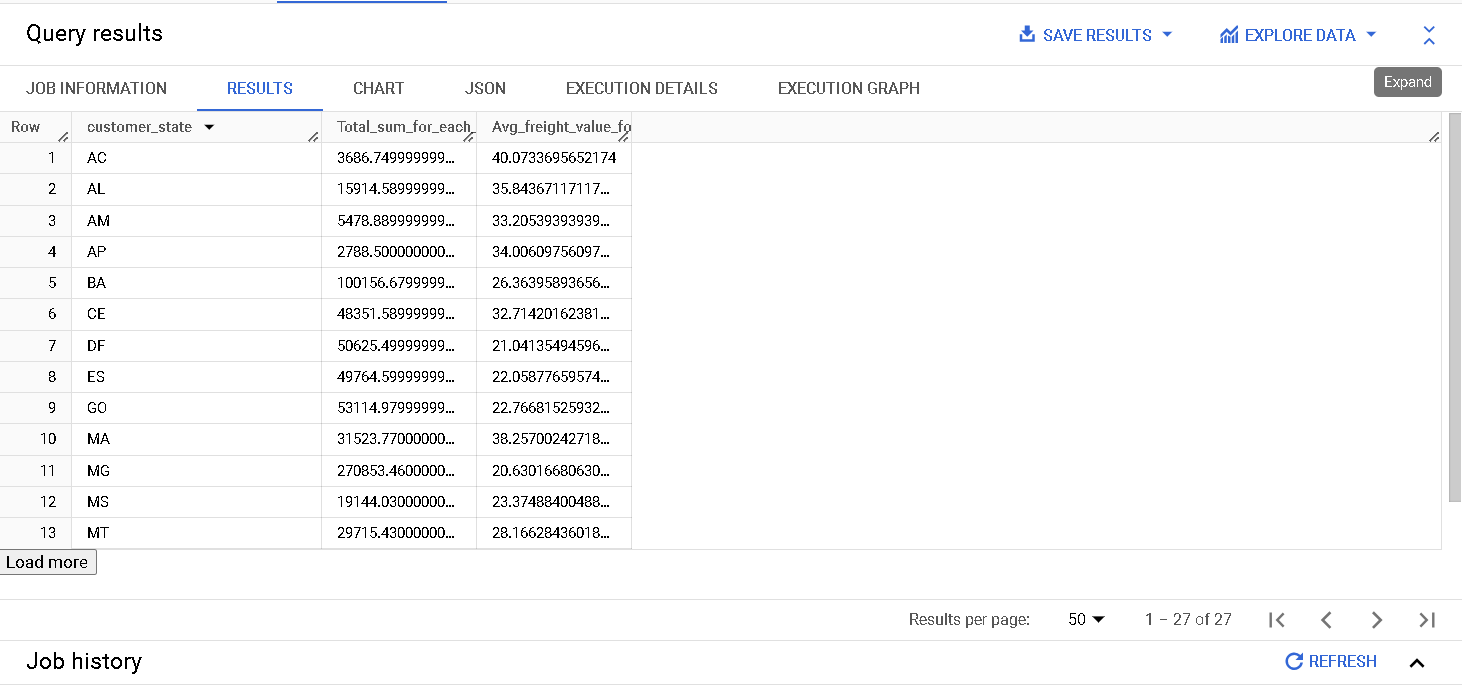
using(order\_id)

right join Target.customers

using(customer\_id)

group by customer\_state

order by customer\_state ;





**Insights:** Provide us the Total & Average value of order freight for each state.

**Recommendations:**

-Helps us in comparing the delivery methods for states with highest Average freight value with the one which has least Average freight value.

