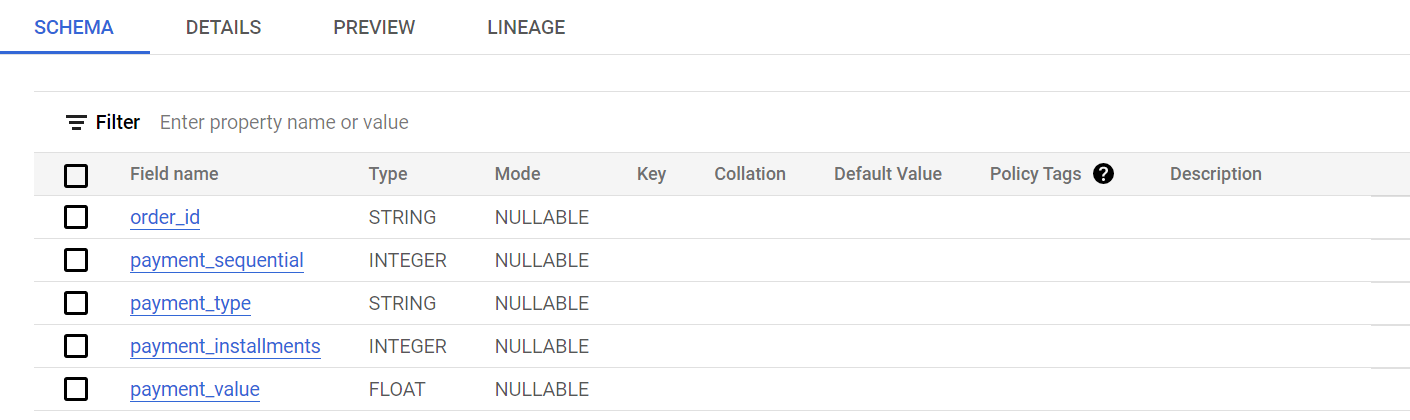
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset**
   1. ***Data type of columns in a table.***



* 1. ***Time period for which the data is given.***

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

SELECT

order\_id,

order\_purchase\_timestamp,

EXTRACT (YEAR FROM order\_purchase\_timestamp) as YEAR,

EXTRACT (QUARTER FROM order\_purchase\_timestamp) as QUARTER,

EXTRACT(MONTH FROM order\_purchase\_timestamp) as MONTH

FROM `Assignment.orders\_dataset`

ORDER BY order\_purchase\_timestamp

LIMIT 10

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

**Insights : -**

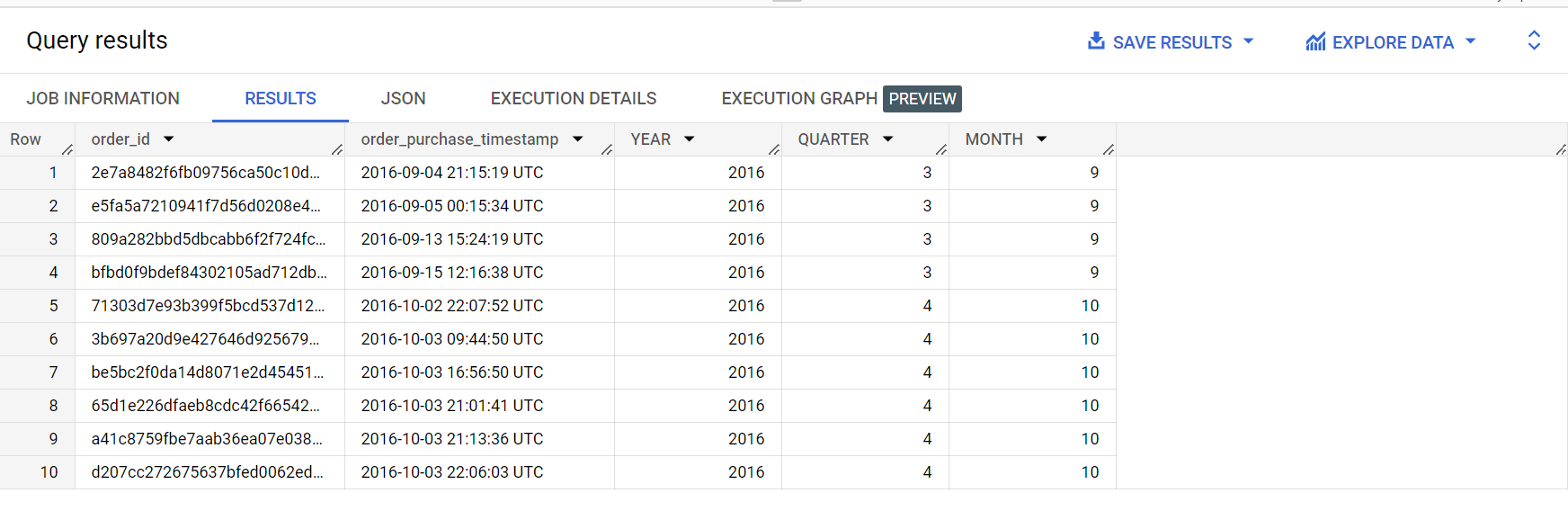
YEAR WISE data from 2016 -2018.

QUARTER WISE – 3rd Quarter in 2016 – 4th Quarter in 2018

Month Wise – Sep 2016 to Oct 2018

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

*Query result screen shot : -*

**

* 1. **Cities and States of customers ordered during the given period.**

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

SELECT

 c.customer\_id,

 order\_id,

 c.customer\_city,

 c.customer\_state

FROM `Assignment.customers\_dataset` c

JOIN `Assignment.orders\_dataset` as o

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

WHERE EXTRACT(YEAR FROM order\_purchase\_timestamp) BETWEEN 2016 and 2018

ORDER BY c.customer\_id

LIMIT 10

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

**Insights :-**

*Incorrect city names have resulted in different rows if data is grouped by city.*

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

**Query result screenshot : -**



1. **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?

