**BUSINESS CASE :- Target SQL**

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

1.Data type of columns in a table :-

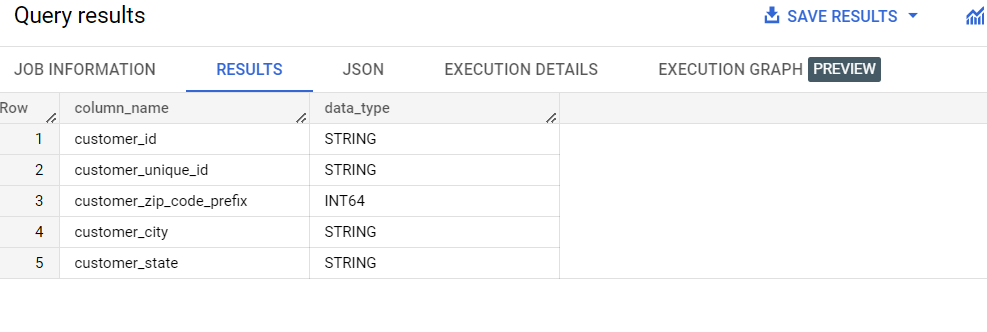
SELECT

  column\_name, data\_type

FROM

  model-bonsai-382608.retail\_data.INFORMATION\_SCHEMA.COLUMNS

  where table\_name='customers'



2.Time period for which the data is given:-

select min(order\_purchase\_timestamp) as opt1,

max(order\_purchase\_timestamp) as opt2 from `retail\_data.orders`;



3. Cities and States of customers ordered during the given period :-

select distinct customer\_city,customer\_state

from  `retail\_data.customers`



**2. In-depth Exploration**:

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 (YEAR FROM order\_purchase\_timestamp) AS year,

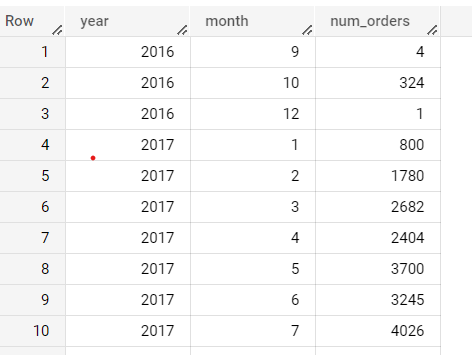
 EXTRACT (MONTH FROM order\_purchase\_timestamp) AS month,

