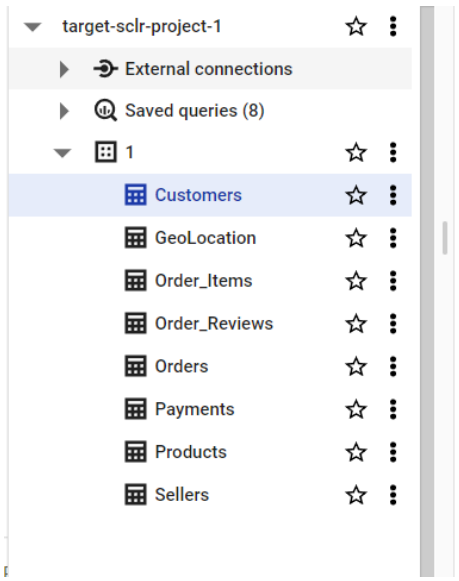
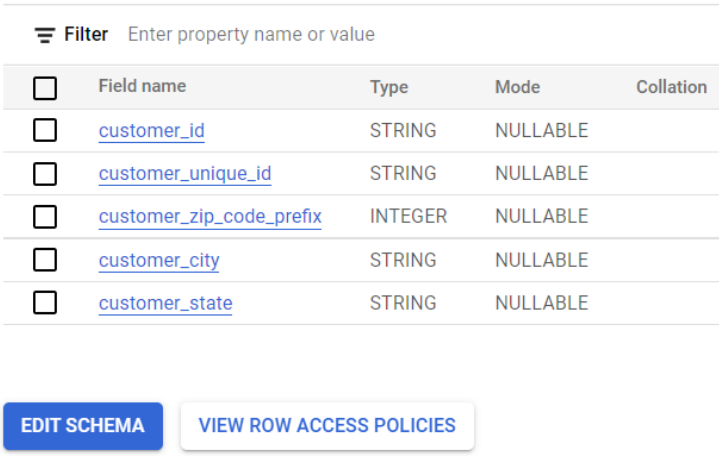
Name : Preetham Tiwari K
Target SQL Project Submission
Submission Date : 26-10-2022

Submission for the Project Target SQL which contains data of orders from Target store in Brazil .There were a total of 8 csv files provided for the project .The csv files were imported in google big query and analysis was performed based on the requirements .

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

        For this question the data type of columns was available upon importing the data into bigquery .



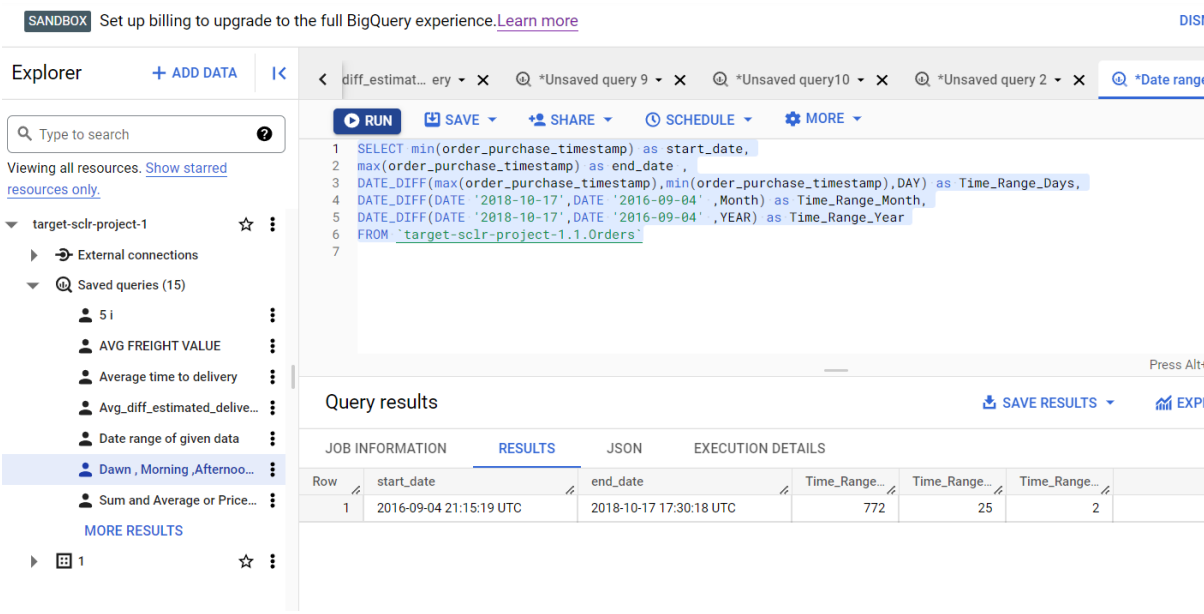


        The data type of the columns in the Customers table was mostly of string data type except the customer_zip_code_prefix column which had the data type as integer.

        The same analysis could be performed on all the tables to understand the data type in each column

2.Time period for which the data is given

To understand the time period of the given data the minimum and maximum date from the order_purchase_timestamp column in the orders table was used .



Query:

```
SELECT min(order_purchase_timestamp) as start_date,

max(order_purchase_timestamp) as end_date ,

DATE_DIFF(max(order_purchase_timestamp),min(order_purchase_timestamp),
DAY) as Time_Range_Days,

DATE_DIFF(DATE '2018-10-17',DATE '2016-09-04' ,Month) as
Time_Range_Month,

DATE_DIFF(DATE '2018-10-17',DATE '2016-09-04' ,YEAR) as
Time_Range_Year

FROM `target-sclr-project-1.1.Orders`
```

Upon selecting the maximum and minimum dates from the order_purchase_timestamp column the dates that were displayed are

Minimum date = 2016-09-04

Maximum date = 2018-10-17

The maximum date column was renamed as start_date and minimum date column was renamed as end_date .Then the DATE_DIFF function was used to calculate the difference between the maximum and minimum dates to get the range of days ,months and year for the given data and the columns were named as Time_Range_Days for range of days ,Time_Range_Months for the range of months and Time_Range_Year for the number of years .From the result the following time ranges were displayed .

Start date of data = 2016-09-04

End date of data = 2018-10-17

Number of days = 772

Number of months = 25

Number of years = 2

So it can be inferred from the result that the date range of the given data is between 2016-09-04 and 2018-10-17 and number of days in the date range is 772 number of months is 25 and number of years is 2

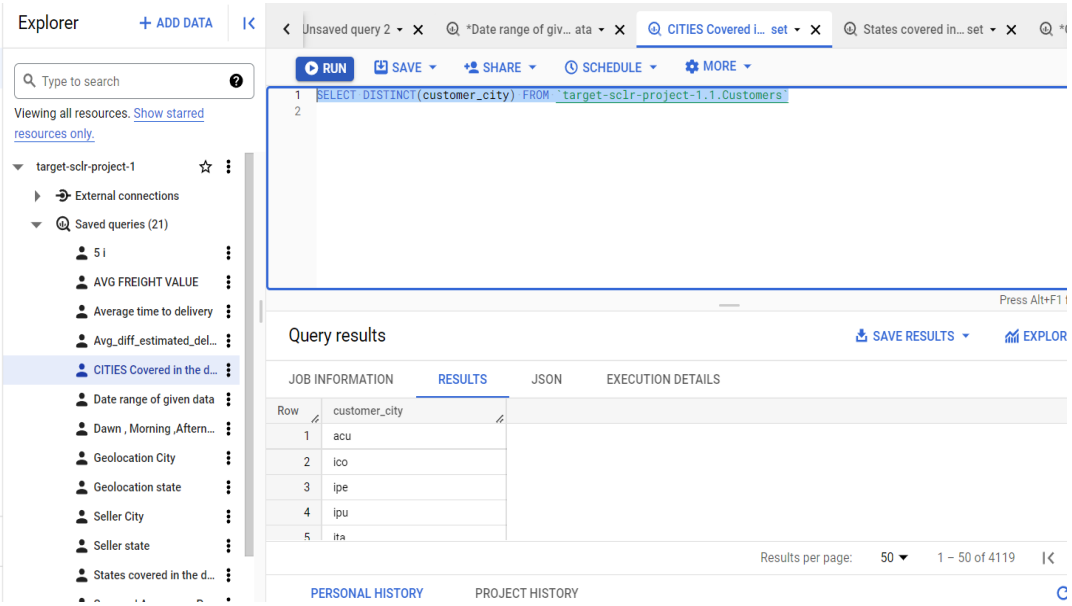
3.Cities and States covered in the dataset

To understand the cities and states covered in the dataset the category had to be split into 3 based on the available tables .

- Customer cities and states
- States and cities from the geolocation
- States and cities from the seller table.

Customer cities and states.

To get the information about the customer cities the following query was run .

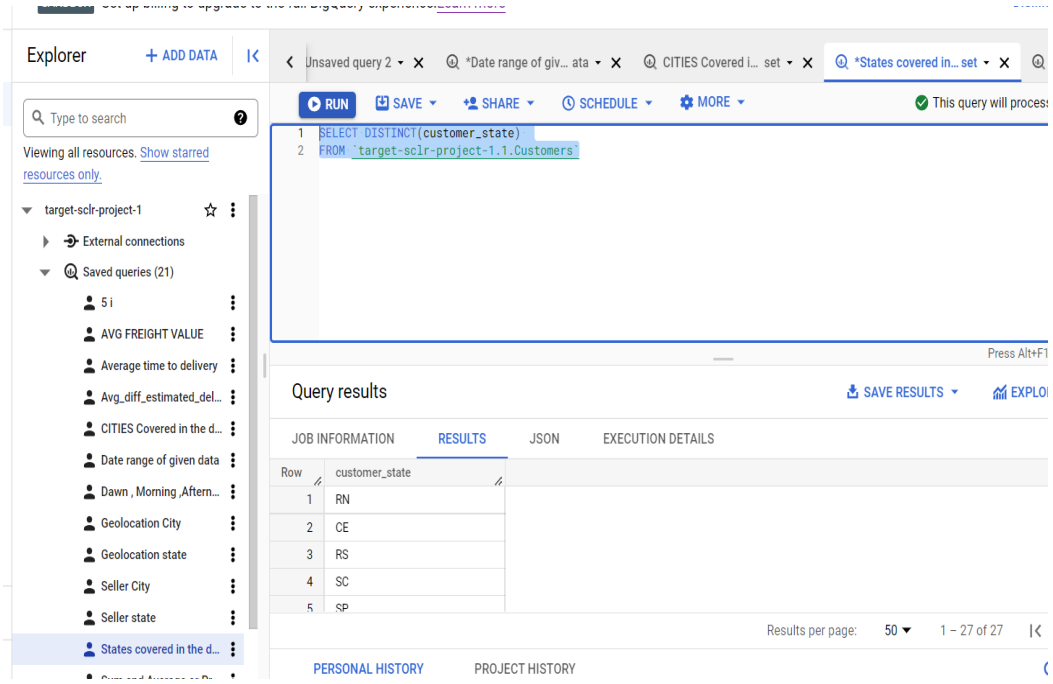


SQL Query :

```
SELECT DISTINCT(customer_city) FROM
`target-sclr-project-1.1.Customers`
```

This query gave a result indicating that the customers were distributed in a total of 4119 cities in Brazil.

To get the information about the customer states the following query was run .



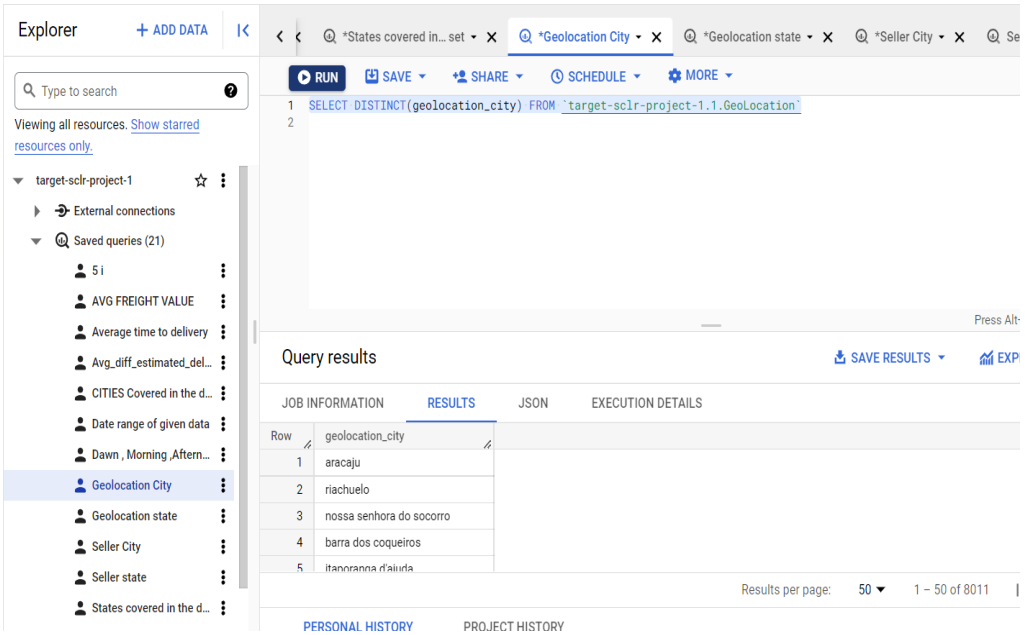
SQL Query :

```
SELECT DISTINCT(customer_state)

FROM `target-sclr-project-1.1.Customers`
```

From the result it can be seen that the customers are distributed in 27 states across Brazil.

