**Business Case: Target SQL**

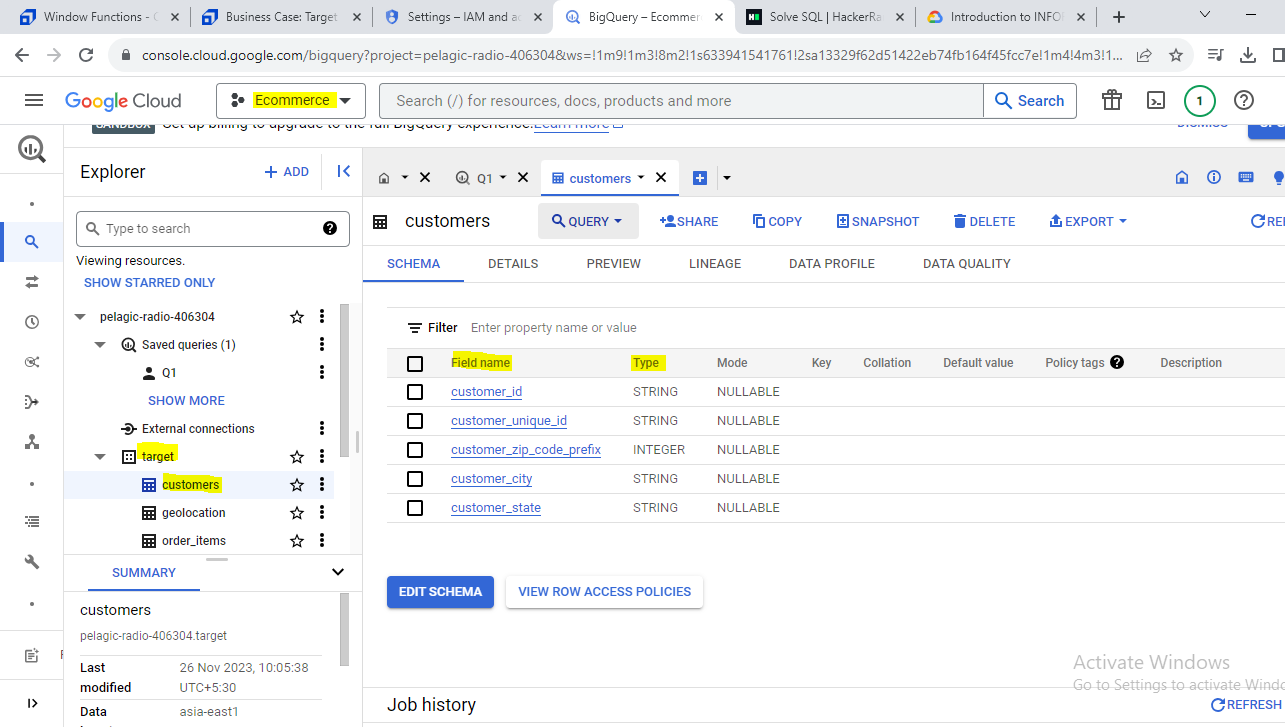
**1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

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

**Approach:** In BigQuery we can get in 2 ways:

1. After uploading tables in the dataset, when we click on table, under schema, we were able to see the data type of all columns.

**Screen shot:**

****

**b. By writing query using information\_schema**

**Query:**

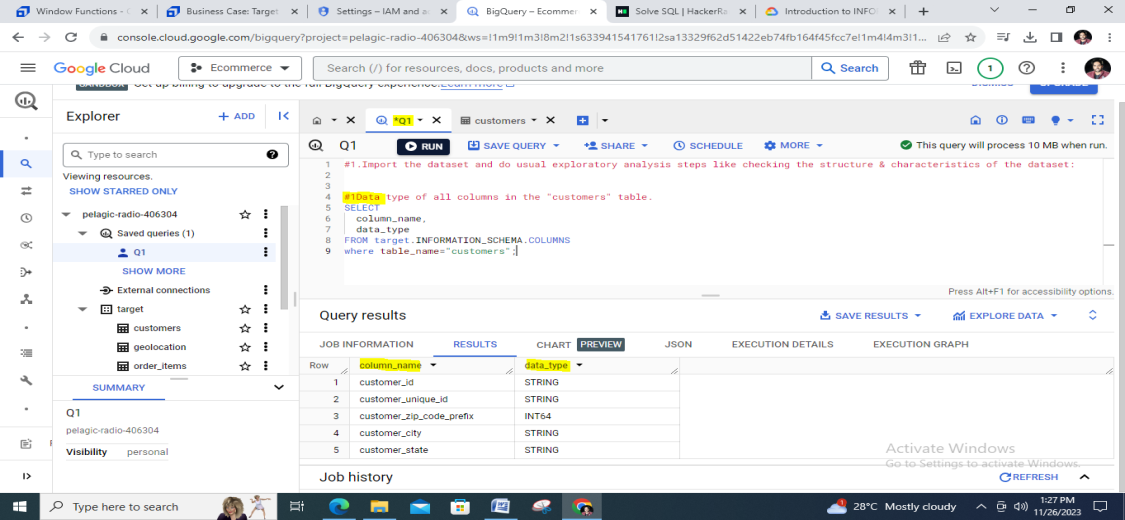
SELECT

  column\_name,

  data\_type

FROM target.INFORMATION\_SCHEMA.COLUMNS

where table\_name="customers";

**Screen shot: **

**Insights:** In customer table of Target we have 5 columns in which **customer\_zip\_code\_prefix** is of integer data type remaining all are of string data type.