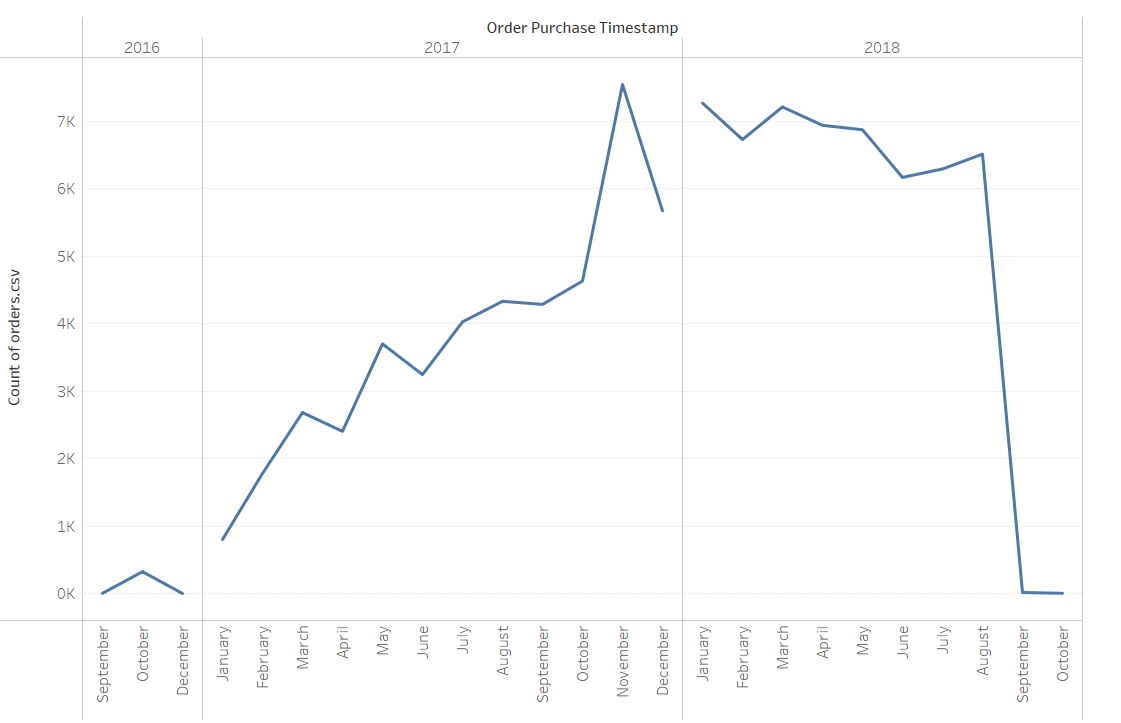
 COUNT(\*) as num\_orders

FROM `retail\_data.orders`

GROUP BY year,month

ORDER BY year,month





Yes, there is growing trend on e commerce in brazil.

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

SELECT

  CASE

    WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) >= 0 AND EXTRACT(HOUR FROM order\_purchase\_timestamp) < 6 THEN 'Dawn'

    WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) >= 6 AND EXTRACT(HOUR FROM order\_purchase\_timestamp) < 12 THEN 'Morning'

    WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) >= 12 AND EXTRACT(HOUR FROM order\_purchase\_timestamp) < 18 THEN 'Afternoon'

    WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) >= 18 AND EXTRACT(HOUR FROM order\_purchase\_timestamp) <= 23 THEN 'Night'

    ELSE 'Unknown'

  END AS time\_period,

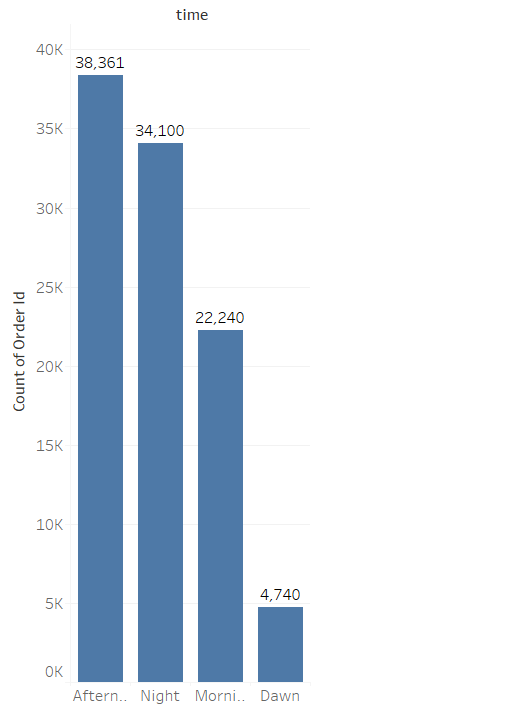
COUNT(\*) AS num\_orders

FROM `retail\_data.orders`

GROUP BY time\_period

ORDER BY num\_orders DESC





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

1. Get month on month orders by states:-

SELECT

    c.customer\_state,

    EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,

    COUNT(\*) AS num\_of\_orders

FROM

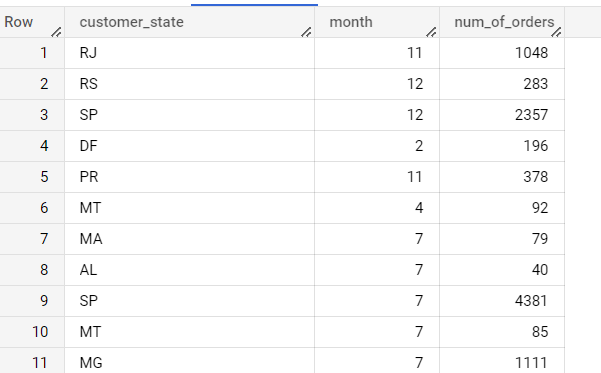
    `retail\_data.orders` AS o

    JOIN `retail\_data.customers` AS c ON c.customer\_id = o.customer\_id

GROUP BY

    c.customer\_state,

    month



2. Distribution of customers across the states in Brazil:-

select t1.customer\_state,t1.num\_customers,t2.num\_orders

from

(select

customer\_state,count(customer\_id) as num\_customers from `retail\_data.customers`

