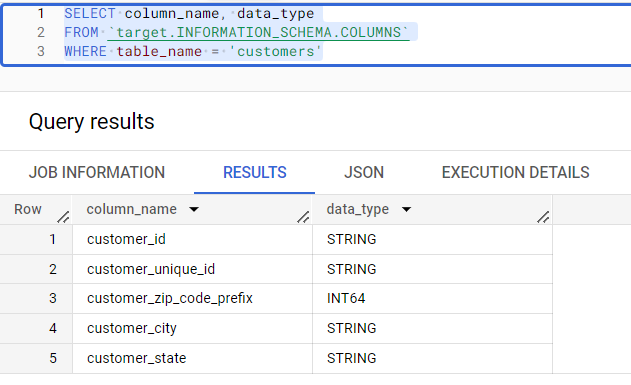
**Target SQL Case Study**

**Q1 :- Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

**Q1.1 Data type of all columns in the "customers" table.**

**Query:-**   
SELECT column\_name, data\_type  
FROM `target.INFORMATION\_SCHEMA.COLUMNS`  
WHERE table\_name = 'customers';

**Screenshot:-**



**Q1.2 Get the time range between which the orders were placed**

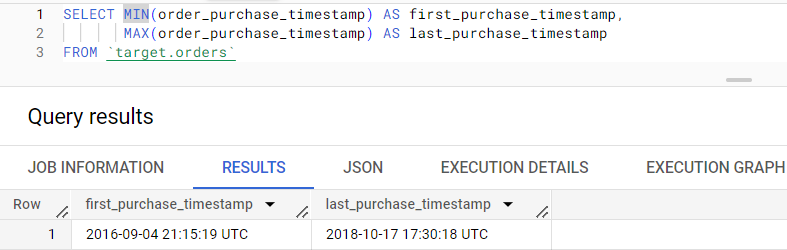
**Query:-**

SELECT   
MIN(order\_purchase\_timestamp) AS first\_purchase\_timestamp,

MAX(order\_purchase\_timestamp) AS last\_purchase\_timestamp

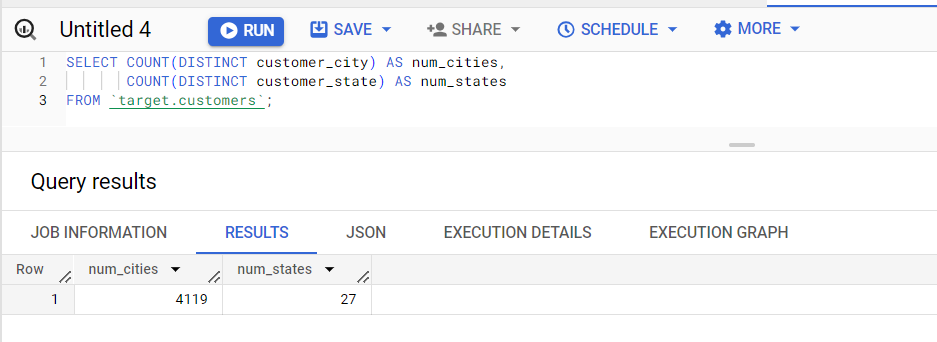
FROM `target.orders`;

**Screenshot:-**

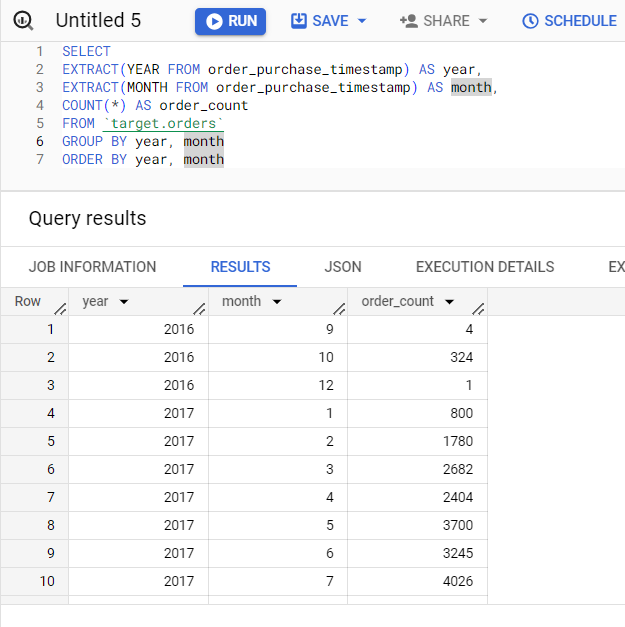


**Q1.3 Count the number of Cities and States in our dataset**

**Query:-**   
SELECT   
COUNT(DISTINCT customer\_city) AS num\_cities,  
COUNT(DISTINCT customer\_state) AS num\_states  
FROM `target.customers`;  
Screenshot:-