**5.Analysis based on sales, freight and delivery time.**

**1. Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.  
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.  
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:**

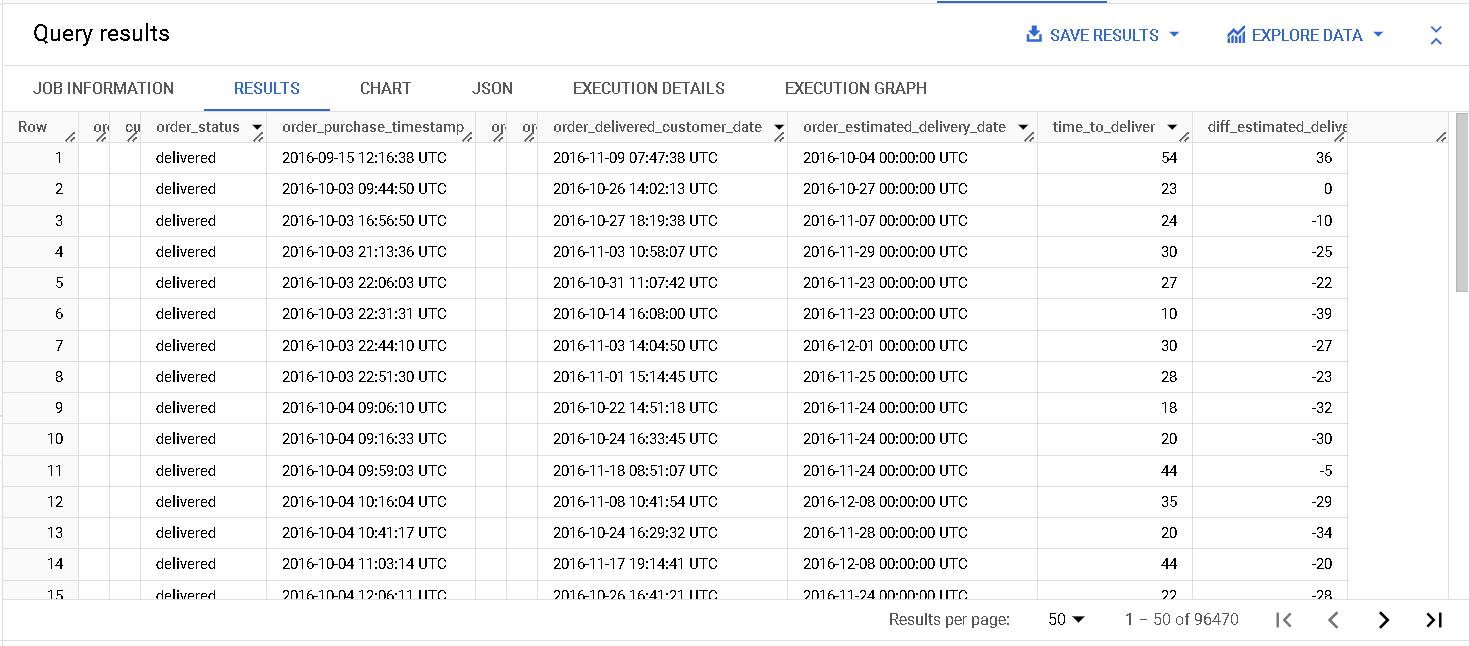
* + **time\_to\_deliver = order\_delivered\_customer\_date - order\_purchase\_timestamp**
  + **diff\_estimated\_delivery = order\_delivered\_customer\_date - order\_estimated\_delivery\_date**

select \*,date\_diff(order\_delivered\_customer\_date,order\_purchase\_timestamp, day) as time\_to\_deliver,date\_diff(order\_delivered\_customer\_date, order\_estimated\_delivery\_date,day) as diff\_estimated\_delivery

from Target.orders

where lower(order\_status)="delivered" and order\_delivered\_customer\_date is not null

order by order\_purchase\_timestamp;



**Insights:** Provide us the comparison between the expected and actual number of days taken for delivering the product.

**Recommendations:**

-Helps in tracking the orders where orders get delayed and try to minimize them

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

select customer\_state,t.Avg\_freight\_value\_for\_each\_state,r2 as ranking\_based\_on\_avg\_freight\_value

from(

select customer\_state,sum(freight\_value) Total\_sum\_for\_each\_state,avg(freight\_value) Avg\_freight\_value\_for\_each\_state,

rank() over(order by avg(freight\_value) ) r1 ,rank() over(order by avg(freight\_value) desc ) r2

from Target.order\_items

join Target.orders

using(order\_id)

