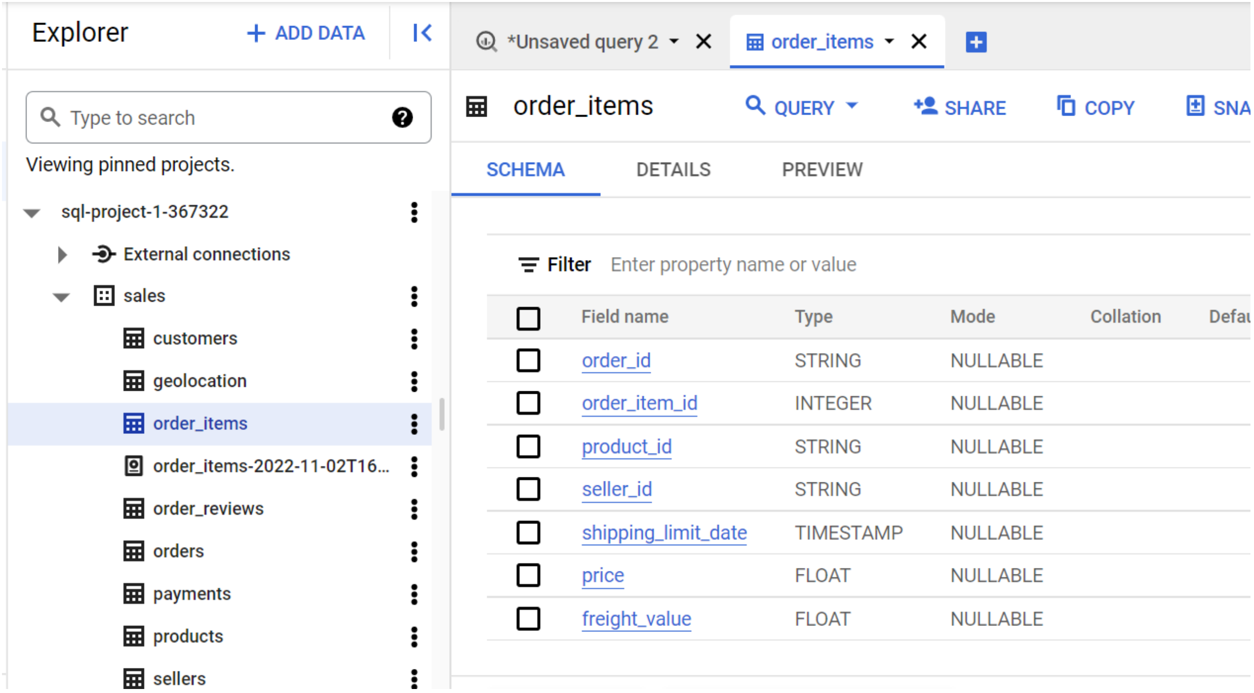
- **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset**

1. Data type of columns in a table

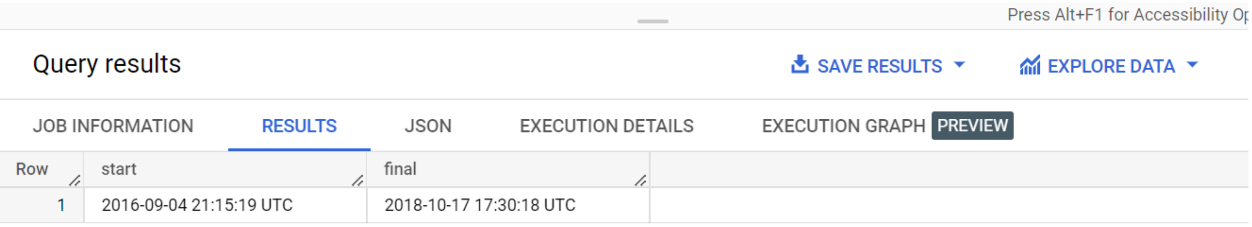
   Ans)_After importing the table , just click on the table name and we can see the column name , data type of column , Nullable means column allows null values.



□	Field name	Type	Mode	Collation	Defau
□	order_id	STRING	NULLABLE		
□	order_item_id	INTEGER	NULLABLE		
□	product_id	STRING	NULLABLE		
□	seller_id	STRING	NULLABLE		
□	shipping_limit_date	TIMESTAMP	NULLABLE		
□	price	FLOAT	NULLABLE		
□	freight_value	FLOAT	NULLABLE		

2. Time period for which the data is given

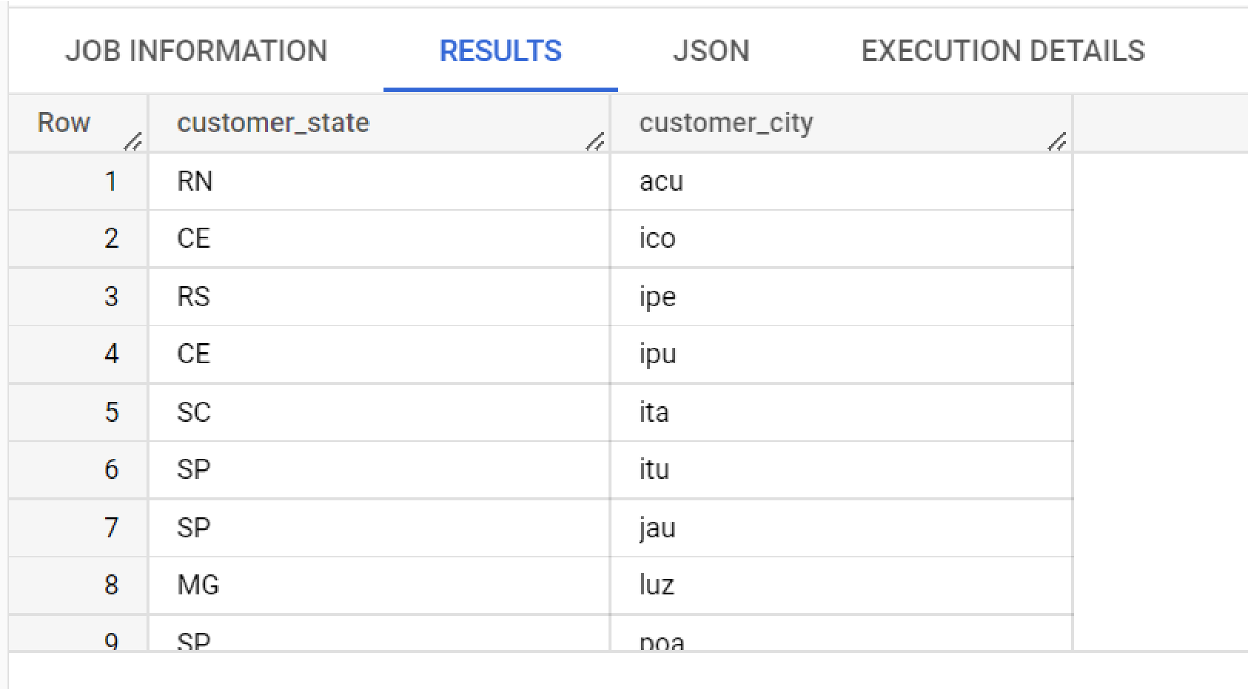
   Ans)
```
with start_date as (select order_purchase_timestamp as start from
`sales.orders`

order by order_purchase_timestamp limit 1 ),



end_date as ( select order_purchase_timestamp as final from `sales.orders`

order by order_purchase_timestamp desc limit 1)



select start,end_date.final

from start_date,end_date;
```



Row	start	final
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

3. Cities and States covered in the dataset

Ans)
```
select customer_state , customer_city
from `sales.customers`
group by customer_state, customer_city;
```