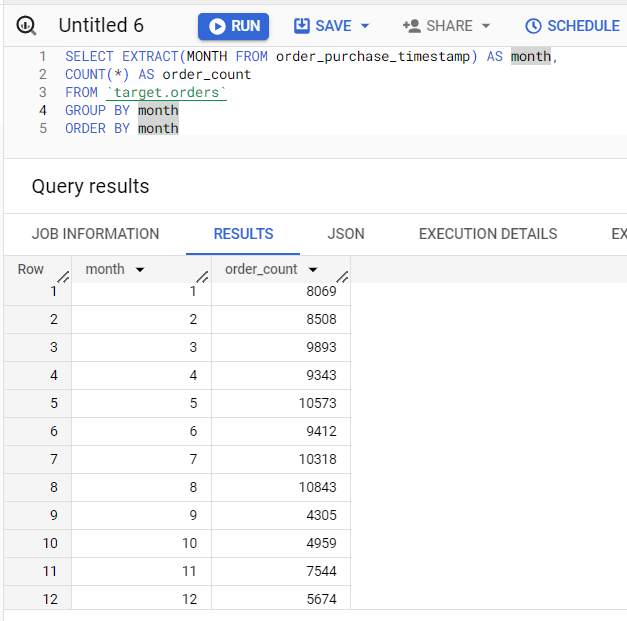
**Q2:-** **In-depth Exploration:  
Q2.1 :- Is there a growing trend in the no. of orders placed over the past years?**

**Query:-**  
SELECT   
EXTRACT(YEAR FROM order\_purchase\_timestamp) AS year,  
EXTRACT(MONTH FROM order\_purchase\_timestamp) AS month,  
COUNT(\*) AS order\_count  
FROM `target.orders`  
GROUP BY year, month  
ORDER BY year, month  
**Screenshot:-**   


**Analysis:-**

1. Max order placed in November 2017 i.e., 7544 orders.
2. Total order in   
   2016 = 329 , 2017 = 45101 , 2018 = 54011
3. We can conclude from the above data that no. of orders is increasing year over year

**Q2.2 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 month,  
COUNT(\*) AS order\_count  
FROM `target.orders`  
GROUP BY month  
ORDER BY month

Screenshot:-   


**Analysis :-**

1. Max order placed in August
2. From the above data we can conclude that the number of orders increases as we reach near the year mid (May, June, July, August), then we can see a decline in the orders during the year end.
3. The above data forms a bell curve having peak in mid year

**Q2.3 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 :-**

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'

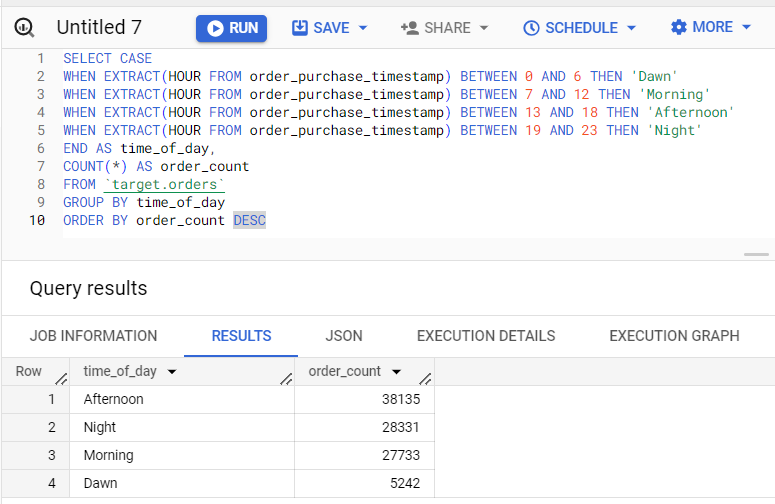
END AS time\_of\_day,

COUNT(\*) AS order\_count

FROM `target.orders`

GROUP BY time\_of\_day

ORDER BY order\_count DESC

**Screenshot :-**  


**Analysis :-**

1. From the above results we can conclude the Maximum orders are places in Afternoon
2. Followed by Night and Morning with a marginal difference of 1k orders
3. Least orders are placed in Dawn

**Note:-** Time zone of the given data set is not converted and used as it is. Incase we have to convert the time zone result may differ.

