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

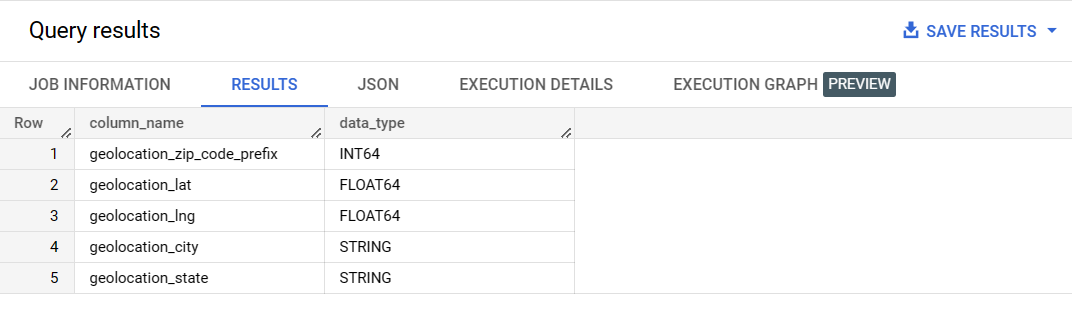
select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

where table\_name = 'customers'

----

--------------------------------------------------------------------------------------------------------------------------------

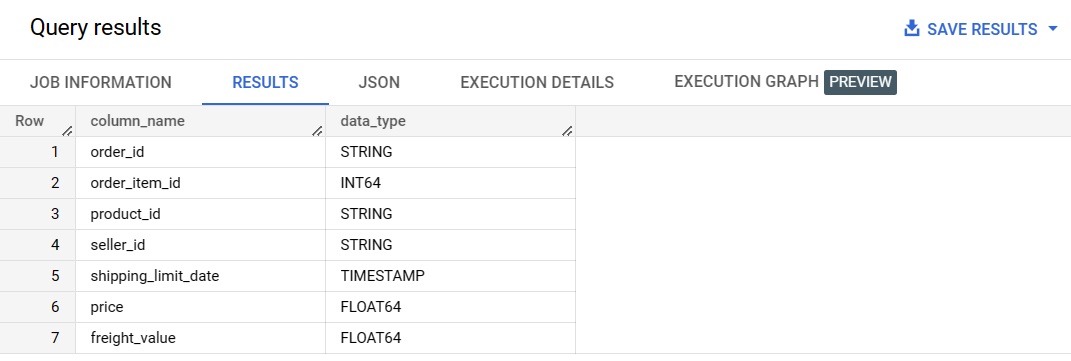
select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS` where table\_name = 'geolocation'



--------------------------------------------------------------------------------------------------------------------------------

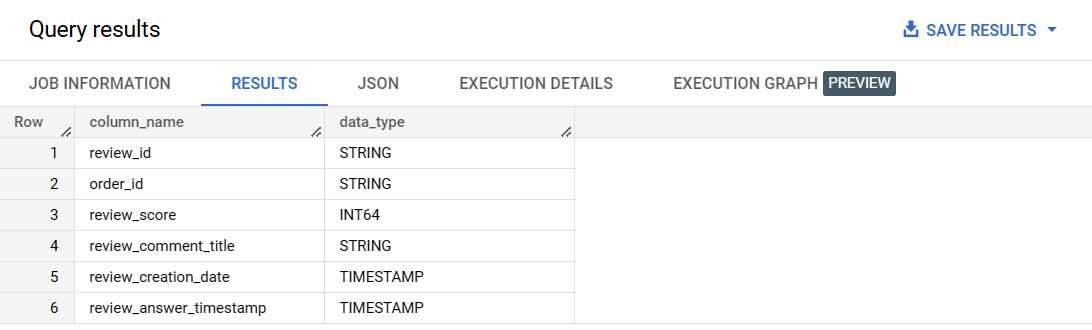
select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

where table\_name = 'order\_items'



--------------------------------------------------------------------------------------------------------------------------------select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

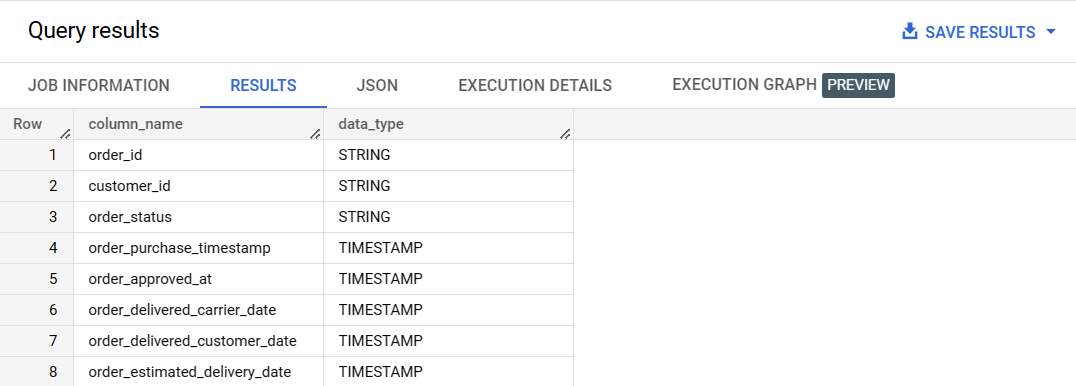
where table\_name = 'order\_reviews'