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

SELECT

 ROUND(SUM(p.payment\_value),2) AS total\_sum,

 EXTRACT(YEAR FROM order\_purchase\_timestamp) as Years

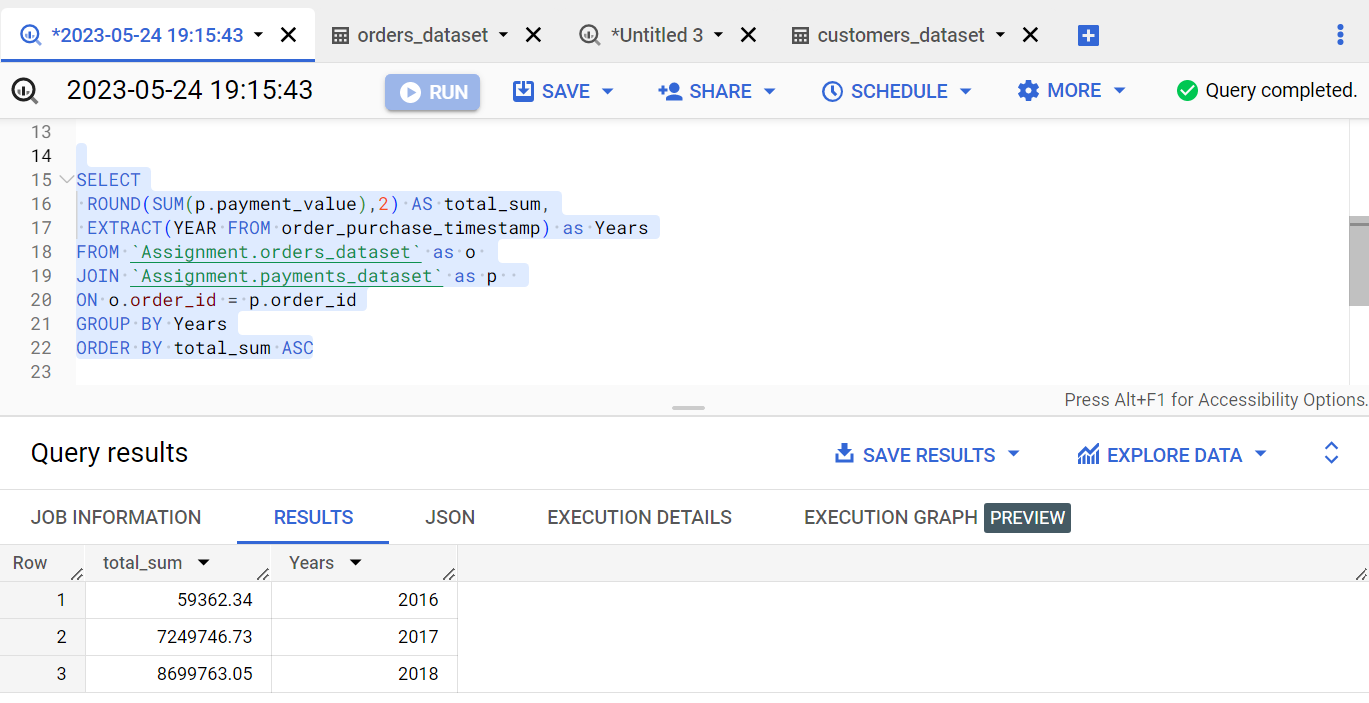
FROM `Assignment.orders\_dataset` as o

JOIN `Assignment.payments\_dataset` as p

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

GROUP BY Years

ORDER BY total\_sum ASC

**

SELECT

 COUNT( DISTINCT c.customer\_id) AS total\_num\_of\_customers\_per\_year,

 EXTRACT(YEAR FROM order\_purchase\_timestamp) as Years

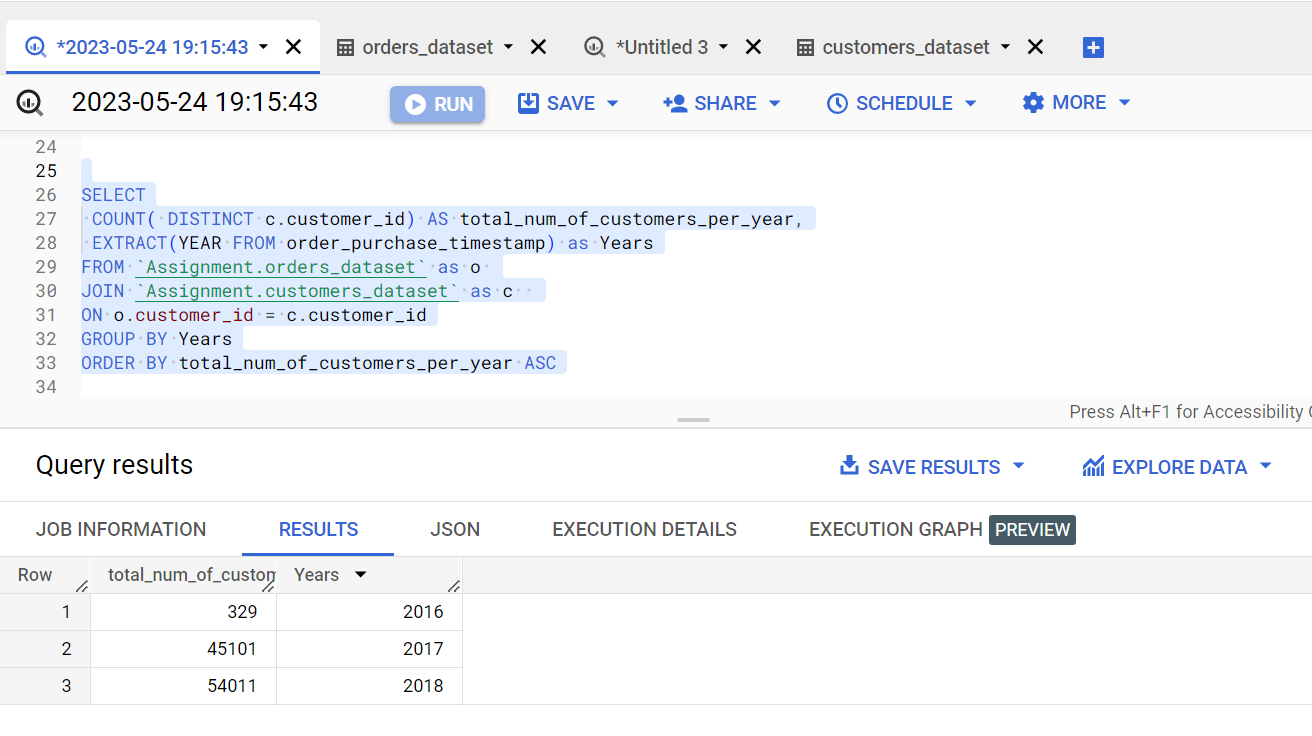
FROM `Assignment.orders\_dataset` as o

JOIN `Assignment.customers\_dataset` as c

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

GROUP BY Years

ORDER BY total\_num\_of\_customers\_per\_year ASC

**

**Insights :-**

*There is a definite increase in e commerce over the duration on which data is processed. We can see an increase in total number of customers per year as well as increase in total sum of payments per year.*

