**Target SQL**

**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. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
2. **Data type of all columns in the "customers" table.**

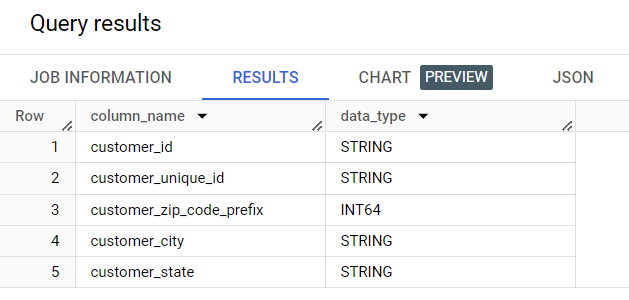
**Answer: SQL Query**

SELECT column\_name, data\_type

FROM

`target-sql-case-407005.target\_sql.INFORMATION\_SCHEMA.COLUMNS`

WHERE table\_name = 'customers';



**INSIGHTS:**

It is observed that 4 out of 5 columns have STRING as their datatype and the remaining 1 column has INTEGER datatype

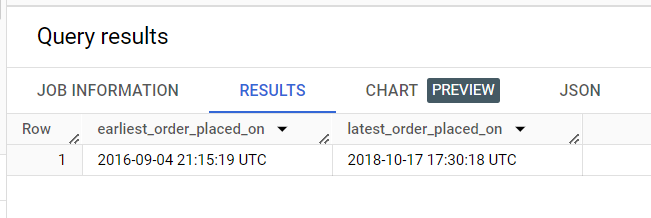
1. **Get the time range between which the orders were placed**

**Answer: SQL Query**

SELECT MIN(order\_purchase\_timestamp) AS earliest\_order\_placed\_on,

      MAX(order\_purchase\_timestamp) AS latest\_order\_placed\_on

FROM `target\_sql.orders`

****

**INSIGHTS:**

It can be seen that the first order was placed on September 4, 2016 while the latest order was placed on October 17, 2018. Hence September 4, 2016 to October 17, 2018 is the time range within which orders were placed.

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

**Answer: SQL Query**

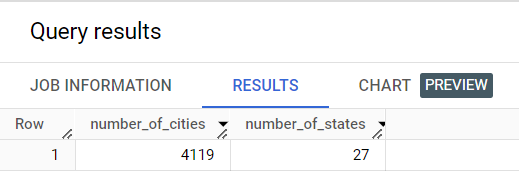
SELECT COUNT(DISTINCT c.customer\_city) AS number\_of\_cities,

COUNT(DISTINCT c.customer\_state) AS number\_of\_states

FROM `target\_sql.orders` o

JOIN `target\_sql.customers` c

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



**INSIGHTS:**

It can be observed that customers from 4119 different cities which are in 27 different states placed orders between the given time frame i.e. from September 4, 2016 to October 17, 2018.

1. **In-depth Exploration:**
2. **Is there a growing trend in the no. of orders placed over the past years?**

**Answer: SQL Query**

SELECT EXTRACT(year FROM order\_purchase\_timestamp) AS year,

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

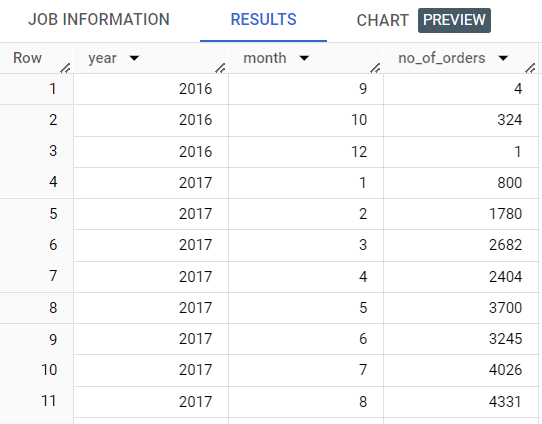
        COUNT(1) AS no\_of\_orders

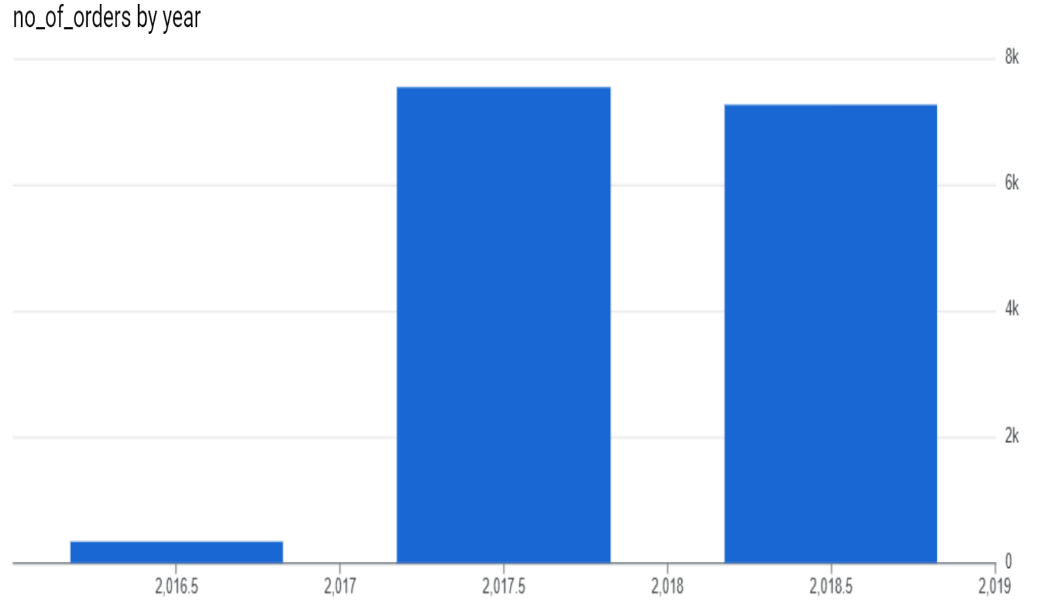
FROM `target\_sql.orders`

GROUP BY year, month

ORDER BY year, month

**Query Output**

****

**INSIGHTS:**

It can be observed that there is a growing trend in terms of number of orders placed. This trend can be observed from December 2016 to March 2018. The numbers of orders remain decrease hence with a slight increase noticed in August 2018. The number of orders slump to only 16 in September 2018 and further to 4 till October 17, 2018.