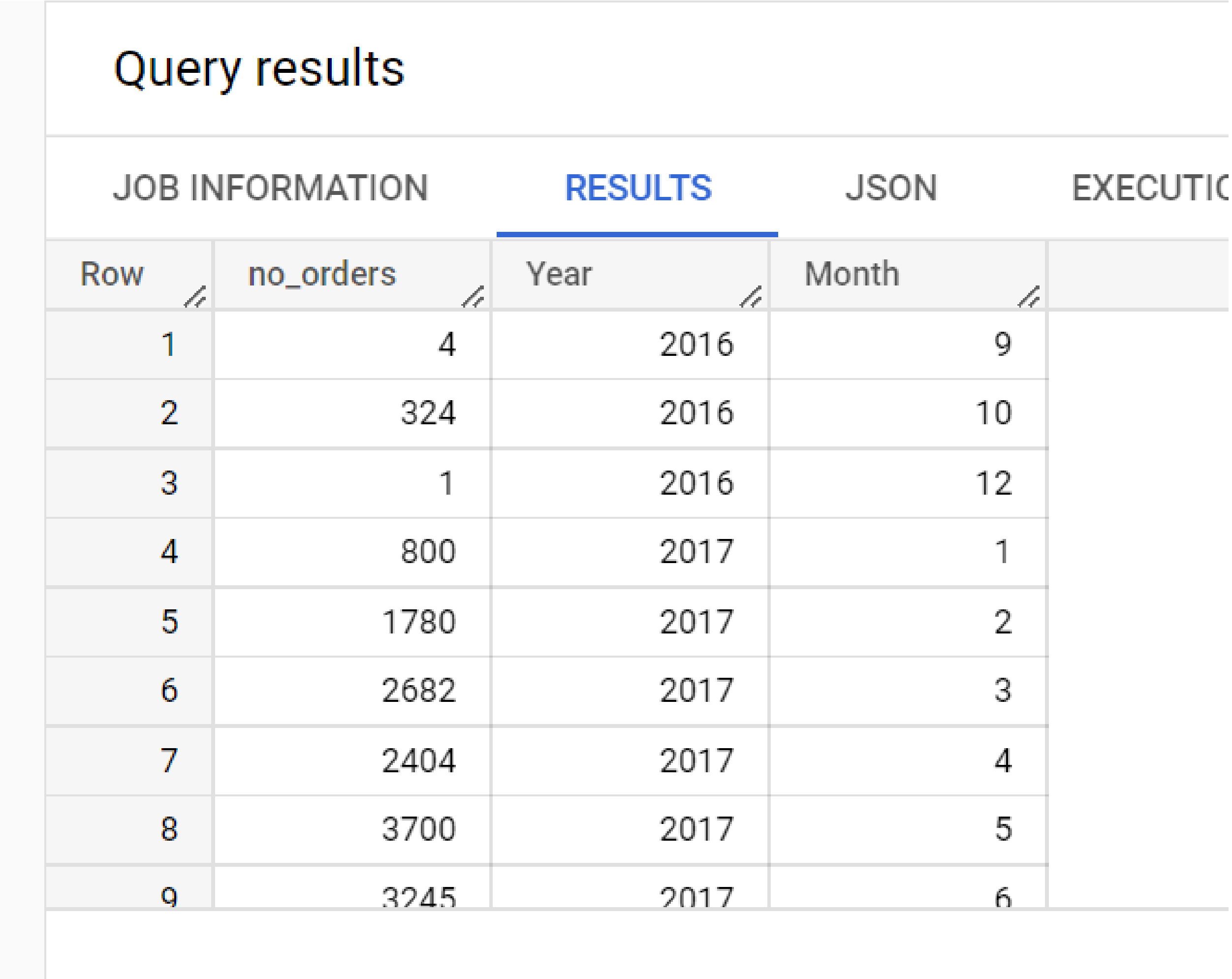


Row	customer_state	customer_city
1	RN	acu
2	CE	ico
3	RS	ipe
4	CE	ipu
5	SC	ita
6	SP	itu
7	SP	jau
8	MG	luz
9	SP	poa

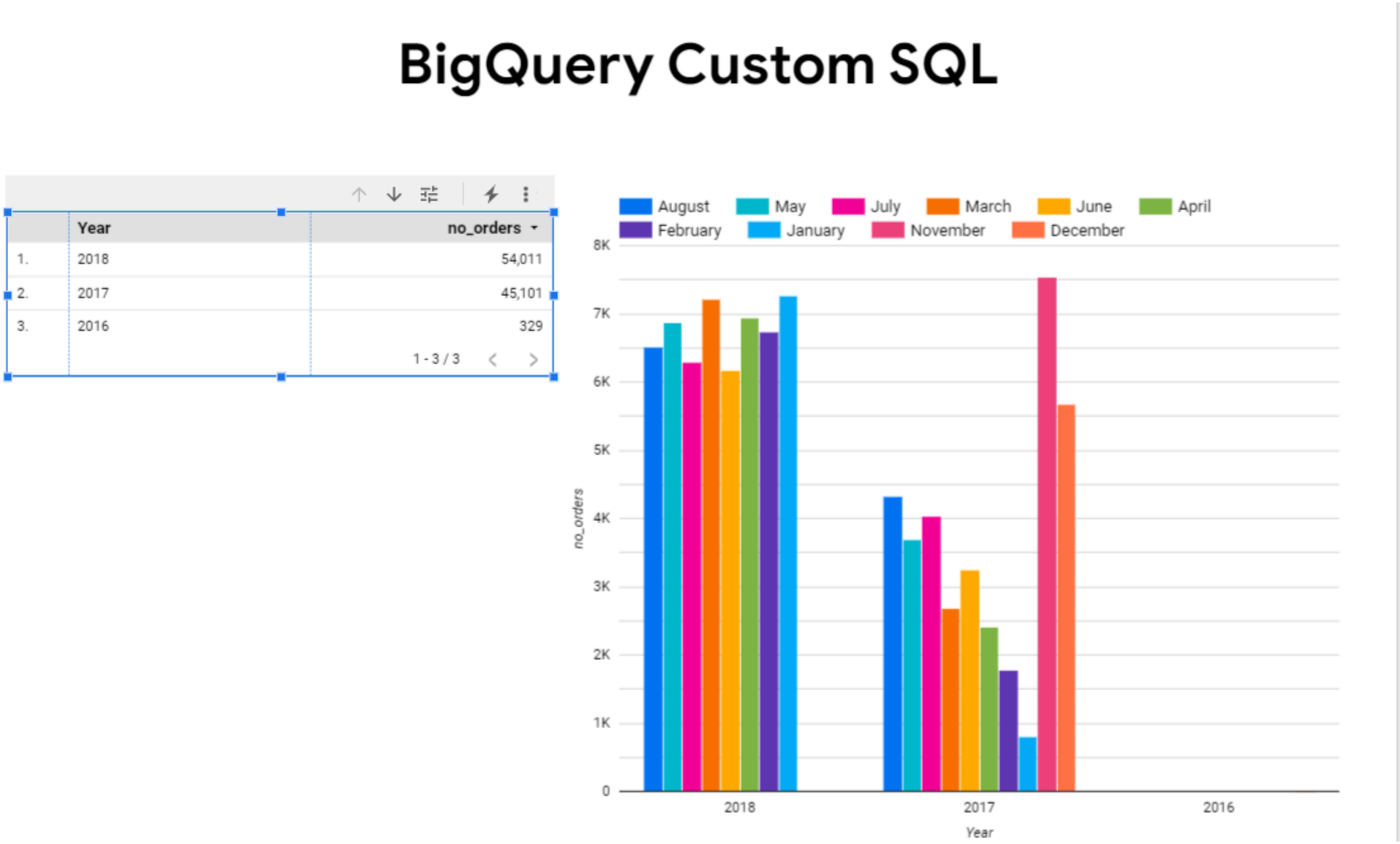
❖ **In-depth Exploration:**

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?

Ans)
```
select count(*) as no_orders,
extract(year from order_purchase_timestamp) as Year,

extract(month from order_purchase_timestamp) as Month

from `sales.orders`

group by Year,Month

order by Year,Month;
```



Row	no_orders	Year	Month
1	4	2016	9
2	324	2016	10
3	1	2016	12
4	800	2017	1
5	1780	2017	2
6	2682	2017	3
7	2404	2017	4
8	3700	2017	5
9	3245	2017	6



BigQuery Custom SQL

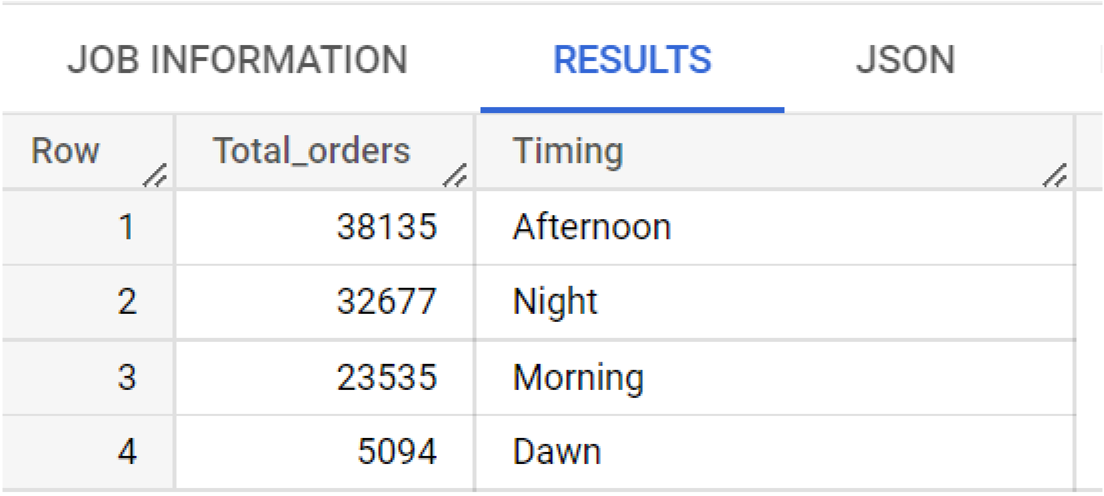
	Year	no_orders ▾
1.	2018	54,011
2.	2017	45,101
3.	2016	329