**Q3. Evolution of E-commerce orders in the Brazil region:**

**Q3.1 Get the month on month no. of orders placed in each state.**

**Query :-**

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

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

c.customer\_state,

COUNT(DISTINCT o.order\_id) AS order\_count

FROM `target.orders` AS o

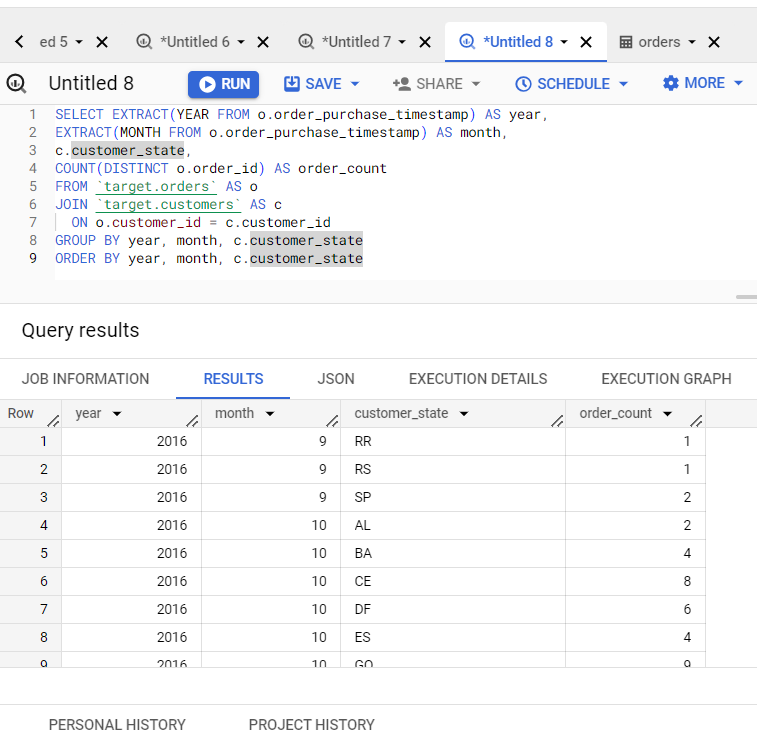
JOIN `target.customers` AS c

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

GROUP BY year, month, c.customer\_state

ORDER BY year, month, c.customer\_state

**Screenshot:-**



**Q3.2 How are the customers distributed across all the states?**

**Query:-**

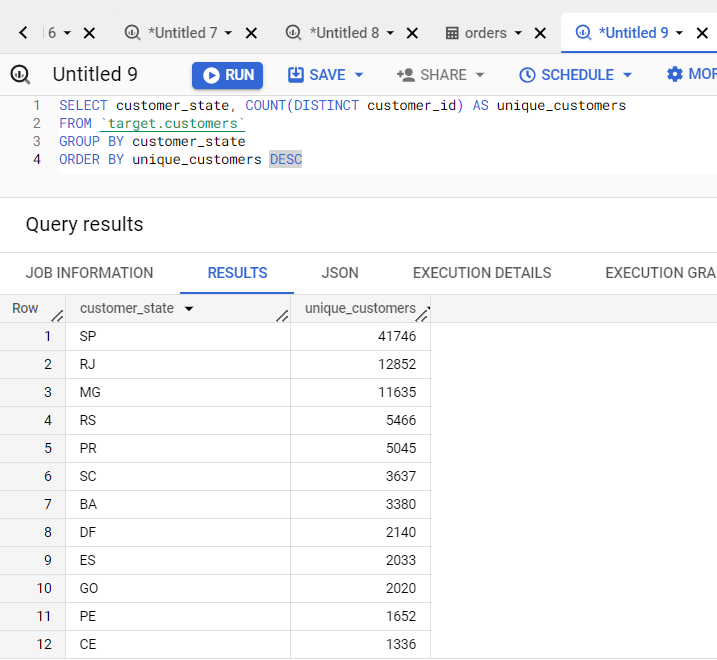
SELECT customer\_state, COUNT(DISTINCT customer\_id) AS unique\_customers