**Recommendation:** Not required

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

**Query:**

SELECT

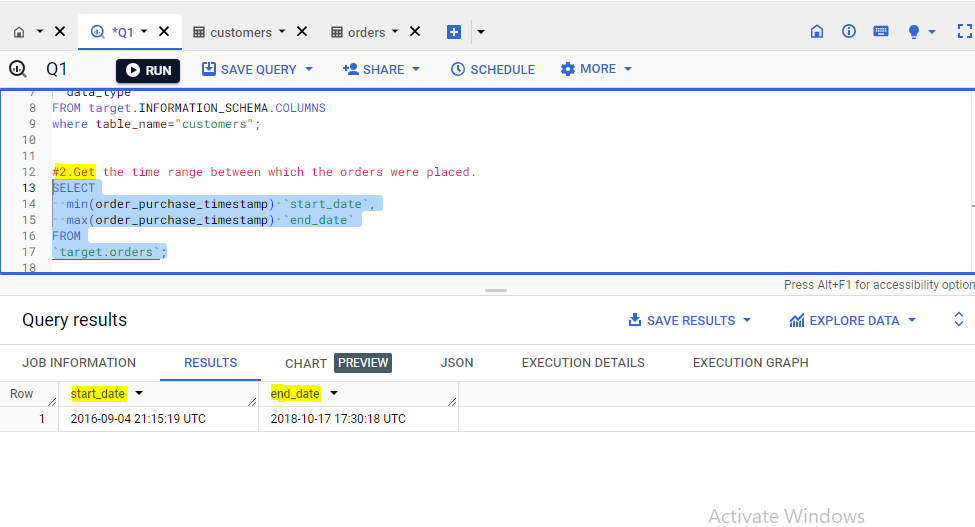
  min(order\_purchase\_timestamp) `start\_date`,

  max(order\_purchase\_timestamp) `end\_date`

FROM

`target.orders`;

**Screenshot:**

****

**Insights:** The time range were orders placed between **‘2016-09-04’ and ‘2018-10-17’.**

**Recommendation:** Not required

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

**Query:**

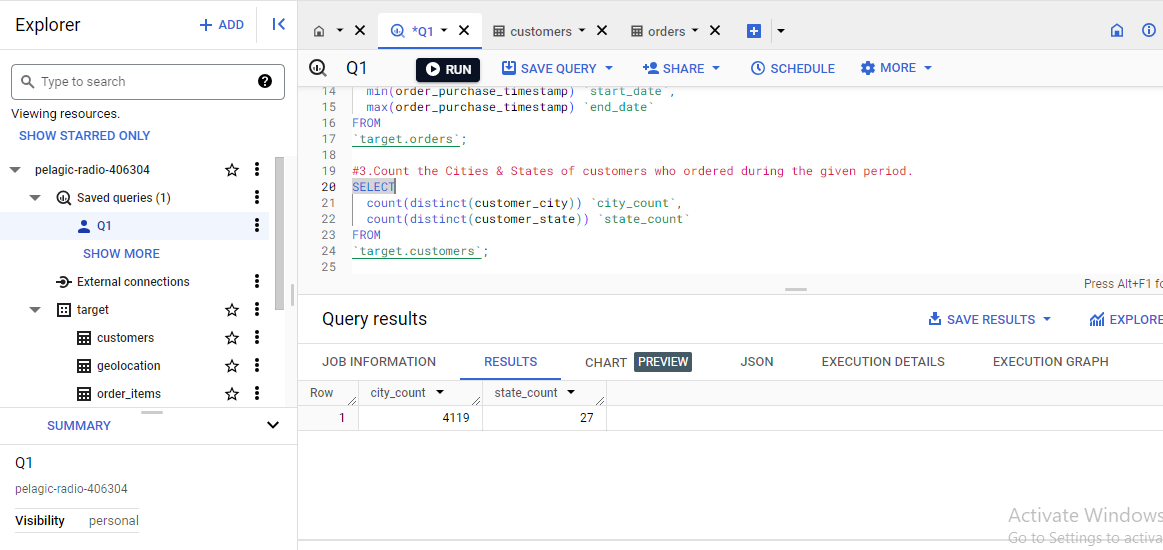
SELECT

  count(distinct(customer\_city)) `city\_count`,

  count(distinct(customer\_state)) `state\_count`

FROM `target.customers`;

**Screen Shot:**

****

**Insight:** During the given period, the customers orderd from 4119 different cities and 27 states.

**Recommendation:** We can get highest number of customers of state and city where we can increase productivity, customer services better.

We can find least number of customers of state and city we can increase order from them by promoting ads and coupon, discounts.