1 - 3 / 3

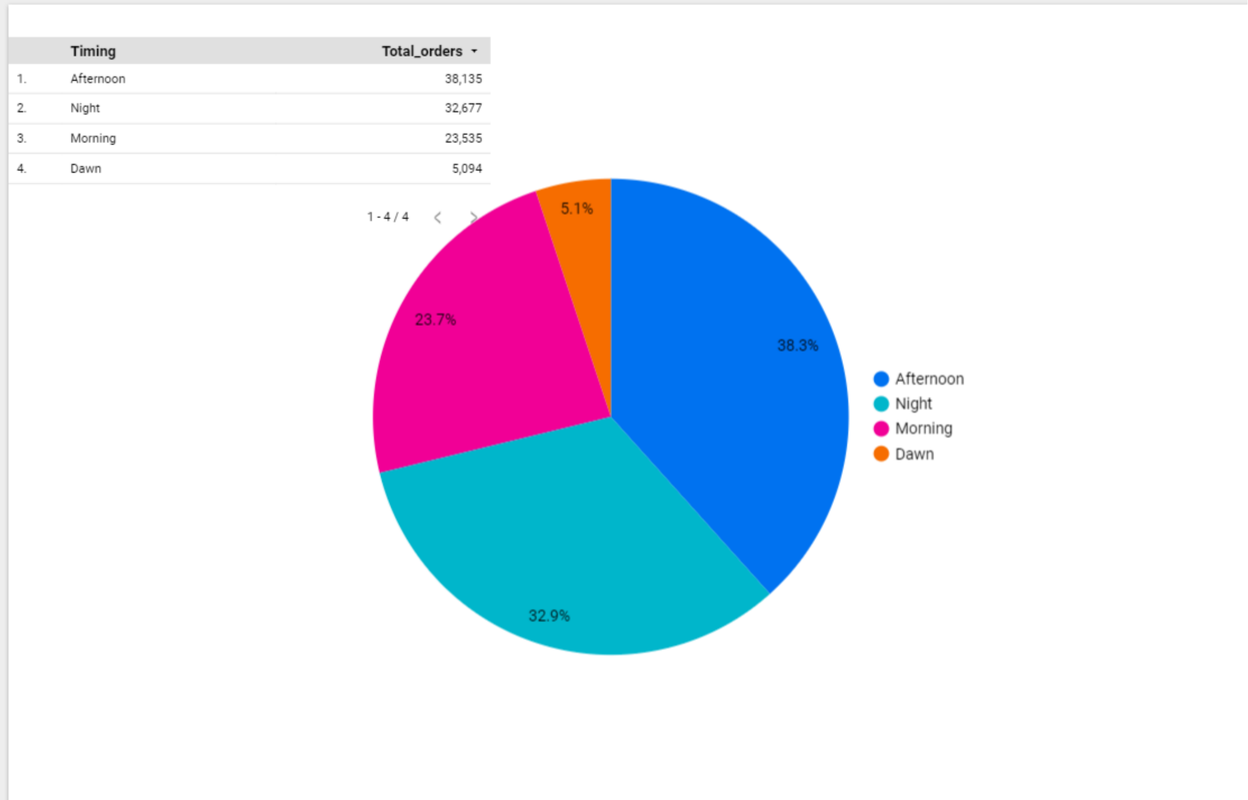
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Ans)

```
with hr as

(select count(*) as No_of_Orders,

extract(hour from order_purchase_timestamp) as Hour

from `sales.orders`

group by Hour)


select sum(No_of_Orders) as Total_orders,

case

when Hour between 4 and 8 then "Dawn"
when Hour between 9 and 12 then "Morning"

when Hour between 13 and 18 then "Afternoon"

else "Night"

end as Timing

from hr

group by Timing

order by Total_orders desc;
```



Row	Total_orders	Timing
1	38135	Afternoon
2	32677	Night
3	23535	Morning
4	5094	Dawn

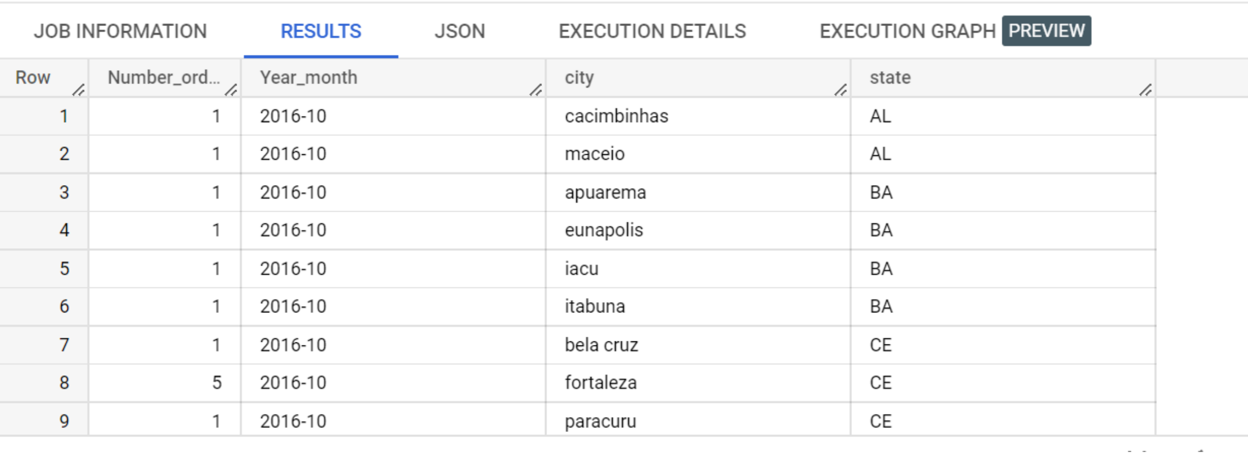


	Timing	Total_orders ▾
1.	Afternoon	38,135
2.	Night	32,677
3.	Morning	23,535
4.	Dawn	5,094

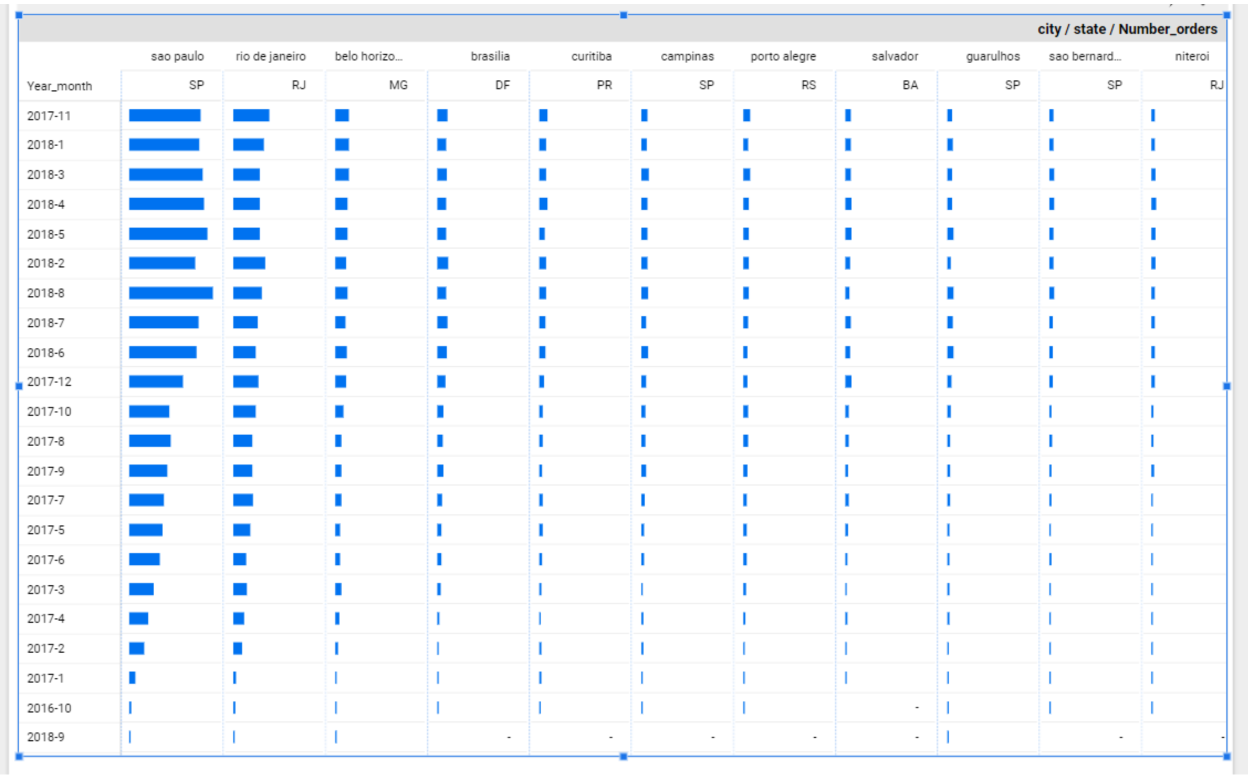
❖ <u>Evolution of E-commerce orders in the Brazil region:</u>

1. Get month on month orders by region, states

Ans)
```
select count(order_id) as Number_orders,

concat(extract(year from order_purchase_timestamp) ,"-",

extract(month from order_purchase_timestamp) ) as Year_month,

customer_city as city,customer_state as state

from `sales.orders` as o

left join sales.customers as c on o.customer_id = c.customer_id

group by Year_month,state,city

order by Year_month ,state,city;
```