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

**Answer: SQL Query**

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

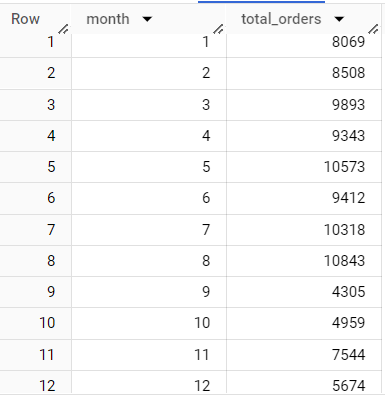
      COUNT(order\_id) AS total\_orders

FROM `target\_sql.orders`

GROUP BY month

ORDER BY month;

**Query Output**

****

**INSIGHTS:**



**x axis: months  
y axis: no of orders**

There is a seasonality in terms of number of orders being placed. The number of orders placed between March to August are significantly greater than the number of orders placed between September to February. The sales are quite low in the last 4 months of an year.

1. **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

    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 'Afternoon'

        WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 19 AND 23 THEN 'Night'

            END AS time\_of\_day,

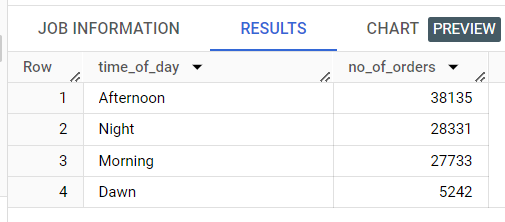
    COUNT(\*) AS no\_of\_orders

FROM `target\_sql.orders`

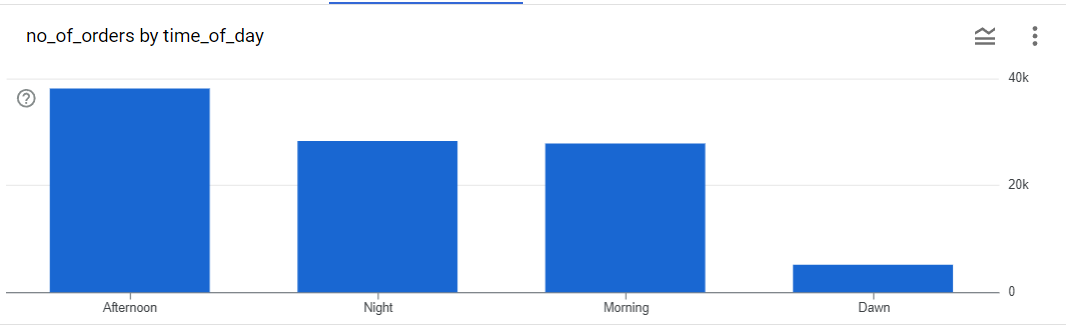
GROUP BY time\_of\_day

ORDER BY no\_of\_orders DESC;

**Query Output**



**INSIGHTS:**

****

It can be derived from the above chart that majority of orders are placed by customers during the afternoon, spending on ad and marketing campaigns can be done accordingly. The number of orders during night and morning are similar while the least number of orders are placed during first 6 hours of the day i.e. dawn in this case.

1. **Evolution of E-commerce orders in the Brazil region:**
2. **Get the month on month no. of orders placed in each state.**

**Answer: SQL Query**

WITH MonthlyOrders AS (

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

        c.customer\_state,

        COUNT(\*) AS num\_orders

    FROM `target\_sql.orders` o

    JOIN `target\_sql.customers` c

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

    GROUP BY c.customer\_state, order\_month

)

SELECT order\_month,

    customer\_state,

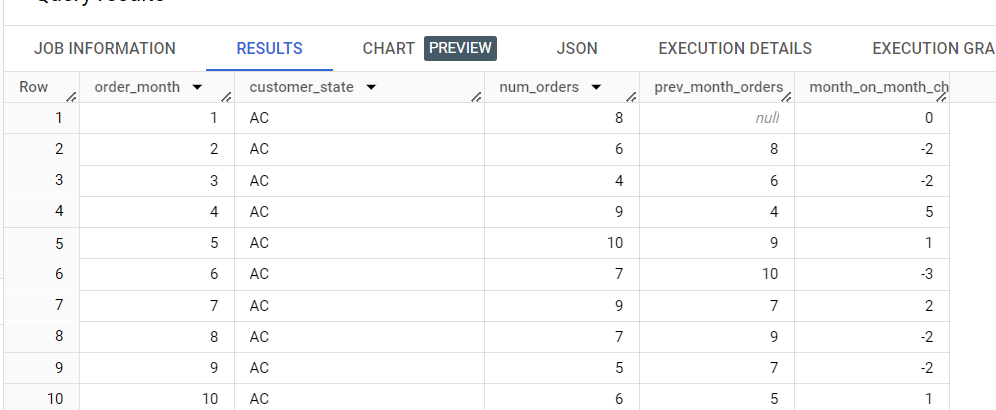
    num\_orders,

    LAG(num\_orders) OVER (PARTITION BY customer\_state ORDER BY order\_month) AS prev\_month\_orders,

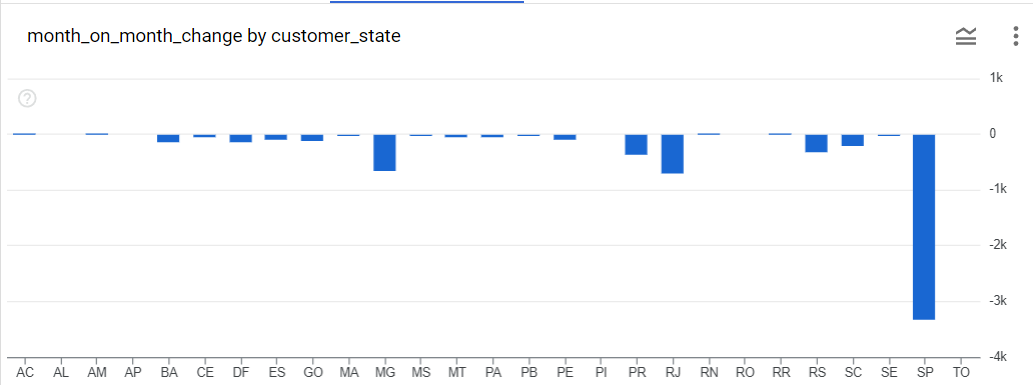
    COALESCE(num\_orders - LAG(num\_orders) OVER (PARTITION BY customer\_state ORDER BY order\_month), 0) AS month\_on\_month\_change

FROM MonthlyOrders

ORDER BY customer\_state, order\_month

**Query Output**

**INSIGHTS:**

****

From the above chart it can be seen that see maximum month on month variation in SP state. The states MG, PR, RS, SC and RJ also have noticeable variation month wise while other states do not display such variation in terms of the number of orders month wise.