- States and cities from the geolocation

  To get the information about the cities from the geolocation table the following queries were run.

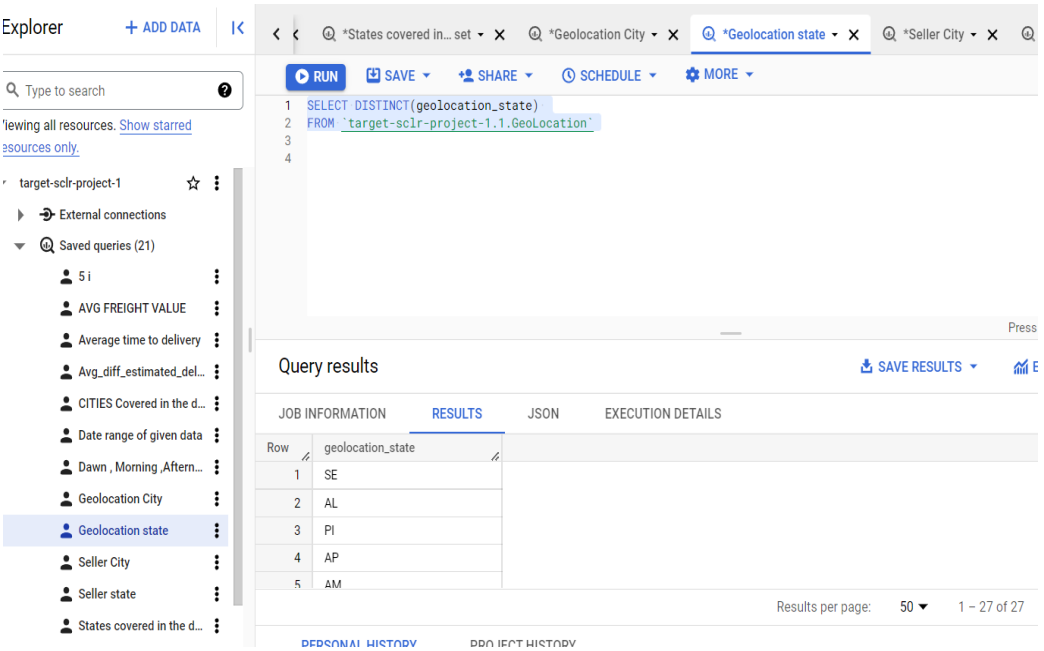


SQL Query :

```
SELECT DISTINCT(geolocation_city)
FROM `target-sclr-project-1.1.GeoLocation`
```

From the result it can be seen that there are a total of 8011 cities listed in the geolocation table .

To get the information about the states from the geolocation table the following queries were run.



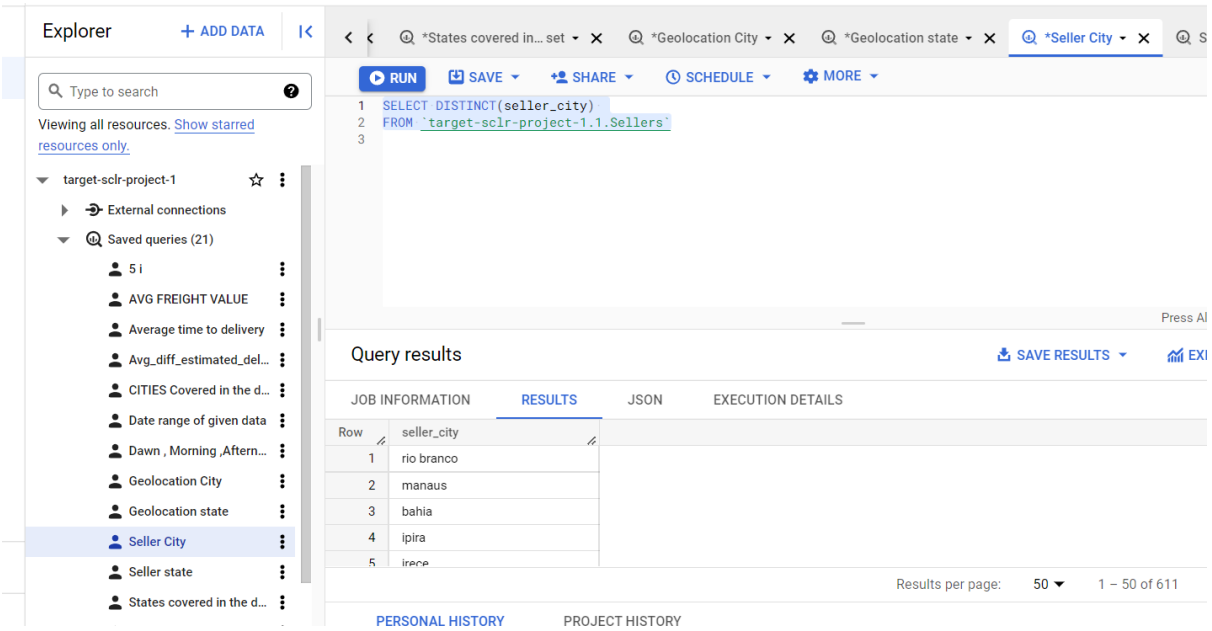
SQL Query:

```
SELECT DISTINCT(geolocation_state)
FROM `target-sclr-project-1.1.GeoLocation`
```

The results from the query indicate that there were a total of 27 states listed in the geolocation table.

States and cities from the seller table.

To get the information about the seller cities from the sellers table the following query was run .

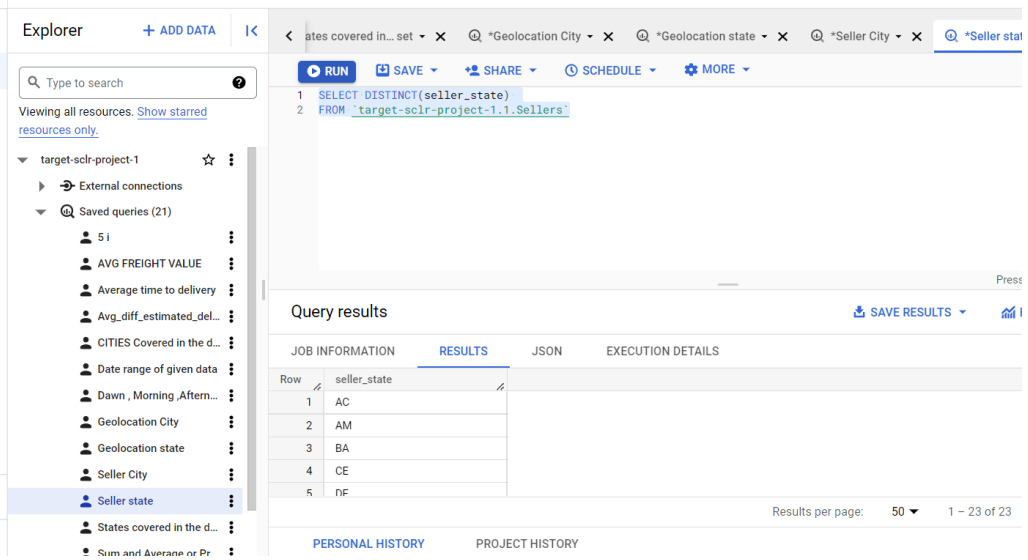


SQL Query :

```
SELECT DISTINCT(seller_city)
FROM `target-sclr-project-1.1.Sellers`
```

The results from this query were interesting as the sellers seemed to be distributed across only 611 cities in comparison to the customers who were distributed across 4119 cities.

To get the information about the seller states from the sellers table the following query was run .



SQL Query

```
SELECT DISTINCT(seller_state)
FROM `target-sclr-project-1.1.Sellers`
```

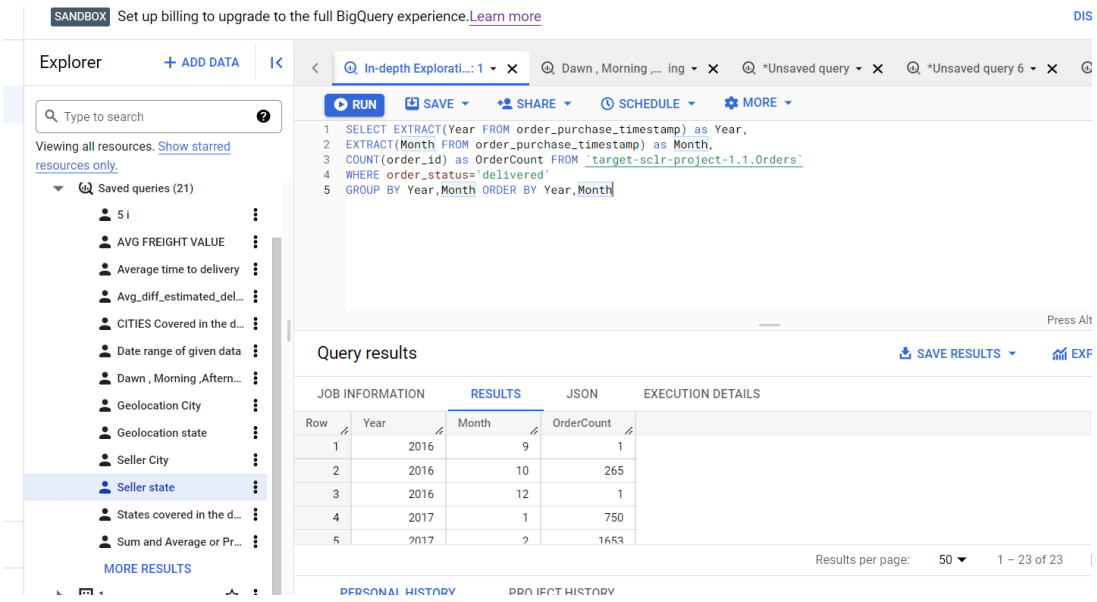
From the results it can be seen that the sellers are distributed across 23 states which is also less in comparison to the distribution of customers across 27 states.

Summary of results .
Count of cities of customer distribution : 4119
Count of states of customer distribution : 27
Count of cities from geolocation table : 8011
Count of states from geolocation table : 27
Count of cities of seller distribution : 611
Count of states of seller distribution : 23

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?

        For this question a query extracting the year and month from the column order_purchase_timestamp was executed by considering order status of only delivered orders and the count of order id as OrderCount.

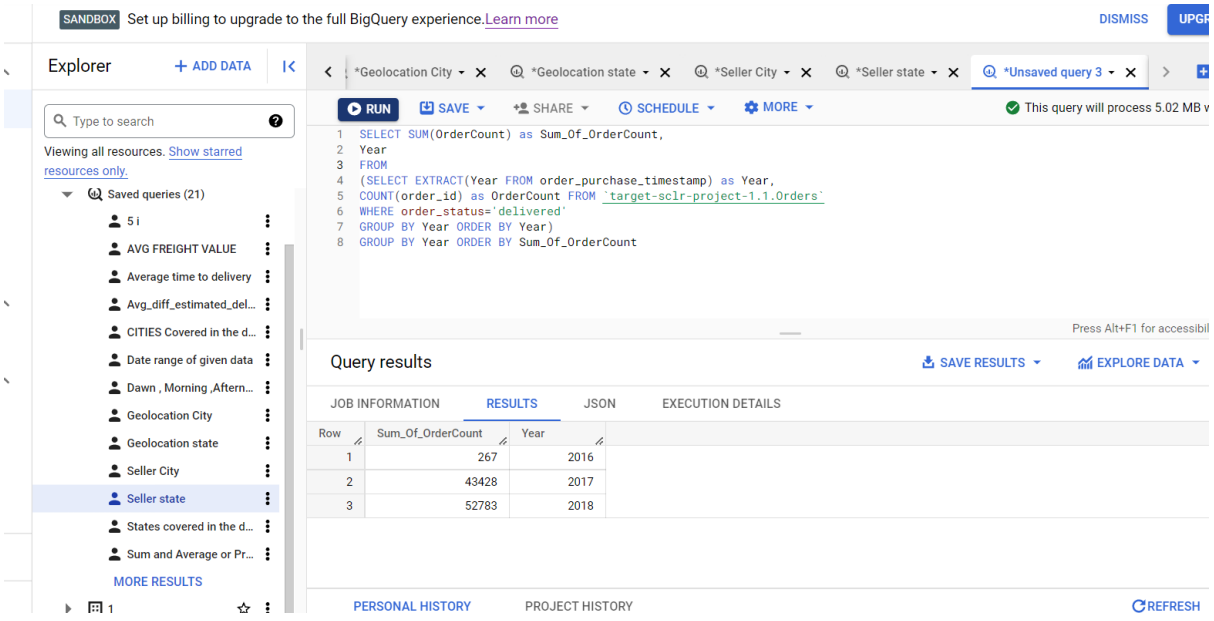


SQL Query:

```
SELECT EXTRACT(Year FROM order_purchase_timestamp) as Year,

EXTRACT(Month FROM order_purchase_timestamp) as Month,

COUNT(order_id) as OrderCount

FROM `target-sclr-project-1.1.Orders`

WHERE order_status='delivered'

GROUP BY Year,Month ORDER BY Year,Month
```

From the result it can be seen that the trend has been on a consistent growth the time period of 2016 september to the last time period of august 2018 .