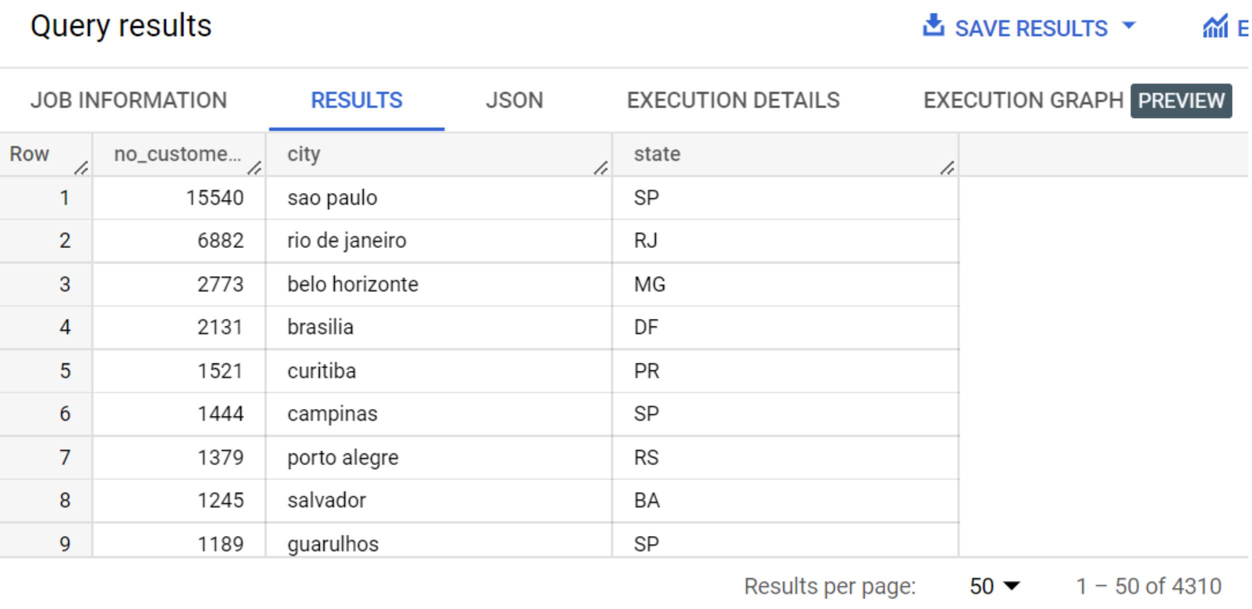
Row	Number_ord...	Year_month	city	state
1	1	2016-10	cacimbinhas	AL
2	1	2016-10	maceio	AL
3	1	2016-10	apuarema	BA
4	1	2016-10	eunapolis	BA
5	1	2016-10	iacu	BA
6	1	2016-10	itabuna	BA
7	1	2016-10	bela cruz	CE
8	5	2016-10	fortaleza	CE
9	1	2016-10	paracuru	CE



2. How are customers distributed in Brazil

Ans)
```
select count(customer_id) as no_customers,

customer_city as city,customer_state as state

from `sales.customers`
```

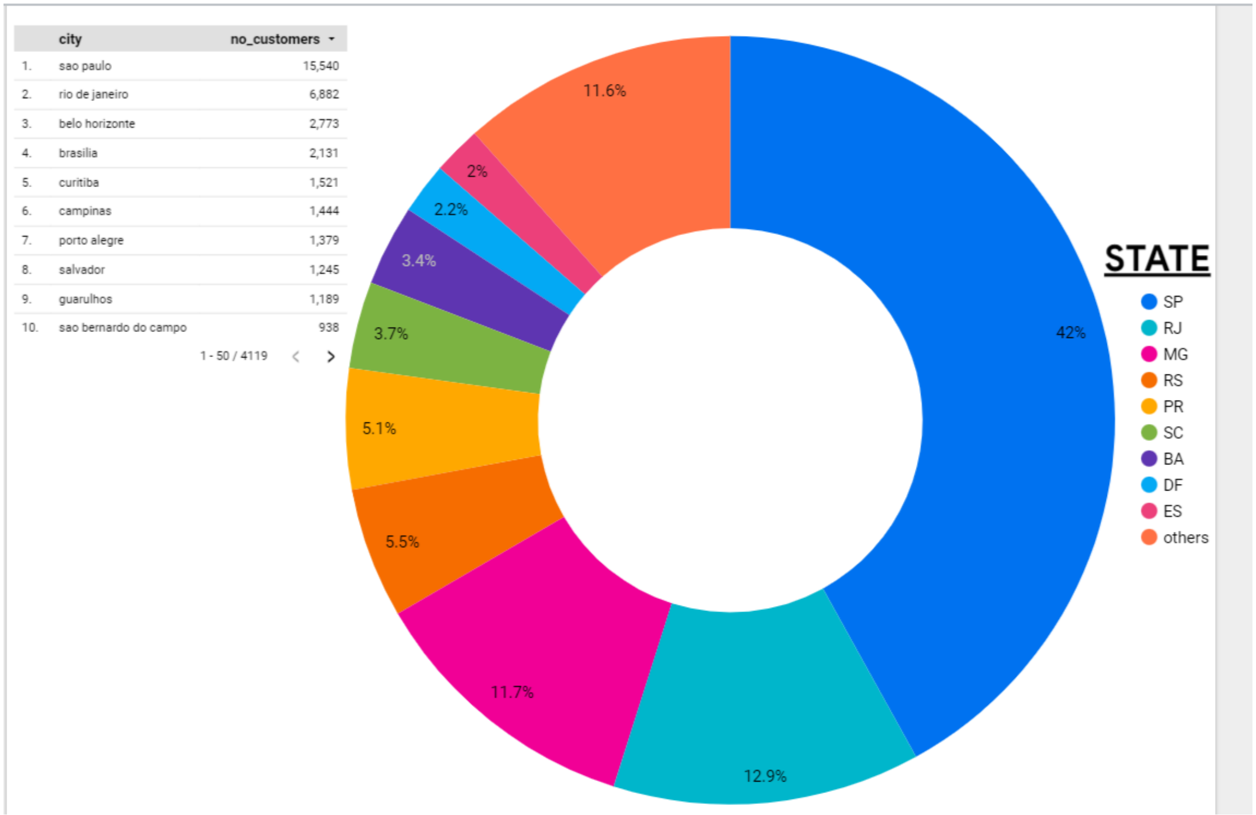
```
group by city,state

order by no_customers desc;
```



Query results

Row	no_custome...	city	state
1	15540	sao paulo	SP
2	6882	rio de janeiro	RJ
3	2773	belo horizonte	MG
4	2131	brasilia	DF
5	1521	curitiba	PR
6	1444	campinas	SP
7	1379	porto alegre	RS
8	1245	salvador	BA
9	1189	guarulhos	SP

Results per page: 50 1 – 50 of 4310



	city	no_customers
1.	sao paulo	15,540
2.	rio de janeiro	6,882
3.	belo horizonte	2,773
4.	brasilia	2,131
5.	curitiba	1,521
6.	campinas	1,444
7.	porto alegre	1,379
8.	salvador	1,245
9.	guarulhos	1,189
10.	sao bernardo do campo	938

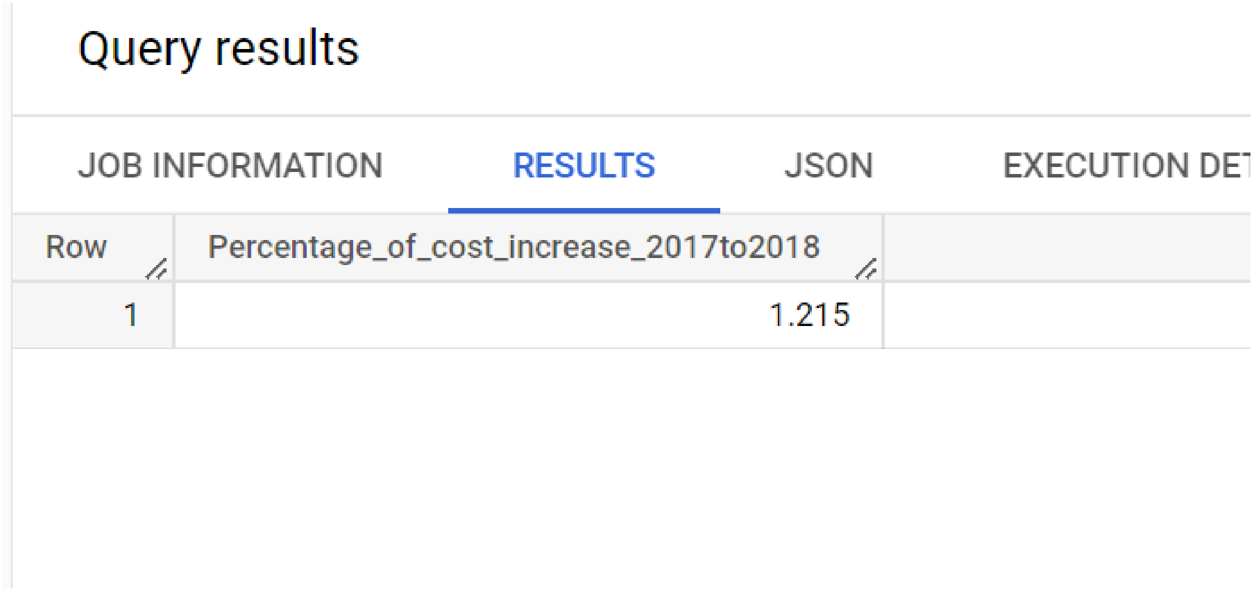
1 - 50 / 4119

- **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)

   Ans)

```
with p1 as ( select price,Month from (select (price+freight_value) as price,

extract(year from order_purchase_timestamp) as Year,

extract(month from order_purchase_timestamp) as Month,



from `sales.order_items`as o1

join `sales.orders`as o2 on o1.order_id=o2.order_id) as x

where Year = 2017 and Month between 1 and 8),



p2 as ( select price,Month from (select (price+freight_value) as price,

extract(year from order_purchase_timestamp) as Year,

extract(month from order_purchase_timestamp) as Month

from `sales.order_items`as o1

join `sales.orders`as o2 on o1.order_id=o2.order_id) as x

where Year = 2018 and Month between 1 and 8)



select round(((sum(p2.price)-sum(p1.price))/ sum(p1.price)) *100,3) as Percentage_of_cost_increase_2017to2018

from  p1 join p2 on p1.Month = p2.Month;
```

