**TARGET Analysis - SQL - Business Case**

**Questions with Answers & Description.**

**Q1.** a). Data type of all columns in the "customers" table.

**Answer:** SHOW COLUMNS FROM `Target.customers`

The above pic is about the Data type of all the columns in “Customers” table, there are 4 String data type and 1 Integer data type with all nullable mode columns.

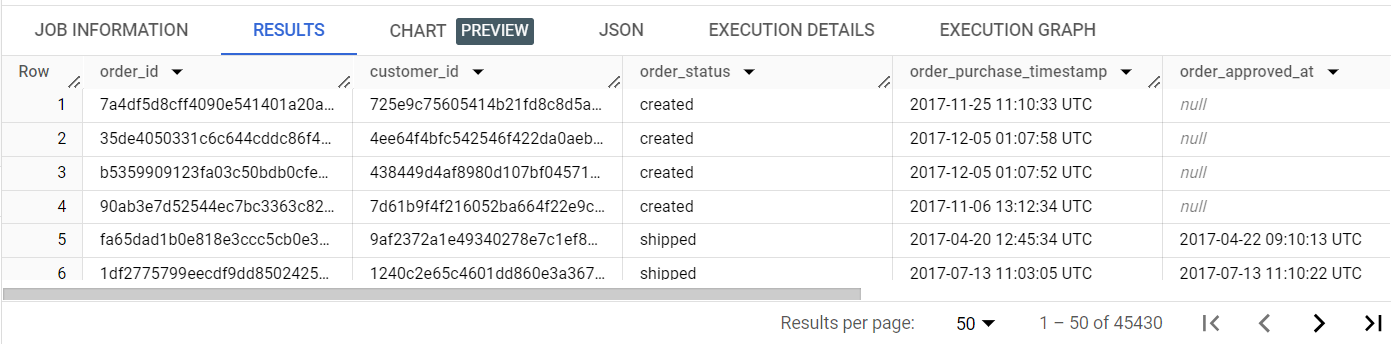
**Q1.** b) Get the time range between which the orders were placed.

Answer:-

SELECT \*

FROM `Target.orders`

WHERE order\_purchase\_timestamp BETWEEN '2016-01-01' AND '2018-01-01'



The query gives the out of order\_purchased between year starting 2016-01-01 AND 2018-01-01, total 45,430 rows

**Q1.** C). Count the Cities & States of customers who ordered during the given period.

Answer:-

SELECT COUNT(c.customer\_city) AS city\_count,

       COUNT(c.customer\_state) AS state\_count,

       MIN(o.order\_purchase\_timestamp) AS start\_purchased\_date,

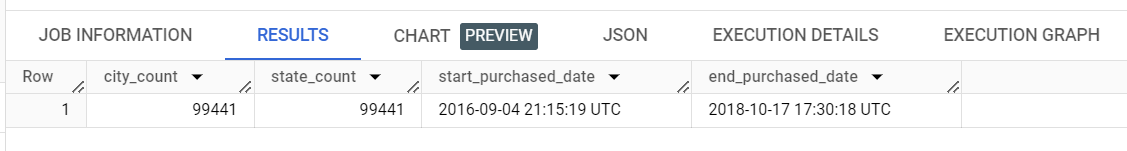
       MAX(o.order\_purchase\_timestamp) AS end\_purchased\_date

FROM `Target.orders` o

INNER JOIN `Target.customers` c

ON o.customer\_id = c.customer\_id

WHERE o.order\_purchase\_timestamp BETWEEN '2016-01-01' AND '2019-01-01';



This query gives the out on count of cities & states of customers who ordered during the given period. I took min date and max purchased date to get starting order date and end order date for where clause.

**Q2.** a). Is there a growing trend in the no. of orders placed over the past years?

**QUERY-**

SELECT DATE(order\_purchase\_timestamp),

       COUNT(order\_id) AS num\_orders

FROM `Target.orders`

GROUP BY DATE(order\_purchase\_timestamp)