There were less than 300 orders places in the three months available for analysis from the year 2016 and in comparison to this the number of orders placed in the year 2017 were 43428 and in just the 8 months available for analysis in the year 2018 the total order count was 52783 which is greater than the total number of orders placed in the entire year of 2017. This can be seen from the query below.



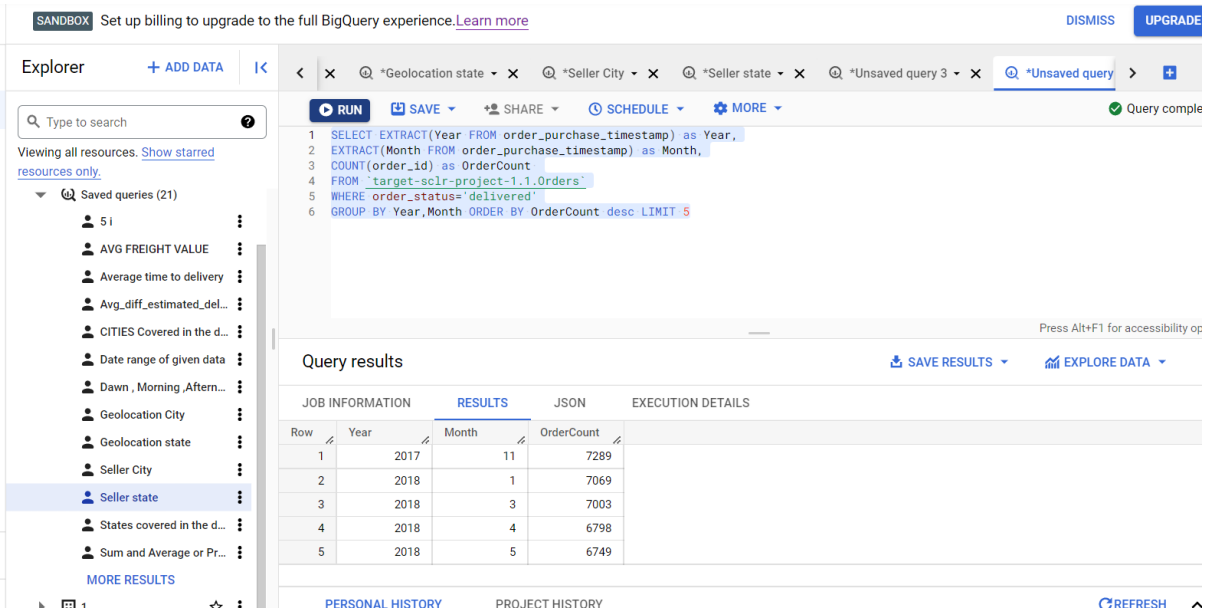
SQL Query:

```
SELECT SUM(OrderCount) as Sum_Of_OrderCount,
Year
FROM
(SELECT EXTRACT(Year FROM order_purchase_timestamp) as Year,
COUNT(order_id) as OrderCount FROM `target-sclr-project-1.1.Orders`
```

```
WHERE order_status='delivered'
GROUP BY Year ORDER BY Year)
GROUP BY Year ORDER BY Sum_Of_OrderCount
```

Such numbers are a clear indication of a growing trend in e-commerce in Brazil.

The month during which the maximum orders were placed is the month of November from the year 2017 with the order count being 7289 .



SQL Query :

```
SELECT EXTRACT(Year FROM order_purchase_timestamp) as Year,
EXTRACT(Month FROM order_purchase_timestamp) as Month,
COUNT(order_id) as OrderCount
FROM `target-sclr-project-1.1.Orders`
WHERE order_status='delivered'
GROUP BY Year,Month ORDER BY OrderCount desc LIMIT 5
```

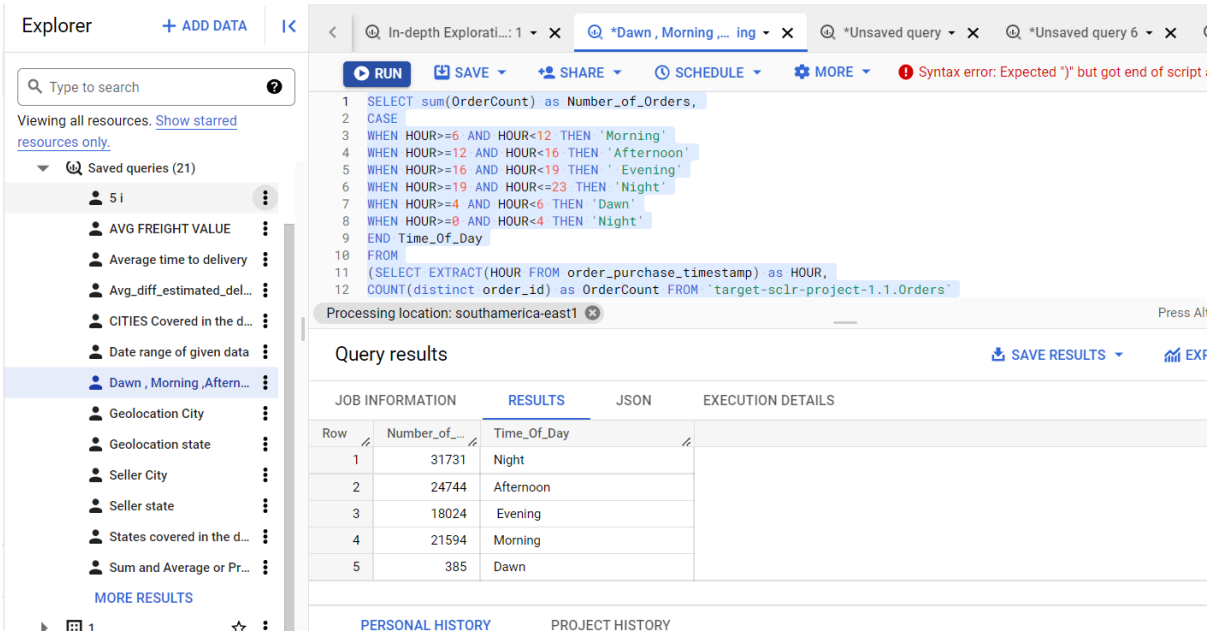
This could be because of 2 major festivals that are celebrated in Brazil in the month of November which are Day of the Dead (November 2) and Proclamation of the Republic (November 15) .

But the other interesting factor to be observed is that in the above query which displays the top 5 months where maximum orders were placed apart from the 1st result which is from the year 2017 all the rest of the results are from the year 2018 .

This is also a clear indicator that the e commerce business is on a stark rise in Brazil.

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

To understand the purchasing pattern of customers in Brazil based on the time of the day the column order_purchase_timestamp from the Orders table was analysed and the following query was executed .



SQL Query :

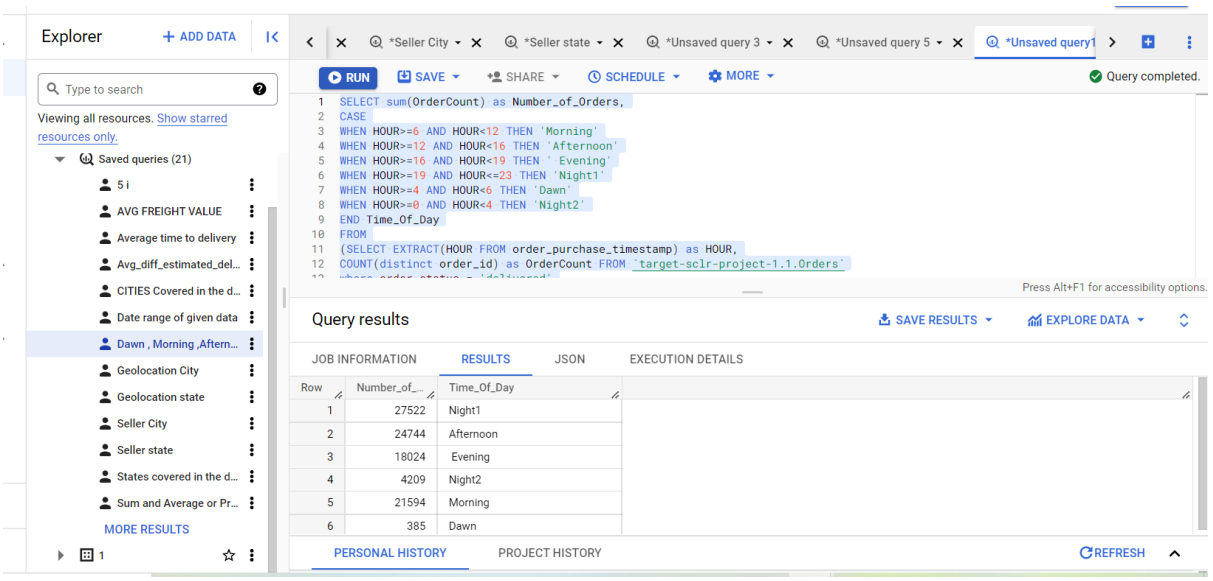
```
SELECT sum(OrderCount) as Number_of_Orders,
 CASE
 WHEN HOUR>=6 AND HOUR<12 THEN 'Morning'
 WHEN HOUR>=12 AND HOUR<16 THEN 'Afternoon'
 WHEN HOUR>=16 AND HOUR<19 THEN ' Evening'
 WHEN HOUR>=19 AND HOUR<=23 THEN 'Night'
 WHEN HOUR>=4 AND HOUR<6 THEN 'Dawn'
 WHEN HOUR>=0 AND HOUR<4 THEN 'Night'
 END Time_Of_Day
 FROM
```

```
  (SELECT EXTRACT(HOUR FROM order_purchase_timestamp) as HOUR,

  COUNT(distinct order_id) as OrderCount FROM `
target-sclr-project-1.1.Orders`

where order_status = 'delivered'

group by HOUR

) group by Time_Of_Day
```

The above query splits of the 24 hour time period in a day into four categories as below .

4am - 6am - Dawn
6am - 12pm -Morning
12pm - 4pm - Afternoon
4pm - 7pm - Evening
7pm-4am -Night

These different time categories are obtained from the order_purchase_timestampe column in the orders table .The count of order id is considered for order count and from the results it can be seen that the maximum number of orders are placed during the night with the count being 31731 orders .The night time ranges from 7pm -4am .further analysis can be performed to understand this trait in detail so as to maximise the orders that be sold to the customers.