**2. In-depth Exploration:**

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

**Query:**

SELECT

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

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

    COUNT(\*) AS num\_orders

FROM

    `target.orders`

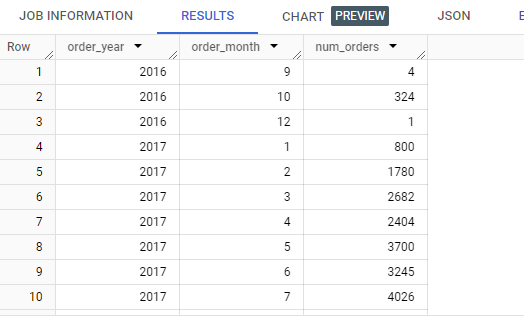
GROUP BY

    order\_year, order\_month

ORDER BY

    order\_year, order\_month;

**Output:**

****

**Insights:** In 2016, the number of orders placed in months are highly fluctuating and also we missed the data of 11 th month, From 2017 from month to month there slight increase and decrease in no. of orders placed.

From year to year there is an increase trend in place of orders.

**Recommendation:**

To know more precisely we can visualize the above data and get where there is high change in trend w.r.t decrease orders so that we can find which month it is and it’s reason and we can improve by finding root cause.

1. 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 order\_month,

    COUNT(\*) AS num\_orders

FROM

    `target.orders`

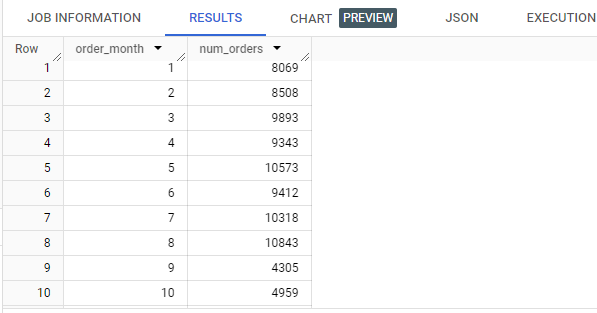
GROUP BY

    order\_month

ORDER BY

    order\_month;

output:



**Insights:** From the data we have, we can say 5th , 7th and 8th month of all years having high number of orders placed and 9th , 10th and 11th months are having low number of orders.

**Recommendation:** We can say that for the highest months there is festival season so the orders are high we can promote items with respect to the seasons in the areas and also provide coupons, discounts on other targeted items and also need to provide efficient delivery.

For lowest months we can provide some extra offers on items.

**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

**Query:**

WITH B AS (

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 "Mornings"

WHEN extract(hour from order\_purchase\_timestamp ) between 13 and 18 THEN "Afternoon"

    ELSE " Night"

  END  `most\_orders\_placed\_time\_of\_day`,

  count(\*) `count\_time\_category`

from `target.orders`

group by 1)

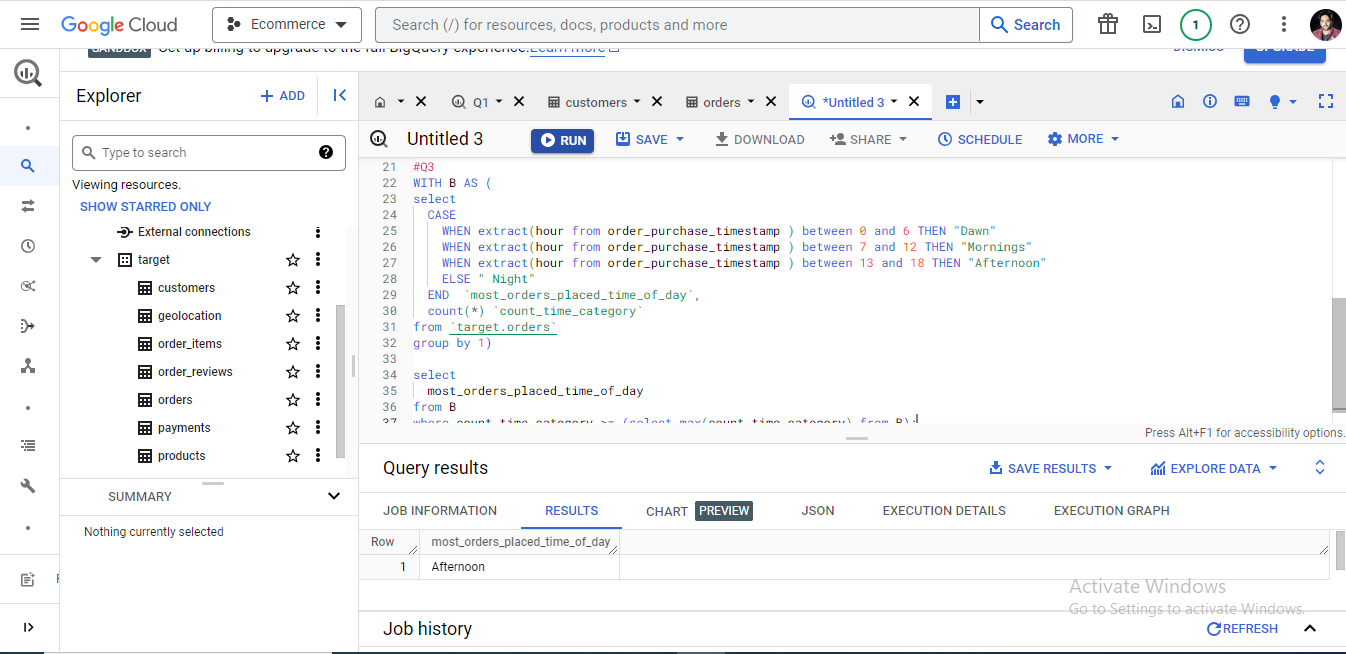
select

  most\_orders\_placed\_time\_of\_day

from B

where count\_time\_category >= (select max(count\_time\_category) from B);