ORDER BY 1

OR

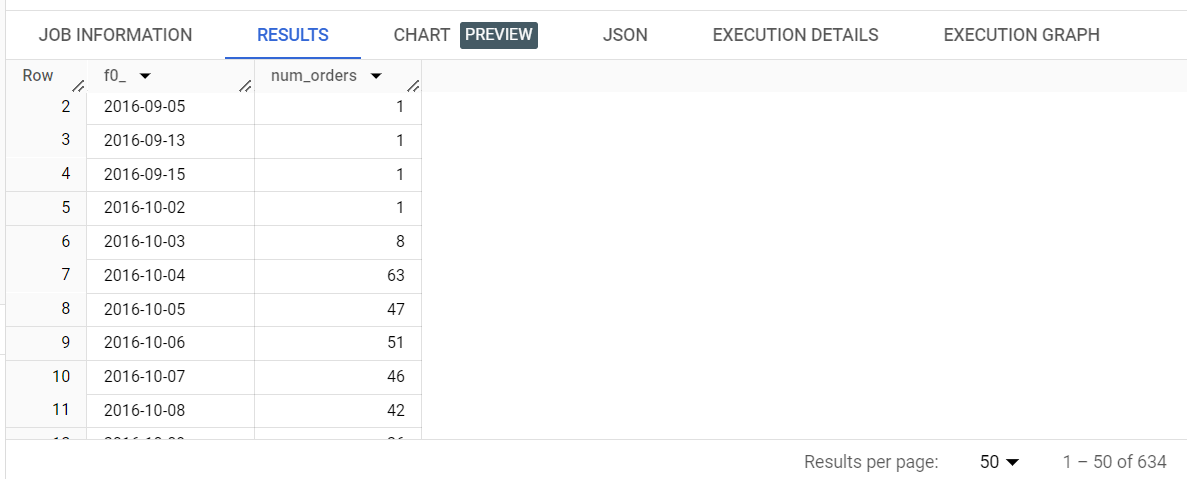
SELECT EXTRACT(YEAR FROM order\_purchase\_timestamp) as order\_over\_years,

       COUNT(order\_id) AS num\_orders

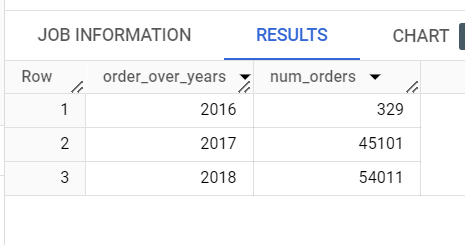
FROM `Target.orders`

GROUP BY order\_over\_years

ORDER BY 1



OR



1. There is a increase and decrease in the num\_orders over years by days the num of orders are fluctuating.
2. But in the order of sum of orders by years are in increasing order.

**Q2.** b). Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

QUERY-

SELECT EXTRACT(MONTH FROM order\_purchase\_timestamp) as orde\_by\_months,

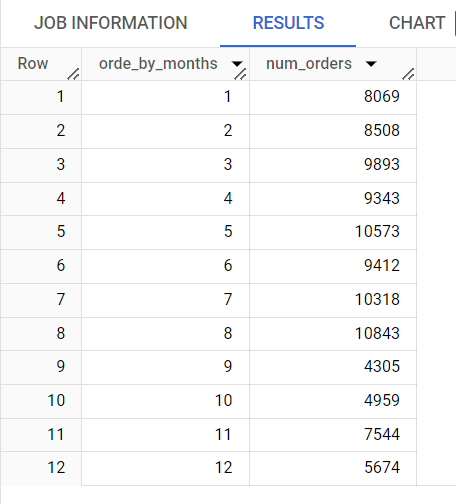
       COUNT(\*) AS num\_orders

FROM `Target.orders`

GROUP BY 1

ORDER BY 1

RESULT-



The orders over months are not stable which is fluctuating over the year, but

Till month 5 may the orders are raising monthly, then there is an sudden drop and rise, and there is an huge drop from 10k to 4k in September.

**Q2.** 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

QUERY-

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 'Evening'

            WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 19 AND 24 THEN 'Night'

       END AS time\_of\_orders,

       COUNT(\*) AS num\_orders

FROM `Target.orders` t

LEFT JOIN `Target.customers` c

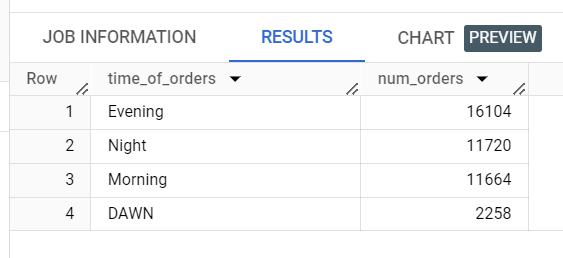
ON t.customer\_id = c.customer\_id

WHERE customer\_state = 'SP'

GROUP BY time\_of\_orders

ORDER BY num\_orders DESC;

Result-



According to the written query the given orders by Brazilian customers during day by time

Is different, the top orders are in evening the orders are more then night, it in next in position after that morning, the lowest stands in dawn with 2258 orders.