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

SELECT

 EXTRACT(YEAR from order\_purchase\_timestamp) as YEAR,

 EXTRACT(MONTH from order\_purchase\_timestamp) as MONTH,

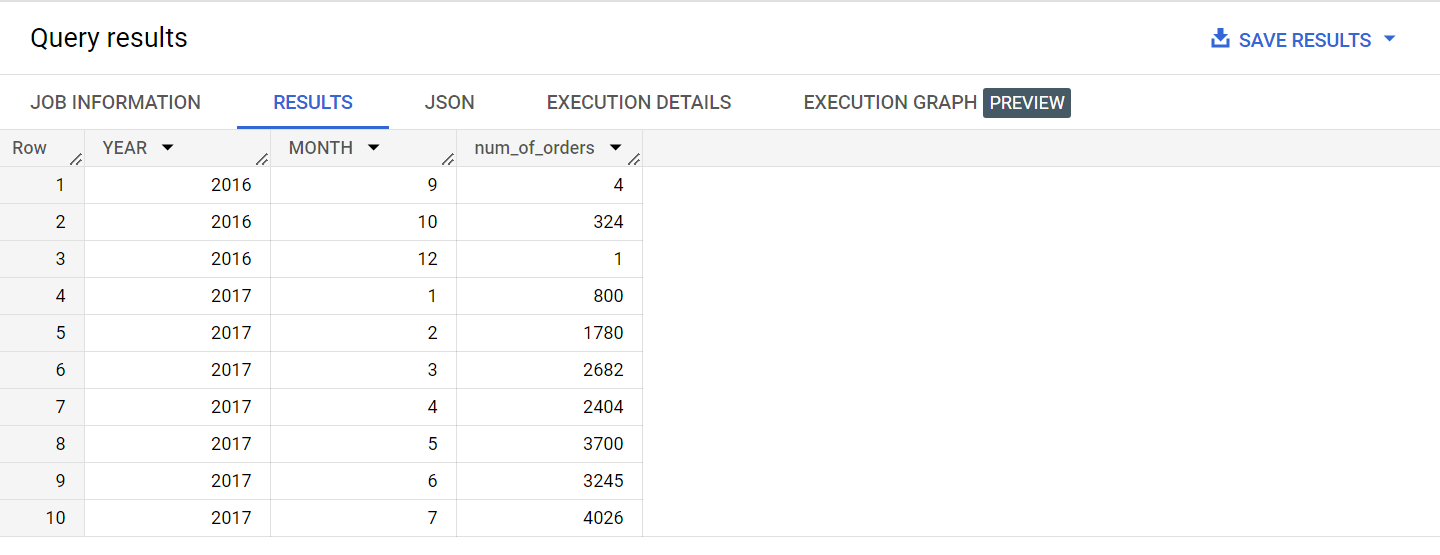
 COUNT(order\_id) as num\_of\_orders

FROM `Assignment.orders\_dataset` as o

GROUP BY YEAR, MONTH

ORDER BY YEAR, MONTH

LIMIT 10

**

**Insights :-**

*With the growth of e commerce in Brazil from 2016 to 2018 more users start to place orders. Hence we see an increase in number of orders each year with the month of Nov 2017 hitting the peak in terms of total orders placed.*

*Max orders each year are placed during the 3nd and 4th Quarter i.e Between the Jul- Dec*

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

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

SELECT

 COUNT(d.order\_id) as num\_orders,

 d.Part\_of\_day

FROM

(

SELECT

 order\_id,

 order\_purchase\_timestamp,

 CASE

    WHEN EXTRACT(TIME FROM order\_purchase\_timestamp) BETWEEN "05:00:00" AND "05:30:00"

    THEN "Dawn"

    WHEN EXTRACT(TIME FROM order\_purchase\_timestamp) BETWEEN "06:00:00" AND "11:59:59"

    THEN "Morning"

    WHEN EXTRACT(TIME FROM order\_purchase\_timestamp) BETWEEN "1:00:00" AND "16:59:59"

    THEN "Afternoon"

    WHEN EXTRACT(TIME FROM order\_purchase\_timestamp) BETWEEN "1:00:00" AND "18:59:59"

    THEN "Evening"

    ELSE "Night"

 END AS Part\_of\_day

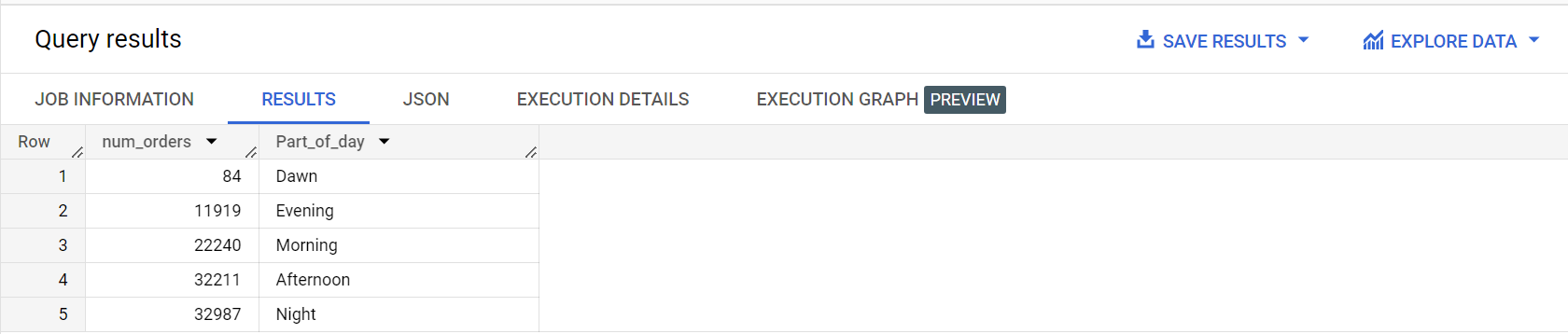
FROM `Assignment.orders\_dataset`

ORDER BY order\_purchase\_timestamp

) as d

GROUP BY d.Part\_of\_day

ORDER BY num\_orders



**Insights : -**

Max customers tend to buy during afternoon and night.

1. ***Evolution of E-commerce orders in the Brazil region:***
   1. Get month on month orders by states.