--------------------------------------------------------------------------------------------------------------------------------

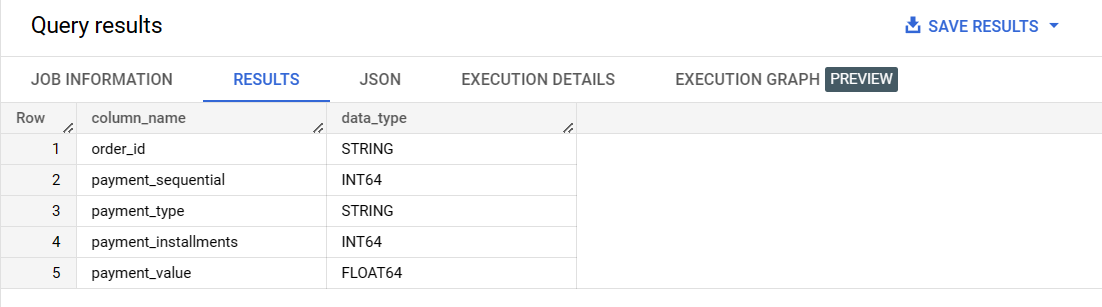
select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

where table\_name = 'orders'



select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

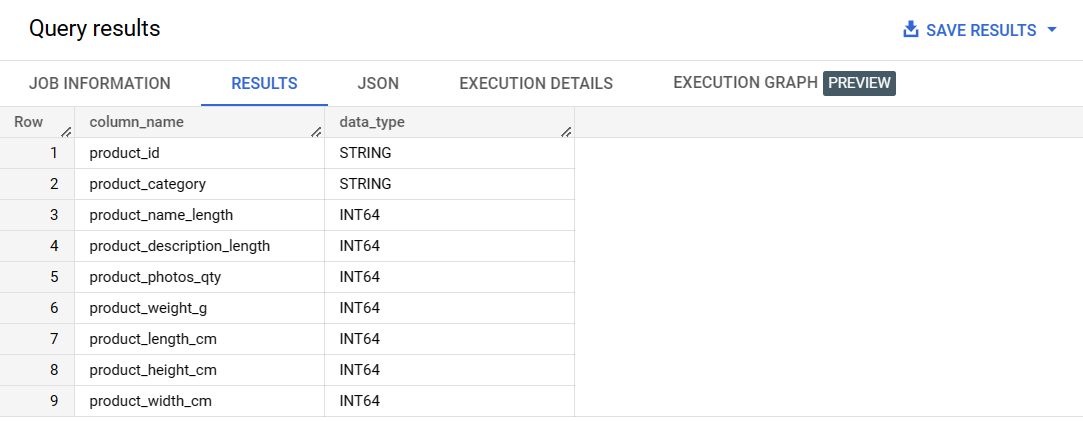
where table\_name = 'payments'



--------------------------------------------------------------------------------------------------------------------------------

select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

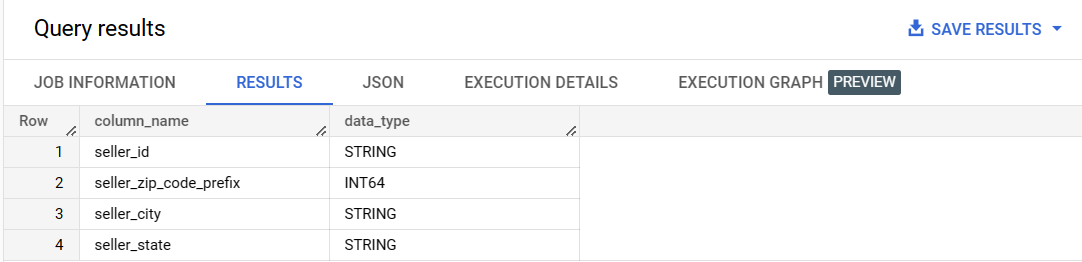
where table\_name = 'products'



--------------------------------------------------------------------------------------------------------------------------------

select column\_name, data\_type from `sqlbusinesscase-384615.SQLAssignement.INFORMATION\_SCHEMA.COLUMNS`

where table\_name = 'sellers'

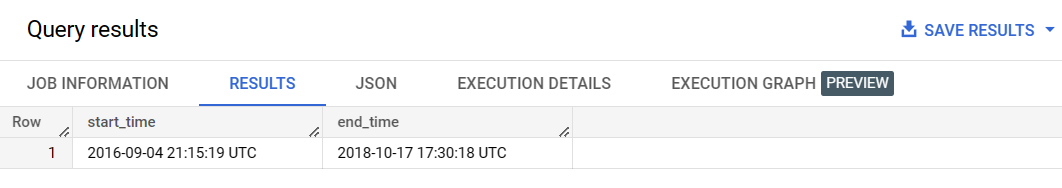


--------------------------------------------------------------------------------------------------------------------------------

1. Time period for which the data is given

select min(order\_purchase\_timestamp)as start\_time, max(order\_purchase\_timestamp) as end\_time

from `sqlbusinesscase-384615.SQLAssignement.orders`



**Insights :**

The time period is calculated based on the data in ‘Orders’ table. start\_time and end\_time gives the total period for which the data is given.