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

**Q2. a).** Get the month on month no. of orders placed in each state.

**QUERY-**

SELECT EXTRACT(MONTH FROM order\_purchase\_timestamp) AS purchase\_month,

       EXTRACT(YEAR FROM order\_purchase\_timestamp) AS purchase\_year,

       c.customer\_state,

       COUNT(\*) as num\_order

FROM `Target.orders` t

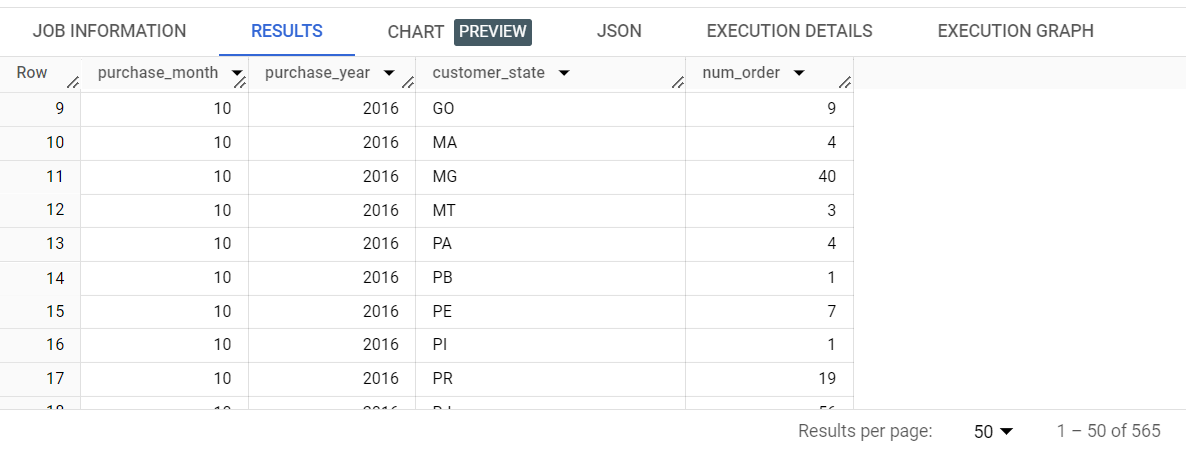
LEFT JOIN `Target.customers` c

ON t.customer\_id = c.customer\_id

GROUP BY purchase\_year, purchase\_month, customer\_state

ORDER BY purchase\_year, purchase\_month, customer\_state;

Result-



Extracted the year and month using “order\_purchase\_timestamp”, from orders table to get the month on month by years in asc order, count the num of orders using count orders, then joined orders table and customers tables using customer\_id, then group the data by year, month, state

To get the desired result.