**Month on month data comparison among all states.**

SELECT

EXTRACT(MONTH from order\_purchase\_timestamp) as MONTH,

c.customer\_state as States,

COUNT(order\_id) as num\_of\_orders

FROM `Assignment.orders\_dataset` as o

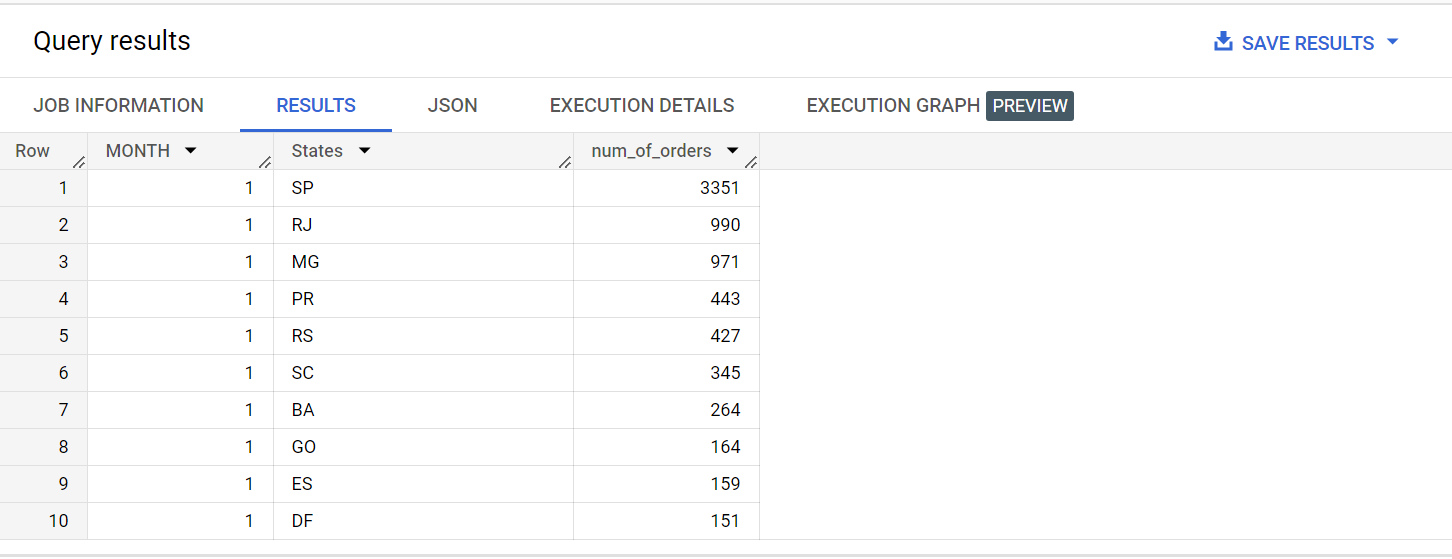
JOIN `Assignment.customers\_dataset` as c

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

GROUP BY States, MONTH

ORDER BY MONTH,num\_of\_orders DESC

LIMIT 10

**

***States : SP, RJ and MG lead in total number of orders per month for each month when data checked over total duration available.***

**Month on month data comparison for each state separately.**

SELECT\*,

 LAG(d.num\_of\_orders,1) OVER (PARTITION BY d.States ORDER BY d.MONTH) as prev\_month\_order,

 LEAD(d.num\_of\_orders,1) OVER (PARTITION BY d.States ORDER BY d.MONTH) as next\_month\_order