**Screenshot:**

****

**Insight:** During Afternoon(13-18 hrs) of the day, the Brazilian customers mostly place their orders

**Recommendation:** We can provide best ways to handle the traffic of orders placed in After noon and also increase the best support teams to handle if customers placing orders are having any issues**.**

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

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

**Query:**

#CTE

WITH

months AS (

select distinct(extract(month from order\_purchase\_timestamp)) `month` from `target.orders`),

filter\_table AS(

select

  c.customer\_id,

  c.customer\_state,

  extract(month from o.order\_purchase\_timestamp) `month`

from `target.customers` c

left join `target.orders` o

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

select

  cs.`customer\_state`,

  m.month,

  coalesce(count(distinct ft.customer\_id),0) as customer\_count

from (

  (select distinct(customer\_state) `customer\_state` from `target.customers`) cs

  cross join

  months `m`

  left join filter\_table `ft` on ft.`customer\_state` = cs.`customer\_state` and ft.`month` = m.month)

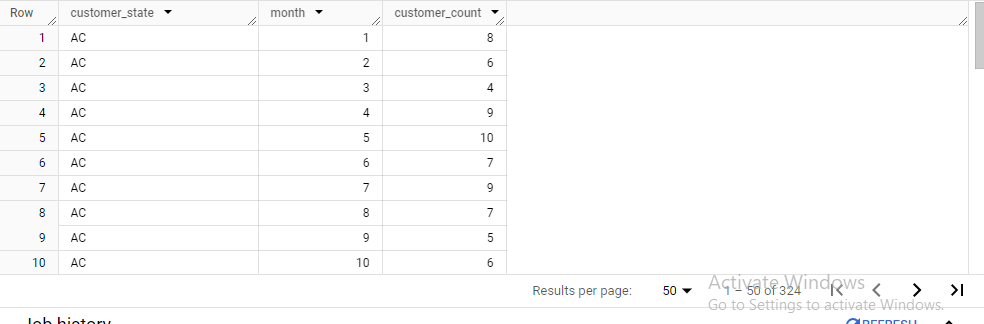
group by

cs.`customer\_state`,`m`.month

order by

cs.`customer\_state`,`m`.month;

**ScreenShot:**



**Insights:** We are getting all months order placed for each state, so we can use this data to conclude, on which month the orders were not placed or least number of orders placed and also we can get which months for which states we are getting highest number of orders.

**Recommendation**: By using this data we can provide delivery efficiency in the state and month with highest orders and also from application p.o.v we need to be more efficient by maintaining best user experience for the traffic of orders.

For lower and zero orders in the state and particular month we need to check the data is not recorded or really the orders were not placed and need to find the root cause analysis by providing details to market or sales executives to get the correct information then we can look into the issue and make decisions accordingly.

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

**Query:**

select

  customer\_state,

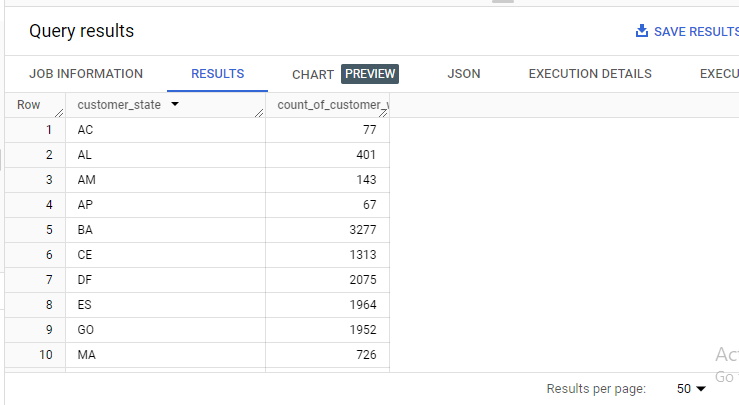
  count(distinct(customer\_unique\_id)) `count\_of\_customer\_wrt\_state`

from `target.customers`

group by customer\_state

order by customer\_state;

**ScreenShot:**

****

**Insights:State** SP has highest number of customers and RR has the lowest number of customers

**Recommendation:** So from the data we can say that for the highest number of customers of that state we can provide best customer experiences by providing some reward points and offers.

In addition we can add items which are least selling by providing certain offers and increase the selling of least selling items.

For the lowest customers of states we can increase offers and discounts and target the customer with items which are frequently buy. Need to make promotions by mentioning maximum offers.