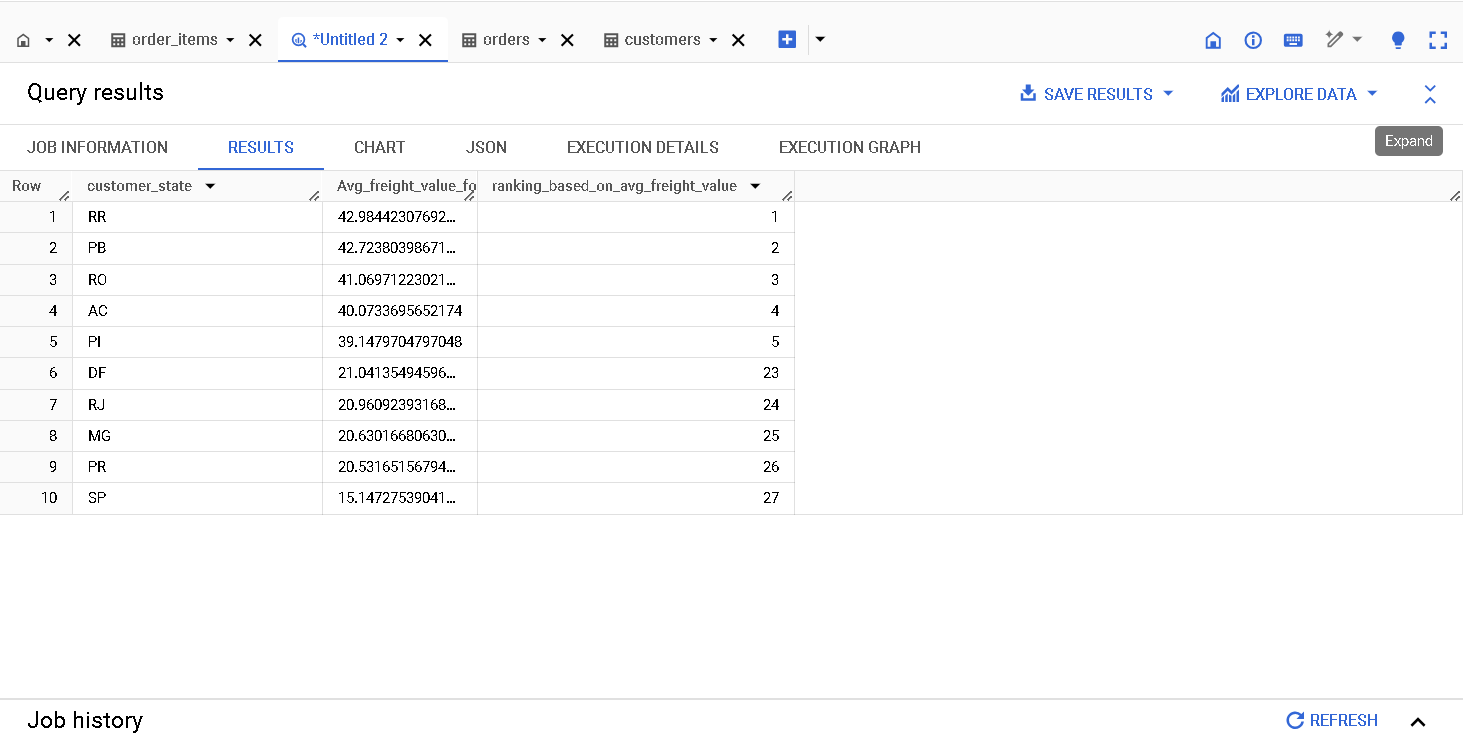
right join Target.customers

using(customer\_id)

group by customer\_state) as t

where r1<=5 or r2<=5

order by t.Avg\_freight\_value\_for\_each\_state desc;



**Insights:** Provide us the top 5 states with the highest & lowest average freight value.

**Recommendations:** Helps us in comparing the delivery models of states with highest average freight value with them of lowest value.