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

select c1.customer\_city, c1.customer\_state from `sqlbusinesscase-384615.SQLAssignement.customers` c1

join

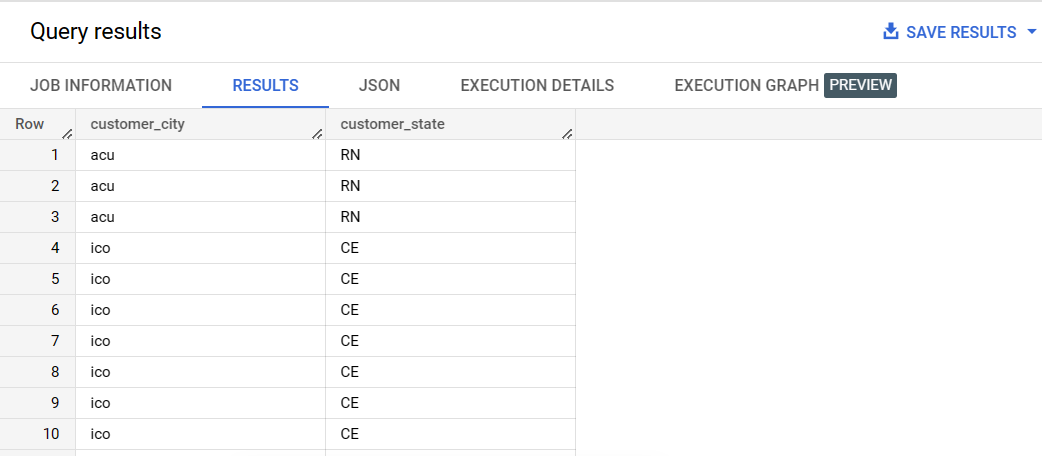
`sqlbusinesscase-384615.SQLAssignement.orders` ord

on c1.customer\_id = ord.customer\_id

where ord.order\_purchase\_timestamp  between

(select min(order\_purchase\_timestamp) from `sqlbusinesscase-384615.SQLAssignement.orders`) and

(select max(order\_purchase\_timestamp) from `sqlbusinesscase-384615.SQLAssignement.orders`)



--------------------------------------------------------------------------------------------------------------------------------

1. In-depth Exploration:
2. 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?

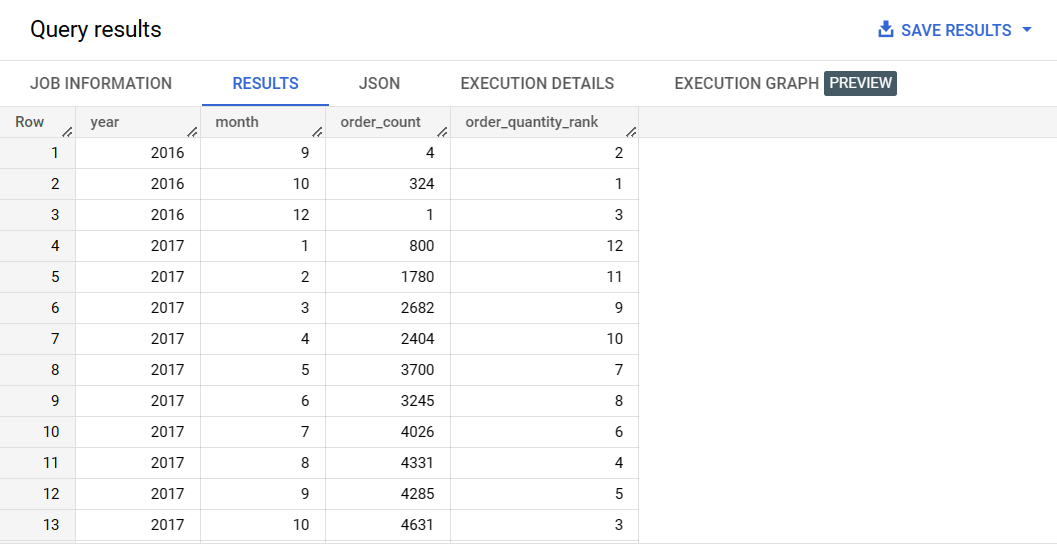
select  temp.year, temp.month,temp.order\_count,rank() over(partition by year order by order\_count desc) as order\_quantity\_rank  from (

select count(order\_id) order\_count , extract(year from order\_purchase\_timestamp) as year,

extract(month from order\_purchase\_timestamp) as month from `sqlbusinesscase-384615.SQLAssignement.orders` o

group by extract(month from order\_purchase\_timestamp),extract(year from order\_purchase\_timestamp)) temp

order by temp.year, temp.month



**Insights:**

We can see the trend of orders/purchase frequency above in both columns ‘order\_count’ and ‘order\_quantity\_rank’. The output is based on every month and every year.