SQL Query:

```
SELECT sum(OrderCount) as Number_of_Orders,

CASE

WHEN HOUR>=6 AND HOUR<12 THEN 'Morning'

WHEN HOUR>=12 AND HOUR<16 THEN 'Afternoon'

WHEN HOUR>=16 AND HOUR<19 THEN ' Evening'

WHEN HOUR>=19 AND HOUR<=23 THEN 'Night1'

WHEN HOUR>=4 AND HOUR<6 THEN 'Dawn'

WHEN HOUR>=0 AND HOUR<4 THEN 'Night2'

END Time_Of_Day

FROM

(SELECT EXTRACT(HOUR FROM order_purchase_timestamp) as HOUR,

COUNT(distinct order_id) as OrderCount FROM `target-sclr-project-1.1.Orders`

where order_status = 'delivered'

group by HOUR

) group by Time_Of_Day
```

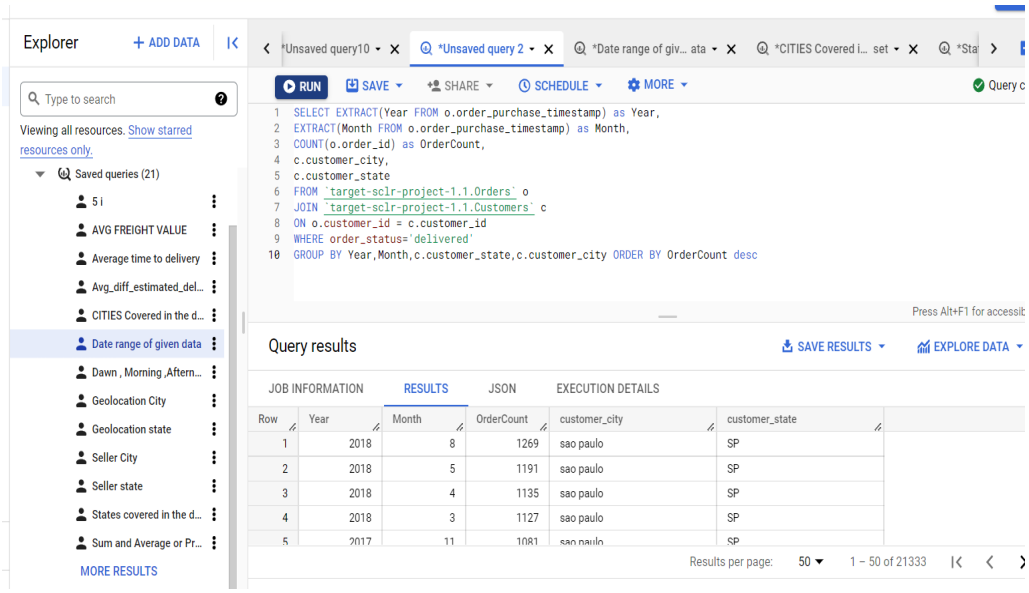
Upon performing further analysis by splitting the night time into parts as Night 1 which is for the time period between 7 pm - 12 am and Night 2 which is for the time period between 12am - 4am it can be seen that the maximum orders are placed between 7pm - 12 pm with the count being 27522 .

From these results it can be seen that the time period between 7pm -12 am is the most preferable time to launch new products and provide attractive offers and discounts to the customers as the traffic of customers is at the peak during these hours .

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

1. Get month on month orders by region, states

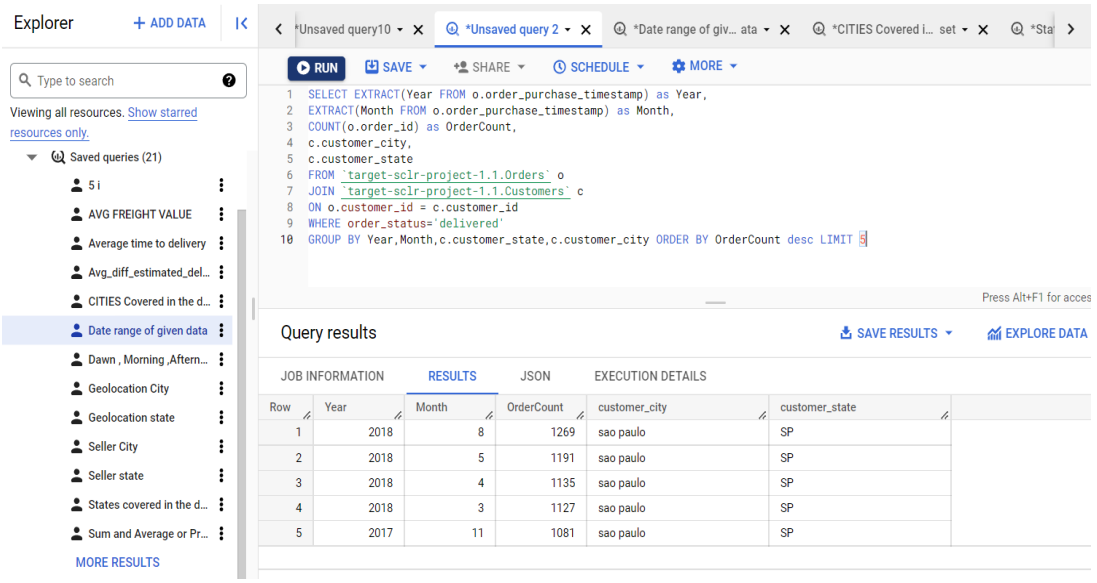
   The following query provides the data for the count of month on month orders by city and state.



SQL Query :

```
SELECT EXTRACT(Year FROM o.order_purchase_timestamp) as Year,

EXTRACT(Month FROM o.order_purchase_timestamp) as Month,

COUNT(o.order_id) as OrderCount,

c.customer_city,

c.customer_state

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

WHERE order_status='delivered'

GROUP BY Year,Month,c.customer_state,c.customer_city ORDER BY
OrderCount desc
```

We can see that the city of Sao Paulo is the region with the highest orders placed in a month .Sao Paulo is the only city in the list of top 5 cities for orders placed in a month .

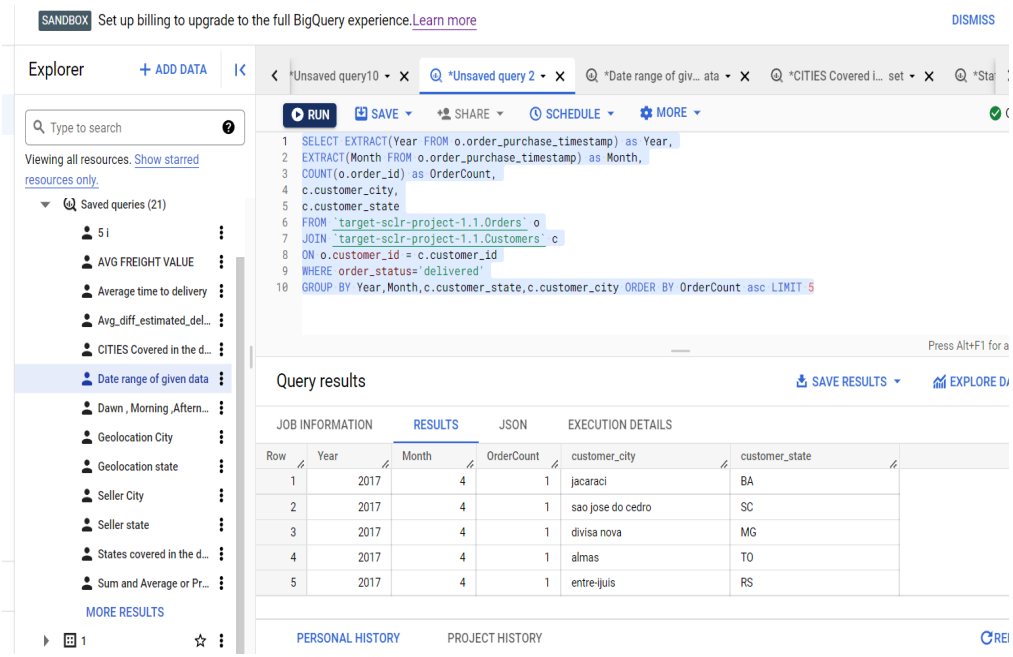


SQL Query :

```
SELECT EXTRACT(Year FROM o.order_purchase_timestamp) as Year,

EXTRACT(Month FROM o.order_purchase_timestamp) as Month,

COUNT(o.order_id) as OrderCount,

C.customer_city,

C.customer_state

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

WHERE order_status='delivered'

GROUP BY Year,Month,c.customer_state,c.customer_city ORDER BY
OrderCount desc LIMIT 5
```

Special offers and features focused for the city of Sao Paulo would bring in more customers and orders .

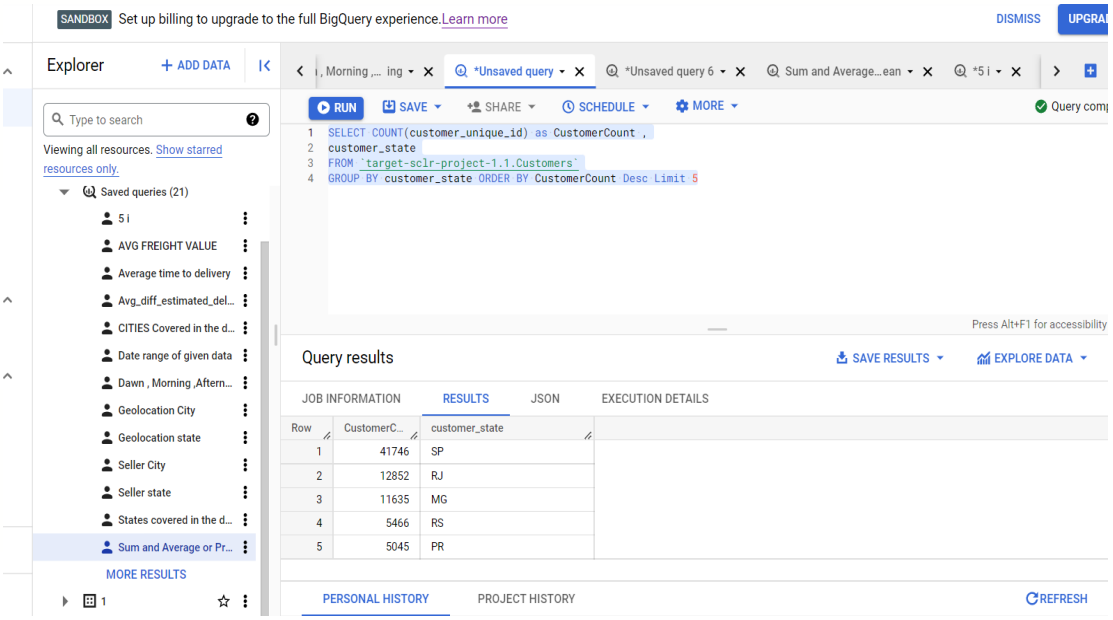
The 5 cities with least number of orders placed in a month can be seen by executing the following query



These cities have a very low order count and do not provide any value to business.

2. How are customers distributed in Brazil

To get the data about the top 5 states with the most customer count the following query was executed .

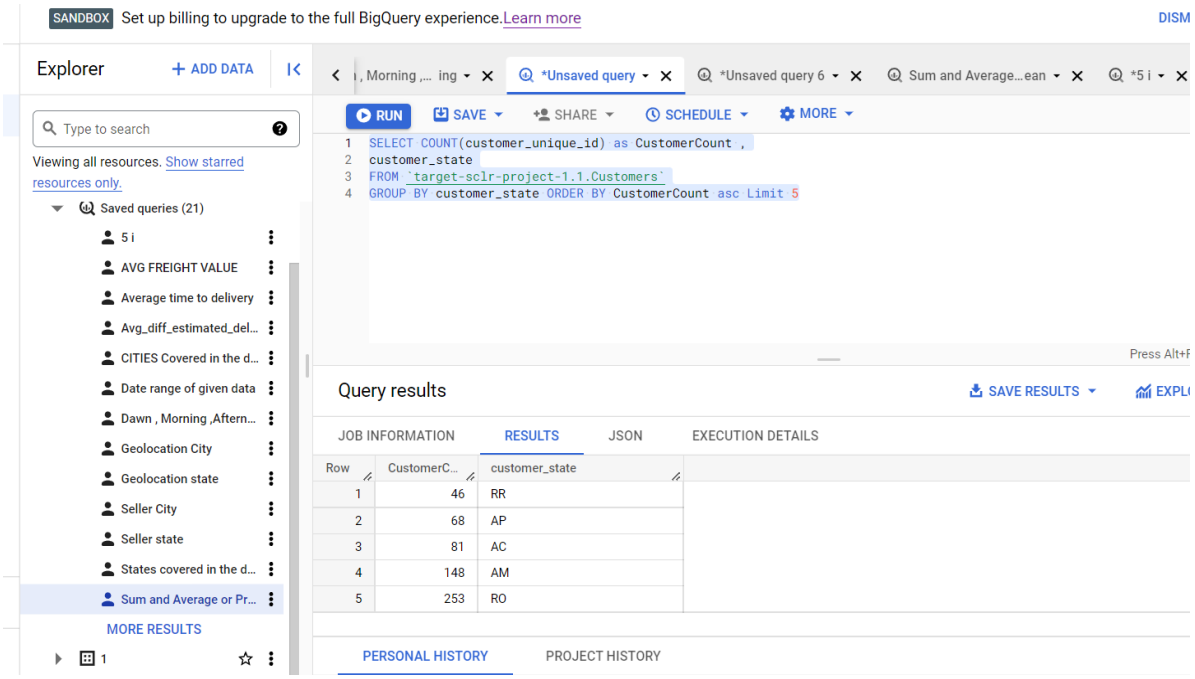


SQL Query :

```
SELECT COUNT(customer_unique_id) as CustomerCount ,
Customer_state
FROM `target-sclr-project-1.1.Customers`
GROUP BY customer_state ORDER BY CustomerCount Desc Limit 5
```

The results indicate that the state with the most customers is Sao Paulo with a count of 41746 customers .