FROM `target.customers`

GROUP BY customer\_state

ORDER BY unique\_customers DESC

Screenshot:-



**Analysis:-**

1. Top three state in terms of customers SP, RJ and MG
2. Bottom three state in terms of customers are RR, AP and AC

**Q4.** **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**

**Q4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).**

**Query :-**

WITH orders\_2017 AS (  
SELECT  
    o.order\_id,  
    o.order\_purchase\_timestamp,    p.payment\_value  
FROM `target.orders` o  
JOIN `target.payments` p ON o.order\_id = p.order\_id  
WHERE  
EXTRACT(YEAR FROM o.order\_purchase\_timestamp) = 2017  
AND EXTRACT(MONTH FROM o.order\_purchase\_timestamp) BETWEEN 1 AND 8),

orders\_2018 AS (

SELECT

    o.order\_id,

    o.order\_purchase\_timestamp,

    p.payment\_value

FROM `target.orders` o

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

WHERE

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

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

SELECT

  (SUM(orders\_2018.payment\_value) - SUM(orders\_2017.payment\_value)) / SUM(orders\_2017.payment\_value) \* 100 AS percentage\_increase

FROM orders\_2017

FULL OUTER JOIN

orders\_2018 ON orders\_2017.order\_id = orders\_2018.order\_id;

**screenshot**



**Q4.2. Calculate the Total & Average value of order price for each state.**

**Query:-**

SELECT

  c.customer\_state,

  ROUND(SUM(oi.price),2) AS total\_order\_price,

  ROUND(AVG(oi.price),2) AS average\_order\_price

FROM `target.orders` o

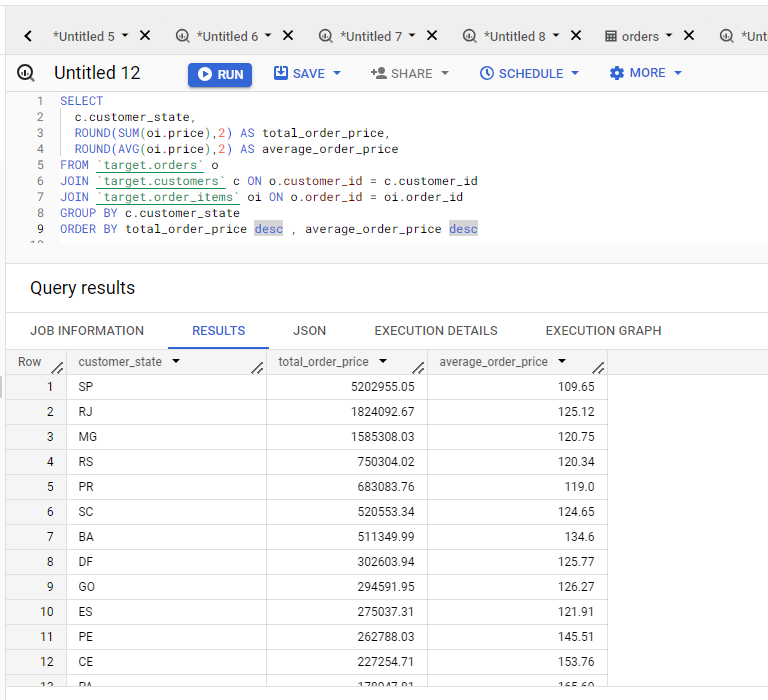
JOIN `target.customers` c ON o.customer\_id = c.customer\_id

JOIN `target.order\_items` oi ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY total\_order\_price desc , average\_order\_price desc