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

**select customer\_state,avg\_delivery\_time\_in\_min,r2 as rank\_in\_fast\_delivery**

**from(**

**select customer\_state,avg(timestamp\_diff(order\_delivered\_customer\_date,order\_purchase\_timestamp,minute)) as avg\_delivery\_time\_in\_min,rank() over(order by avg(timestamp\_diff(order\_delivered\_customer\_date,order\_purchase\_timestamp,minute)) desc ) as r1,rank() over(order by avg(timestamp\_diff(order\_delivered\_customer\_date,order\_purchase\_timestamp,minute))) as r2**

**from Target.order\_items**

**join Target.orders**

**using(order\_id)**

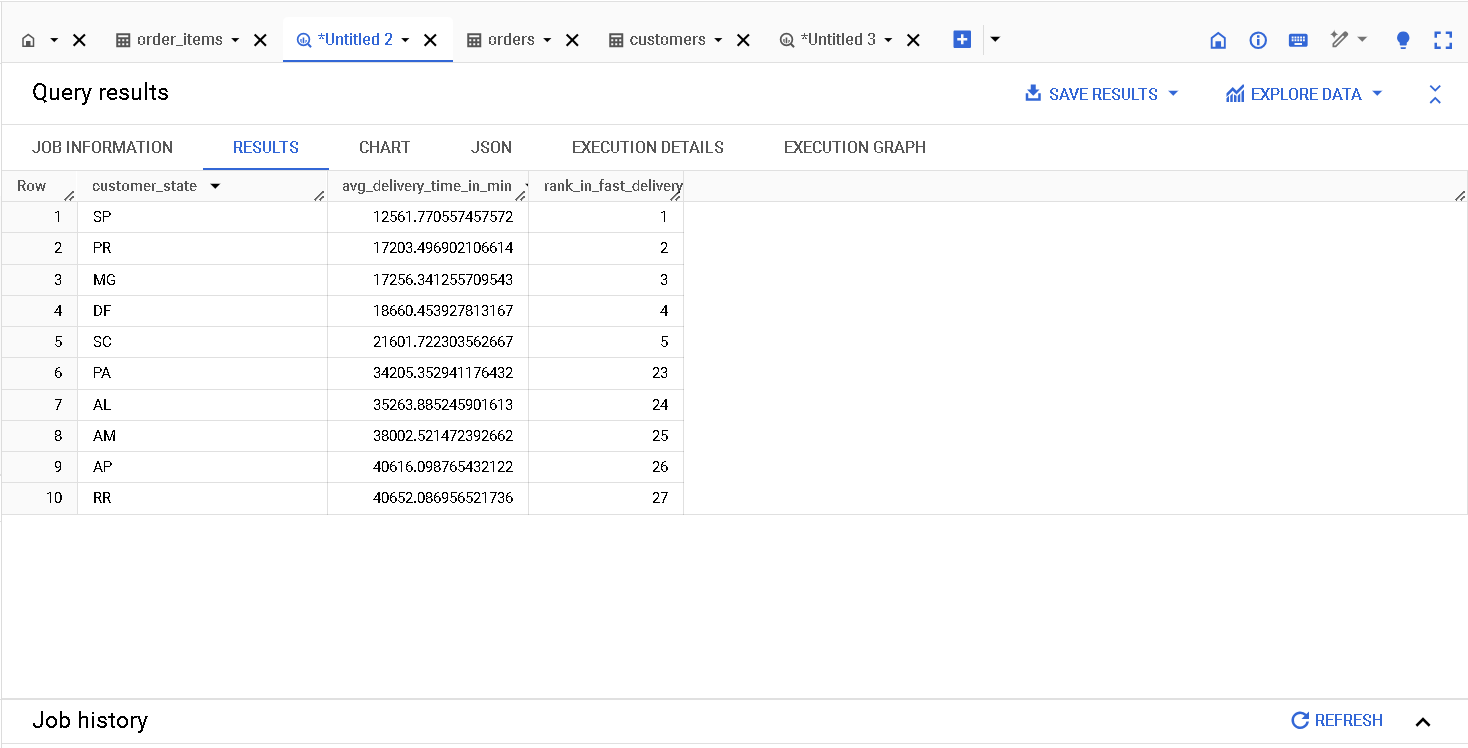
**right join Target.customers**

**using(customer\_id)**

**group by customer\_state) as t**

**where r1<=5 or r2<=5**

**order by rank\_in\_fast\_delivery;**



**Insights:** Provide us the top 5 states with the highest & lowest average delivery time

**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.**

select customer\_state,