**Q2. b).** How are the customers distributed across all the states?

**QUERY-**

SELECT customer\_state,

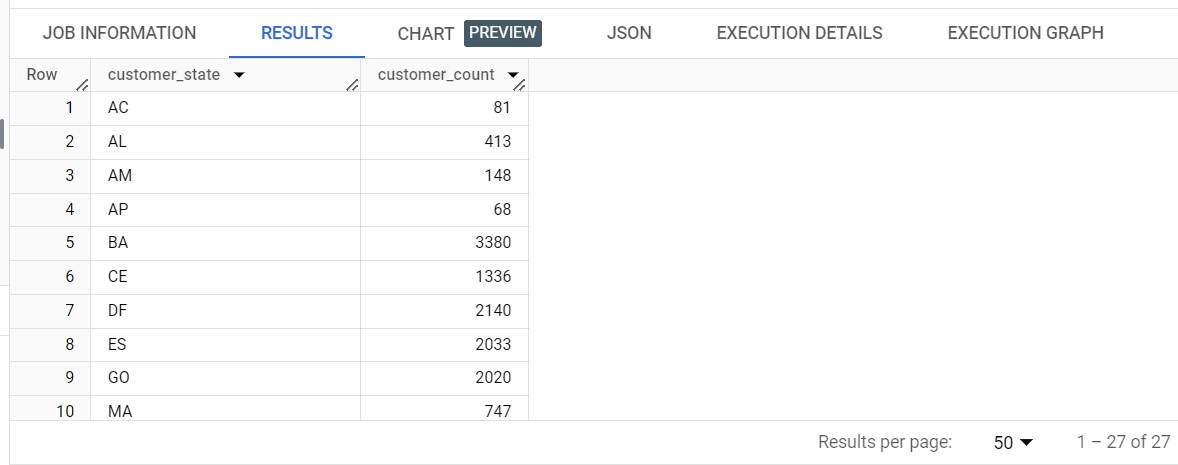
       COUNT(\*) as customer\_count

FROM `Target.customers`

GROUP BY 1

ORDER BY 1

RESULT-



This query gives us the customer distribution across all the states to see the num of customers in different state.

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

**Q3. 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.

**QUERY-**

SELECT ROUND((

        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 ELSE 0 END) -

        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 ELSE 0 END )) /

        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 ELSE 0 END),4)\*100 AS percent\_increase

FROM `Target.orders` o

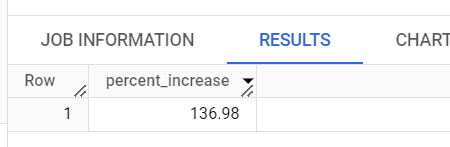
LEFT JOIN `Target.payments` p

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

ORDER BY 1

RESULT-



**Q3. b).** Calculate the Total & Average value of order price for each state.

**QUERY-**

SELECT

       ROUND(SUM(payment\_value),2) AS total\_order\_value,

       ROUND(AVG(payment\_value),2) AS avg\_order\_value,

       c.customer\_state

FROM `Target.payments` p

INNER JOIN

          `Target.orders` o ON p.order\_id = o.order\_id

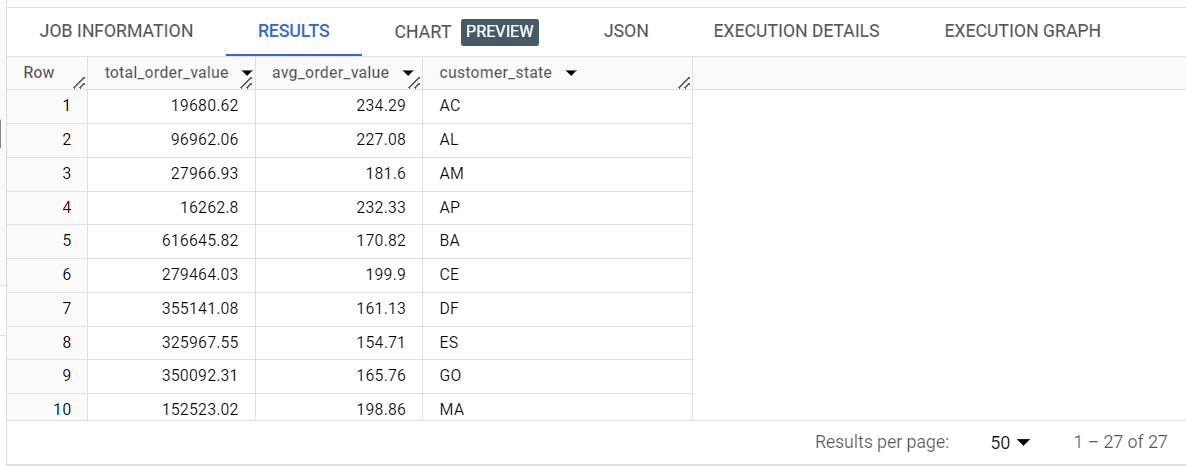
INNER JOIN

          `Target.customers` c ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state ASC;