--------------------------------------------------------------------------------------------------------------------------------

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

select max(t1), buyTime from

select customer\_id,temp.buyTime, count(temp.buyTime) over(partition by temp.buyTime) t1 from

(select customer\_id,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'

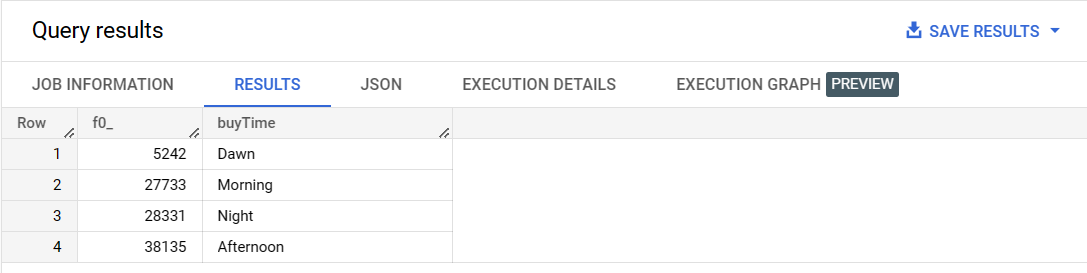
when (extract(hour from order\_purchase\_timestamp)) between 13 and 18 then 'Afternoon'

else 'Night'

end  as buyTime

from `sqlbusinesscase-384615.SQLAssignement.orders`) temp) temp1

group by temp1.buyTime;



**Insights :** Brazilian customers tend to buy more during ‘Afternoon’

--------------------------------------------------------------------------------------------------------------------------------

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

select \* from

(select extract(month from o.order\_purchase\_timestamp) as month, c.customer\_state, count(o.order\_id) order\_count from `sqlbusinesscase-384615.SQLAssignement.orders` o

join

`sqlbusinesscase-384615.SQLAssignement.customers` c

on o.customer\_id = c.customer\_id

group by c.customer\_state,month) temp

order by temp.month, temp.customer\_state



Insights : Month on month orders by states is sorted for month and state in ascending order.

--------------------------------------------------------------------------------------------------------------------------------

1. Distribution of customers across the states in Brazil

select c.customer\_state, count(c.customer\_id) as customer from `sqlbusinesscase-384615.SQLAssignement.customers` c

group by c.customer\_state

order by c.customer\_state



1. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
2. 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 temp1.\*,LAG(order\_sum) over(order by temp1.year,temp1.month)as prev\_order\_sum,

round(((order\_sum-LAG(order\_sum) over(order by temp1.year,temp1.month))/LAG(order\_sum) over(order by temp1.year,temp1.month))\*100,2) as percentage\_increase from

(select temp.month,temp.year,round(sum(temp.payment\_value),0) order\_sum from

(select \*, extract(month from o.order\_purchase\_timestamp) as month, extract(year from o.order\_purchase\_timestamp) as year  from

 `sqlbusinesscase-384615.SQLAssignement.orders` o

join

`sqlbusinesscase-384615.SQLAssignement.payments` p

on p.order\_id = o.order\_id

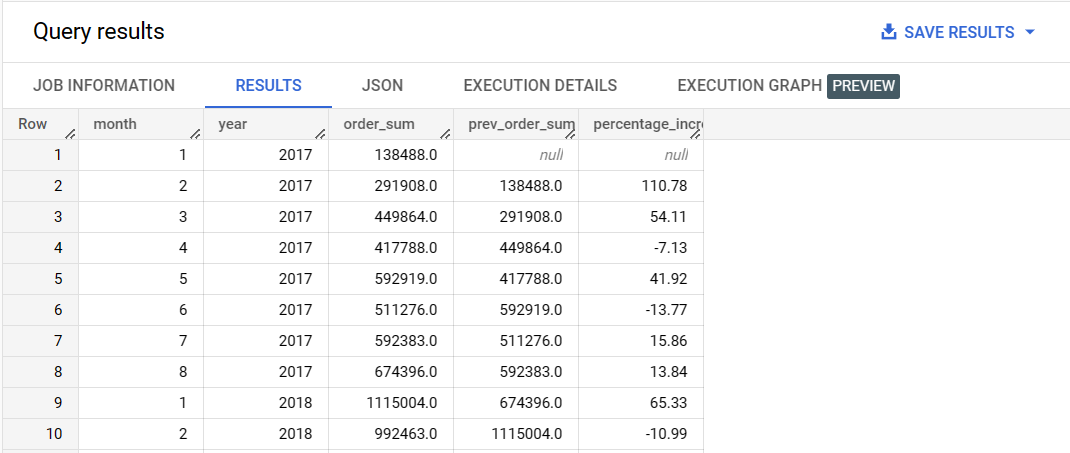
where extract(month from o.order\_purchase\_timestamp) between 1 and 8

order by o.order\_purchase\_timestamp) temp

group by  temp.year,temp.month

) temp1

order by temp1.year, temp1.month



--------------------------------------------------------------------------------------------------------------------------------