Screenshot:-



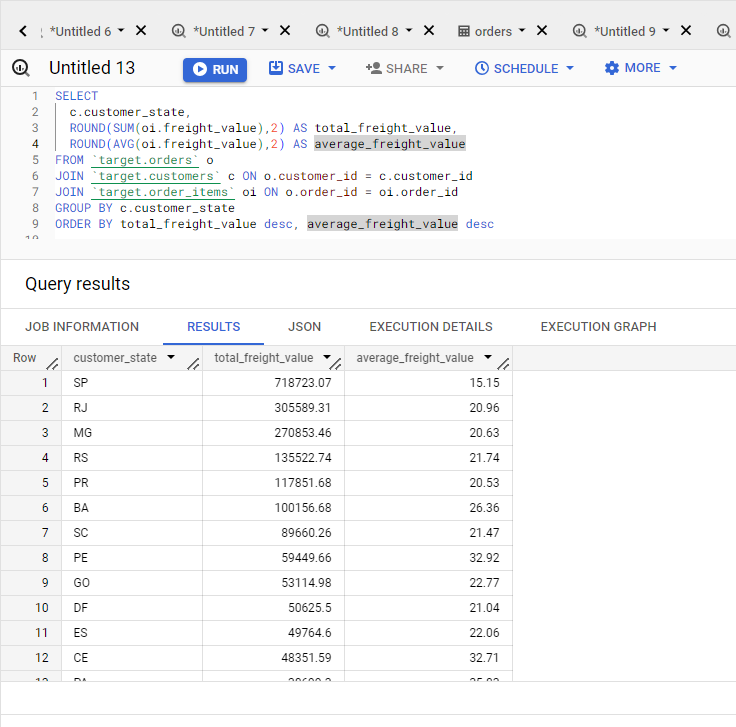
**Analysis :-**

1. Top three state in terms of max total order price are SP, RJ, MG
2. Top three state in terms of avg order price are PB, AL, AC
3. Bottom three state in terms of total order price are RR, AP, AC
4. Bottom three state in terms of avg order price are SP, PR, RS

**Q4.3 Calculate the Total & Average value of order freight for each state**

**Query :-**

SELECT  
  c.customer\_state,  
  ROUND(SUM(oi.freight\_value),2) AS total\_freight\_value,  
  ROUND(AVG(oi.freight\_value),2) AS average\_freight\_value  
FROM `target.orders` o  
JOIN `target.customers` c ON o.customer\_id = c.customer\_id  
JOIN `target.order\_items` oi ON o.order\_id = oi.order\_id  
GROUP BY c.customer\_state  
ORDER BY total\_freight\_value desc, average\_freight\_value desc

**Screenshot:-**   


**Analysis :-**

1. Top three state in terms of max total freight value are SP, RJ, MG
2. Top three state in terms of avg freight value are RR, PB, RO
3. Bottom three state in terms of total freight value are RR, AP, AC
4. Bottom three state in terms of avg freight value are SP, PR, MG

**Q5**. **Analysis based on sales, freight and delivery time.**

**Q5.1 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.**

**Query :-**

SELECT order\_id,

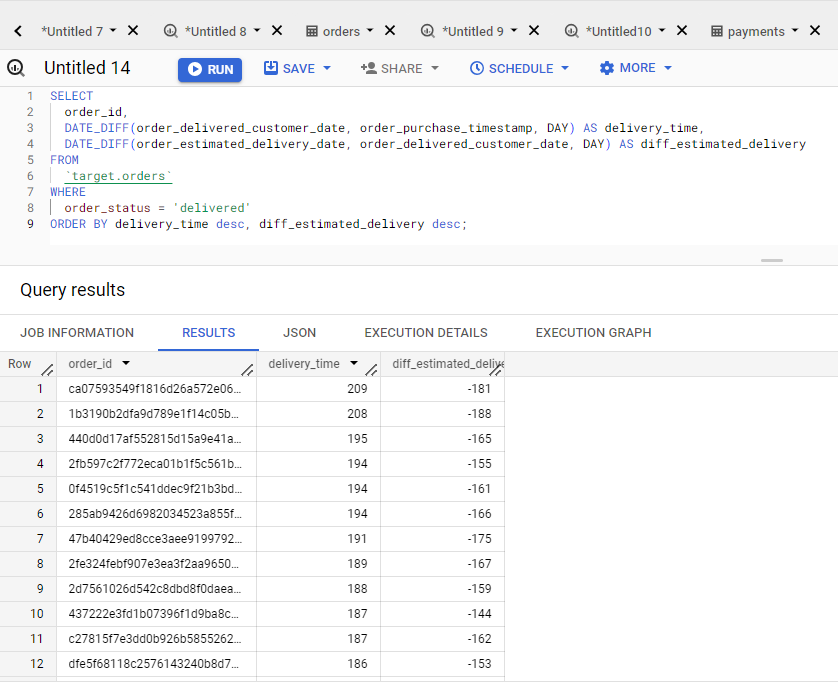
DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY) AS delivery\_time,