avg(timestamp\_diff(order\_delivered\_customer\_date,order\_estimated\_delivery\_date,minute)) as avg\_delivery\_time\_in\_min,

rank() over(order by avg(timestamp\_diff(order\_delivered\_customer\_date,order\_estimated\_delivery\_date,minute))) as rank\_for\_fast\_delivery

from Target.order\_items

join Target.orders

using(order\_id)

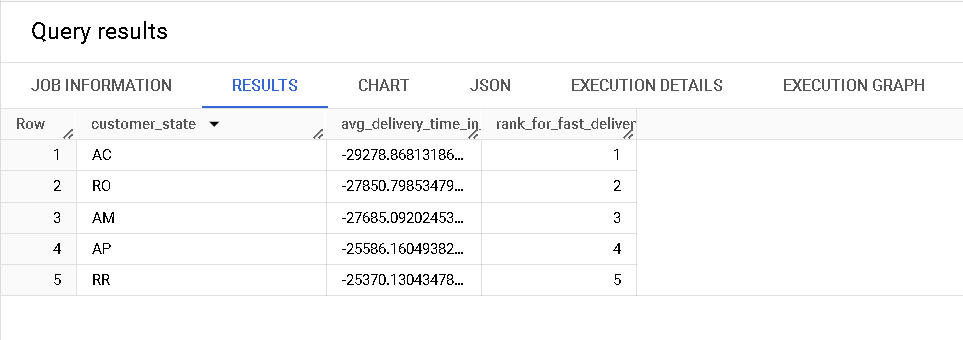
right join Target.customers

using(customer\_id)

group by customer\_state

order by rank\_for\_fast\_delivery

limit 5;



I

**Insights:** Provide us the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

**Recommendations:**

Helps us in tracking the delivery models in these states and implement them in the states where average delivery time is high.

**6. Analysis based on the payments:**

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

select extract(year from order\_purchase\_timestamp) as year,extract(month from order\_purchase\_timestamp) as month,payment\_type,count(8) as no\_of\_orders