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

**Answer: SQL Query**

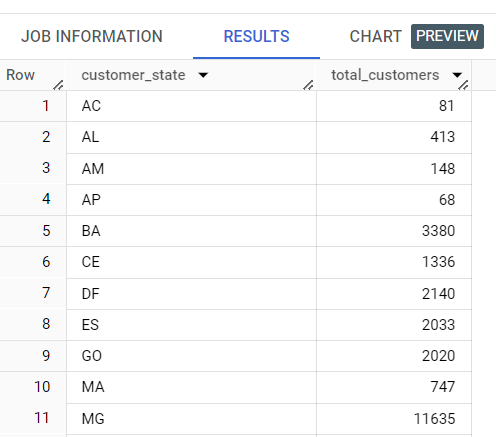
SELECT customer\_state, COUNT(\*) AS total\_customers

FROM `target\_sql.customers`

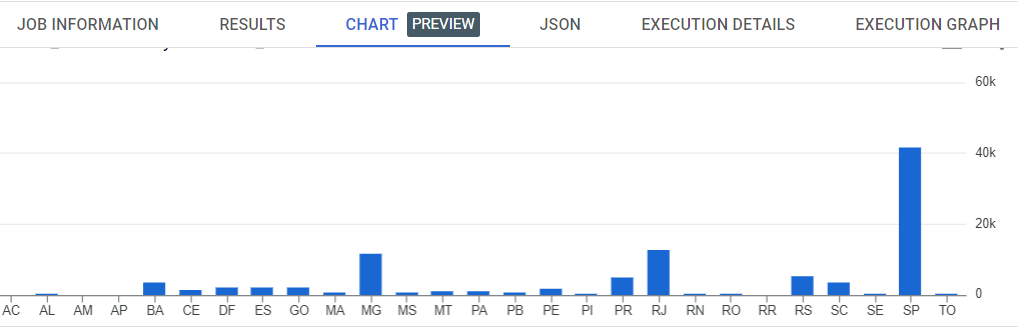
GROUP BY customer\_state

ORDER BY customer\_state;

**Query Output**

****

**INSIGHTS:**

****

**x axis: states  
 y axis: no of customers**

From the above result and chart, it can be seen that most of the customers reside in MG, PR, RS, SC and RJ. This can also explain the variation in month on month orders in the previous question which is witnessed in these states

1. **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**
2. **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.**

**Answer: SQL Query**

WITH payment\_date AS (

  SELECT EXTRACT (DATE FROM o.order\_purchase\_timestamp) AS date\_of\_order,

    EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS year\_of\_order,

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

    p.payment\_value

  FROM `target\_sql.orders` o

  JOIN `target\_sql.payments` p

  ON p.order\_id = o.order\_id

),

order\_value AS (

  SELECT

    FORMAT\_DATE('%Y', date\_of\_order) AS order\_year,

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

  FROM payment\_date

  WHERE month\_of\_order BETWEEN 1 AND 8 AND year\_of\_order BETWEEN 2017 AND 2018

  GROUP BY 1

  ORDER BY 1

),

order\_year\_value\_lag AS (

  SELECT \*,

    LAG(total\_order\_value) OVER (ORDER BY order\_year) AS prev\_year\_sale

  FROM order\_value

  ORDER BY order\_year

)

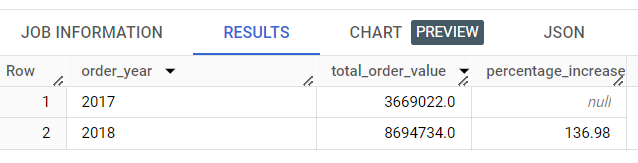
SELECT order\_year,

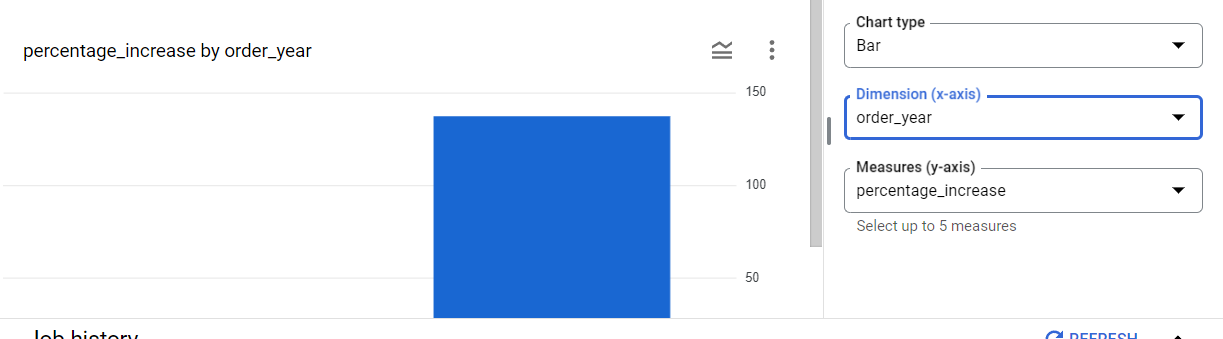
  total\_order\_value,

  ROUND((((total\_order\_value - prev\_year\_sale) / prev\_year\_sale) \* 100), 2) AS percentage\_increase

FROM order\_year\_value\_lag

**Query Output**

****

**  
INSIGHTS:**

It can be observed that the number of orders placed between January to August in the year 2018 are 136% more than those placed between January to August in 2017.

This indicates the increasing trend over the years between the months January to August. There is also a noticeable seasonality in terms of number of orders placed.