RESULT-



This query give the total and average value of order price placed in each state

**Q3. c).** Calculate the Total & Average value of order freight for each state.

**Query-**

SELECT

       ROUND(SUM(p.payment\_value\*0.10),2) AS total\_order\_value,

       ROUND(AVG(p.payment\_value\*0.10),2) AS avg\_order\_value,

       c.customer\_state

FROM `Target.payments` p

INNER JOIN

          `Target.orders` o ON p.order\_id = o.order\_id

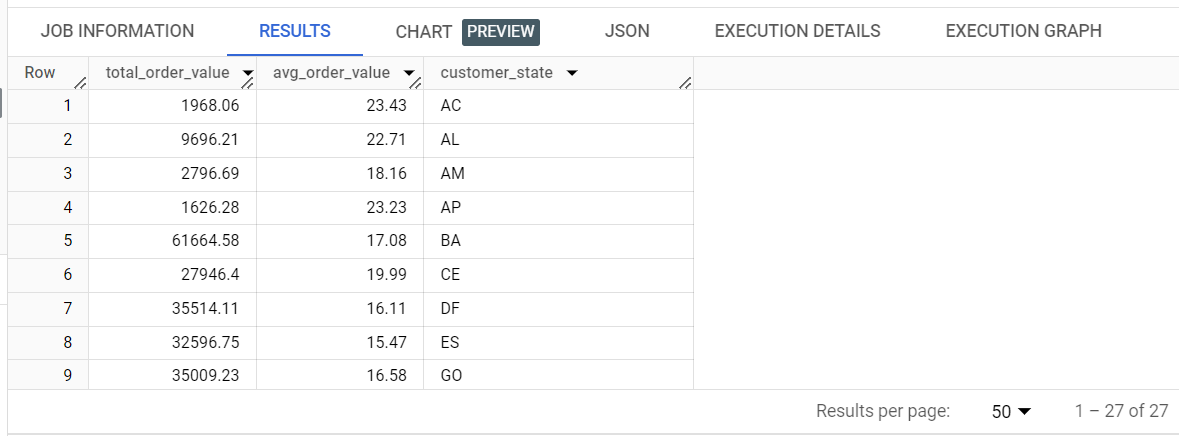
INNER JOIN

          `Target.customers` c ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state ASC;

RESULT-



As I don’t have a freight value column I have used a factor of 0.10 to estimate the freight value.

It is a rough freight value on each state

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

**Q4. 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\_delivered\_customer\_date - order\_estimated\_delivery\_date

**QUERY-**

SELECT

    order\_id,

    order\_purchase\_timestamp,

    order\_estimated\_delivery\_date,

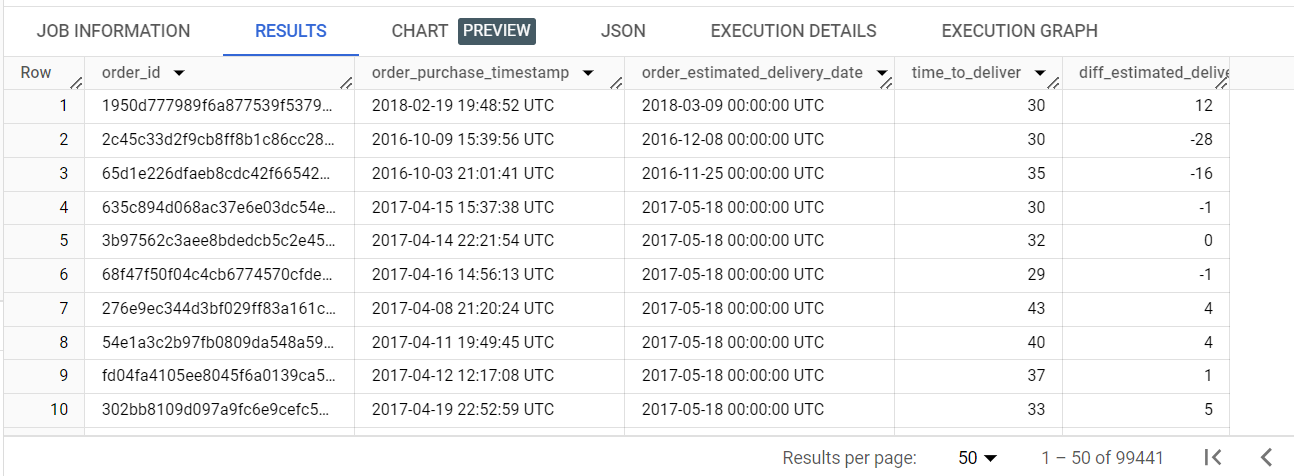
    DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY) AS time\_to\_deliver,

    DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date, DAY) AS diff\_estimated\_delivery

FROM

    `Target.orders`;