To get the data for the states with the least number of customers the following query was executed .

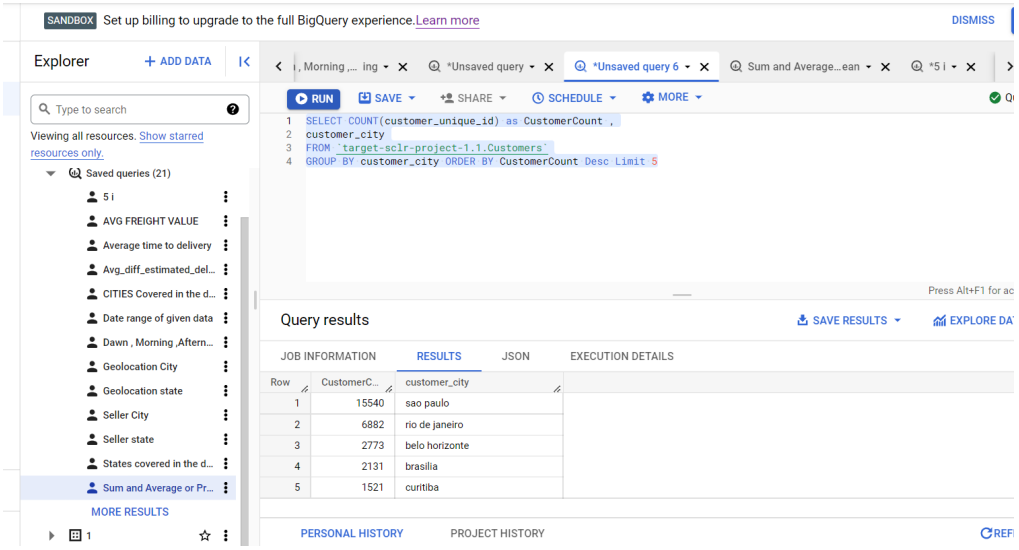


SQL Query :

```
SELECT COUNT(customer_unique_id) as CustomerCount ,
customer_state
FROM `target-sclr-project-1.1.Customers`
GROUP BY customer_state ORDER BY CustomerCount asc Limit 5
```

The query indicates that the state Roraima indicated as RR has the least number of customers with the count being just 46.

To understand the maximum count of customers across cities the following query was executed .

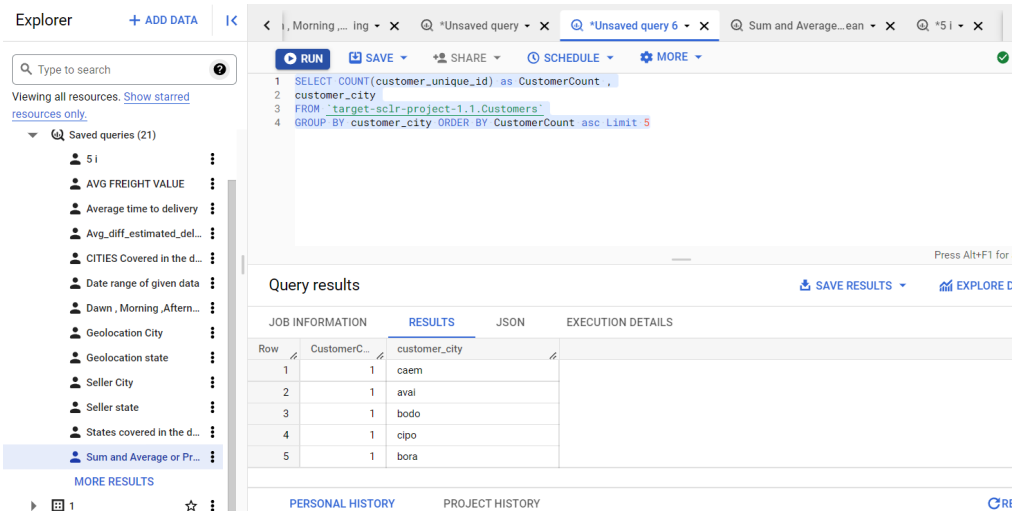


SQL Query :

```
SELECT COUNT(customer_unique_id) as CustomerCount ,
customer_city
FROM `target-sclr-project-1.1.Customers`
GROUP BY customer_city ORDER BY CustomerCount Desc Limit 5
```

Sao Paulo is the city with the most number of customers with the count being 15540

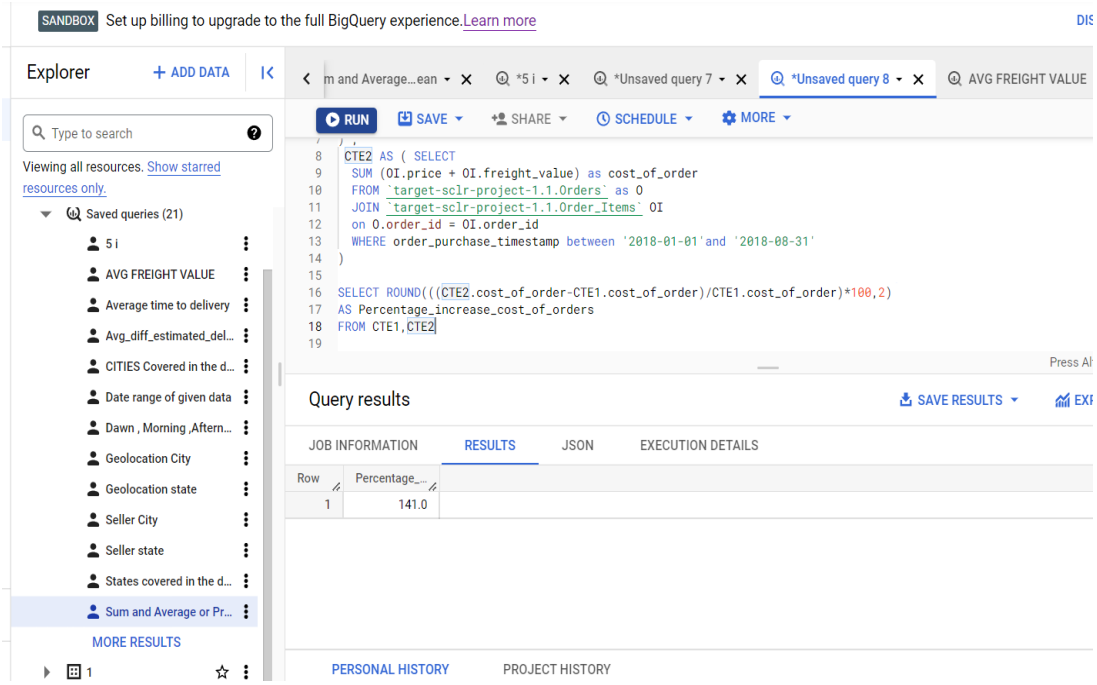
To understand the least count of customers across cities the following query was executed .



All the 5 cities in the result have a customer count of just 1 and are the cities with the least count of customers .

1. Impact on Economy: Analyze the money movemented 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)

        To calculate the percent in increase of cost of orders from the year 2017 - 2018 for the first 8 months the following query was executed.



SQL Query :

```
WITH CTE1 AS ( SELECT

 SUM (OI.price + OI.freight_value) as cost_of_order

 FROM `target-sclr-project-1.1.Orders` as O

JOIN `target-sclr-project-1.1.Order_Items` OI

on O.order_id = OI.order_id

WHERE order_purchase_timestamp between '2017-01-01'and '2017-08-31'

) ,

 CTE2 AS ( SELECT

SUM (OI.price + OI.freight_value) as cost_of_order

 FROM `target-sclr-project-1.1.Orders` as O
```

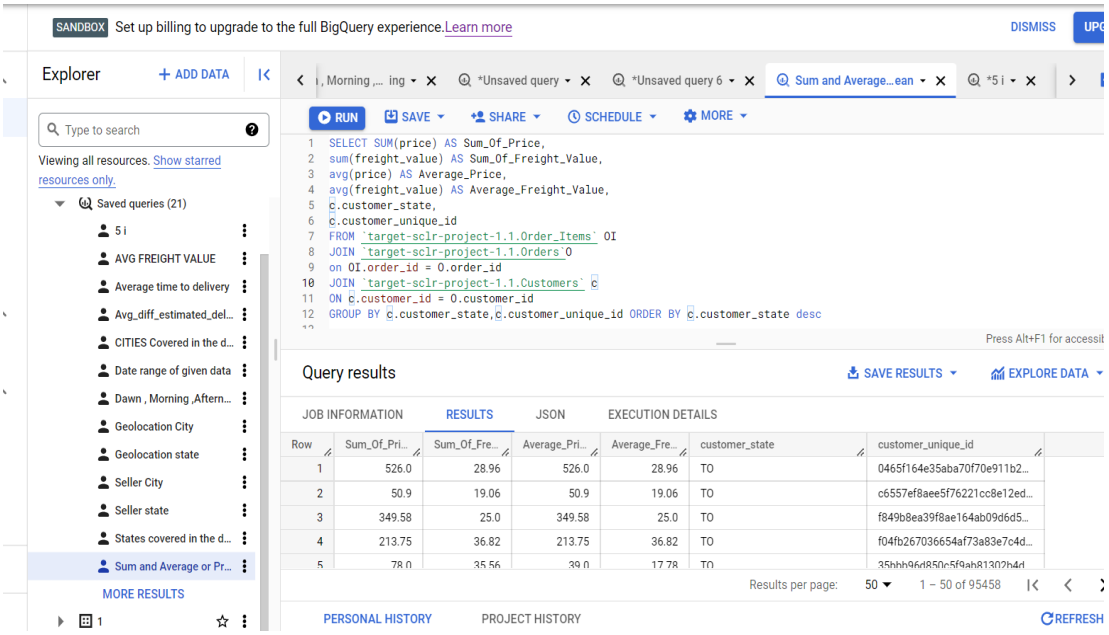
```
JOIN `target-sclr-project-1.1.Order_Items` OI

 on O.order_id = OI.order_id

WHERE order_purchase_timestamp between '2018-01-01'and '2018-08-31'

)



      SELECT
ROUND(((CTE2.cost_of_order-CTE1.cost_of_order)/CTE1.cost_of_order)*100
,2)

AS Percentage_increase_cost_of_orders

FROM CTE1,CTE2
```

The query is executed across 2 tables which are orders and order items tables .From the order items table the columns price and freight_value are summed and from the table orders the column order_purchase_timestamp is considered to choose the time period between january and august. The sum of price and freight value for the period between january and august for the year 2017 and the same is calculated for the year 2018 and the difference of the result between the year 2018 and 2017 is dividing by the value obtained for 2017 and the result is rounded to 2 to get the percentage increase in cost of orders .

Percentage increase in cost of orders = 141.0

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

The following query provides the mean and sum of price and freight value by customer state .