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

**Answer: SQL Query**

SELECT c.customer\_state AS state,

    SUM(oi.price) AS total\_order\_price,

    AVG(oi.price) AS average\_order\_price

FROM `target\_sql.order\_items` oi

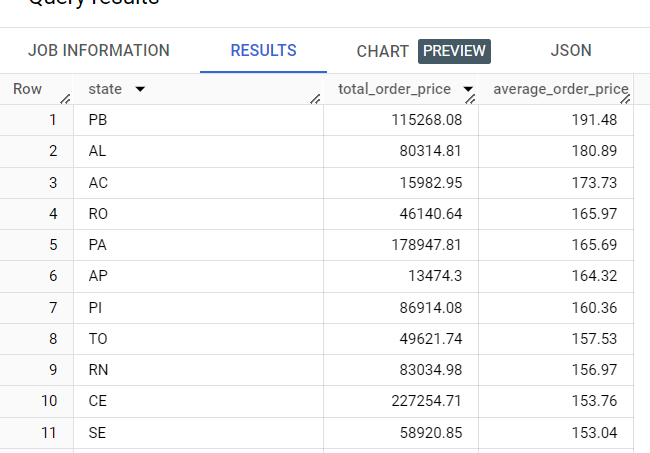
JOIN `target\_sql.orders` o ON oi.order\_id = o.order\_id

JOIN `target\_sql.customers` c ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

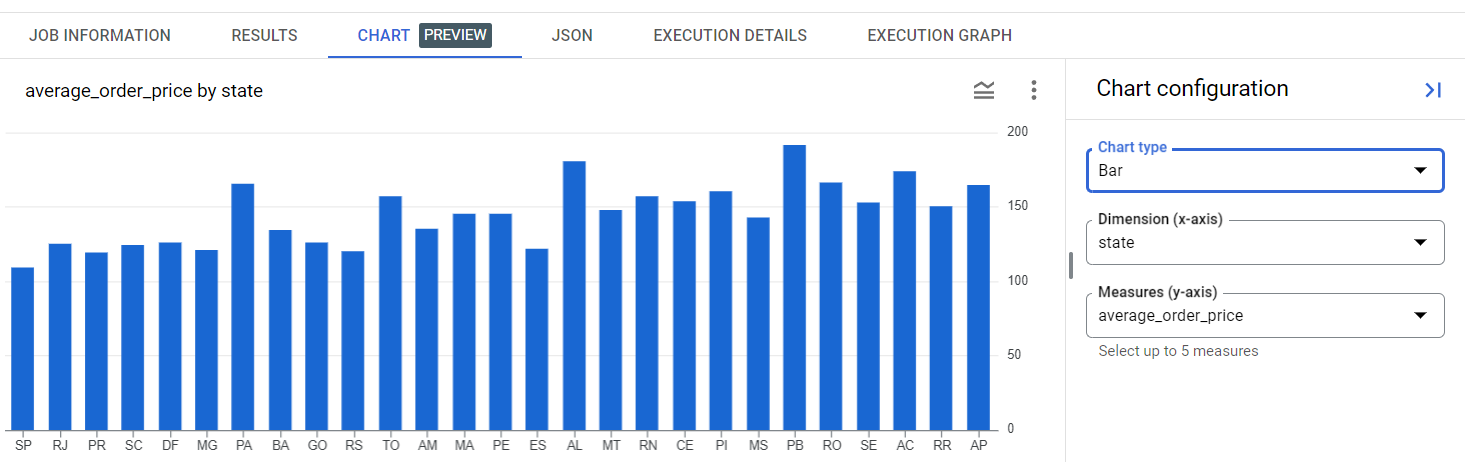
ORDER BY average\_order\_price DESC

**Query Output**

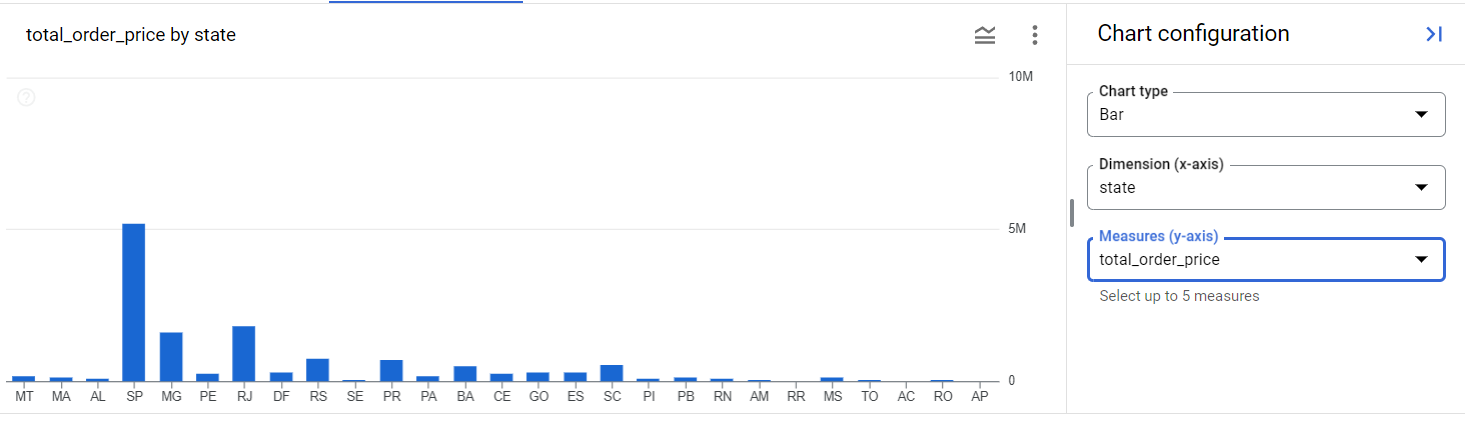
****

**INSIGHTS:**

Above is the list of states with total and average order price for each state ordered by average order price in desceding order

****

It is observed that states PB, AL, AC, RO and PA have the highest average order price



States SP, RJ, MG, RS, PR are the top 5 states with highest total order price.