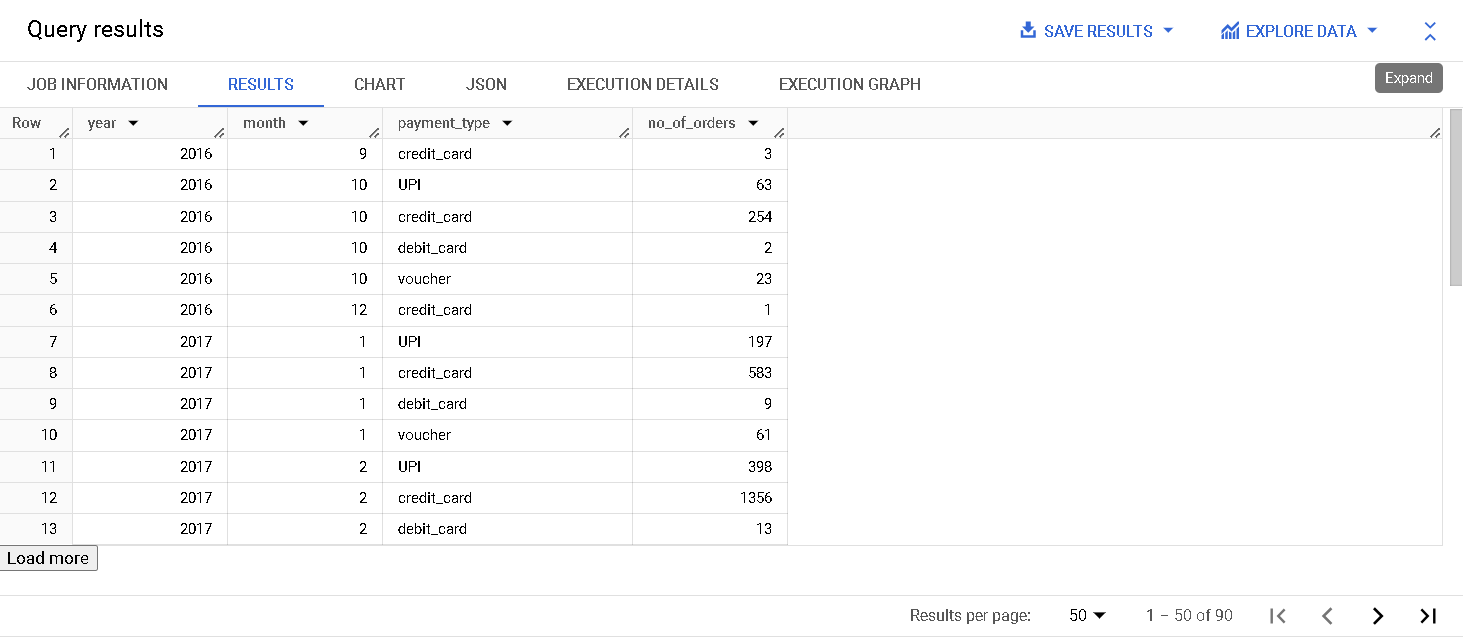
from Target.orders

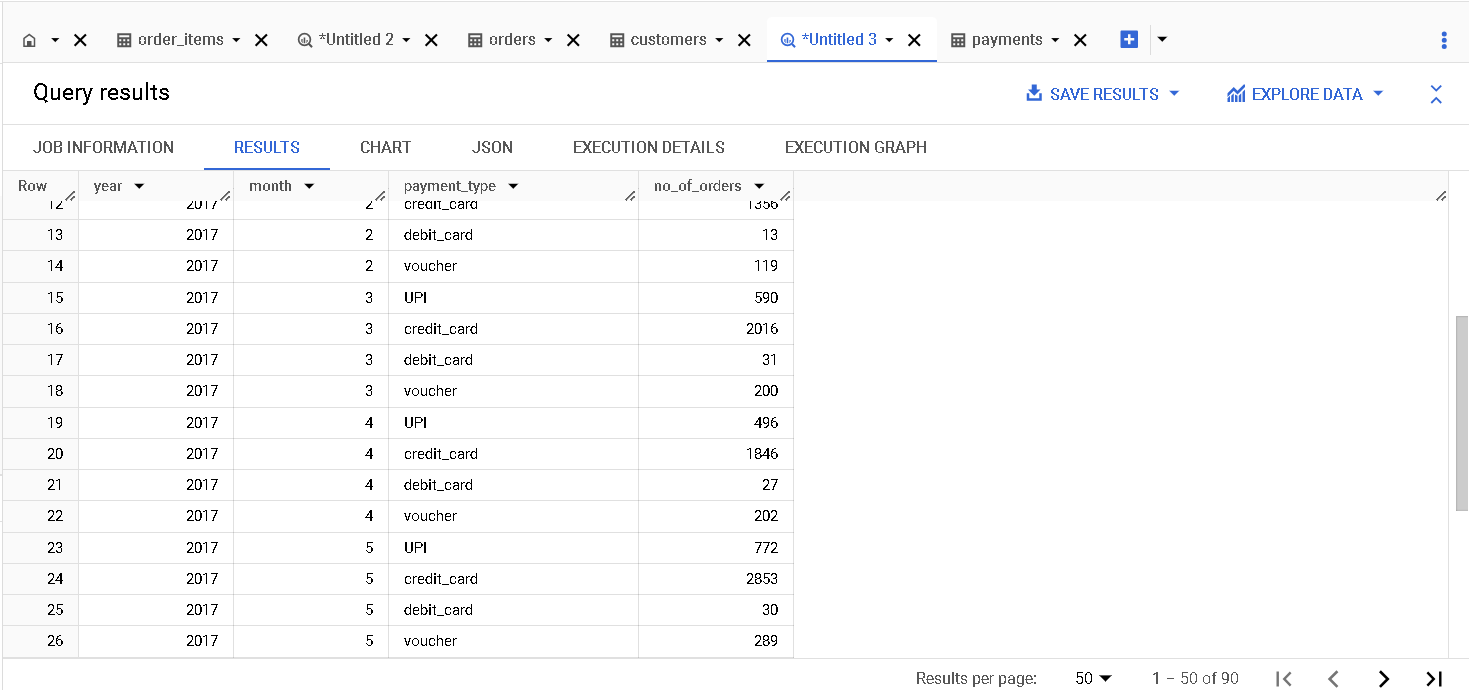
join Target.payments

using(order\_id)

group by extract(year from order\_purchase\_timestamp),extract(month from order\_purchase\_timestamp),payment\_type

order by year,month,payment\_type;