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

FROM `target.orders`

WHERE order\_status = 'delivered'

ORDER BY delivery\_time desc, diff\_estimated\_delivery desc;

**Screenshot :-**  


**Analysis :-**

1. Maximum delivery time = 209 days
2. Minimum delivery time = 0 days
3. Maximum diff estimated delivery = 146 days
4. Minimum diff estimated delivery = -188 days (delivered before time)

**Q5.2 Find out the top 5 states with the highest & lowest average freight value.**

1. **Highest**

**Query**

SELECT

  customer\_state,

  AVG(freight\_value) AS average\_freight

FROM

  `target.orders` o

JOIN

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

JOIN

  `target.order\_items` oi ON o.order\_id = oi.order\_id

GROUP BY

  customer\_state

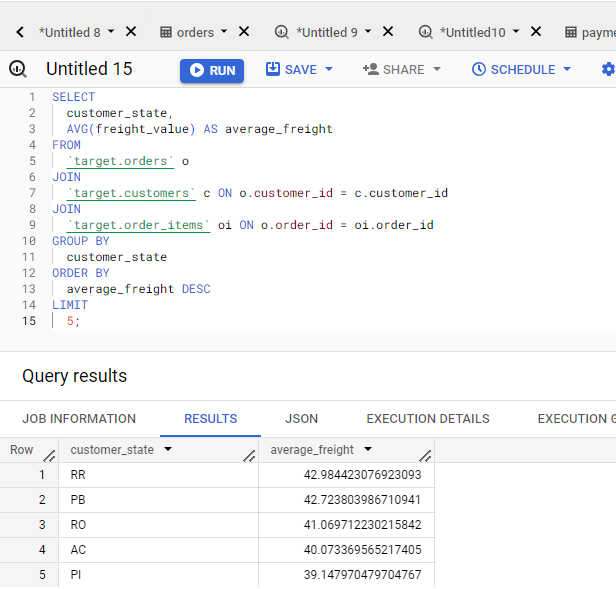
ORDER BY

  average\_freight DESC

LIMIT

  5;

**Screenshot**



1. **Lowest**

**Query :-**

SELECT

  customer\_state,

  AVG(freight\_value) AS average\_freight

FROM

  `target.orders` o

JOIN

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

JOIN

  `target.order\_items` oi ON o.order\_id = oi.order\_id

GROUP BY

  customer\_state

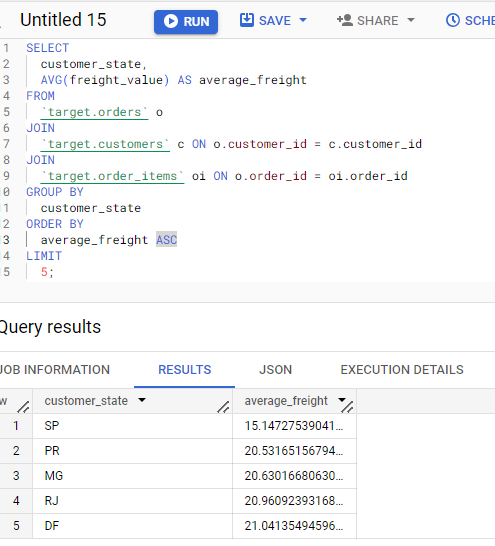
ORDER BY

  average\_freight ASC

LIMIT

  5;

**Screenshot:-**



**Q 5.3 Find out the top 5 states with the highest & lowest average delivery time.**

**A. highest**

**Query :-**

SELECT

  customer\_state,

  AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS average\_delivery\_time

FROM

  `target.orders` o

JOIN

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

WHERE

  order\_status = 'delivered'