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

**Answer: SQL Query**

SELECT c.customer\_state AS state,

    ROUND(SUM(oi.freight\_value),2) AS total\_freight\_value,

    ROUND(AVG(oi.freight\_value),2) AS average\_freight\_value

FROM `target\_sql.order\_items` oi

JOIN `target\_sql.orders` o ON oi.order\_id = o.order\_id

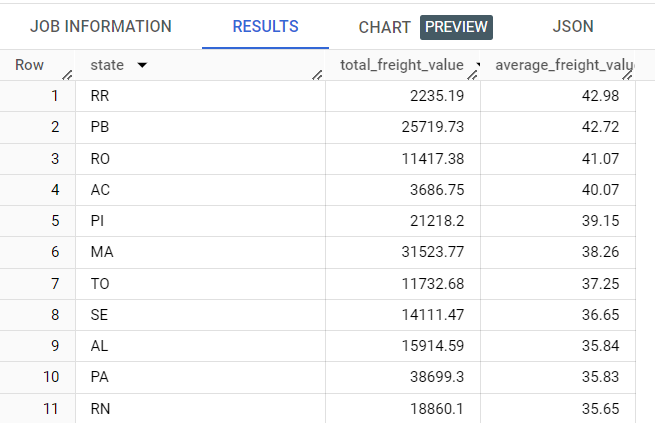
JOIN `target\_sql.customers` c ON o.customer\_id = c.customer\_id

GROUP BY c.customer\_state

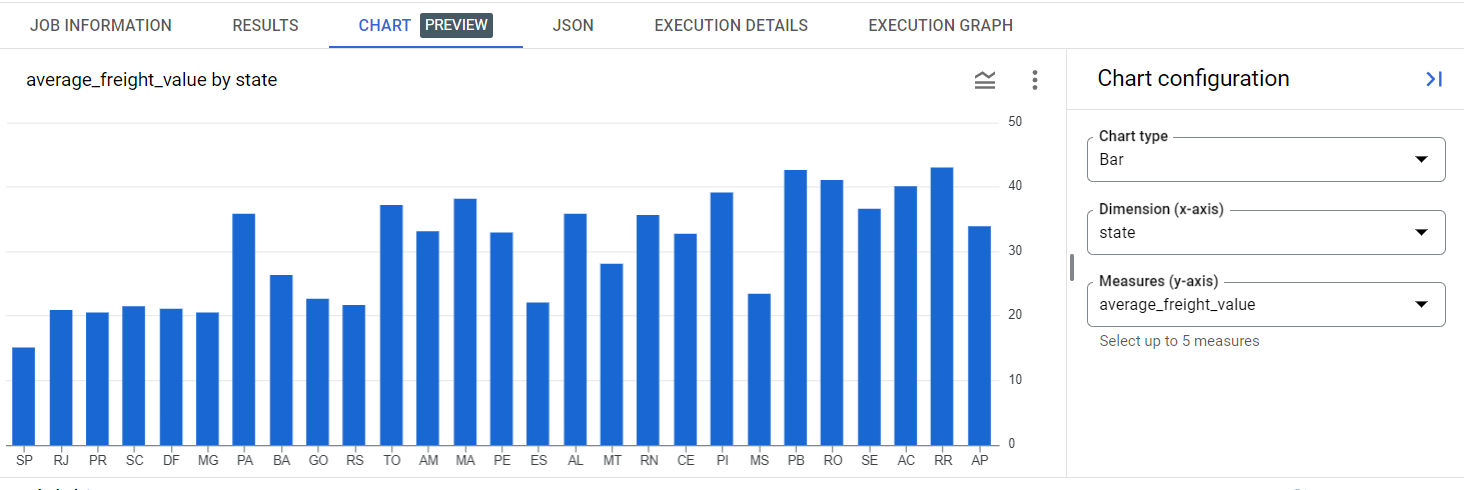
ORDER BY average\_freight\_value DESC

**Query Output**

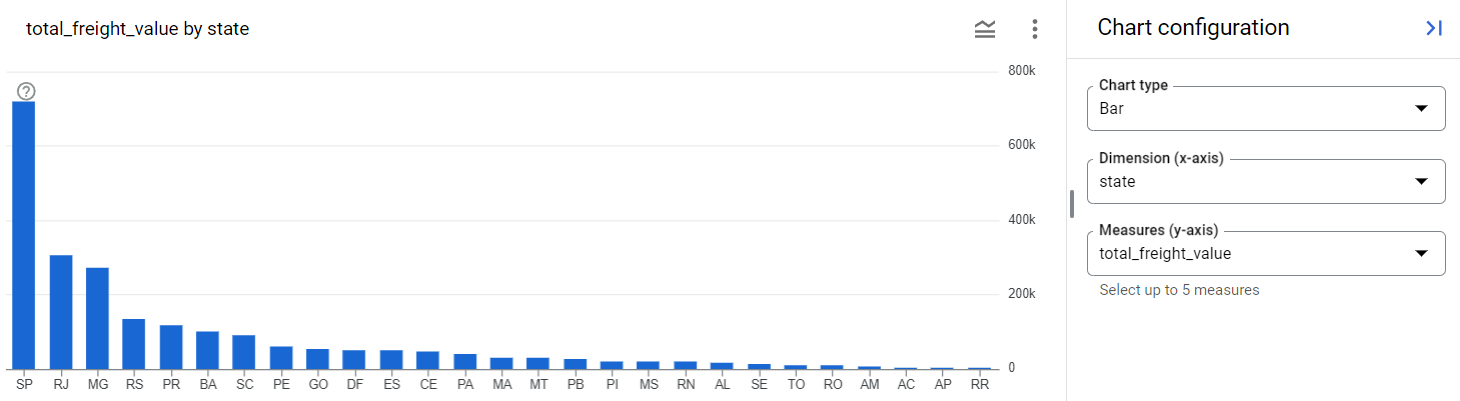
Following is the list of states with total and average freight value ordered by average freight value



**INSIGHTS:**

****

It is observed that states RR, PB, RO, AC and PI have highest average freight value.



States SP, RJ, MG, RS and PR have highest total freight value.