**Insights:** Provide us the month on month no. of orders placed using different payment

Types.

**Recommendations:**

-Helps us tracking which payment method was most used and adding more flexibility to those payment methods.

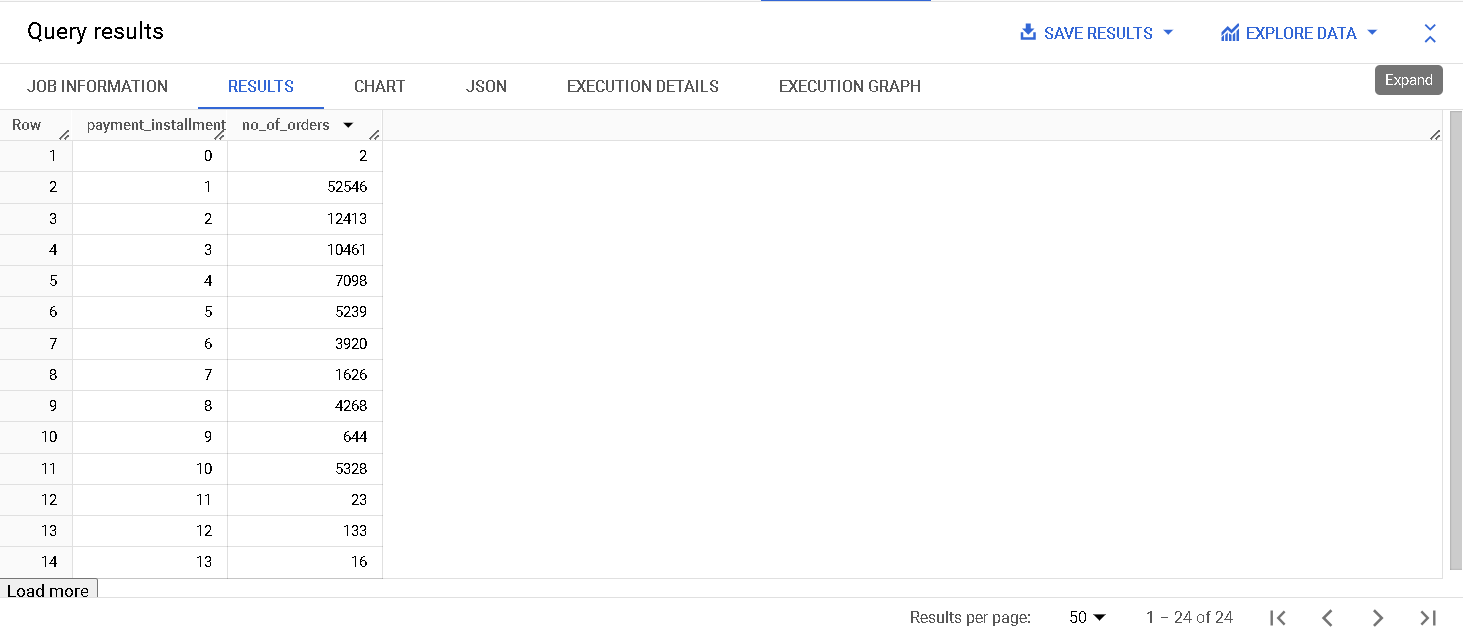
**2.Find the no. of orders placed on the basis of the payment installments that have been paid.**