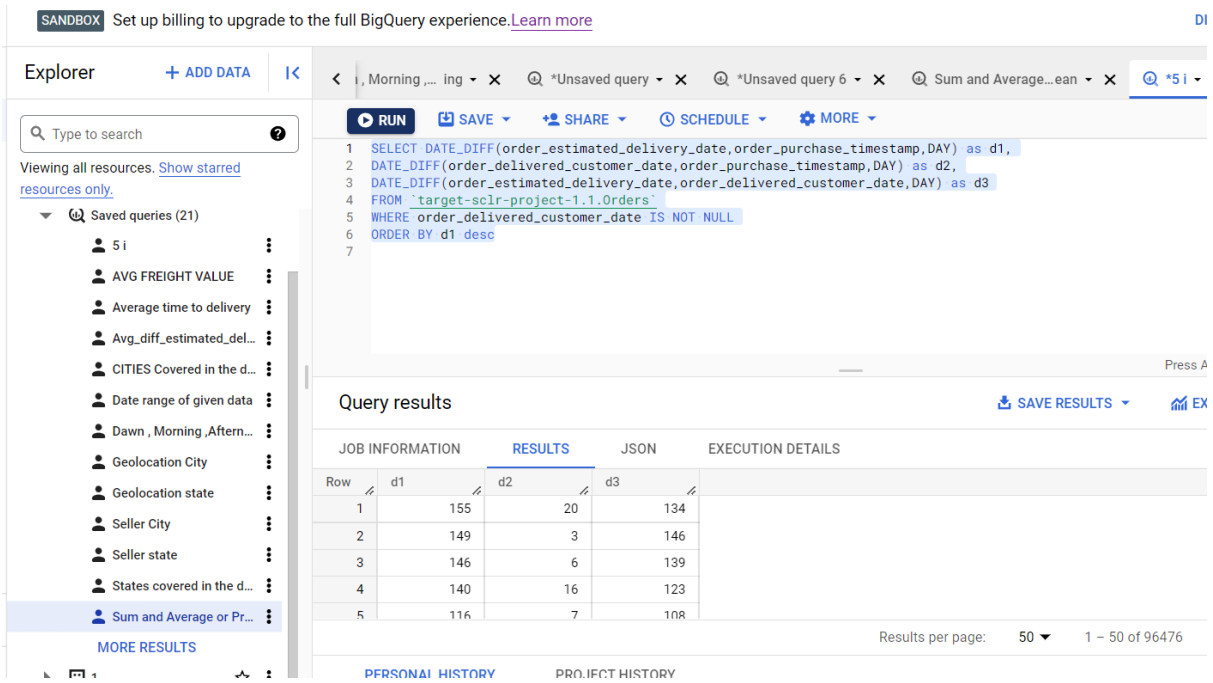
SQL Query:

```
SELECT SUM(price) AS Sum_Of_Price,
sum(freight_value) AS Sum_Of_Freight_Value,
avg(price) AS Average_Price,
avg(freight_value) AS Average_Freight_Value,
c.customer_state,
c.customer_unique_id
FROM `target-sclr-project-1.1.Order_Items` OI
JOIN `target-sclr-project-1.1.Orders`O
on OI.order_id = O.order_id
JOIN `target-sclr-project-1.1.Customers` c
ON c.customer_id = O.customer_id
GROUP BY c.customer_state,c.customer_unique_id
ORDER BY c.customer_state desc
```

5.Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

   The following query provides the required information .



   SQL Query :

```
SELECT DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,DAY)
as d1,

DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) as d2,

DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY)
as d3

FROM `target-sclr-project-1.1.Orders`

WHERE order_delivered_customer_date IS NOT NULL

ORDER BY d1 desc
```

   In the above query d1,d2 and d3 indicate the following.

   d1 = Difference between order_estimated_delivery_date and order_purchase_timestamp

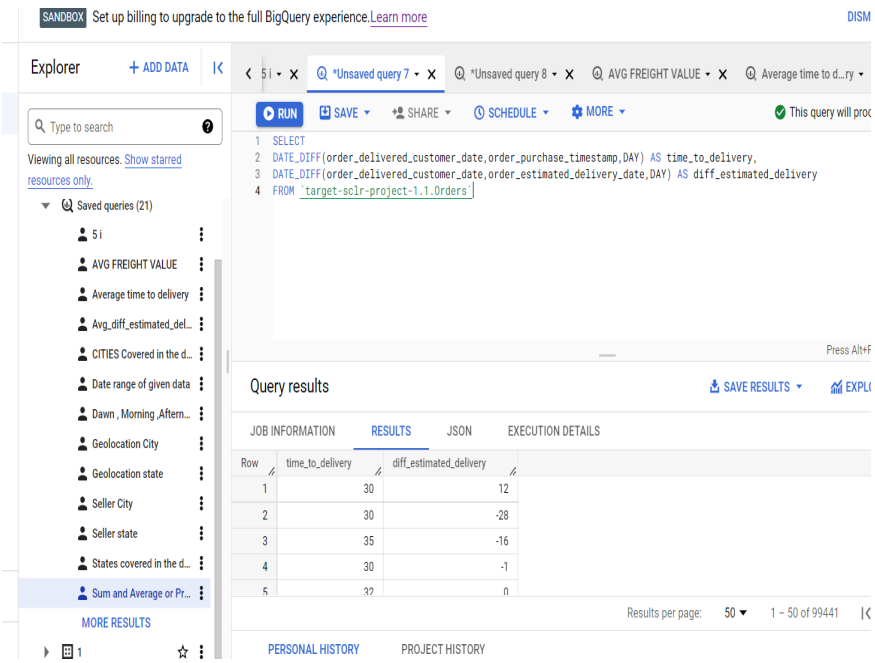
d2 = Difference between order_delivered_customer_date and order_purchase_timestamp

d3 = Difference between order_estimated_delivery_date and order_delivered_customer_date

2.Create columns:

- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

To create the 2 columns time_to_delivery and diff_estimated_delivery the following query is executed.

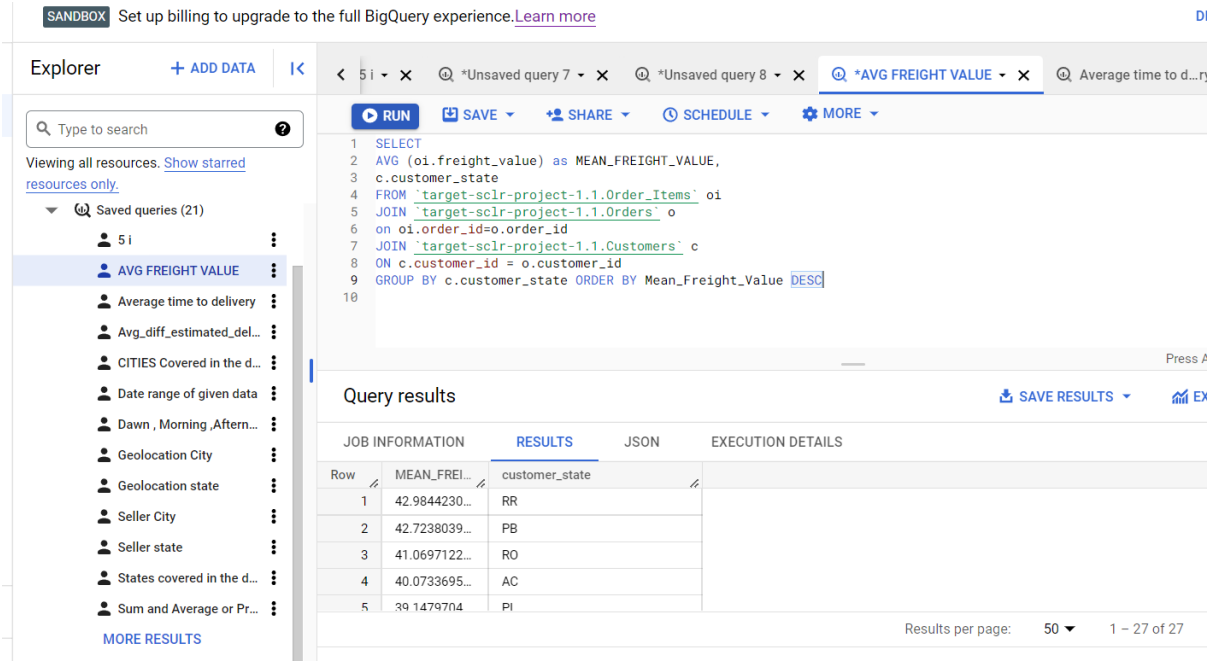


SQL Query:

```
SELECT

DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)
AS time_to_delivery,

DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date,
DAY) AS diff_estimated_delivery

FROM `target-sclr-project-1.1.Orders`
```

3.Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

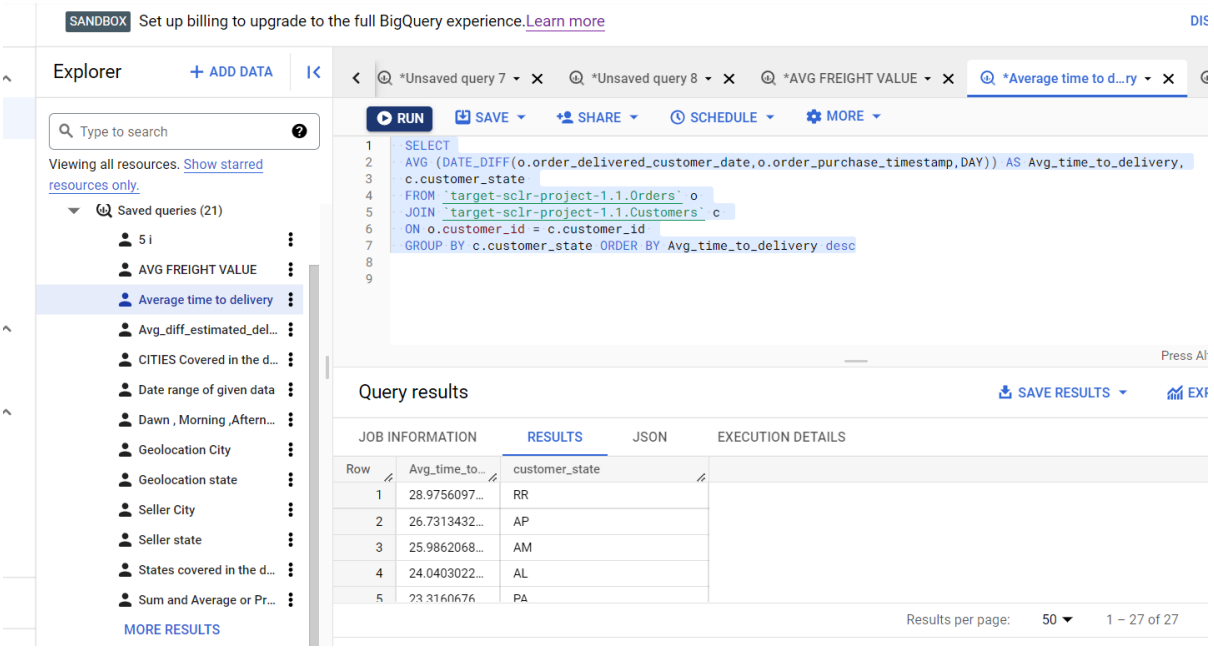
Mean of freight grouped by state .



SQL Query :

```
SELECT

AVG (oi.freight_value) as MEAN_FREIGHT_VALUE,

c.customer_state

FROM `target-sclr-project-1.1.Order_Items` oi

JOIN `target-sclr-project-1.1.Orders` o

on oi.order_id=o.order_id

JOIN `target-sclr-project-1.1.Customers` c

ON c.customer_id = o.customer_id

GROUP BY c.customer_state ORDER BY Mean_Freight_Value DESC
```

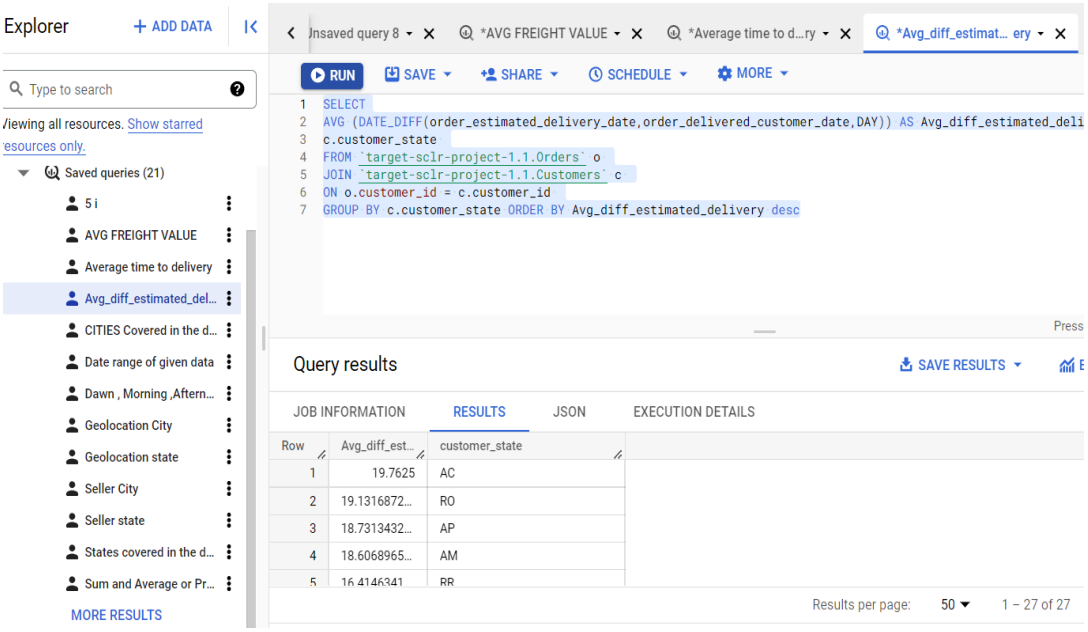
Mean of time to delivery grouped by state .