1. **Analysis based on sales, freight and delivery time.**
2. **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 d elivery date using the given formula:**

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

**Answer: SQL Query**

SELECT order\_id,

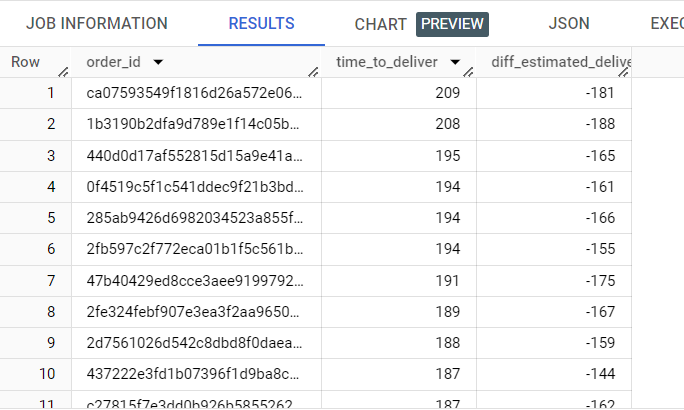
  IFNULL(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY), NULL) AS delivery\_time,

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

ORDER BY time\_to\_deliver DESC

FROM `target\_sql.orders`

**Query Output**



**INSIGHTS:**

Above is the list of orders with time taken to deliver order and difference between actual and estimated date of order delivery in days. The output has been ordered by time taken to deliver order in descending order

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

**Answer: SQL Query**

WITH average\_freight\_per\_state AS (

  SELECT s.seller\_state AS State,

    ROUNd(AVG(oi.freight\_value),2) AS Average\_Freight\_Value

  FROM `target\_sql.order\_items` oi

  JOIN `target\_sql.sellers` s ON oi.seller\_id = s.seller\_id

  GROUP BY s.seller\_state

)

SELECT State,

  Average\_Freight\_Value

FROM (

  SELECT State,

    Average\_Freight\_Value,

    ROW\_NUMBER() OVER (ORDER BY Average\_Freight\_Value DESC) AS rank\_high

  FROM average\_freight\_per\_state

)

WHERE rank\_high <= 5

UNION ALL

SELECT State,

  Average\_Freight\_Value

FROM (

  SELECT State,

    Average\_Freight\_Value,

    ROW\_NUMBER() OVER (ORDER BY Average\_Freight\_Value ASC) AS rank\_low

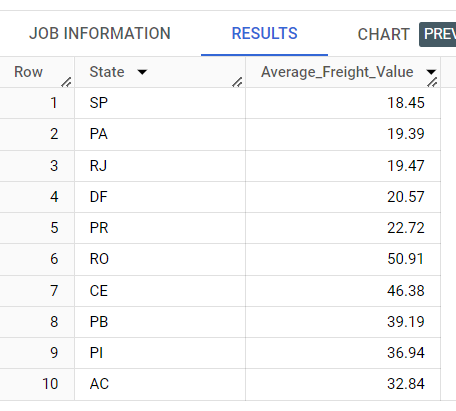
  FROM

    average\_freight\_per\_state

)

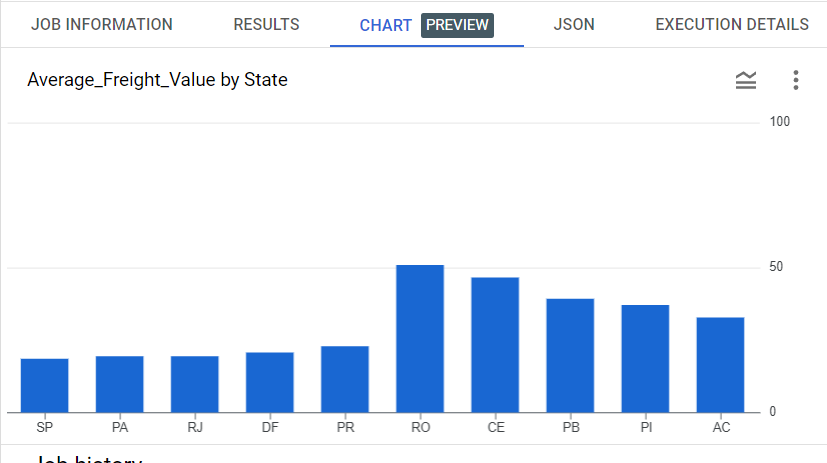
WHERE rank\_low <= 5;

**Query Output**

****

**INSIGHTS:**

Here are the states with 5 states (top 5 in output) with lowest average freight values in ascending order and 5 states(bottom 5 in output) with highest average freight values in descending order

****

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

**Answer: SQL Query**

WITH DeliveryTimes AS (

  SELECT c.customer\_state,

    TIMESTAMP\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY) AS delivery\_time

  FROM `target\_sql.orders` o

  JOIN `target\_sql.customers` c ON o.customer\_id = c.customer\_id

)

SELECT \* FROM (

  SELECT customer\_state,

    ROUND(AVG(delivery\_time),2) AS average\_delivery\_time

  FROM DeliveryTimes

  GROUP BY customer\_state

  ORDER BY average\_delivery\_time DESC

  LIMIT 5

) top\_5\_states

UNION ALL

SELECT \* FROM (

  SELECT customer\_state,

    ROUND(AVG(delivery\_time),2) AS average\_delivery\_time

  FROM DeliveryTimes

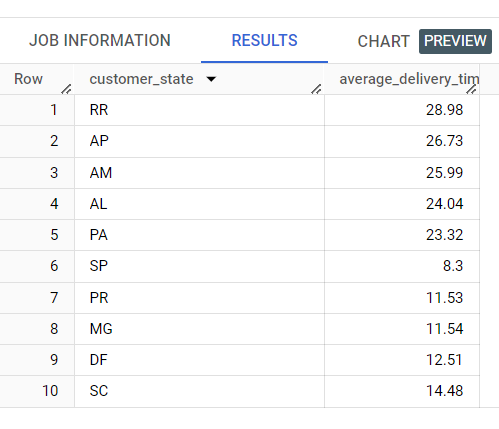
  GROUP BY customer\_state

  ORDER BY average\_delivery\_time ASC

  LIMIT 5

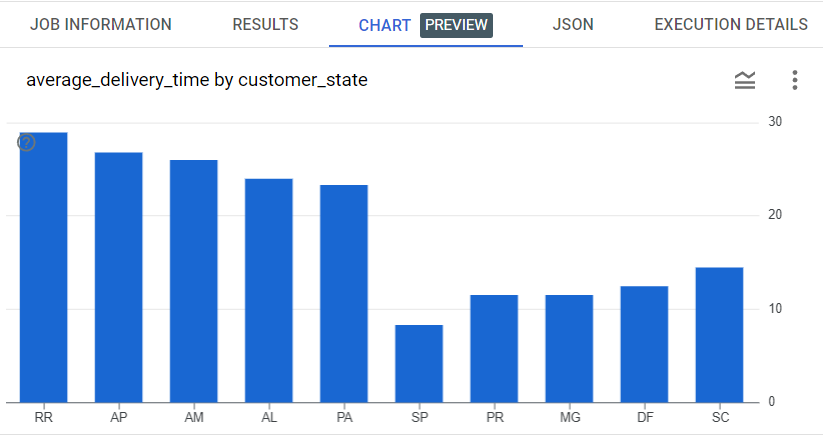
) bottom\_5\_states

**Query Output**

****

**INSIGHTS:**

Here are the states with 5 states(top 5 in output) with highest average delivery time in descending order and 5 states(bottom 5 in output) with lowest average delivery time in ascending order