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

select c.customer\_state,round(sum(i.price),2) price\_sum, round(sum(i.freight\_value),2) freight\_sum,

round(avg(i.price),2) price\_mean,

round(avg(i.freight\_value),2) freight\_mean,

from `sqlbusinesscase-384615.SQLAssignement.customers` c

join

`sqlbusinesscase-384615.SQLAssignement.orders` o

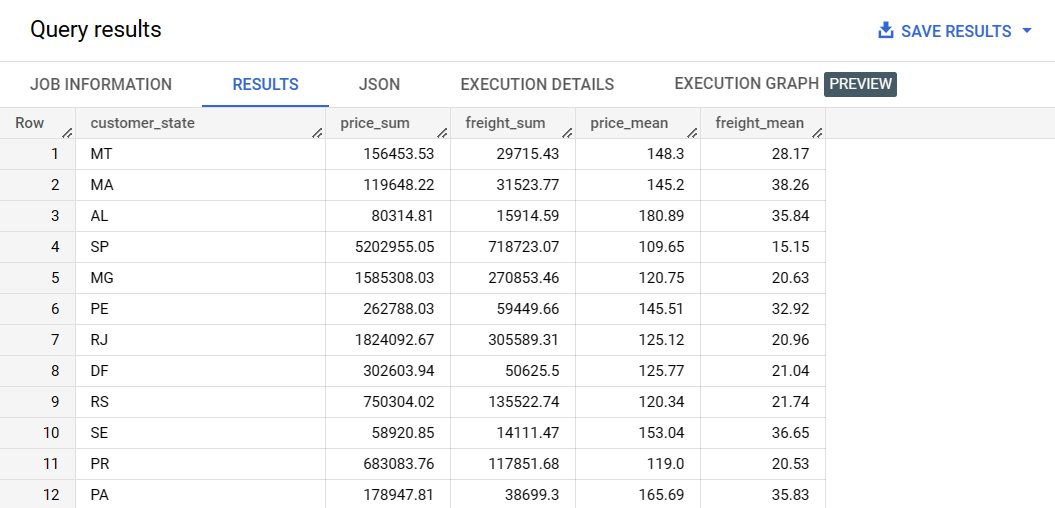
on c.customer\_id = o.customer\_id

join

`sqlbusinesscase-384615.SQLAssignement.order\_items` i

on i.order\_id = o.order\_id

group by c.customer\_state



--------------------------------------------------------------------------------------------------------------------------------

1. Analysis on sales, freight and delivery time
2. Calculate days between purchasing, delivering and estimated delivery

select abs(date\_diff(order\_estimated\_delivery\_date,order\_purchase\_timestamp,day)) as purchase\_estimated\_delivery\_diff,

abs(date\_diff(order\_delivered\_customer\_date,order\_purchase\_timestamp,day)) as purchase\_delivery\_diff,

abs(date\_diff(order\_delivered\_customer\_date,order\_estimated\_delivery\_date,day)) as actual\_estimated\_delivery\_diff,

from `sqlbusinesscase-384615.SQLAssignement.orders`

order by purchase\_estimated\_delivery\_diff desc;



**Insights :** ABS() is used to get the difference(positive) in days. ORDER BY() is included to display values other than NULL as first few rows gave NULL values for ‘purchase\_delivery\_diff’ and ‘actual\_estimated\_delivery\_diff’

--------------------------------------------------------------------------------------------------------------------------------

1. Find time\_to\_delivery & diff\_estimated\_delivery

select abs(date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)) as diff\_estimated\_delivery,from `sqlbusinesscase-384615.SQLAssignement.orders`



--------------------------------------------------------------------------------------------------------------------------------

1. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

(select round(avg(t.freight\_value),2) as freight\_avg,

round(avg(t.time\_to\_delivery),2) as time\_to\_delivery\_avg,

round(avg(t.diff\_estimated\_delivery),2) as diff\_estimated\_delivery\_avg,

t.customer\_state from

(select abs(date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

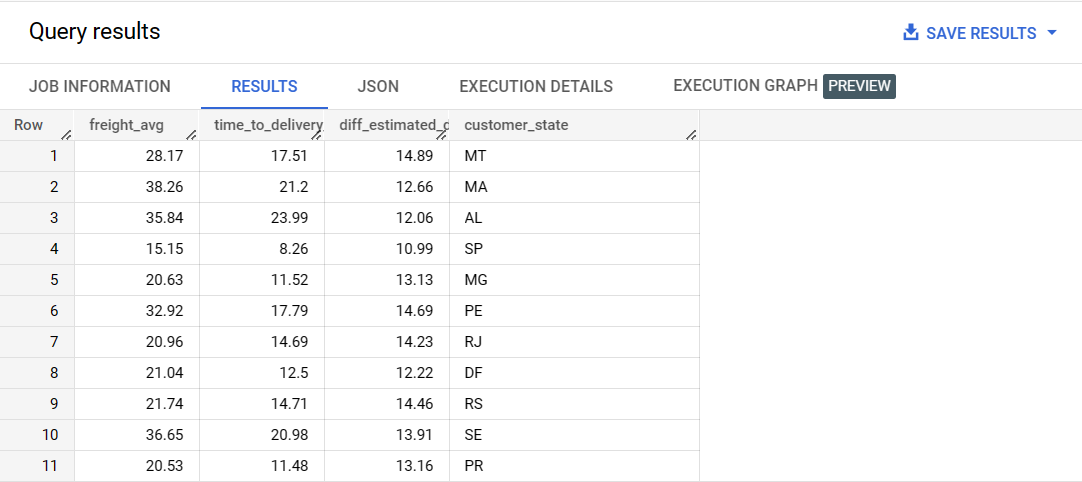
on o1.order\_id = o2.order\_id

join

`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state)



--------------------------------------------------------------------------------------------------------------------------------

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

**==> Highest 5 average freight value**

select round(avg(t.freight\_value),2) as freight\_avg,

t.customer\_state as state from

(select abs(date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

on o1.order\_id = o2.order\_id

join

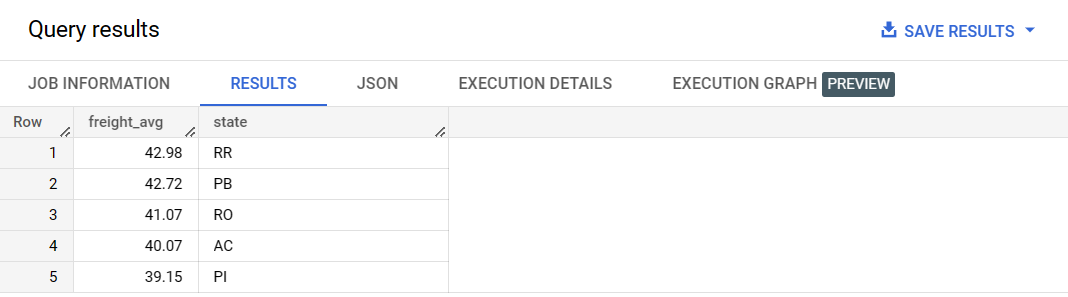
`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state

order by freight\_avg desc

limit 5;



**==> Lowest 5 average freight value**

select round(avg(t.freight\_value),2) as freight\_avg,

t.customer\_state as state from

(select abs(date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

on o1.order\_id = o2.order\_id

join

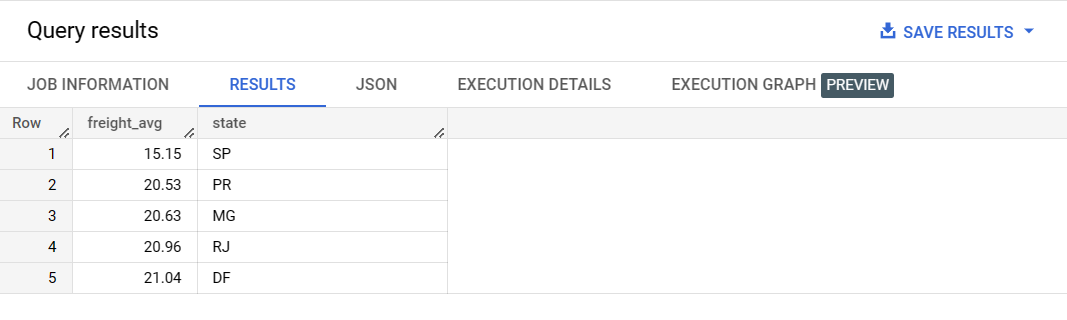
`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state

order by freight\_avg

limit 5;



--------------------------------------------------------------------------------------------------------------------------------

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

**==> Top 5 states with highest average time to delivery**

select

round(avg(t.time\_to\_delivery),2) as time\_to\_delivery\_avg,

t.customer\_state as state from

(select abs(date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date,  day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

on o1.order\_id = o2.order\_id

join

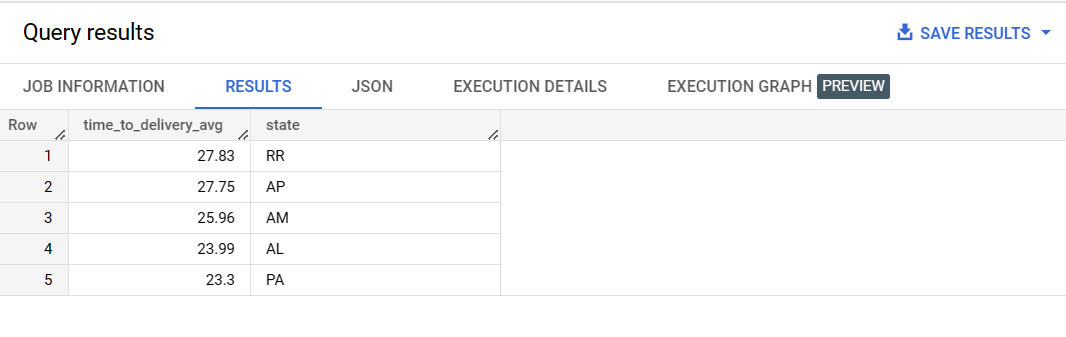
`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state

order by time\_to\_delivery\_avg DESC

limit 5;