GROUP BY

  customer\_state

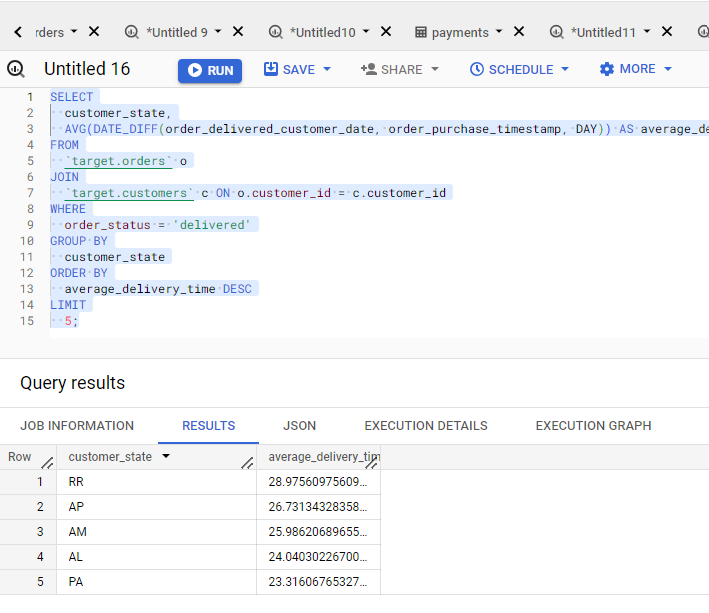
ORDER BY

  average\_delivery\_time DESC

LIMIT

  5;

**Screenshot:-**



**B. Lowest**

**Query :-**

SELECT

  customer\_state,

  AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS average\_delivery\_time

FROM

  `target.orders` o

JOIN

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

WHERE

  order\_status = 'delivered'

GROUP BY

  customer\_state

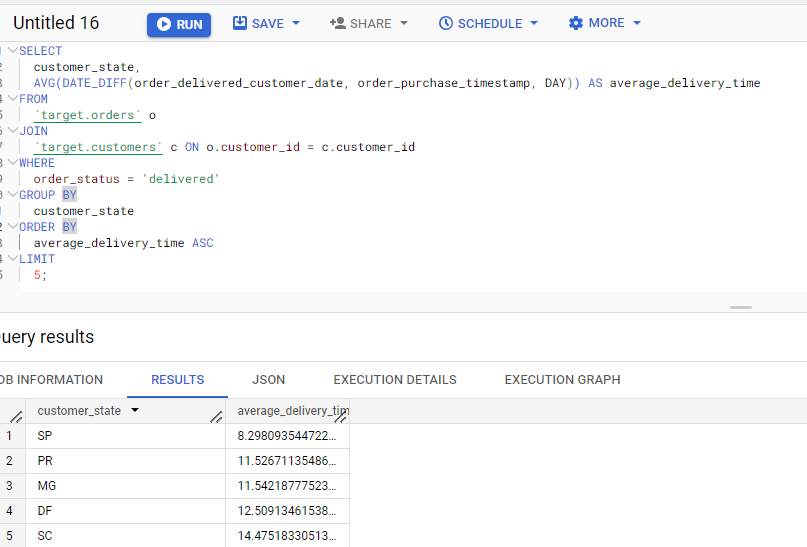
ORDER BY

  average\_delivery\_time ASC

LIMIT

  5;

**Screenshot:-**



**Q 5.4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**

**Query :-**

SELECT

  customer\_state,

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

FROM

  `target.orders` o

JOIN

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

WHERE

  order\_status = 'delivered'

GROUP BY

  customer\_state

HAVING

  diff\_average\_delivery > 0

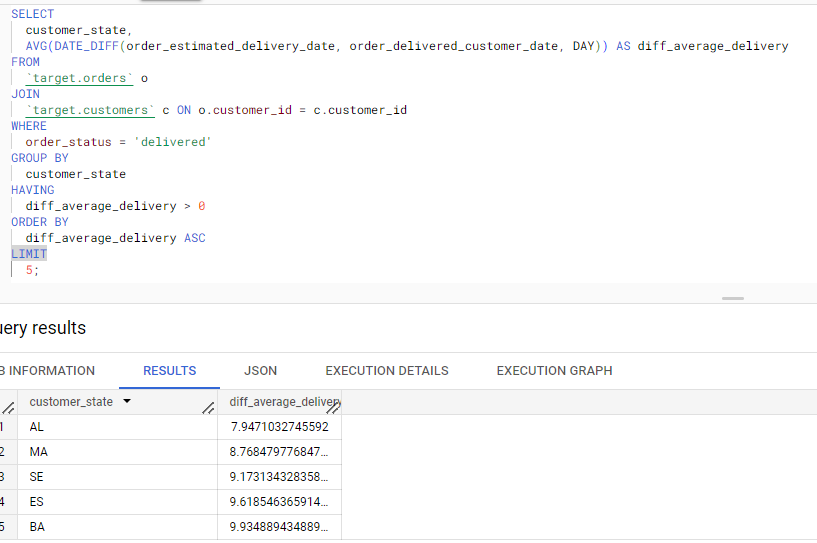
ORDER BY

  diff\_average\_delivery ASC

LIMIT

  5;