**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.

Query:

with

  `data\_2017` AS (

    select

      extract(month from `orders`.order\_purchase\_timestamp) as `order\_month`,

      sum(`payments`.payment\_value) as `cost\_of\_orders\_2017`

    from `target.orders` `orders`

    left join `target.payments` `payments`

    on `orders`.order\_id = `payments`.order\_id

    where extract(month from `orders`.order\_purchase\_timestamp) IN (1,2,3,4,5,6,7,8)

    and

    extract(year from `orders`.order\_purchase\_timestamp)=2017

    group by `order\_month`

    order by `order\_month`

  ),

  `data\_2018` AS (

    select

      extract(month from `orders`.order\_purchase\_timestamp) as `order\_month`,

      sum(`payments`.payment\_value) as `cost\_of\_orders\_2018`

    from `target.orders` `orders`

    left join `target.payments` `payments`

    on `orders`.order\_id = `payments`.order\_id

    where extract(month from `orders`.order\_purchase\_timestamp) IN (1,2,3,4,5,6,7,8)

    and

    extract(year from `orders`.order\_purchase\_timestamp)=2018

    group by `order\_month`

    order by `order\_month`

  ),

  `data\_2017\_2018` AS (

    select

     `data\_2017`.order\_month,

      `cost\_of\_orders\_2017`,

      `cost\_of\_orders\_2018`

    from `data\_2017`

    inner join `data\_2018`

    on `data\_2017`.order\_month =`data\_2018`.order\_month

  )

select

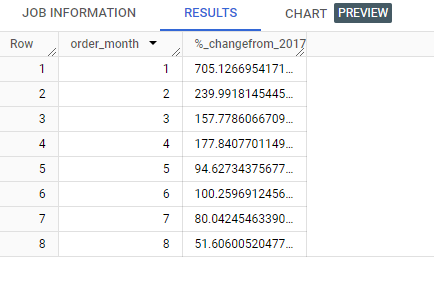
    order\_month,

    ((`cost\_of\_orders\_2018`-`cost\_of\_orders\_2017`)/`cost\_of\_orders\_2017`)\*100 as `%\_changefrom\_2017\_to\_2018`

from `data\_2017\_2018`

order by order\_month asc;

**output:**



**Insight:** From January to August, from 2017 to 2018, January has high change in terms of percentage and August has least change.

**Recommendation: Identify Trends, Provide Promotions and Discounts.**

Evaluate the factors contributing to increased order costs. Assess whether it's due to higher product prices, increased shipping costs, or other factors.

Consider adjusting product prices or shipping fees based on the analysis. Ensure that pricing strategies align with customer expectations and market conditions.

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

**Query:**

select `tc`.customer\_state,

  round(SUM(`oi`.price),2) as `sum\_order\_price`,

  round(AVG(`oi`.price),2) as `avg\_order\_price`

from `target.customers` `tc`

left join `target.orders` `to`

on `tc`.customer\_id = `to`.customer\_id

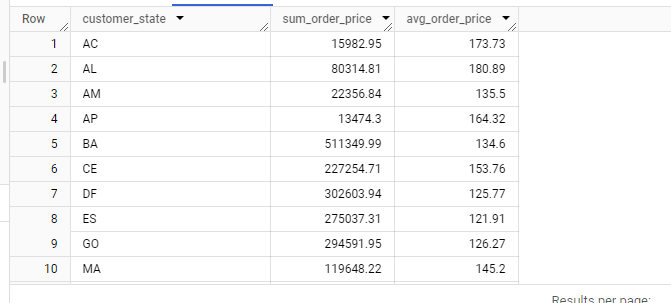
left join `target.order\_items` `oi`

on `oi`.order\_id = `to`.order\_id

group by `tc`.customer\_state

order by `tc`.customer\_state;

**output:**



**Insight:** For state SP the sum of order price is maximum, value is 5202955.05 and average\_order price is max for state PE.

For state AP the sum of orders price is minimum, value is 13474.3 and average\_order\_price is minimum for state SP

**Recommendation**: Analyze the states with the highest total order prices to identify states with strong sales performance.

Segment customers based on states to understand preferences and behaviors. Adjust marketing and product strategies accordingly.

Analyze the types of products that contribute significantly to total order prices. Consider expanding or promoting these product categories.

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

**Query:**

select `tc`.customer\_state,

  round(SUM(`oi`.freight\_value),2) as `sum\_order\_freight`,

  round(AVG(`oi`.freight\_value),2) as `avg\_order\_freight`

from `target.customers` `tc`

left join `target.orders` `to`

on `tc`.customer\_id = `to`.customer\_id

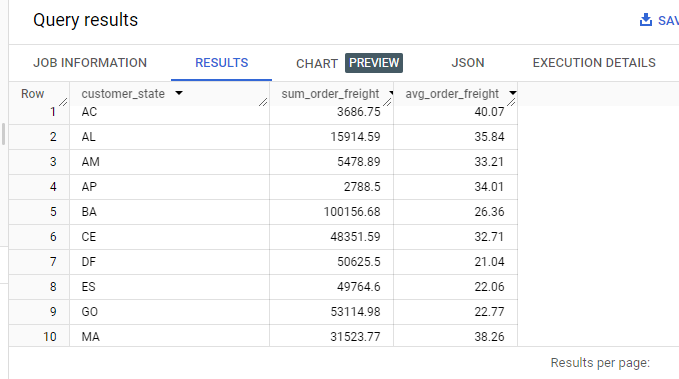
left join `target.order\_items` `oi`

on `oi`.order\_id = `to`.order\_id

group by `tc`.customer\_state

order by `tc`.customer\_state;

**output:**



**Insight:**

Total value of order freight for each state, indicating the overall shipping cost volume.

Average value of order freight for each state, providing insights into customer shipping cost patterns.

**Recommendation:**Analyze the states with the highest total order freight to identify regions with higher shipping costs and vice versa.

Explore opportunities to optimize shipping costs, especially in states with high shipping expenses. Negotiate with logistics partners for better rates.