SQL Query :

```
SELECT

    AVG
(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,
DAY)) AS Avg_time_to_delivery,

c.customer_state

 FROM `target-sclr-project-1.1.Orders` o

 JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

GROUP BY c.customer_state ORDER BY Avg_time_to_delivery desc
```

Mean of difference in estimated delivery grouped by state.



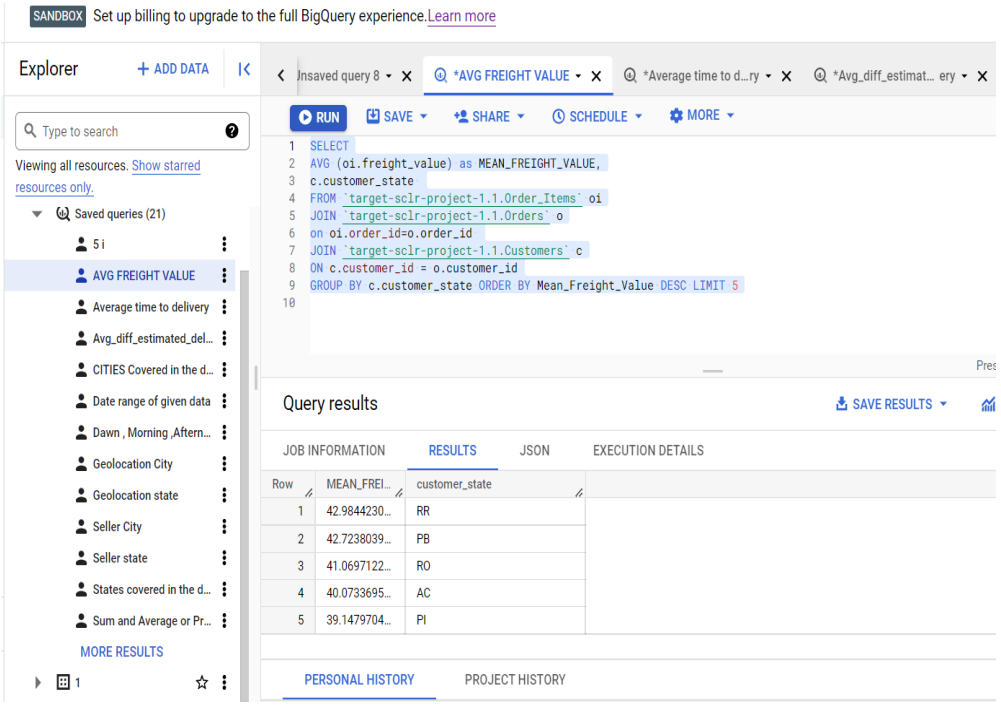
SQL Query :

```
SELECT

AVG
(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY))
AS Avg_diff_estimated_delivery,

c.customer_state

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

GROUP BY c.customer_state ORDER BY Avg_diff_estimated_delivery desc
```

4.Sort the data to get the following:

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

Top 5 states with highest average freight value.

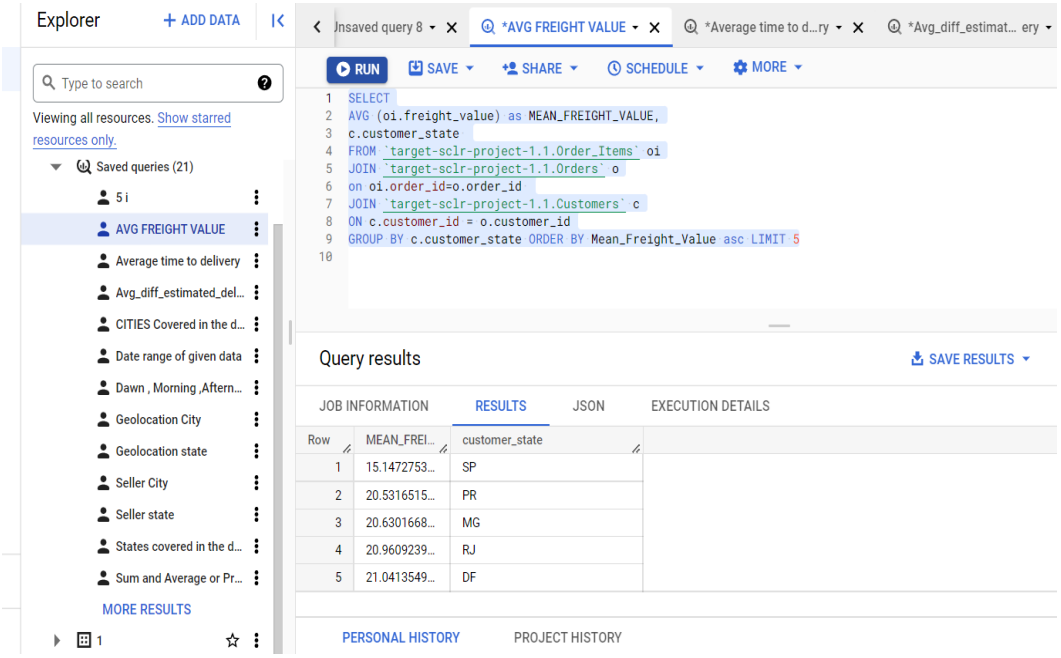


SQL Query :

```
SELECT

AVG (oi.freight_value) as MEAN_FREIGHT_VALUE,

c.customer_state

FROM `target-sclr-project-1.1.Order_Items` oi

JOIN `target-sclr-project-1.1.Orders` o

on oi.order_id=o.order_id

JOIN `target-sclr-project-1.1.Customers` c

ON c.customer_id = o.customer_id

GROUP BY c.customer_state ORDER BY Mean_Freight_Value DESC LIMIT 5
```

From the results it can be seen that the state Roraima has the highest average freight value .

Top 5 states with lowest average freight value.

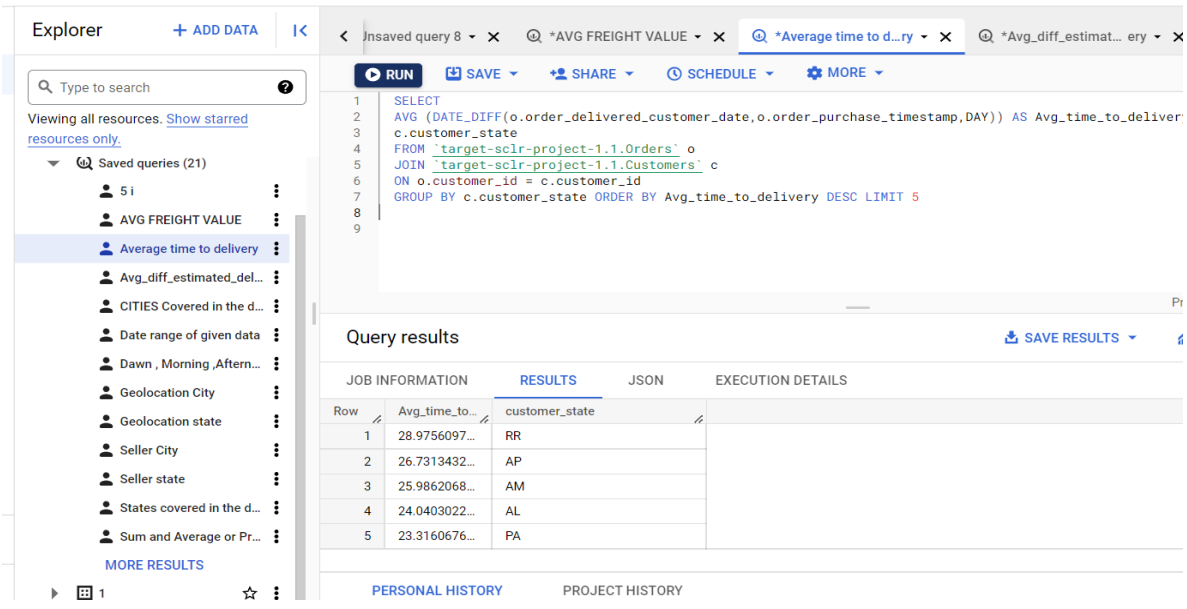


The state with the lowest average freight value is the state of Sao Paulo.

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

Top 5 states with highest average time to delivery

The following query provides the top 5 states with highest average time to delivery .

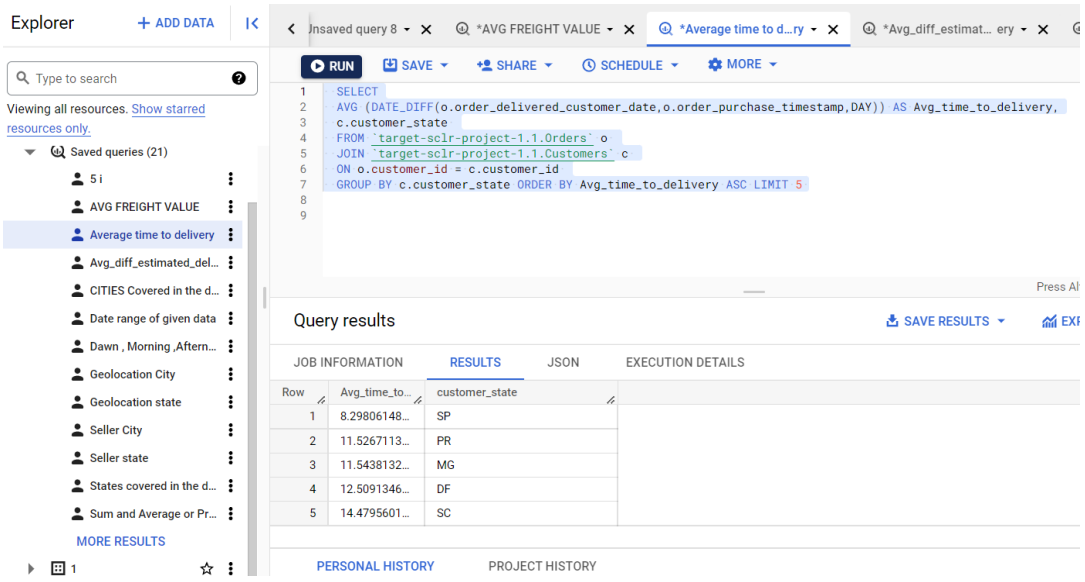


SQL Query :

```
  SELECT

      AVG
(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS Avg_time_to_delivery,

c.customer_state

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

GROUP BY c.customer_state ORDER BY Avg_time_to_delivery DESC LIMIT 5
```

From the results it can be seen that the state Roraima has the highest average time to delivery .

Top 5 states with lowest average time to delivery



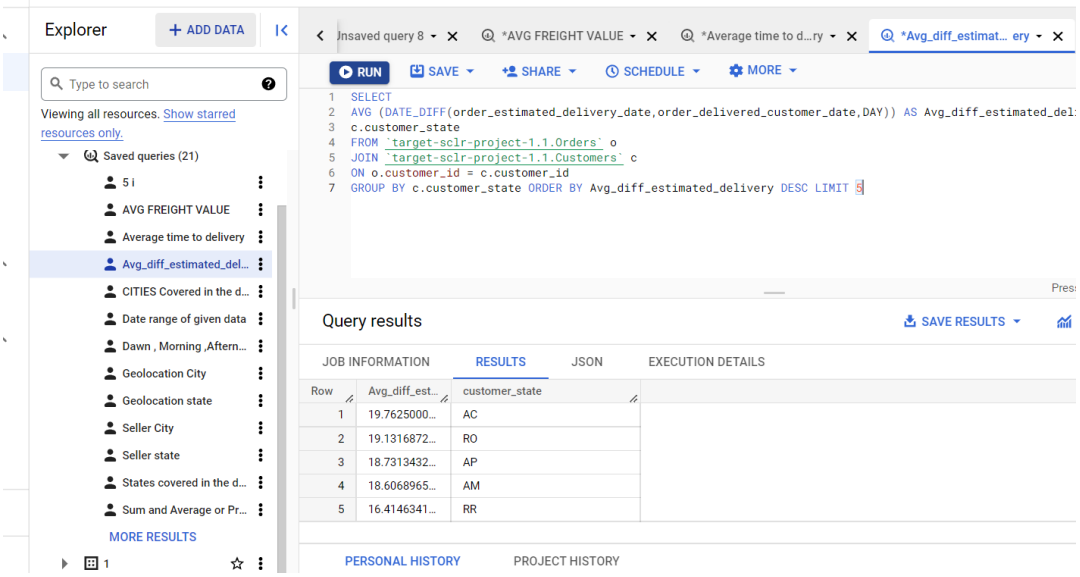
SQL Query :

```
  SELECT

  AVG
(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS Avg_time_to_delivery,

  c.customer_state

  FROM `target-sclr-project-1.1.Orders` o

  JOIN `target-sclr-project-1.1.Customers` c

  ON o.customer_id = c.customer_id

  GROUP BY c.customer_state ORDER BY Avg_time_to_delivery ASC LIMIT 5
```

From the results it can be seen that the state with the lowest average time to delivery is the state of Sao Paulo.