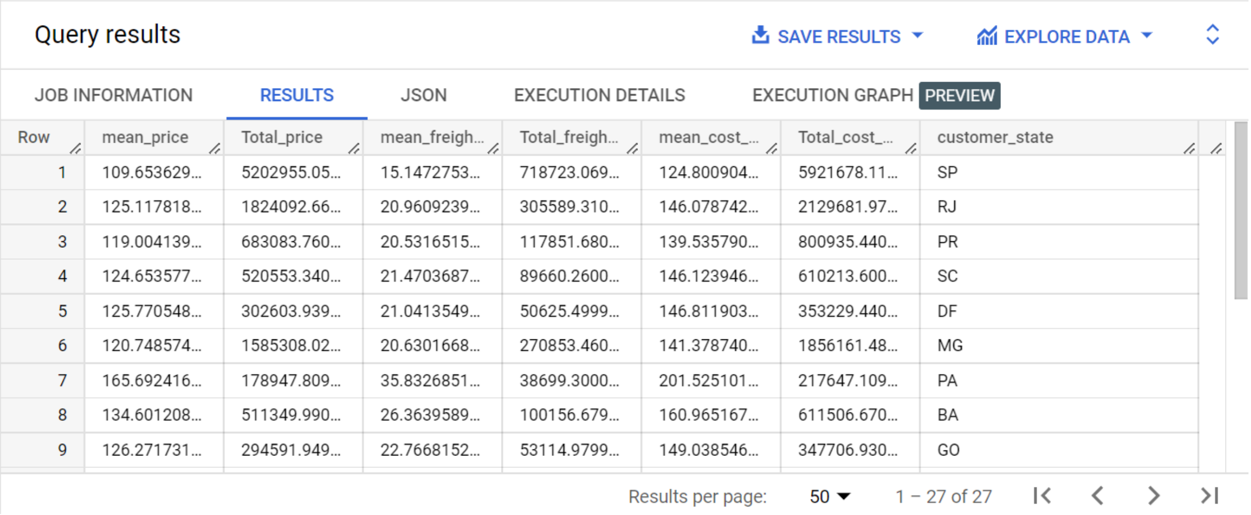


Query results

Row	Percentage_of_cost_increase_2017to2018
1	1.215

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

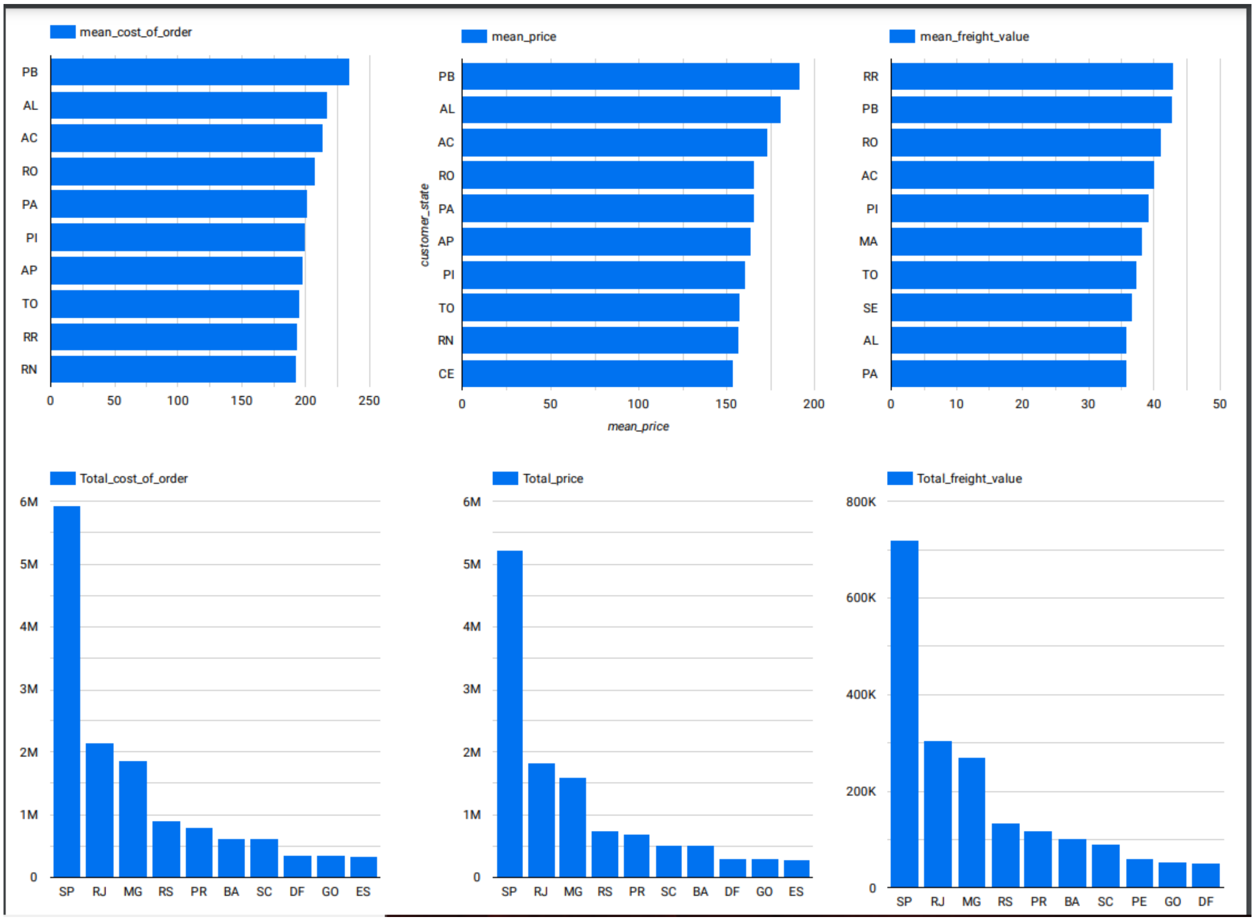
```
Ans)select avg(price) as mean_price ,sum(price) as Total_price,

avg(freight_value) as mean_freight_value , sum(freight_value) as
Total_freight_value,

avg(price+freight_value) as mean_cost_of_order , sum(price+freight_value) as
Total_cost_of_order,

customer_state

from `sales.order_items` as o1

join `sales.orders` as o2 on o1.order_id=o2.order_id

join `sales.customers` as c on o2.customer_id=c.customer_id

group by customer_state;
```



Query results

Row	mean_price	Total_price	mean_freigh...	Total_freigh...	mean_cost_...	Total_cost_...	customer_state
1	109.653629...	5202955.05...	15.1472753...	718723.069...	124.800904...	5921678.11...	SP
2	125.117818...	1824092.66...	20.9609239...	305589.310...	146.078742...	2129681.97...	RJ
3	119.004139...	683083.760...	20.5316515...	117851.680...	139.535790...	800935.440...	PR
4	124.653577...	520553.340...	21.4703687...	89660.2600...	146.123946...	610213.600...	SC
5	125.770548...	302603.939...	21.0413549...	50625.4999...	146.811903...	353229.440...	DF
6	120.748574...	1585308.02...	20.6301668...	270853.460...	141.378740...	1856161.48...	MG
7	165.692416...	178947.809...	35.8326851...	38699.3000...	201.525101...	217647.109...	PA
8	134.601208...	511349.990...	26.3639589...	100156.679...	160.965167...	611506.670...	BA
9	126.271731...	294591.949...	22.7668152...	53114.9799...	149.038546...	347706.930...	GO

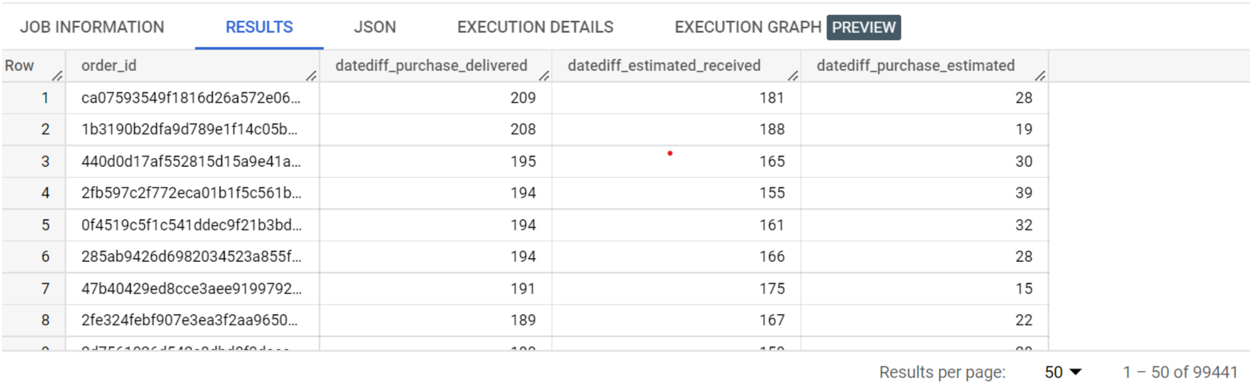
Results per page: 50 1 – 27 of 27



❖ Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

Ans)
```
select
order_id,date_diff(order_delivered_customer_date,order_purchase_timestamp ,
day) as datediff_purchase_delivered,

date_diff(order_delivered_customer_date,order_estimated_delivery_date, day) as
datediff_estimated_received,

date_diff(order_estimated_delivery_date,order_purchase_timestamp, day) as
datediff_purchase_estimated

from `sales.orders`

order by datediff_purchase_delivered
desc,datediff_estimated_received,datediff_purchase_estimated ;
```