FROM

(

SELECT DISTINCT

c.customer\_state as States,

EXTRACT(MONTH from order\_purchase\_timestamp) as MONTH,

COUNT(order\_id) as num\_of\_orders,

FROM `Assignment.orders\_dataset` as o

JOIN `Assignment.customers\_dataset` as c

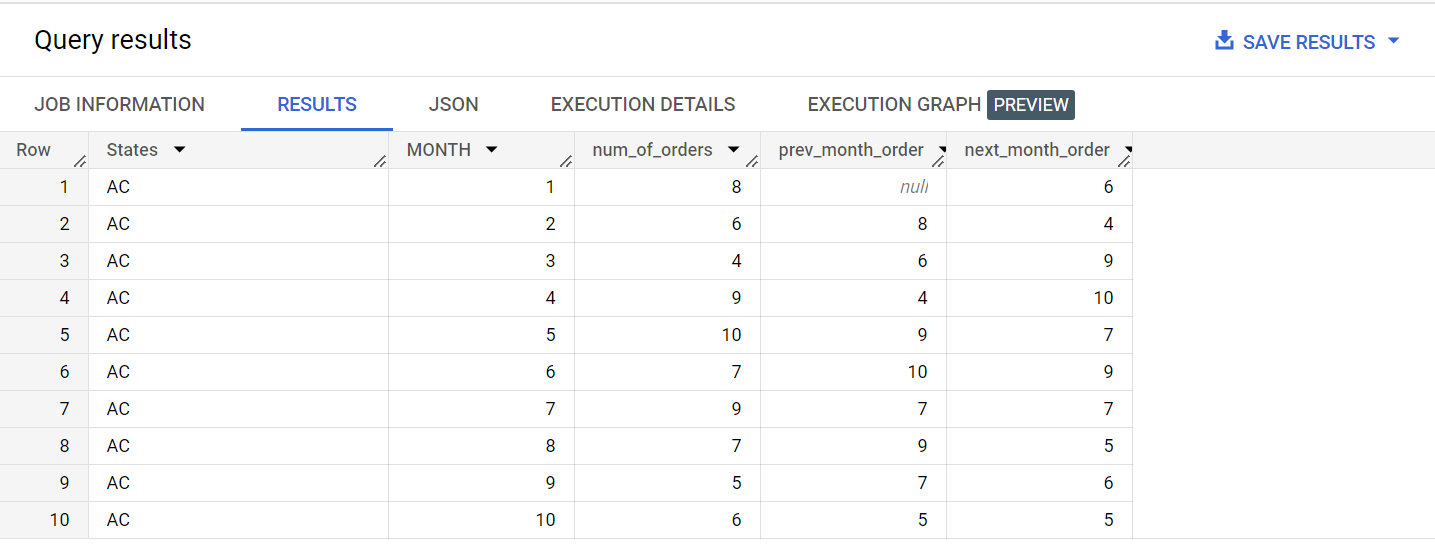
ON c.customer\_id = o.customer\_id

GROUP BY States, MONTH

) as d

ORDER BY d.States, d.MONTH

LIMIT 10

**

* 1. **Distribution of customers across the states in Brazil.**

SELECT

 COUNT(DISTINCT customer\_id) as num\_of\_customers\_per\_state,

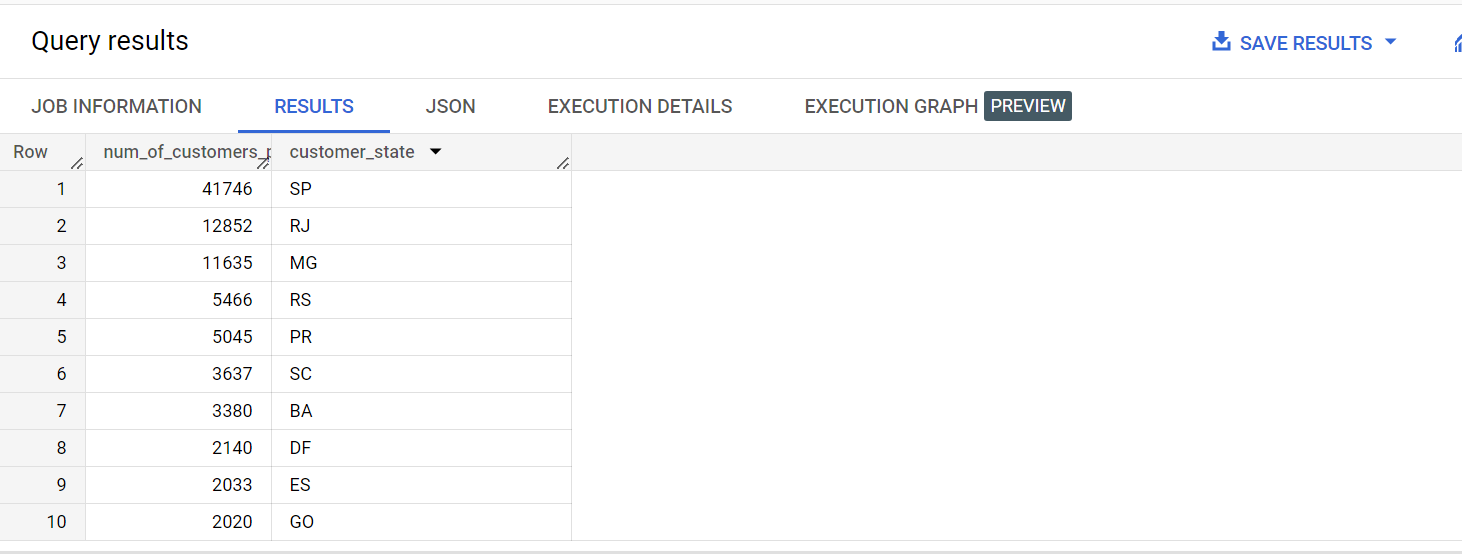
 customer\_state

FROM `Assignment.customers\_dataset`

GROUP BY customer\_state

ORDER BY num\_of\_customers\_per\_state DESC

LIMIT 10

**

1. ***Impact on Economy: Analyze the money movement 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) - You can use “payment\_value” column in payments table**

SELECT \*,

 ROUND(((e.total\_payments - e.previous\_year\_cost)/e.total\_payments) \* 100 ,2) as cost\_inc\_precentage

FROM

(

SELECT \*,

 LAG(d.total\_payments,1) OVER (ORDER BY d.Year) as previous\_year\_cost

FROM

(

SELECT

 EXTRACT(YEAR from order\_purchase\_timestamp) as Year,

 SUM(payment\_value) as total\_payments

FROM `Assignment.orders\_dataset` as o

JOIN `Assignment.payments\_dataset` as p

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

WHERE EXTRACT(YEAR from order\_purchase\_timestamp) IN(2017,2018) AND

EXTRACT(MONTH from order\_purchase\_timestamp) BETWEEN 1 AND 8

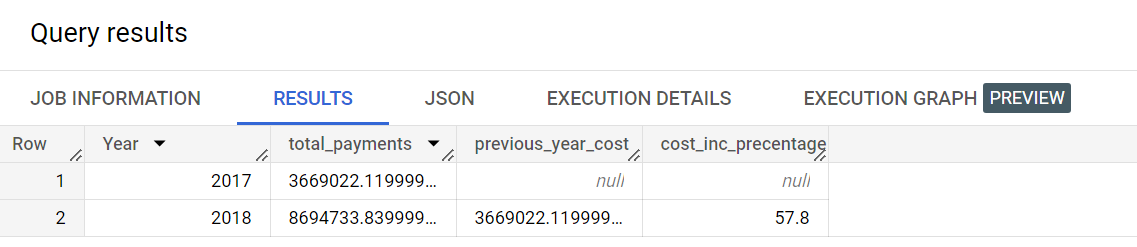
GROUP BY Year

ORDER BY Year

) as d

) as e

ORDER BY e.Year

**

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

SELECT

  c.customer\_state,

  SUM(oi.price) as sum\_price,

  AVG(oi.price) as mean\_price,

  SUM(oi.freight\_value) as freight\_value\_per\_state

FROM `Assignment.order\_items\_dataset` as oi

JOIN `Assignment.orders\_dataset` as o

ON oi.order\_id = o.order\_id

JOIN `Assignment.customers\_dataset` as c

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

GROUP BY c.customer\_state

ORDER BY c.customer\_state

LIMIT 10

**

***5. Analysis on sales, freight and delivery time***

SELECT

 order\_status,

 COUNT(order\_id) as num\_of\_orders

FROM `Assignment.orders\_dataset`

GROUP BY order\_status

ORDER BY order\_status

******

*We see that we have different number of orders based on their order status. It is found when the date in order\_dataset is ordered by order status. Only orders with status as delivered, shipped and some orders which were cancelled after the product was shipped have an actual date value* ***order\_deliver\_carrier\_date*** *populated for them in the table. Rest all orders have NULL. This means only the above mentioned orders have freight value populated for them. Hence in the below query while calculating freight value we have filtered the status condition to meet this requirement****.***