RESULT-



This query give the output of date difference delivery date and purchase date

**Q4. b).** Find out the top 5 states with the highest & lowest average freight value.

**QUERY-**

SELECT

       ROUND(SUM(p.payment\_value\*0.10),2) AS total\_order\_value,

       ROUND(AVG(p.payment\_value\*0.10),2) AS avg\_order\_value,

       c.customer\_state

FROM `Target.payments` p

INNER JOIN

          `Target.orders` o ON p.order\_id = o.order\_id

INNER JOIN

          `Target.customers` c ON o.customer\_id = c.customer\_id

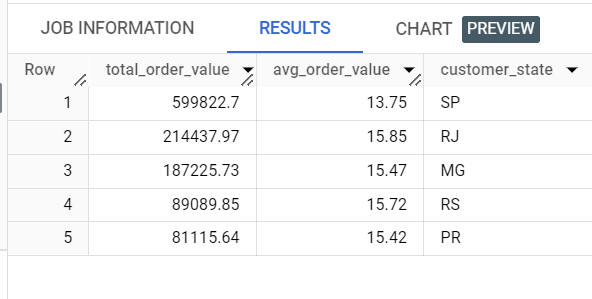
GROUP BY c.customer\_state

ORDER BY total\_order\_value DESC,

         avg\_order\_value ASC

LIMIT 5;

RESULT-



I tried to get the total and average freight value

**Q4. c).** Find out the top 5 states with the highest & lowest average delivery time.

**QUERY-**

SELECT

    AVG(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY)) AS avg\_time\_to\_deliver,

    AVG(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_estimated\_delivery\_date, DAY)) AS avg\_diff\_estimated\_delivery,

    c.customer\_state

FROM

    `Target.payments` p

INNER JOIN

    `Target.orders` o ON p.order\_id = o.order\_id

INNER JOIN

    `Target.customers` c ON o.customer\_id = c.customer\_id

GROUP BY

    c.customer\_state

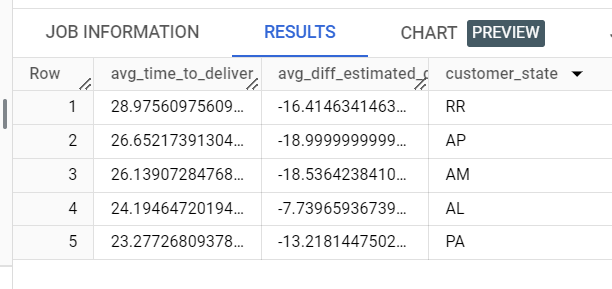
ORDER BY

    avg\_time\_to\_deliver DESC,

    avg\_diff\_estimated\_delivery ASC

LIMIT 5;

RESULT-



I tried to get the highest avg of time to deliver and lowest avg of estimated delivery date diff

**Q4. 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.

QUERY-

SELECT

    ROUND(AVG(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY)),0) AS avg\_time\_to\_deliver,

    c.customer\_state

FROM

    `Target.payments` p

INNER JOIN

    `Target.orders` o ON p.order\_id = o.order\_id

INNER JOIN

    `Target.customers` c ON o.customer\_id = c.customer\_id

GROUP BY

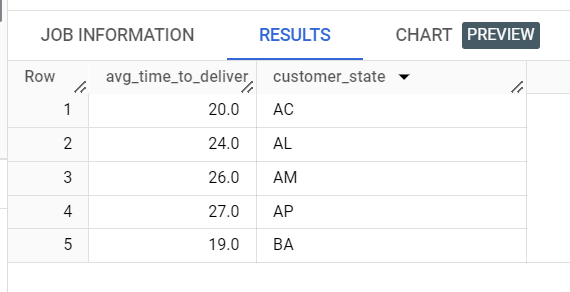
    c.customer\_state

ORDER BY

     c.customer\_state

LIMIT 5;