group by customer\_state) as t1

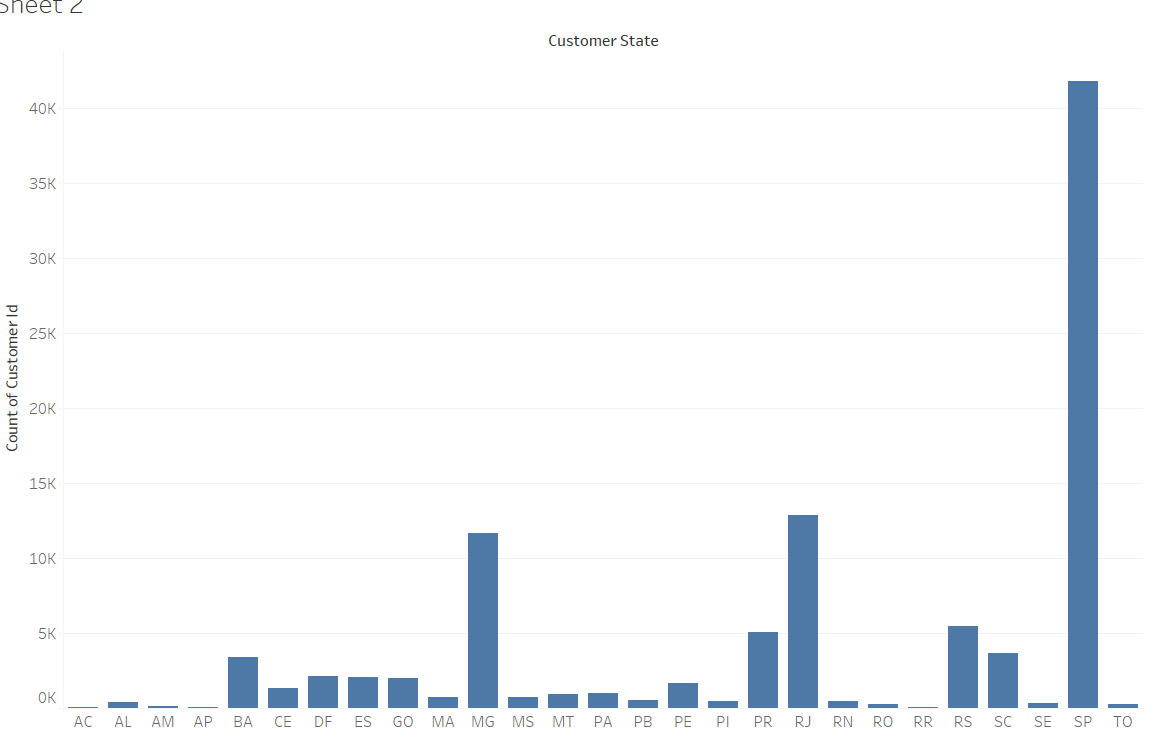
join

(select c.customer\_state,count(distinct o.order\_id) as num\_orders

from `retail\_data.orders` o join `retail\_data.customers` c on o.customer\_id=c.customer\_id group by c.customer\_state ) as t2

on t1.customer\_state=t2.customer\_state





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

* 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

WITH orders\_2017 AS (

  SELECT

    EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,

    round(SUM(p.payment\_value),2) AS total\_payment\_value\_2017

  FROM `retail\_data.orders` o

  INNER JOIN `retail\_data.payments` p ON o.order\_id = p.order\_id

  WHERE EXTRACT(YEAR FROM o.order\_purchase\_timestamp) = 2017

    AND EXTRACT(MONTH FROM o.order\_purchase\_timestamp) BETWEEN 1 AND 8

  GROUP BY EXTRACT(MONTH FROM o.order\_purchase\_timestamp)

),

orders\_2018 AS (

  SELECT

    EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS month,

    round(SUM(p.payment\_value),2) AS total\_payment\_value\_2018

  FROM `retail\_data.orders` o

  INNER JOIN `retail\_data.payments` p ON o.order\_id = p.order\_id

  WHERE EXTRACT(YEAR FROM o.order\_purchase\_timestamp) = 2018

    AND EXTRACT(MONTH FROM o.order\_purchase\_timestamp) BETWEEN 1 AND 8

  GROUP BY EXTRACT(MONTH FROM o.order\_purchase\_timestamp )