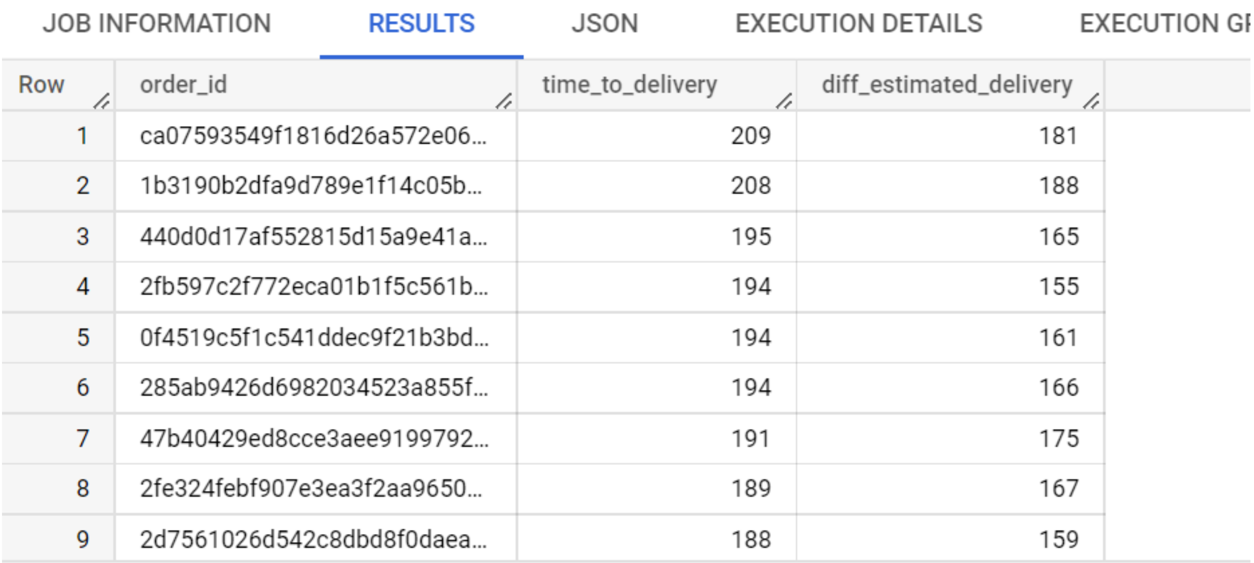


Row	order_id	datediff_purchase_delivered	datediff_estimated_received	datediff_purchase_estimated
1	ca07593549f1816d26a572e06...	209	181	28
2	1b3190b2dfa9d789e1f14c05b...	208	188	19
3	440d0d17af552815d15a9e41a...	195	165	30
4	2fb597c2f772eca01b1f5c561b...	194	155	39
5	0f4519c5f1c541ddec9f21b3bd...	194	161	32
6	285ab9426d6982034523a855f...	194	166	28
7	47b40429ed8cce3aee9199792...	191	175	15
8	2fe324febf907e3ea3f2aa9650...	189	167	22

NOTE - If the difference is in positive integer then it shows the no.of days order delayed, If the difference is in Negative integer then it shows the no.of days order received early.

2. Create columns:
    a. time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
    b. diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Ans)
```
select
order_id,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 `sales.orders`

order by time_to_delivery desc,diff_estimated_delivery;
```



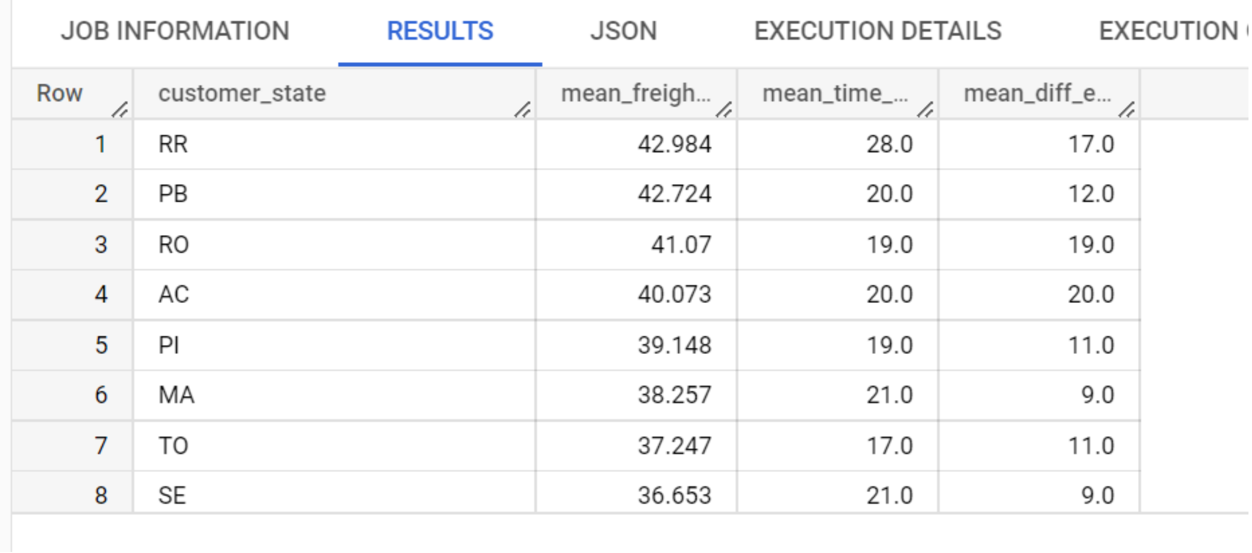
Row	order_id	time_to_delivery	diff_estimated_delivery
1	ca07593549f1816d26a572e06...	209	181
2	1b3190b2dfa9d789e1f14c05b...	208	188
3	440d0d17af552815d15a9e41a...	195	165
4	2fb597c2f772eca01b1f5c561b...	194	155
5	0f4519c5f1c541ddec9f21b3bd...	194	161
6	285ab9426d6982034523a855f...	194	166
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324febf907e3ea3f2aa9650...	189	167
9	2d7561026d542c8dbd8f0daea...	188	159

NOTE - If the difference is in positive integer then it shows the no.of days order delayed, If the difference is in Negative integer then it shows the no.of days order received early

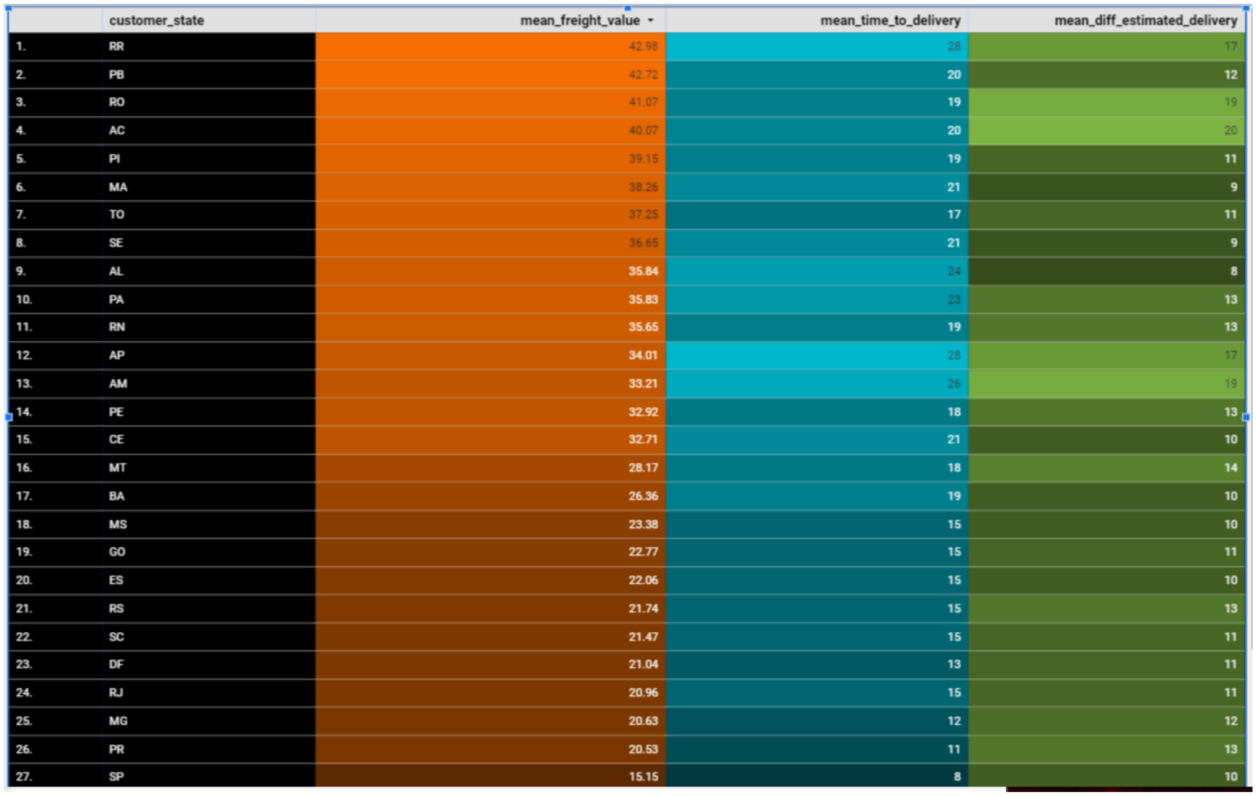
3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Ans)
```
select customer_state,round(avg(freight_value),3) as mean_freight_value,

round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp ,
day))) as mean_time_to_delivery,

round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date
, day))) as mean_diff_estimated_delivery

from `sales.order_items` as o1

join `sales.orders` as o2 on o1.order_id=o2.order_id

join `sales.customers` as c on o2.customer_id=c.customer_id
```

```
group by customer_state

order by mean_freight_value desc;
```