Clearly communicate shipping costs to customers, especially in states with higher average order freight. Manage customer expectations regarding shipping expenses.

Consider adjusting product prices or introducing shipping promotions in states with higher shipping costs to remain competitive.

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

**Query:**

select

  \*,

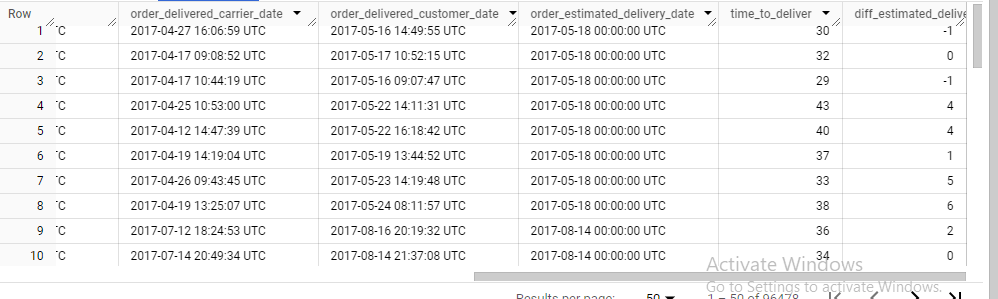
  DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,day) as `time\_to\_deliver`,

  DATE\_DIFF(order\_delivered\_customer\_date,order\_estimated\_delivery\_date,day) as `diff\_estimated\_delivery` #negative number indicates no. of days before estimated delivered, postive indicates delivered after estimated date

from `target.orders`

where order\_status="delivered";

**output:**

****

**Insight:** We can analyze the distribution of delivery times to understand the average and variation in the time taken to deliver orders.

Identify orders that were delivered within the estimated delivery time. Determine the percentage of orders that are delivered on time.

Analyze whether the geo location of customers has an impact on delivery times. Identify regions with consistently longer or shorter delivery times and **Seasonal Patterns and Delivery Delays.**

**Recommendation:**

If there are consistent delays, evaluate and optimize the logistics and order fulfillment processes to improve operational efficiency.

Implement effective communication strategies to inform customers about estimated delivery times and any potential delays. Transparency can enhance customer satisfaction.

Implement predictive analytics to forecast delivery times more accurately. Use historical data to identify patterns and improve delivery estimates.

**Collaboration with Logistics Partners and focus on Customer Feedback Analysis.**

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

CREATE VIEW target.`ranks\_cities` AS

select

  `tc`.customer\_state,

  round(AVG(freight\_value)) as `avg\_freight\_value`,

  dense\_rank() over(order by round(AVG(freight\_value))) as `rank\_high`,

  dense\_rank() over(order by round(AVG(freight\_value)) desc) as `rank\_low`

from `target.customers` `tc`

left join `target.orders` `to`

on `tc`.customer\_id = `to`.customer\_id

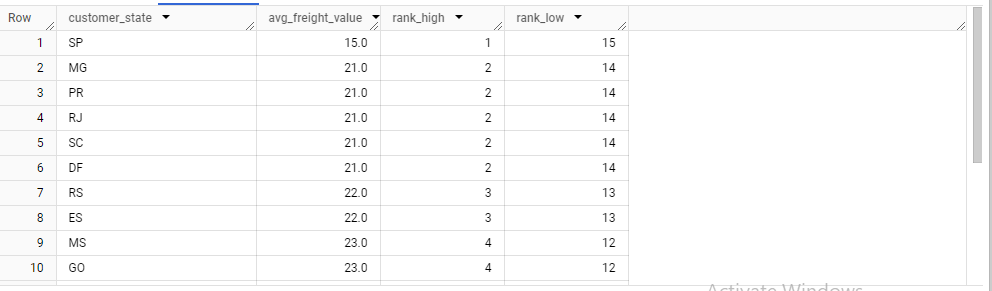
left join `target.order\_items` `oi`

on `oi`.order\_id = `to`.order\_id

group by `tc`.customer\_state

order by `avg\_freight\_value`;

**screenshot:**

****

#Top 5 state with the lowest average freight value

select

  `ranks\_cities`.customer\_state,

  avg\_freight\_value,

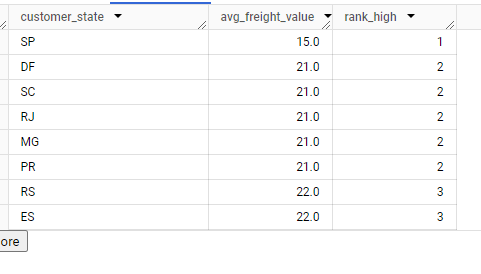
  rank\_high

from target.`ranks\_cities`

where `rank\_high`<=5

order by avg\_freight\_value;

**screenshot:**

****

#Top 5states with the highest average freight value

select

  ranks\_cities.customer\_state,

  avg\_freight\_value,

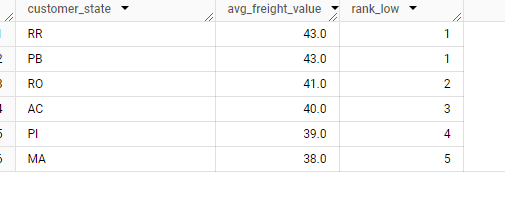
  rank\_low

from target.`ranks\_cities`

where `rank\_low`<=5

order by rank\_low ;

**screen** shot:



**Insight:**

**Top 5 States with Highest Average Freight:**

This information can help understand states where shipping costs are generally higher.