RESULT-



I tried to get the average day to get delivery

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

**Q5. a).** Find the month on month no. of orders placed using different payment types.

QUERY- SELECT

     EXTRACT(MONTH FROM o.order\_purchase\_timestamp) as purchase\_month,

     COUNT(\*) as count\_of\_orders,

     p.payment\_type

FROM `Target.orders` o

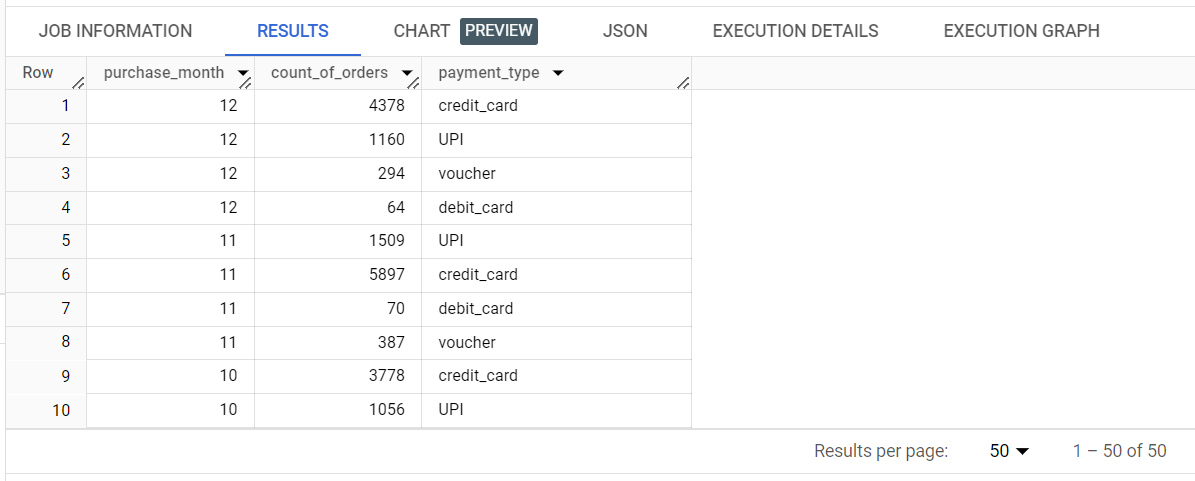
INNER JOIN `Target.payments` p

ON o.order\_id = p.order\_id

GROUP BY purchase\_month, p.payment\_type

ORDER BY purchase\_month DESC

RESULT-



**Q5. b).** Find the no. of orders placed on the basis of the payment installments that have been paid.

**QUERY-**

SELECT payment\_installments,

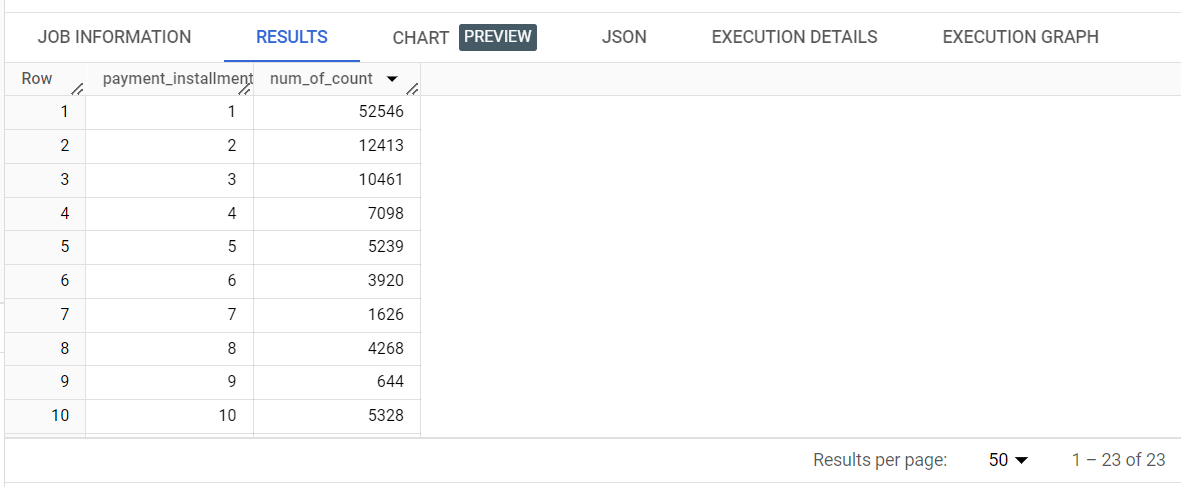
       count(\*) AS num\_of\_count

FROM `Target.payments`

WHERE payment\_installments >= 1

GROUP BY payment\_installments;

**RESULT-**



This query gives the query based on payment instalments and count of instalments paid in that payment