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

Top 5 states where delivery is really fast compared to estimated date



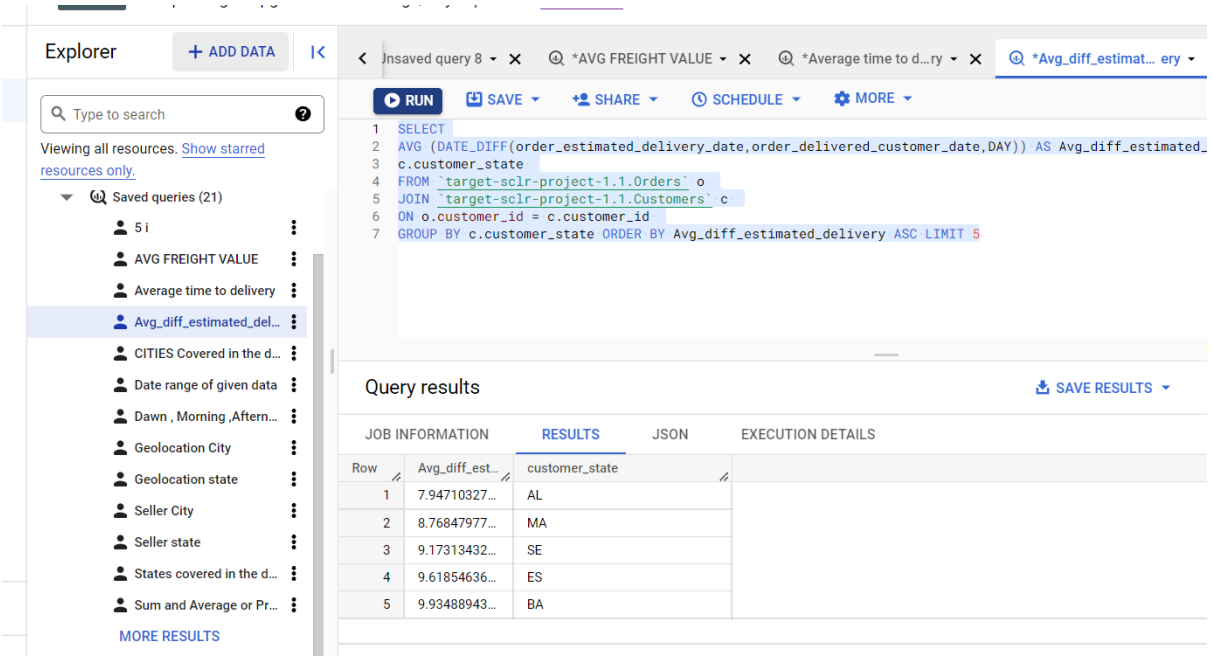
SQL Query :

```
SELECT

AVG
(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY))
AS Avg_diff_estimated_delivery,

c.customer_state

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

GROUP BY c.customer_state ORDER BY Avg_diff_estimated_delivery DESC LIMIT 5
```

From the result it can be seen that the state where delivery is really fast compared to estimated date is Acre .

Top 5 states where delivery is really slow compared to estimated date

The following query provides the list of top 5 states where delivery is really slow.



SQL Query :

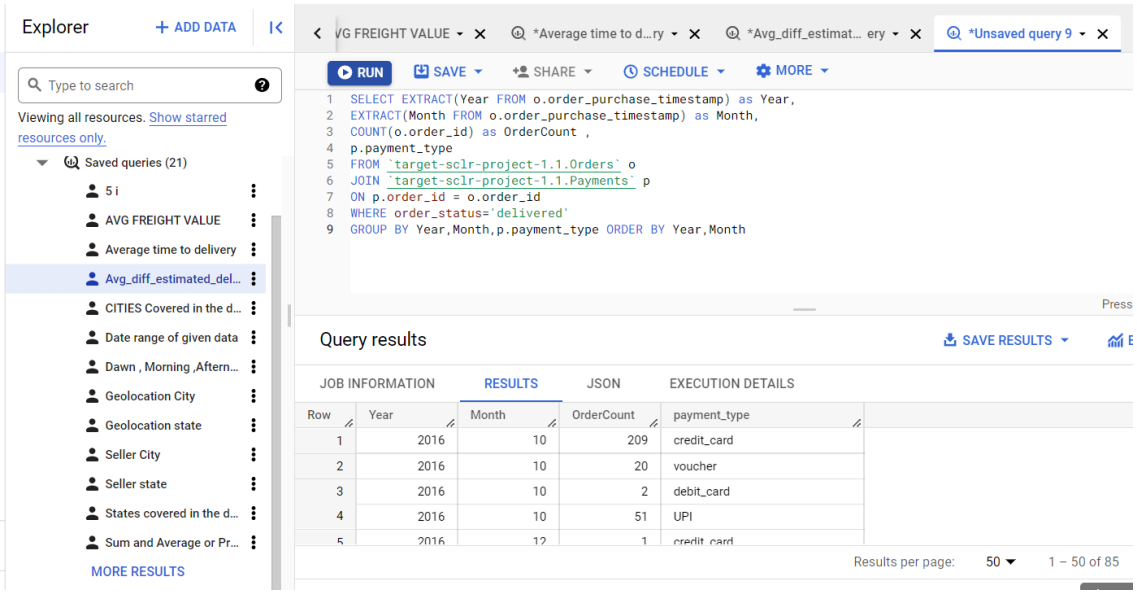
```
SELECT

AVG
(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY))
AS Avg_diff_estimated_delivery,

c.customer_state

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Customers` c

ON o.customer_id = c.customer_id

GROUP BY c.customer_state ORDER BY Avg_diff_estimated_delivery ASC LIMIT 5
```

From the result it can be seen that the state Alagoas is the state where the delivery is the slowest compared to estimated date

6.Payment type analysis:

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

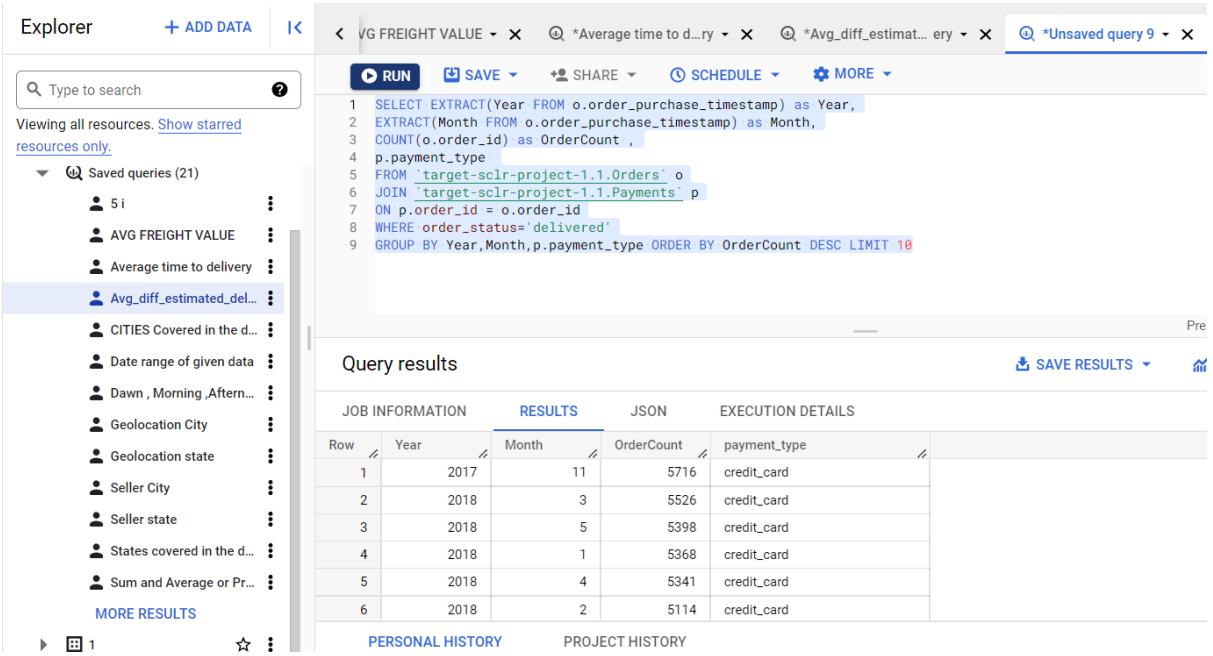
   The following query provides the month over month count of orders for different payment types .



SQL Query :

```sql
SELECT EXTRACT(Year FROM o.order_purchase_timestamp) as Year,

EXTRACT(Month FROM o.order_purchase_timestamp) as Month,

COUNT(o.order_id) as OrderCount ,

p.payment_type

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Payments` p

ON p.order_id = o.order_id

WHERE order_status='delivered'

GROUP BY Year,Month,p.payment_type ORDER BY Year,Month
```

By the running the same query and ordering the result by OrderCount in descending order and limiting the result to 10 it can be seen that the most used payment type is credit card as all the payment type for the top 10 results belong to the category of credit card .

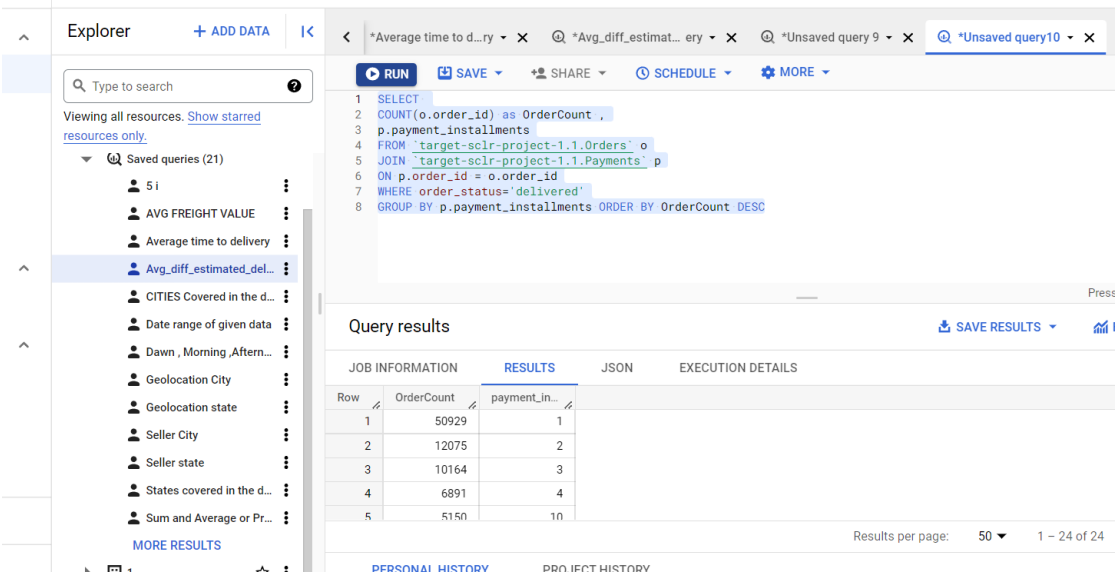


SQL Query :

```sql
SELECT EXTRACT(Year FROM o.order_purchase_timestamp) as Year,

EXTRACT(Month FROM o.order_purchase_timestamp) as Month,

COUNT(o.order_id) as OrderCount ,

p.payment_type

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Payments` p

ON p.order_id = o.order_id

WHERE order_status='delivered'

GROUP BY Year,Month,p.payment_type ORDER BY OrderCount DESC LIMIT 10
```

2. Distribution of payment instalments and count of orders

The following query provides the distribution of payment instalments and the respective count of orders .



SQL Query :

```
SELECT

COUNT(o.order_id) as OrderCount ,

p.payment_installments

FROM `target-sclr-project-1.1.Orders` o

JOIN `target-sclr-project-1.1.Payments` p

ON p.order_id = o.order_id

WHERE order_status='delivered'

GROUP BY p.payment_installments ORDER BY OrderCount DESC
```

From the results it can be seen that the instalment type of 1 has the highest orders with the count of orders as 50929 followed by instalments of 2 , 3 , 4 and 10 in the top 5 category .

This concludes the submission of the project .