**Top 5 States with Lowest Average Freight:**

This can be indicative of regions with lower shipping costs or efficient logistics.

### Recommendations:

### For states with the highest average freight, explore opportunities to optimize shipping routes, carrier negotiations, or packaging strategies to reduce costs.

### Consider adjusting product prices or introducing promotions for states with higher average freight values to offset shipping costs and remain competitive.

### Evaluate and negotiate partnerships with logistics providers to secure more favorable rates, especially in regions with higher freight costs.

### Explore the possibility of establishing regional warehouses or fulfillment centers in states with higher average freight costs to minimize shipping distances.

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

**Query:**

CREATE VIEW target.`avg\_delivery\_states\_table` AS

SELECT

    customer\_state,

    ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp,day))) AS avg\_delivery\_time,

    DENSE\_RANK() OVER (ORDER BY ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp,day)))) AS rank\_high,

    DENSE\_RANK() OVER (ORDER BY ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp,day)))DESC) AS rank\_low

FROM

    `target.orders` AS `to`

LEFT JOIN `target.customers` AS `tc`

ON `to`.customer\_id = `tc`.customer\_id

WHERE

    order\_status ="delivered"

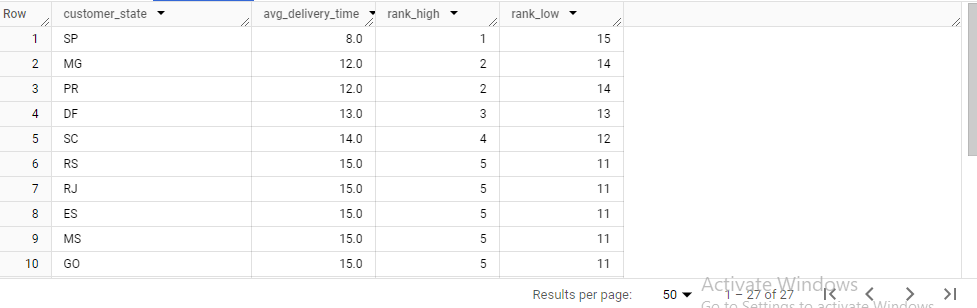
GROUP BY

    customer\_state

ORDER BY

    avg\_delivery\_time ;

**Output:**



**#top 5 states with lowest avg\_delivery\_time**

**Query:**

SELECT

    customer\_state,

    avg\_delivery\_time,

    rank\_high

FROM

target.`avg\_delivery\_states\_table`

WHERE `rank\_high`<=5;

**Output:**



**# top 5 states with highest avg\_time**

**Query:**

SELECT

    customer\_state,

    avg\_delivery\_time,

    rank\_low

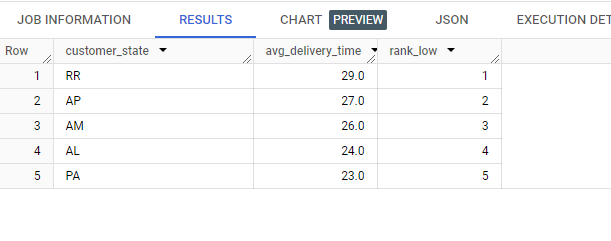
FROM

target.`avg\_delivery\_states\_table`

WHERE `rank\_low`<=5

order by `rank\_low`;

**output:**



**Insights:**

**Top 5 States with Highest Average Delivery Time:**

Identify the states where customers, on average, experience the longest delivery times. This information can highlight states where delivery logistics may need improvement.

**Top 5 States with Lowest Average Delivery Time:**

Identify the states where customers, on average, experience the shortest delivery times. This can indicate states with efficient delivery logistics.

### Recommendations:

### For states with the highest average delivery time, analyze and optimize the logistics and transportation processes to reduce delivery times.

### Explore the possibility of regional warehousing or fulfillment centers in states with longer average delivery times to shorten shipping distances.

### Enhance the overall customer experience by ensuring timely deliveries, particularly in regions with longer average delivery times.

### Work on improving the overall efficiency of the supply chain, from order processing to delivery, to reduce lead times

**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.**

**Query:**

WITH DeliverySpeeds AS (

    SELECT

        `tc`.customer\_state,

        ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date,day))) AS `avg\_delivery\_speed`,

        DENSE\_RANK() OVER (ORDER BY ROUND(AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date,       day)))ASC) as `delivery\_speed\_rank`

    FROM

    `target.orders` AS `to`

    LEFT JOIN `target.customers` AS `tc`

    ON `to`.customer\_id = `tc`.customer\_id

    WHERE

        order\_status = 'delivered'

    GROUP BY

        customer\_state

    ORDER BY

        `avg\_delivery\_speed`

)

SELECT

    customer\_state,

    avg\_delivery\_speed,

    delivery\_speed\_rank

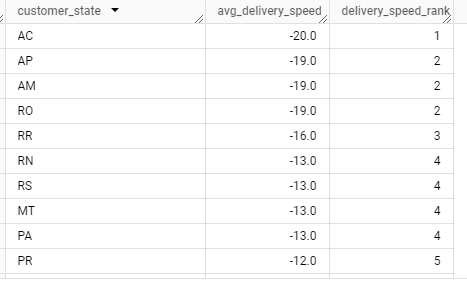
FROM

    DeliverySpeeds

WHERE

    `delivery\_speed\_rank` <= 5;

**Output:**

****

**Insights:** This information highlights states with efficient and expedited delivery services. The state with highest average delivery speed is ‘AC’.