****

1. **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.**

**Answer: SQL Query**

WITH OrderDeliveryTimes AS (

  SELECT c.customer\_state,

    AVG(TIMESTAMP\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS avg\_days\_purchase\_to\_delivery,

    AVG(TIMESTAMP\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, DAY)) AS avg\_days\_delivery\_to\_estimated

  FROM `target\_sql.orders` o

  JOIN `target\_sql.customers` c

  ON o.customer\_id = c.customer\_id

  WHERE o.order\_status = 'delivered'

  GROUP BY c.customer\_state

)

SELECT customer\_state,

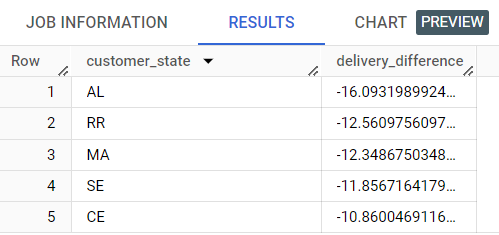
  (avg\_days\_delivery\_to\_estimated - avg\_days\_purchase\_to\_delivery) AS delivery\_difference

FROM OrderDeliveryTimes

ORDER BY delivery\_difference ASC

LIMIT 5;

**Query Output**

****

**INSIGHTS:**

States AL, RR, MA, SE and CE are the top states where delivery is fast when compared with the estimated date of delivery

1. **Analysis based on the payments:**
2. **Find the month on month no. of orders placed using different payment types.**

**Answer: SQL Query**

SELECT

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

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

  p.payment\_type,

  COUNT(DISTINCT o.order\_id) AS num\_orders

FROM `target\_sql.payments` p

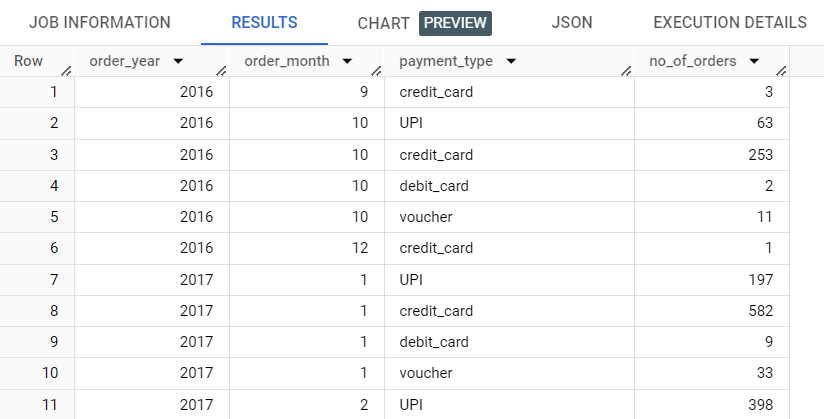
JOIN `target\_sql.orders` o

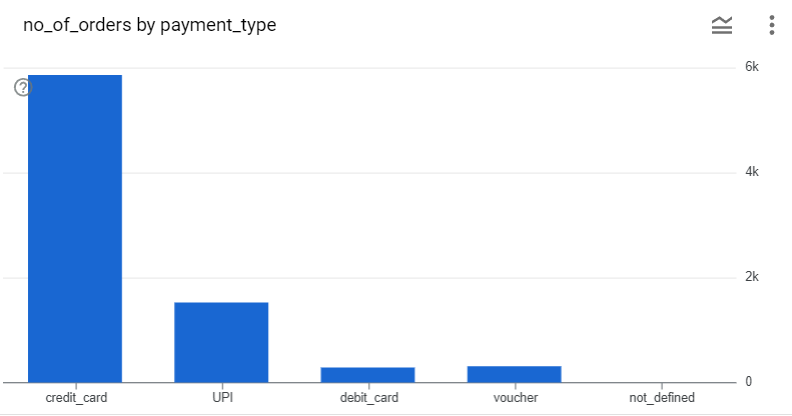
ON p.order\_id = o.order\_id

GROUP BY order\_year, order\_month, p.payment\_type

ORDER BY order\_year, order\_month, p.payment\_type

**Query Output**

****

**INSIGHTS:**

From the above result it can be observed that most orders have been placed using credit card, followed by UPI and voucher. Least no of orders purchased through debit card

**2nd Approach(Question 6A)**

**Answer: SQL Query**

WITH MonthlyOrders AS (

  SELECT EXTRACT(YEAR FROM TIMESTAMP(order\_purchase\_timestamp)) AS year,

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

    payment\_type,

    COUNT(DISTINCT o.order\_id) AS num\_orders

  FROM `target\_sql.orders` o

  JOIN `target\_sql.payments` p ON o.order\_id = p.order\_id

  GROUP BY year, month,payment\_type

)