Row	customer_state	mean_freigh...	mean_time_...	mean_diff_e...
1	RR	42.984	28.0	17.0
2	PB	42.724	20.0	12.0
3	RO	41.07	19.0	19.0
4	AC	40.073	20.0	20.0
5	PI	39.148	19.0	11.0
6	MA	38.257	21.0	9.0
7	TO	37.247	17.0	11.0
8	SE	36.653	21.0	9.0



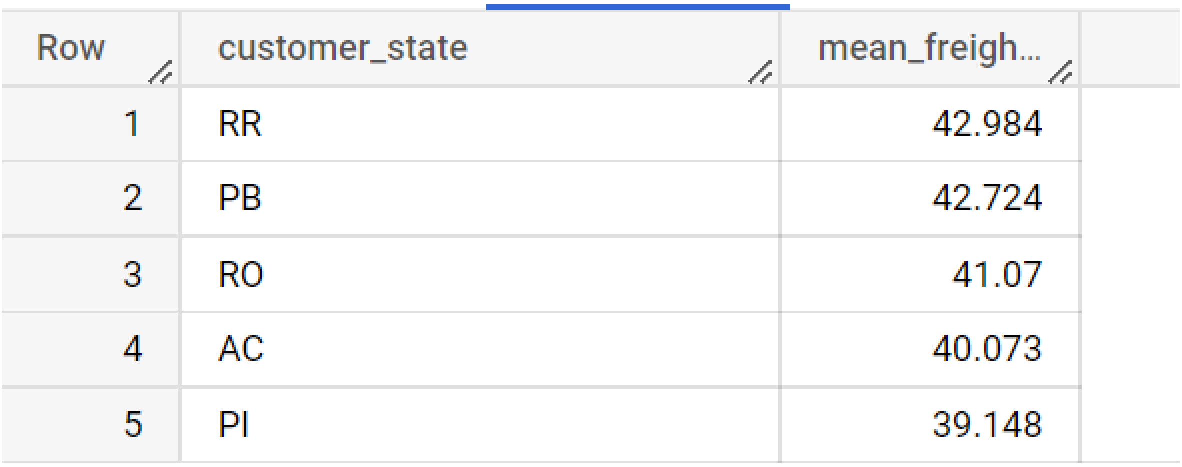
	customer_state	mean_freight_value ▾	mean_time_to_delivery	mean_diff_estimated_delivery
1.	RR	42.98	28	17
2.	PB	42.72	20	12
3.	RO	41.07	19	19
4.	AC	40.07	20	20
5.	PI	39.15	19	11
6.	MA	38.26	21	9
7.	TO	37.25	17	11
8.	SE	36.65	21	9
9.	AL	35.84	24	8
10.	PA	35.83	23	13
11.	RN	35.65	19	13
12.	AP	34.01	28	17
13.	AM	33.21	26	19
14.	PE	32.92	18	13
15.	CE	32.71	21	10
16.	MT	28.17	18	14
17.	BA	26.36	19	10
18.	MS	23.38	15	10
19.	GO	22.77	15	11
20.	ES	22.06	15	10
21.	RS	21.74	15	13
22.	SC	21.47	15	11
23.	DF	21.04	13	11
24.	RJ	20.96	15	11
25.	MG	20.63	12	12
26.	PR	20.53	11	13
27.	SP	15.15	8	10

4. Sort the data to get the following:

   Ans)

a) Top 5 states with highest average freight value - sort in desc/asc limit 5

```
select customer_state,round(avg(freight_value),3) as mean_freight_value
from `sales.order_items` as o1
join `sales.orders` as o2 on o1.order_id=o2.order_id
join `sales.customers` as c on o2.customer_id=c.customer_id
group by customer_state
order by mean_freight_value desc
limit 5;
```

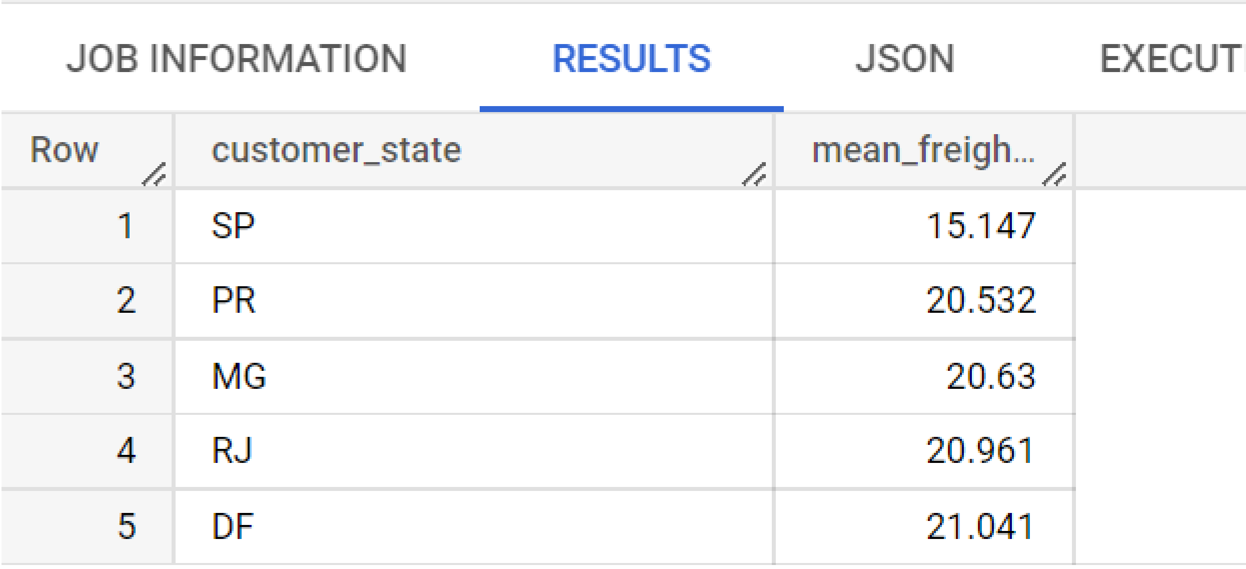


Row	customer_state	mean_freigh...
1	RR	42.984
2	PB	42.724
3	RO	41.07
4	AC	40.073
5	PI	39.148

b) Top 5 states with lowest average freight value - sort in desc/asc limit 5

```
select customer_state,round(avg(freight_value),3) as mean_freight_value
from `sales.order_items` as o1
join `sales.orders` as o2 on o1.order_id=o2.order_id
join `sales.customers` as c on o2.customer_id=c.customer_id
group by customer_state
order by mean_freight_value
```

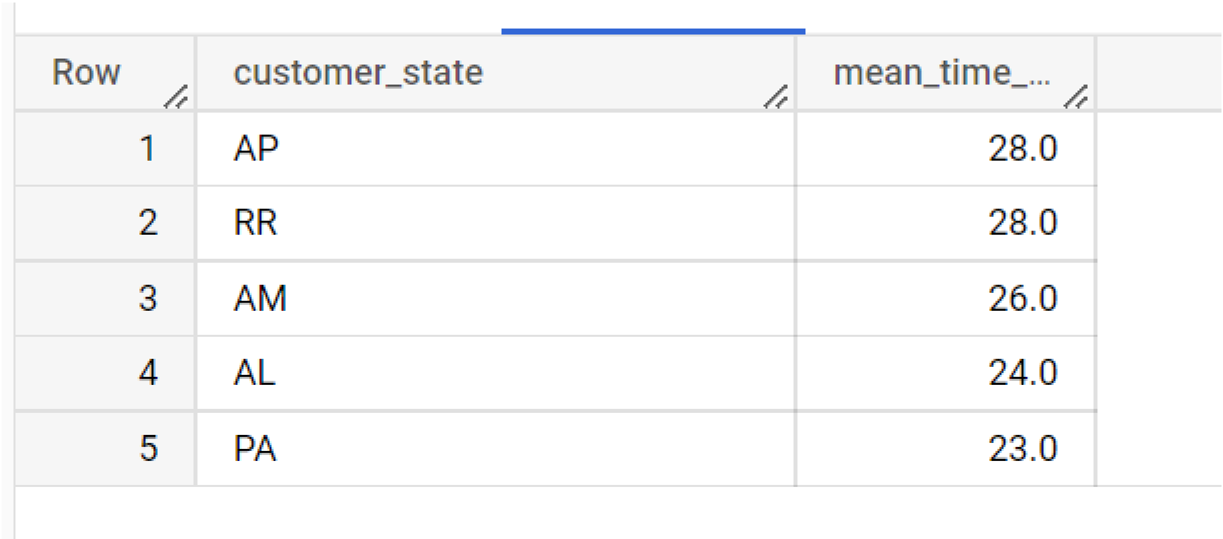
```
limit 5;
```



Row	customer_state	mean_freigh...
1	SP	15.147
2	PR	20.532
3	MG	20.63
4	RJ	20.961
5	DF	21.041