1. **Calculate days between purchasing, delivering and estimated delivery**

***SELECT***

***order\_status,***

***order\_purchase\_timestamp,***

***order\_estimated\_delivery\_date,***

***DATE\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp, DAY) as est\_del\_time,***

***order\_delivered\_customer\_date,***

***DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, DAY) as act\_del\_time,***

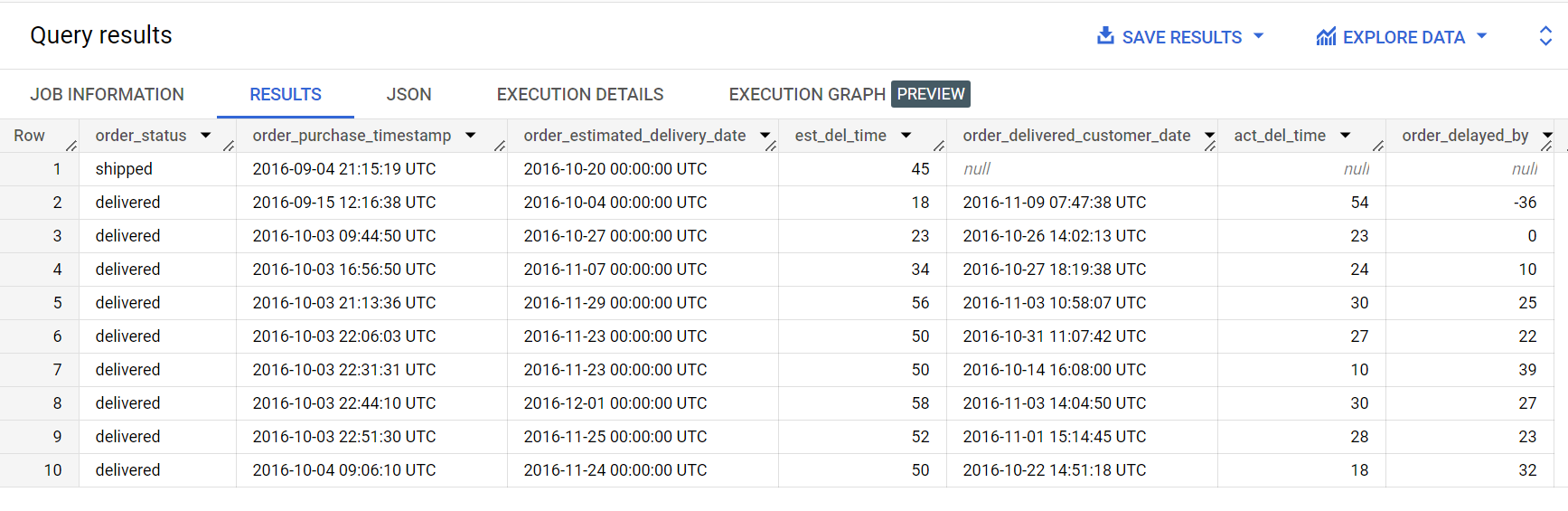
***DATE\_DIFF(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, DAY) as order\_delayed\_by***

***FROM `Assignment.orders\_dataset`***

***WHERE order\_status IN ("delivered", "shipped", "cancelled") AND order\_delivered\_carrier\_date IS NOT NULL***

***ORDER BY order\_purchase\_timestamp***

***LIMIT 10***



1. Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:
   * time\_to\_delivery = order\_delivered\_customer\_date-order\_purchase\_timestamp
   * diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date
2. **Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery.**

***SELECT***

***c.customer\_state,***

***ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp, DAY)),2) as time\_to\_delivery,***

***ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, DAY)),2) as diff\_estimated\_delivery,***

***ROUND(AVG(oi.freight\_value),2) as avg\_freight\_value***

***FROM `Assignment.orders\_dataset` as o***

***INNER JOIN `Assignment.order\_items\_dataset` as oi***

***ON o.order\_id = oi.order\_id***

***INNER JOIN `Assignment.customers\_dataset` as c***

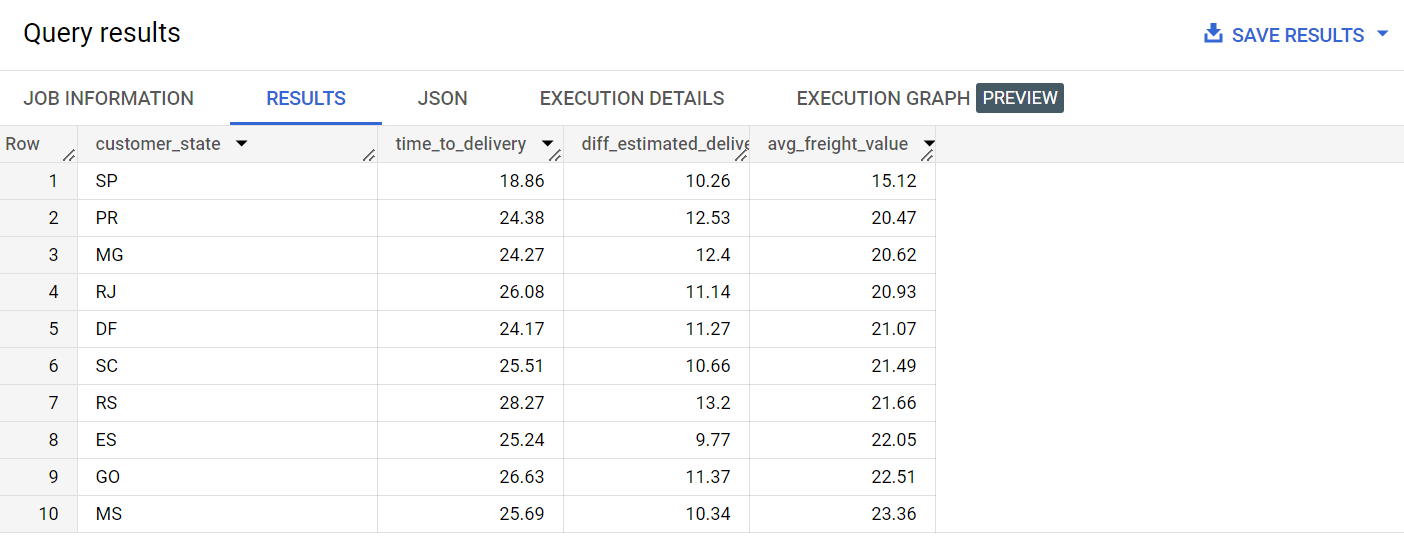
***ON o.customer\_id = c.customer\_id***