### Recommendations: Analyze the processes and logistics in states with the fastest delivery speeds. Identify factors contributing to the efficiency and replicate successful strategies in other regions.

### Share best practices and optimize logistics processes across the entire delivery network to ensure consistent and fast delivery times.

### Implement or enhance real-time tracking features to keep customers informed about the status and location of their orders, contributing to a positive delivery experience.

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

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

**Query:**

WITH

  `payment\_mode` AS (

      select

          IF (payment\_type IS NULL,"others",payment\_type) AS payment\_type

      from `target.orders`

      LEFT JOIN `target.payments`

      ON `target.orders`.order\_id = `target.payments`.order\_id

      group by payment\_type),

  `month` AS(

      select

          extract(month from order\_purchase\_timestamp) as `month`

      from `target.orders`

      group by `month`),

  `filtered\_date\_orders\_count` AS (

      SELECT

          extract(month from order\_purchase\_timestamp) AS `month`,

          IF (payment\_type IS NULL,"others",payment\_type) AS payment\_type,

          count(\*) AS `total\_orders\_placed`

      FROM `target.orders`

      LEFT JOIN `target.payments`

      ON `target.orders`.order\_id = `target.payments`.order\_id

      group by `month`,payment\_type)

SELECT

        `month`.month,

        `payment\_mode`.payment\_type,

        COALESCE(`filtered\_date\_orders\_count`.`total\_orders\_placed`,0) AS `order\_count`

    FROM `month`

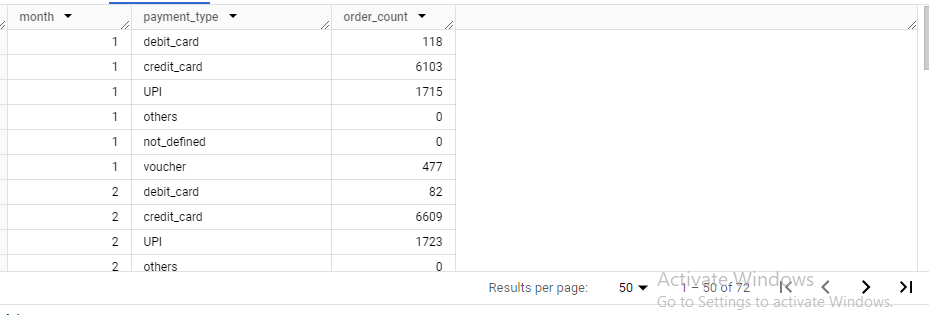
    CROSS JOIN `payment\_mode`

    LEFT JOIN `filtered\_date\_orders\_count`

    ON`month`.month = `filtered\_date\_orders\_count`.month and `payment\_mode`.payment\_type= `filtered\_date\_orders\_count`.   payment\_type

ORDER BY month;

**ScreenShot:**

****

**Insight:** It is observed that for few order payments are not done using any of the payment methods , so for that I added as others as payment \_type for each month. order\_count.

We have almost 6 types of payment types, we are considering: debit\_card, credit\_card, UPI, not\_defined, voucher and others.

So for 12 months- overall we have 72 rows, we can analyze as required.

**Recommendation:** we can do the following things**- Payment Method Analysis, Seasonal Patterns, Customer Preferences, Payment Processing Efficiency, User Experience**

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

select

  payment\_installments,

  count(distinct(order\_id)) AS `count1`

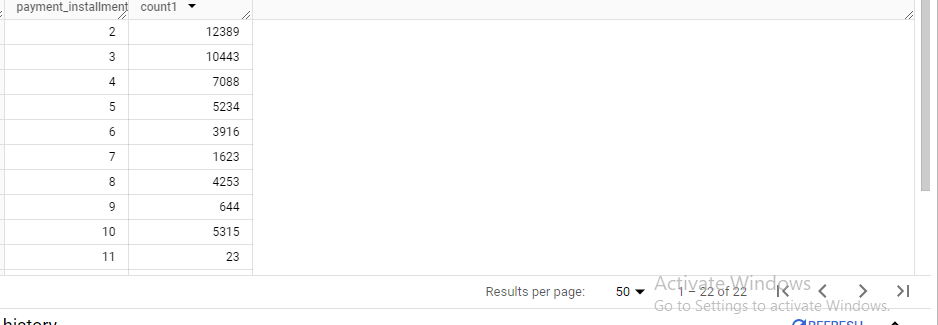
from `target.payments`

where payment\_installments>1

group by payment\_installments

order by payment\_installments;

**output:**

****

**Insight:** We can identify which payment installment plans are most commonly chosen by customers,

We can assess whether customers tend to choose a specific number of installments or if there is variability in their choices

**Recommendations:** If certain payment installment options are more popular, consider offering flexible payment plans to attract a broader customer base.

Increase marketing strategies to highlight the availability and benefits of installment payment options. Consider promoting installment plans during peak shopping periods.

Understand customer behavior regarding installment payments. Determine if there are preferred installment plans or if customers tend to pay in full.