SELECT year,

  month,

  payment\_type,

  num\_orders,

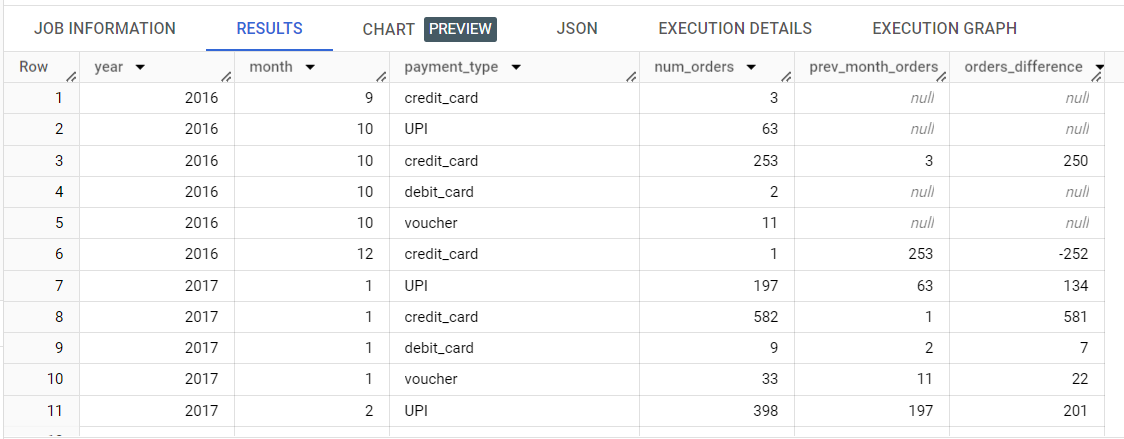
  LAG(num\_orders) OVER (PARTITION BY payment\_type ORDER BY year, month) AS prev\_month\_orders,

  num\_orders - LAG(num\_orders) OVER (PARTITION BY payment\_type ORDER BY year, month) AS orders\_difference

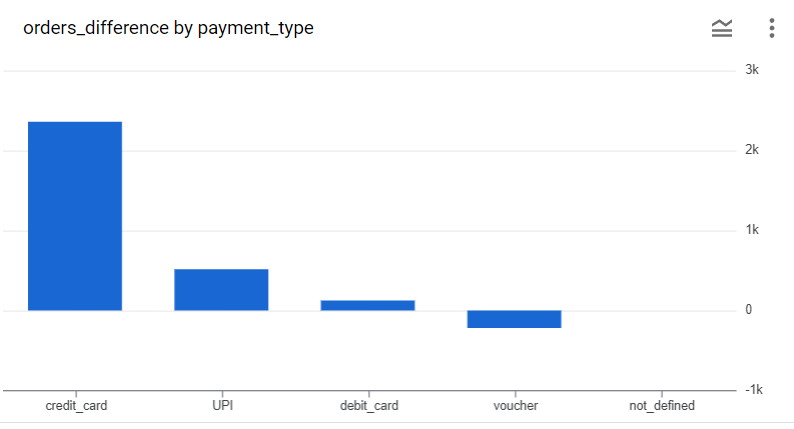
FROM MonthlyOrders

ORDER BY year, month, payment\_type

**Query Output**



**INSIGHTS:**

****

From the above chart we can see that there is a lot of variation in the transcations made using credit cards month on month which may be due to various credit card offers and the payment through vouchers has decreased.

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

**Answer: SQL Query**

SELECT \*,SUM(num\_orders) OVER() AS total\_installment\_orders

FROM(

SELECT payment\_installments, COUNT(DISTINCT order\_id) AS num\_orders

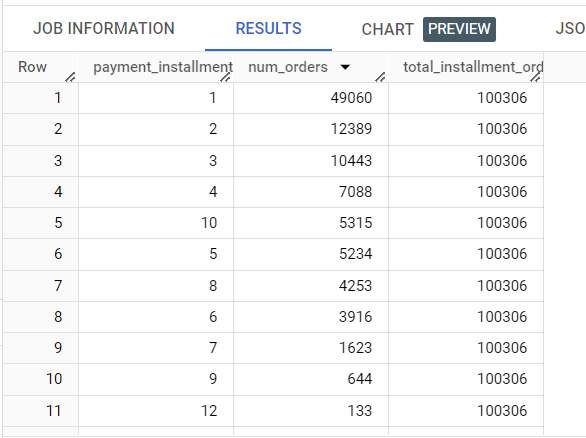
FROM `target\_sql.payments`

GROUP BY payment\_installments

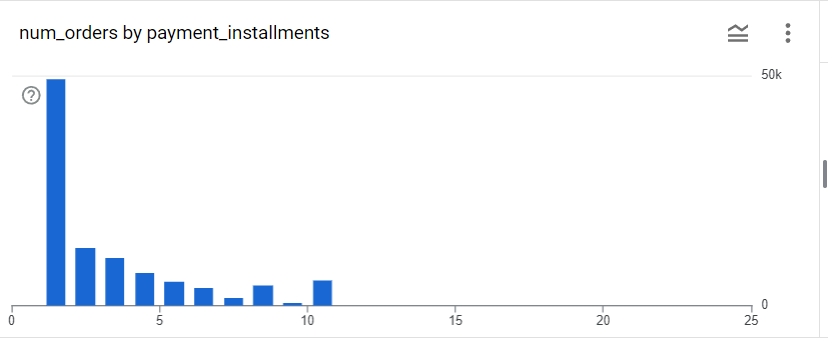
HAVING payment\_installments >= 1)

ORDER BY num\_orders DESC

**Query Output**

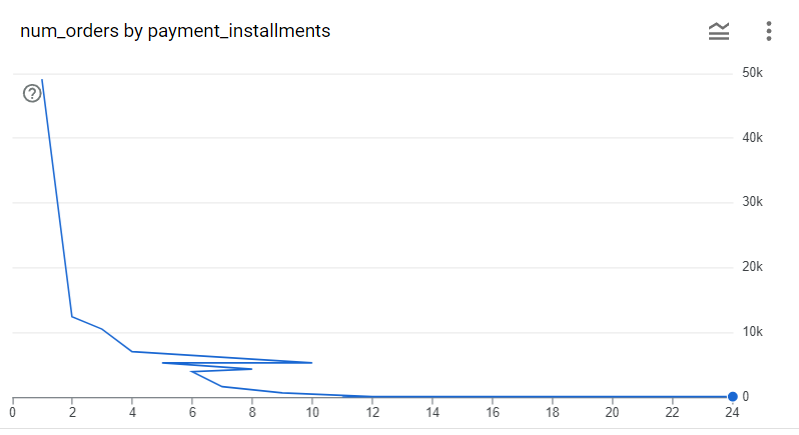
****

**INSIGHTS:**



X axis: No of instalments  
 y axis: No of orders

From the above chart and output, it can be seen that maximum orders (49060 orders) are paid in 1 installment only. Most of the payments are made in 1,2,3, and 4 installments which can also be observed from the below graph.



**Thank you**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***