**Screenshot:-**



**Q6.** **Analysis based on the payments:**

**Q6.1 Find the month on month no. of orders placed using different payment types.**

**Query:-**

SELECT

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

  p.payment\_type,

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

FROM

  `target.orders` o

JOIN

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

GROUP BY

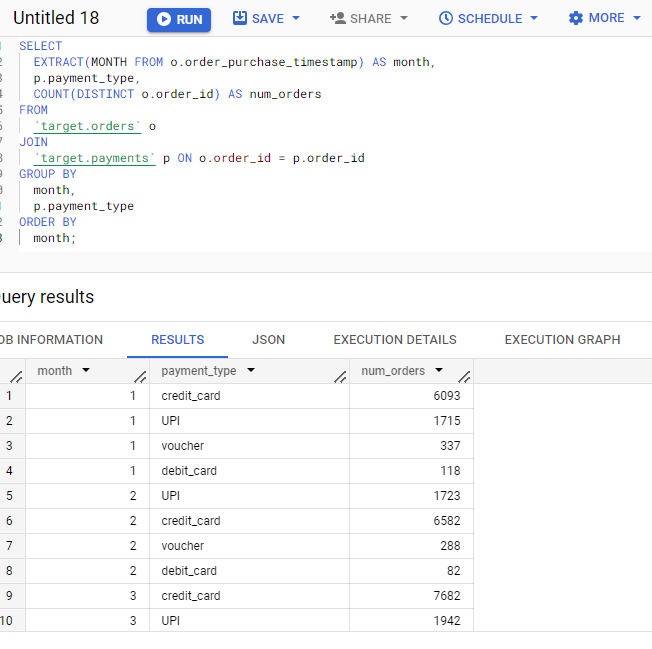
  month,

  p.payment\_type

ORDER BY

  month;

**Screenshot:-**



**Q6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.**

**Query**:-

SELECT

  payment\_installments,

  COUNT(DISTINCT order\_id) AS num\_orders

FROM

  `target.payments`

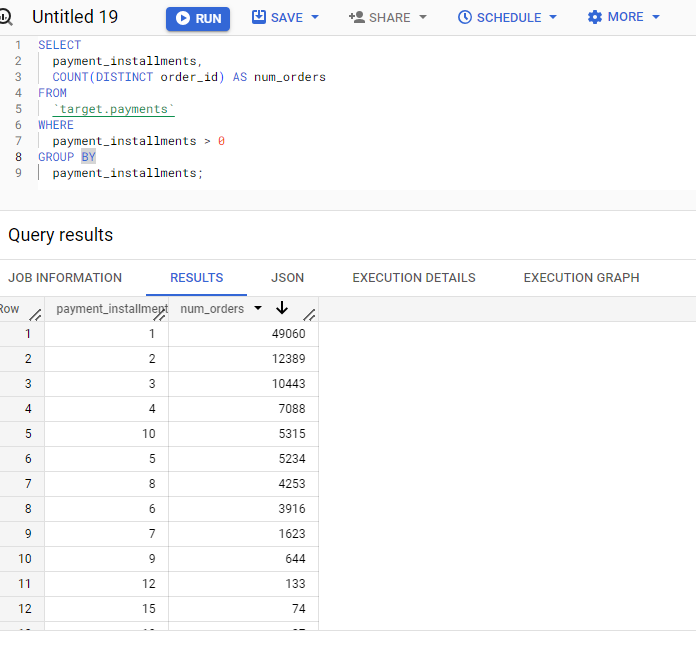
WHERE

  payment\_installments > 0

GROUP BY

  payment\_installments;

**Screenshot** :-



**Analysis :-**

1. Maximum order places with single instalment i.e., full payment
2. Minimum orders are placed with 22 and 23 instalments (single order each)