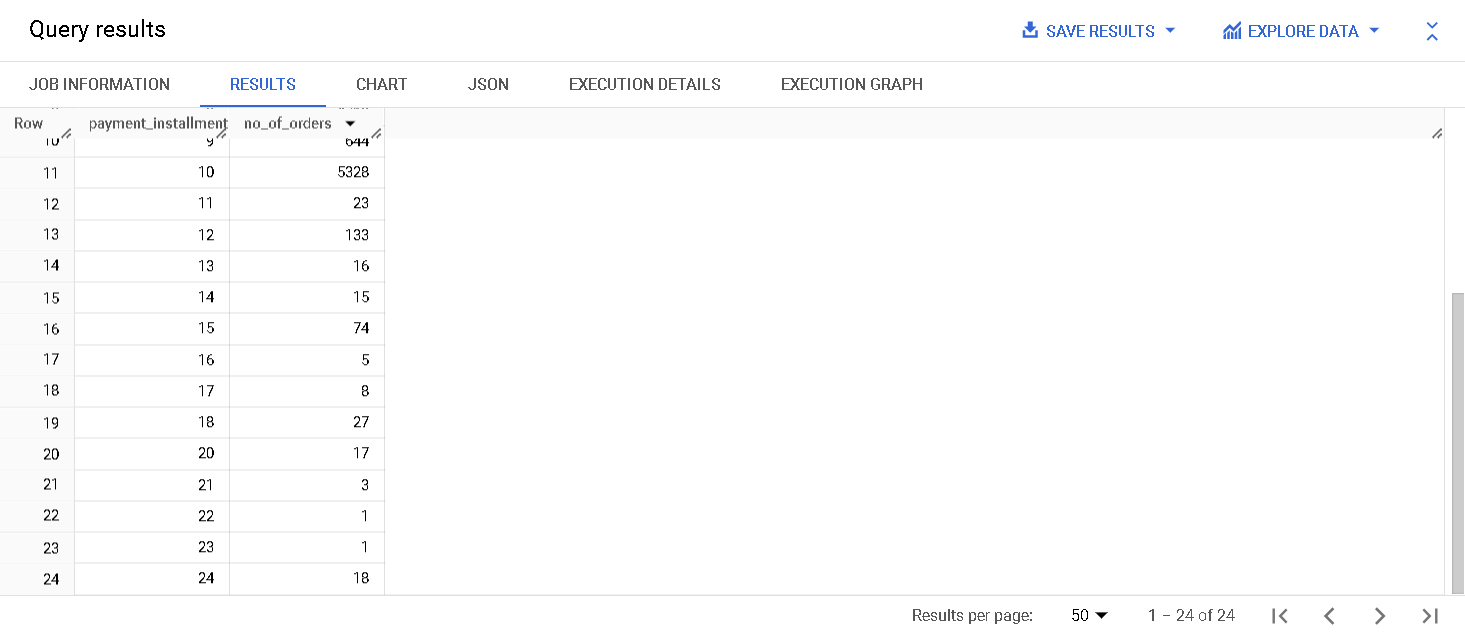
select payment\_installments,count(\*) as no\_of\_orders

from Target.payments

group by payment\_installments

order by payment\_installments;





**Insights:** Provide us the number of orders placed on the basis of the payment installments that have been paid.

**Recommendations:**

Here most orders are placed with payment installments from 1 to 10. So provide more flexibility like lesser interest rate or no cost emi to those instalments which can grab more customers.