)

SELECT

  o1.month,

  o1.total\_payment\_value\_2017,

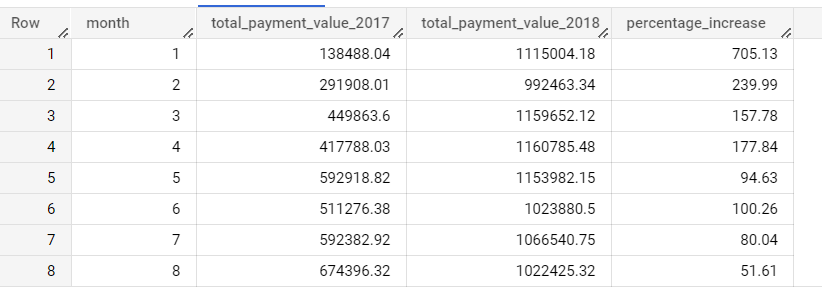
  o2.total\_payment\_value\_2018,

  round(((o2.total\_payment\_value\_2018 - o1.total\_payment\_value\_2017) / o1.total\_payment\_value\_2017) \* 100,2) AS percentage\_increase

FROM orders\_2017 as o1

INNER JOIN orders\_2018 as o2 ON o1.month = o2.month

order by o1.MONTH;



2. Mean & Sum of price and freight value by customer state

select

c.customer\_state,

avg(oi.price) as mean,

sum(oi.price) as sum\_of\_price,

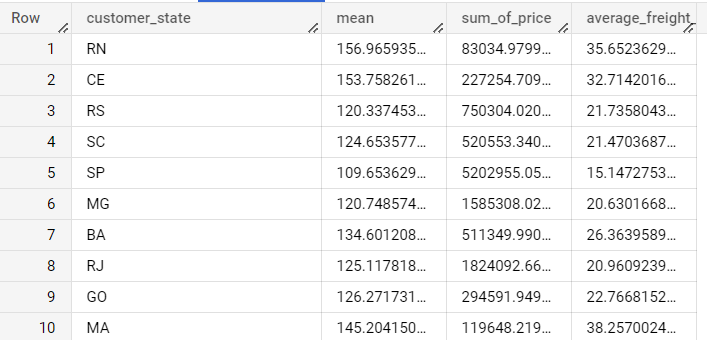
avg(oi.freight\_value) as average\_freight\_value

from `retail\_data.customers`as c  join `retail\_data.orders`

as o on o.customer\_id = c.customer\_id

join `retail\_data.order\_items` as oi on oi.order\_id=o.order\_id

group by c.customer\_state





**5. Analysis on sales, freight and delivery time:-**

1. Calculate days between purchasing, delivering and estimated delivery:-

select

order\_id,

DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,DAY) AS time\_to\_deliver,

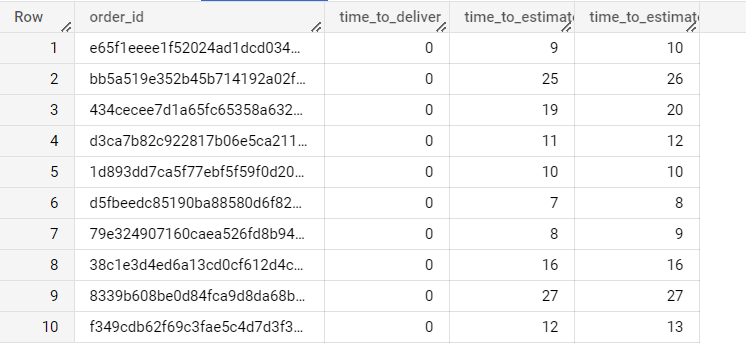
DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,DAY) AS time\_to\_estimate\_deliver,

DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_carrier\_date,DAY) AS time\_to\_estimate\_deliver

from `retail\_data.orders`

where order\_delivered\_customer\_date is not null

order by time\_to\_deliver



2. Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:

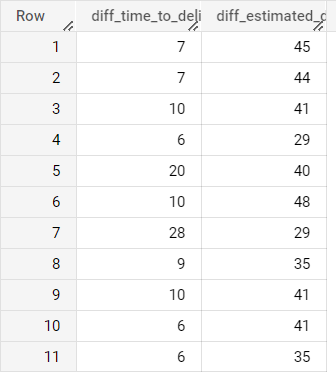
* + time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date
  + diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

select

abs(DATE\_DIFF(order\_purchase\_timestamp,order\_delivered\_customer\_date,day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date,order\_delivered\_customer\_date,day)) as diff\_estimated\_delivery

from `retail\_data.orders`



3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

select

c.customer\_state,

round(avg(oi.freight\_value),2) as mean,

round(Avg(date\_diff(o.order\_delivered\_customer\_date,o.order\_purchase\_timestamp,day)),2) as time\_to\_delivery,

round(avg(date\_diff(o.order\_estimated\_delivery\_date,o.order\_delivered\_carrier\_date,day)),2) as time\_for\_estimated\_delivery

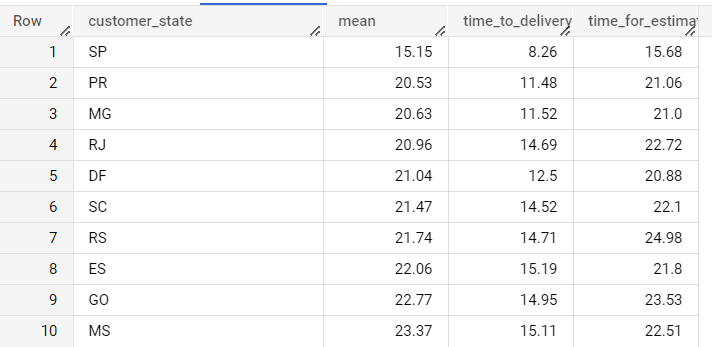
from `retail\_data.customers` as c

 join `retail\_data.orders` as o on c.customer\_id=o.customer\_id

join `retail\_data.order\_items` as oi on oi.order\_id=o.order\_id

group by c.customer\_state

order by mean



4.Sort the data to get the following:

5.Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

HIGHEST AVERAGE FREIGHT VALUE

select c.customer\_state,AVG(oi.freight\_value) as avg\_freight

from `retail\_data.orders` as o

join `retail\_data.customers` as c on c.customer\_id=o.customer\_id

join `retail\_data.order\_items` as oi on oi.order\_id=o.order\_id

group by c.customer\_state

order by avg\_freight desc

limit 5



LOWEST AVERAGE FREIGHT VALUE

select c.customer\_state,AVG(oi.freight\_value) as avg\_freight

from `retail\_data.orders` as o

join `retail\_data.customers` as c on c.customer\_id=o.customer\_id

join `retail\_data.order\_items` as oi on oi.order\_id=o.order\_id

group by c.customer\_state

order by avg\_freight ASC

limit 5



6.Top 5 states with highest/lowest average time to delivery

HIGHEST AVERAGE TIME FOR DELIVERY

select c.customer\_state,AVG(date\_diff(o.order\_delivered\_customer\_date,

o.order\_purchase\_timestamp,day)) as time\_to\_delivery

from `retail\_data.orders` as o

join `retail\_data.customers` as c on c.customer\_id=o.customer\_id

group by c.customer\_state

order by time\_to\_delivery desc

limit 5



LOWEST AVERAGE TIME FOR DELIVERY

select c.customer\_state,AVG(date\_diff(o.order\_delivered\_customer\_date,

o.order\_purchase\_timestamp,day)) as time\_to\_delivery

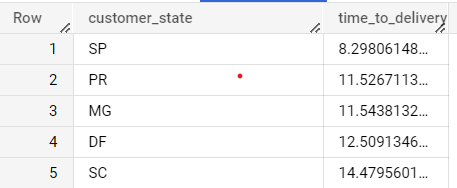
from `retail\_data.orders` as o

join `retail\_data.customers` as c on c.customer\_id=o.customer\_id

group by c.customer\_state

order by time\_to\_delivery desc

limit 5



7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

WITH delivery\_time AS (

  SELECT

    c.customer\_state,

    CASE

      WHEN o.order\_delivered\_customer\_date <= o.order\_estimated\_delivery\_date THEN 'Fast Delivery'