***WHERE order\_status IN ("delivered", "shipped", "cancelled") AND order\_delivered\_carrier\_date IS NOT NULL***

***GROUP BY c.customer\_state***

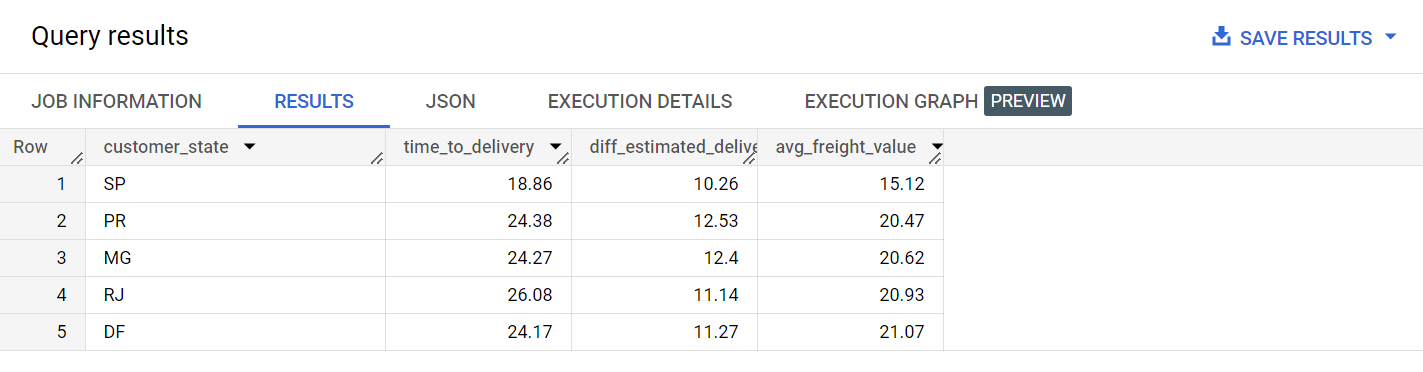
***ORDER BY avg\_freight\_value***

***LIMIT 10***



1. Sort the data to get the following:
2. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

***Same query as above except ORDER BY avg\_freight\_value DESC in last second line and LIMIT 5 after that.***



***Same query as above except ORDER BY avg\_freight\_value DESC in last second line and LIMIT 5 after that.***



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

***SELECT***

***c.customer\_state,***

***ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp, DAY)),2) as avg\_time\_to\_delivery,***

***ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, DAY)),2) as avg\_diff\_estimated\_delivery,***

***ROUND(AVG(oi.freight\_value),2) as avg\_freight\_value***

***FROM `Assignment.orders\_dataset` as o***

***INNER JOIN `Assignment.order\_items\_dataset` as oi***

***ON o.order\_id = oi.order\_id***

***INNER JOIN `Assignment.customers\_dataset` as c***

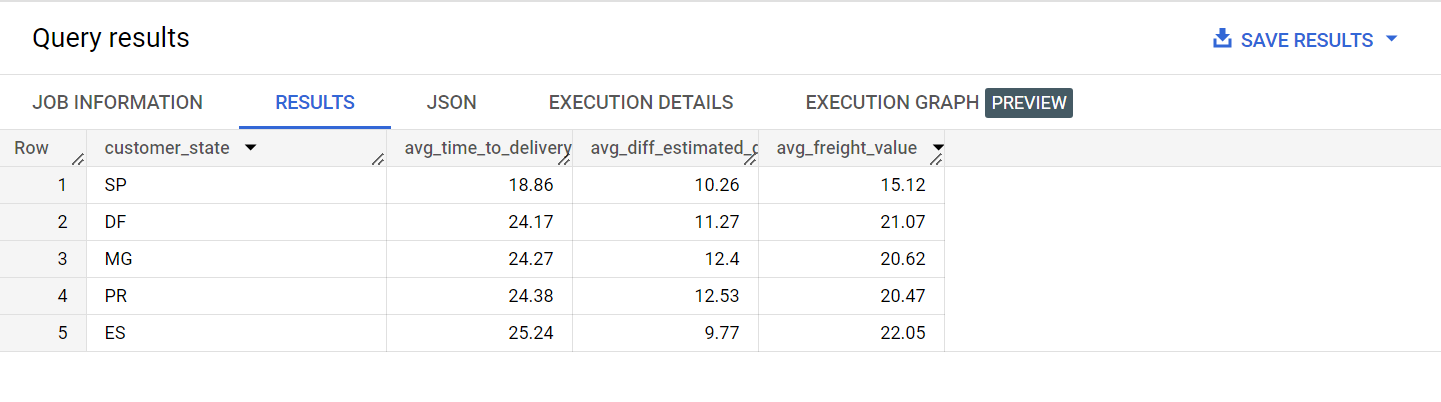
***ON o.customer\_id = c.customer\_id***

***WHERE order\_status IN ("delivered", "shipped", "cancelled") AND order\_delivered\_carrier\_date IS NOT NULL***

***GROUP BY c.customer\_state***

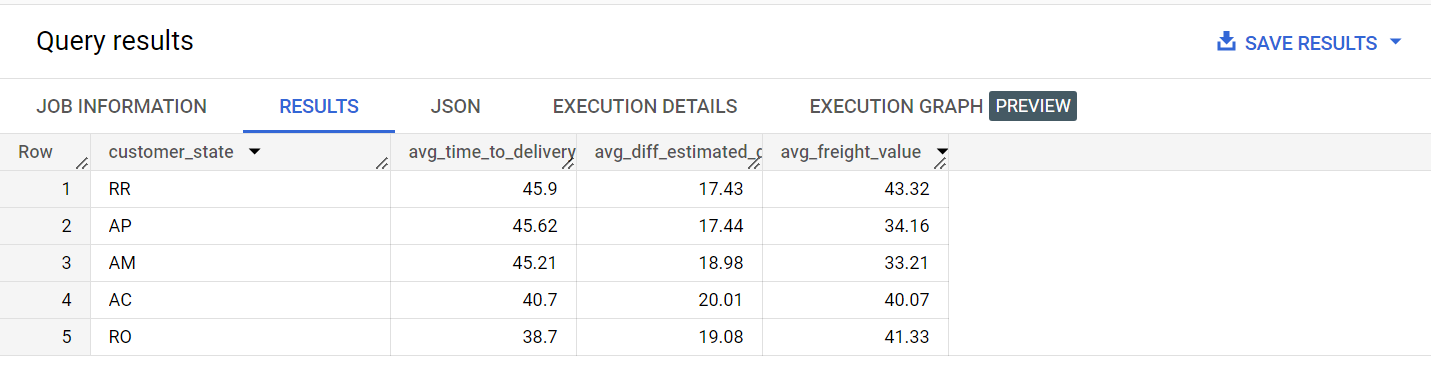
***ORDER BY avg\_time\_to\_delivery***

***LIMIT 5***

******

***Highest time to delivery. Same query as above, except ORDER BY avg\_time\_to\_delivery***

***DESC in last second line.***

******

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

***SELECT***

***c.customer\_state,***

***ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp, DAY)),2) as avg\_time\_to\_delivery,***

***ROUND(AVG(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, DAY)),2) as avg\_diff\_estimated\_delivery,***

***ROUND(AVG(oi.freight\_value),2) as avg\_freight\_value***

***FROM `Assignment.orders\_dataset` as o***

***INNER JOIN `Assignment.order\_items\_dataset` as oi***

***ON o.order\_id = oi.order\_id***

***INNER JOIN `Assignment.customers\_dataset` as c***

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

***WHERE order\_status IN ("delivered", "shipped", "cancelled") AND order\_delivered\_carrier\_date IS NOT NULL***

***GROUP BY c.customer\_state***

***ORDER BY avg\_diff\_estimated\_delivery***

***LIMIT 5***

******

***Delivery not so fast : -***

***Same query except ORDER BY avg\_diff\_estimated\_deliveryDESC in last second line.***

******

**6**. **Payment type analysis:**

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

***SELECT\*,***

***LAG(d.num\_orders,1) OVER (PARTITION BY d.payment\_type ORDER BY d.payment\_type,d.Month) as previous\_month\_orders,***