**==> Top 5 states with lowest average time to delivery**

select

round(avg(t.time\_to\_delivery),2) as time\_to\_delivery\_avg,

t.customer\_state as state from

(select abs(date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day)) as time\_to\_delivery,

abs(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

on o1.order\_id = o2.order\_id

join

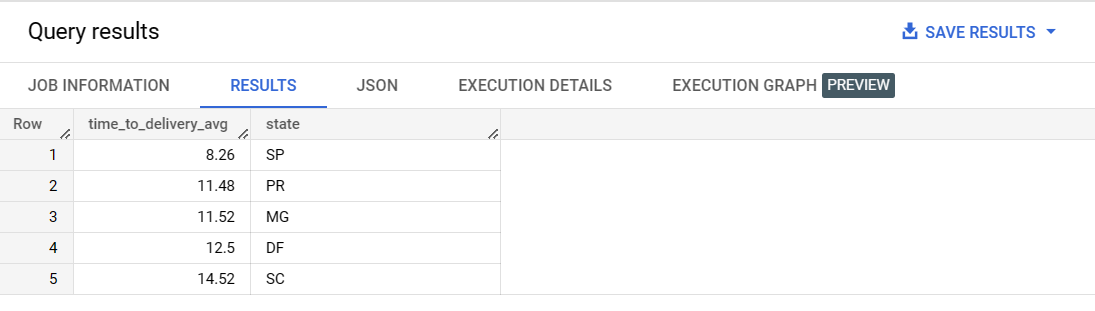
`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state

order by time\_to\_delivery\_avg

limit 5;



--------------------------------------------------------------------------------------------------------------------------------

1. 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**

select

round(avg(t.diff\_estimated\_delivery),2) as diff\_estimated\_delivery\_avg,

t.customer\_state as state from

(select date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day) as time\_to\_delivery,

date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

on o1.order\_id = o2.order\_id

join

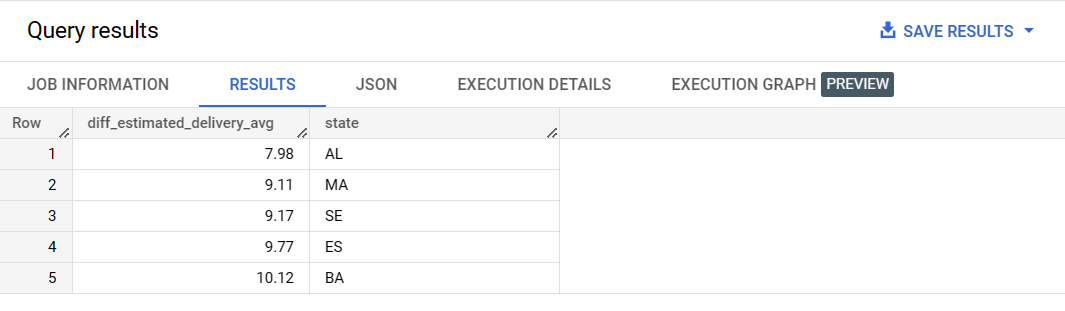
`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state

ORDER BY diff\_estimated\_delivery\_avg

LIMIT 5;



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

select

round(avg(t.diff\_estimated\_delivery),2) as diff\_estimated\_delivery\_avg,

t.customer\_state as state from

(select date\_diff(order\_purchase\_timestamp, order\_delivered\_customer\_date, day) as time\_to\_delivery,

date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day) as diff\_estimated\_delivery,

o2.freight\_value, c1.customer\_state

from `sqlbusinesscase-384615.SQLAssignement.orders` o1 join

`sqlbusinesscase-384615.SQLAssignement.order\_items` o2

on o1.order\_id = o2.order\_id

join

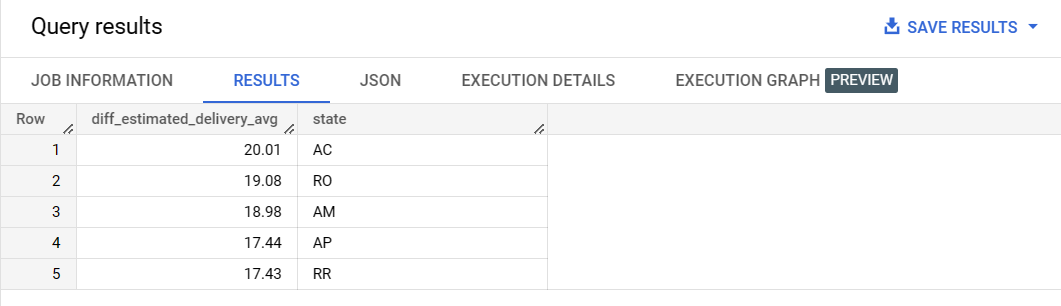
`sqlbusinesscase-384615.SQLAssignement.customers` c1

on o1.customer\_id = c1.customer\_id) t

group by t.customer\_state

ORDER BY diff\_estimated\_delivery\_avg desc

LIMIT 5;