      ELSE 'Not so Fast Delivery'

    END AS delivery\_status,

  FROM `retail\_data.orders` as o join `retail\_data.customers` as c on o.customer\_id=c.customer\_id

  WHERE o.order\_delivered\_customer\_date IS NOT NULL

  GROUP BY c.customer\_state, delivery\_status

),

delivery\_times\_rank AS (

  SELECT

    customer\_state,

    delivery\_status,

    ROW\_NUMBER() OVER (PARTITION BY delivery\_status ) AS rank\_of

  FROM delivery\_time

)

SELECT

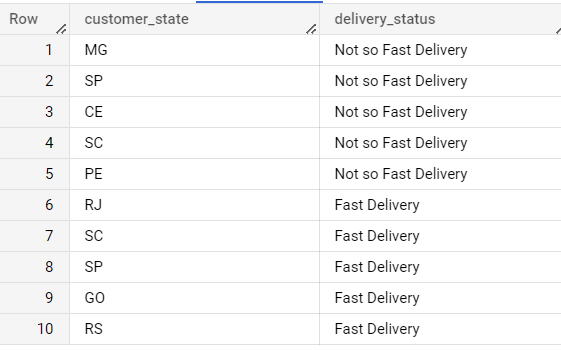
  customer\_state,

  delivery\_status,

FROM delivery\_times\_rank

WHERE rank\_of <= 5

ORDER BY delivery\_status desc



**6. Payment type analysis:**

1. Month over Month count of orders for different payment types:-

SELECT

  p.payment\_type,

  EXTRACT(MONTH from o.order\_purchase\_timestamp) AS month,

  count(\*) AS order\_count

FROM

  `retail\_data.orders` as o

JOIN `retail\_data.payments` p ON o.order\_id = p.order\_id

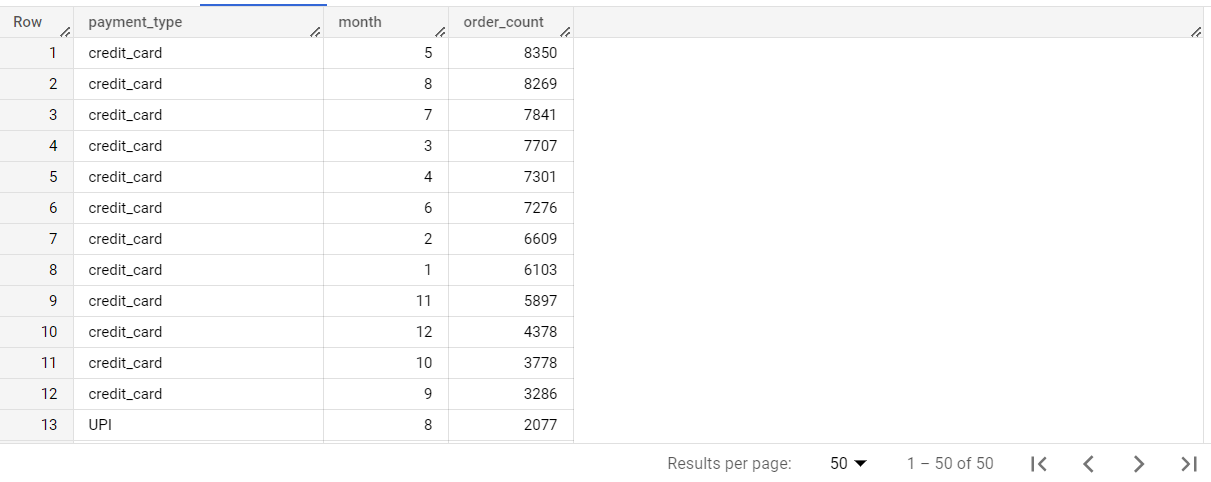
GROUP BY

p.payment\_type,

  month

ORDER BY

  order\_count desc



We can see that orders are more ordered by payment type credit card

1. Count of orders based on the no. of payment installments

select payment\_installments,count(order\_id) as num\_of\_orders

from `retail\_data.payments` as p

group by payment\_installments