***LEAD(d.num\_orders,1) OVER (PARTITION BY d.payment\_type ORDER BY d.payment\_type,d.Month) as next\_month\_orders***

***FROM***

***(***

***SELECT***

***payment\_type,***

***EXTRACT(MONTH FROM o.order\_purchase\_timestamp) as Month,***

***COUNT(o.order\_id) as num\_orders***

***FROM `Assignment.payments\_dataset` as p***

***JOIN `Assignment.orders\_dataset` as o***

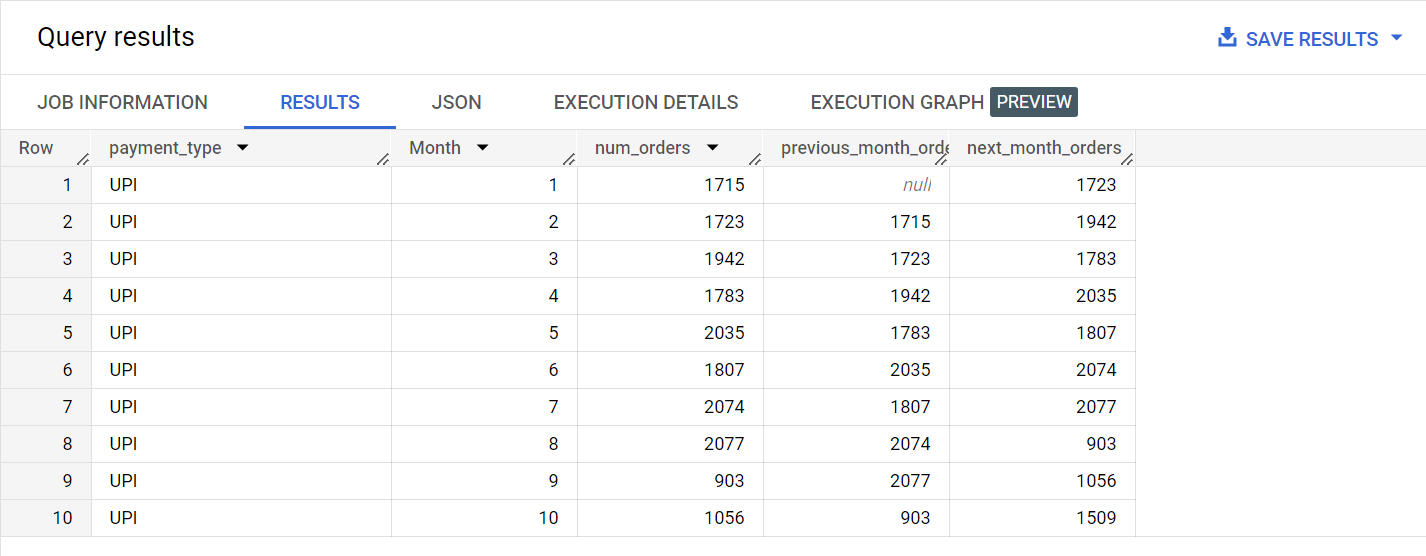
***ON o.order\_id = p.order\_id***

***GROUP BY payment\_type, Month***

***ORDER BY payment\_type, Month***

***) as d***

***ORDER by d.payment\_type, d.Month***

******

1. Count of orders based on the no. of payment instalments.

***SELECT***

***payment\_installments,***

***COUNT(o.order\_id) as num\_orders***

***FROM `Assignment.payments\_dataset` as p***

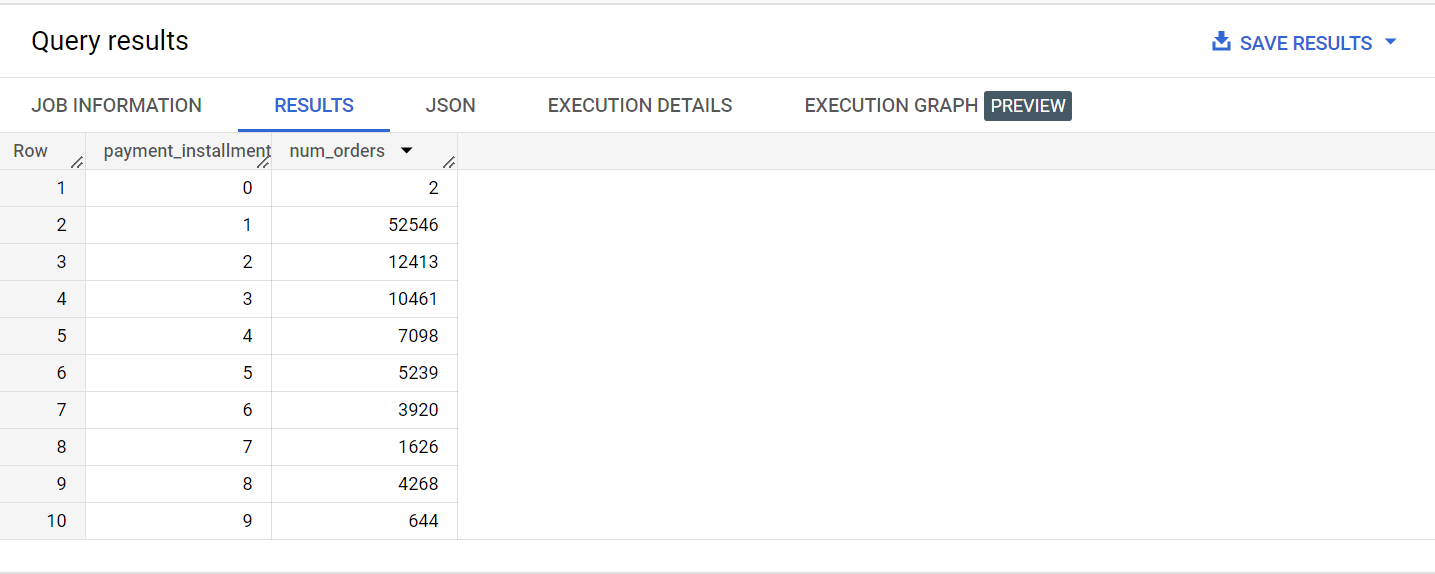
***JOIN `Assignment.orders\_dataset` as o***

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

***GROUP BY payment\_installments***

***ORDER BY payment\_installments***

***LIMIT 10***

******