--------------------------------------------------------------------------------------------------------------------------------

1. Payment type analysis:
2. Month over Month count of orders for different payment types

select \* from (

select count(o.order\_id) as order\_count,(extract(month from o.order\_purchase\_timestamp)) as month, p.payment\_type from `sqlbusinesscase-384615.SQLAssignement.orders` o

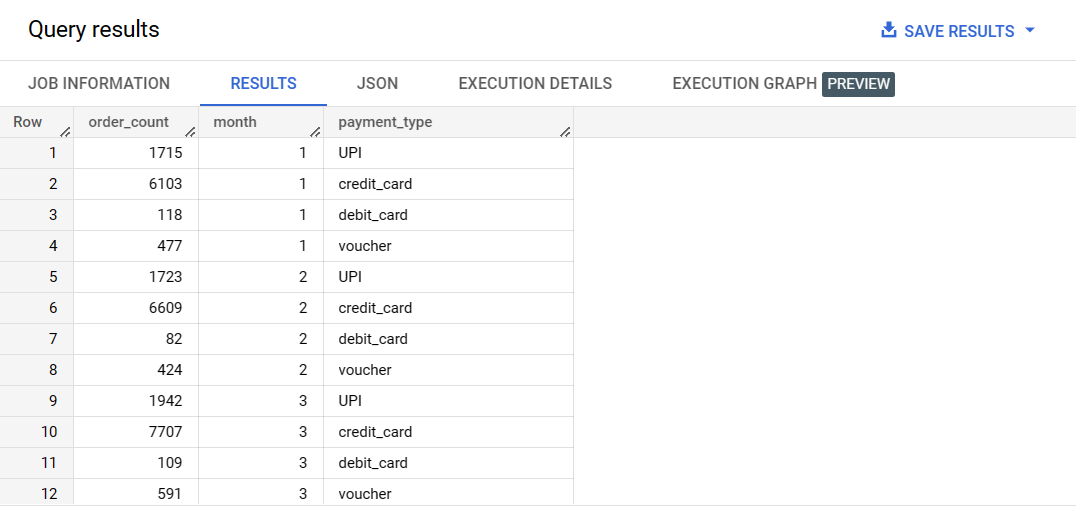
join

`sqlbusinesscase-384615.SQLAssignement.payments` p

on o.order\_id = p.order\_id

group by extract(month from o.order\_purchase\_timestamp),p.payment\_type) as temp

order by temp.month, temp.payment\_type



**Insights :** the result shows the total number of orders for every month for every payment mode. Result is sorted on both month wise and payment\_type wise in ascending order

--------------------------------------------------------------------------------------------------------------------------------

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

select count(o.order\_id) as order\_count, p.payment\_installments from `sqlbusinesscase-384615.SQLAssignement.orders` o

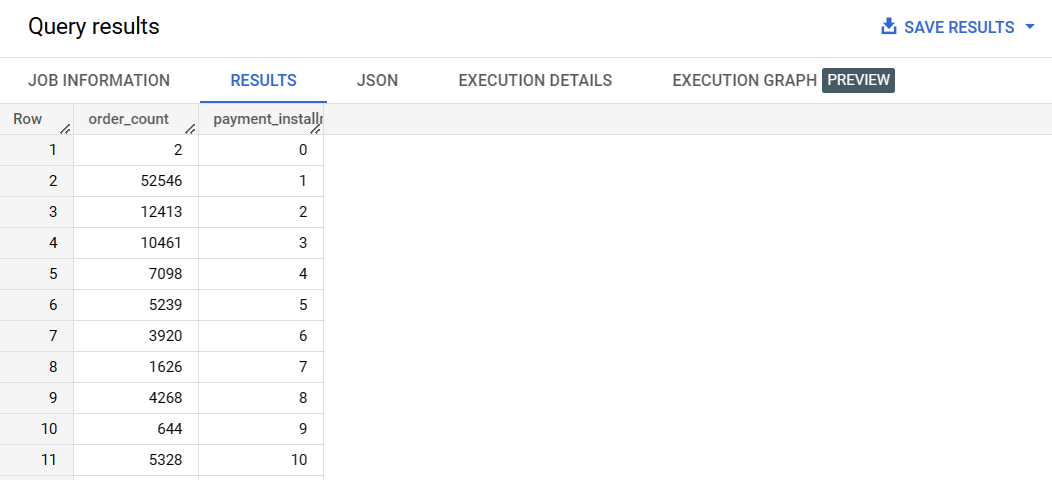
join

`sqlbusinesscase-384615.SQLAssignement.payments` p

on o.order\_id = p.order\_id

group by p.payment\_installments

order by p.payment\_installments



**Insights :** The query result gives the order count based on no of payment installments in ascending order