C) Top 5 states with highest average time to delivery

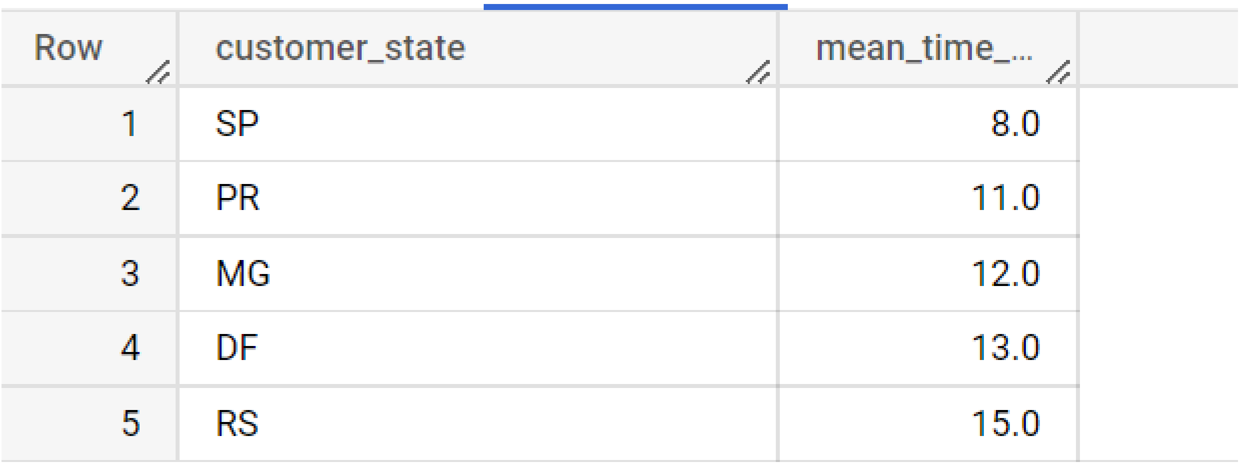
```
select customer_state,

round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp ,
day))) as mean_time_to_delivery,

from `sales.order_items` as o1

join `sales.orders` as o2 on o1.order_id=o2.order_id

join `sales.customers` as c on o2.customer_id=c.customer_id

group by customer_state

order by mean_time_to_delivery desc

limit 5;
```



Row	customer_state	mean_time_...
1	AP	28.0
2	RR	28.0
3	AM	26.0
4	AL	24.0
5	PA	23.0

d)Top 5 states with lowest average time to delivery

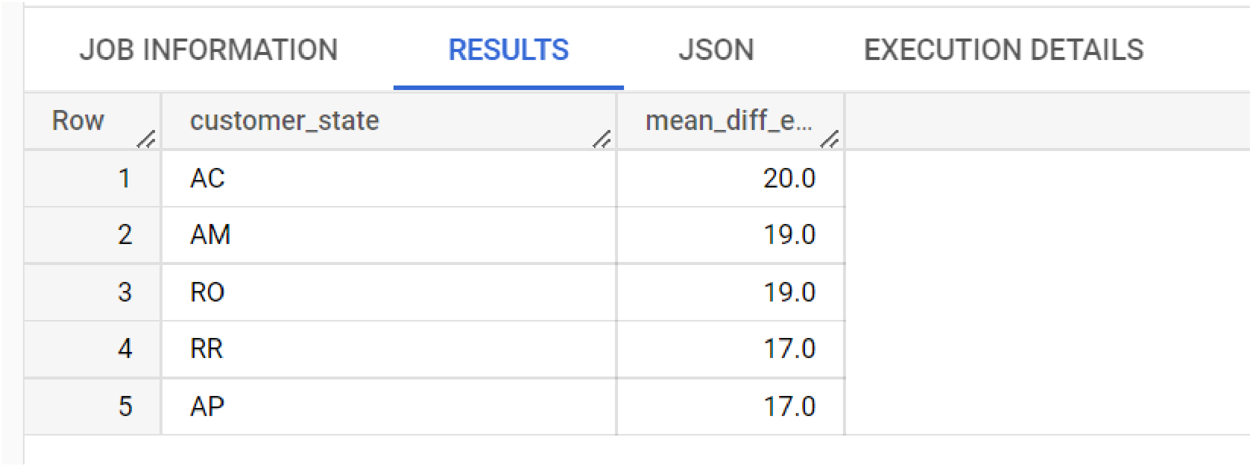
```
select customer_state,

round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp , day))) as mean_time_to_delivery,

from `sales.order_items` as o1

join `sales.orders` as o2 on o1.order_id=o2.order_id

join `sales.customers` as c on o2.customer_id=c.customer_id

group by customer_state

order by mean_time_to_delivery

limit 5;
```



Row	customer_state	mean_time_...
1	SP	8.0
2	PR	11.0
3	MG	12.0
4	DF	13.0
5	RS	15.0

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

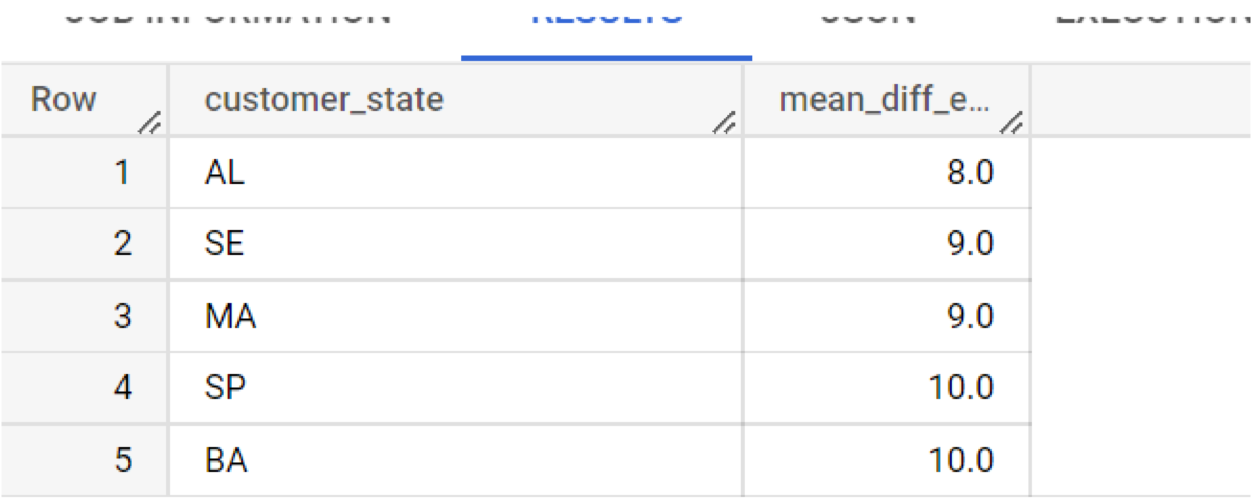
```
select customer_state,

round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day))) as mean_diff_estimated_delivery

from `sales.order_items` as o1

join `sales.orders` as o2 on o1.order_id=o2.order_id

join `sales.customers` as c on o2.customer_id=c.customer_id

group by customer_state

order by mean_diff_estimated_delivery desc

limit 5;
```



Row	customer_state	mean_diff_e...
1	AC	20.0
2	AM	19.0
3	RO	19.0
4	RR	17.0
5	AP	17.0

f) Top 5 states where delivery is not so fast compared to estimated date

```
select customer_state,

round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_d
ate, day))) as mean_diff_estimated_delivery

from `sales.order_items` as o1

join `sales.orders` as o2 on o1.order_id=o2.order_id

join `sales.customers` as c on o2.customer_id=c.customer_id

group by customer_state

order by mean_diff_estimated_delivery

limit 5;
```



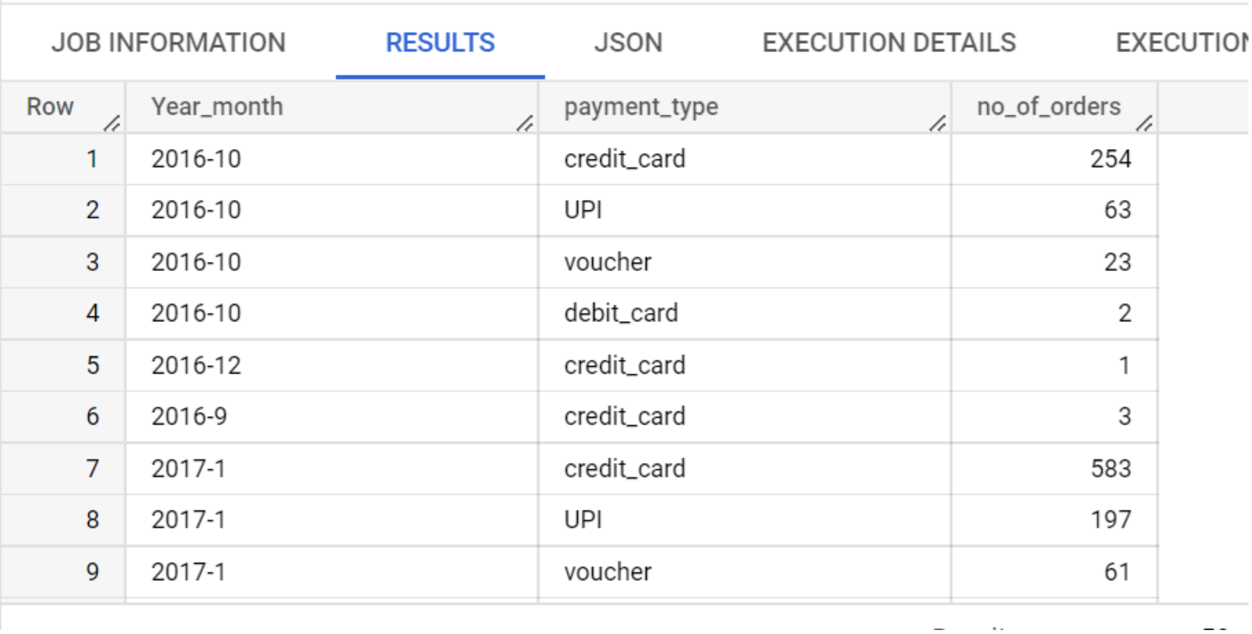
Row	customer_state	mean_diff_e...
1	AL	8.0
2	SE	9.0
3	MA	9.0
4	SP	10.0
5	BA	10.0

- **Payment type analysis:**

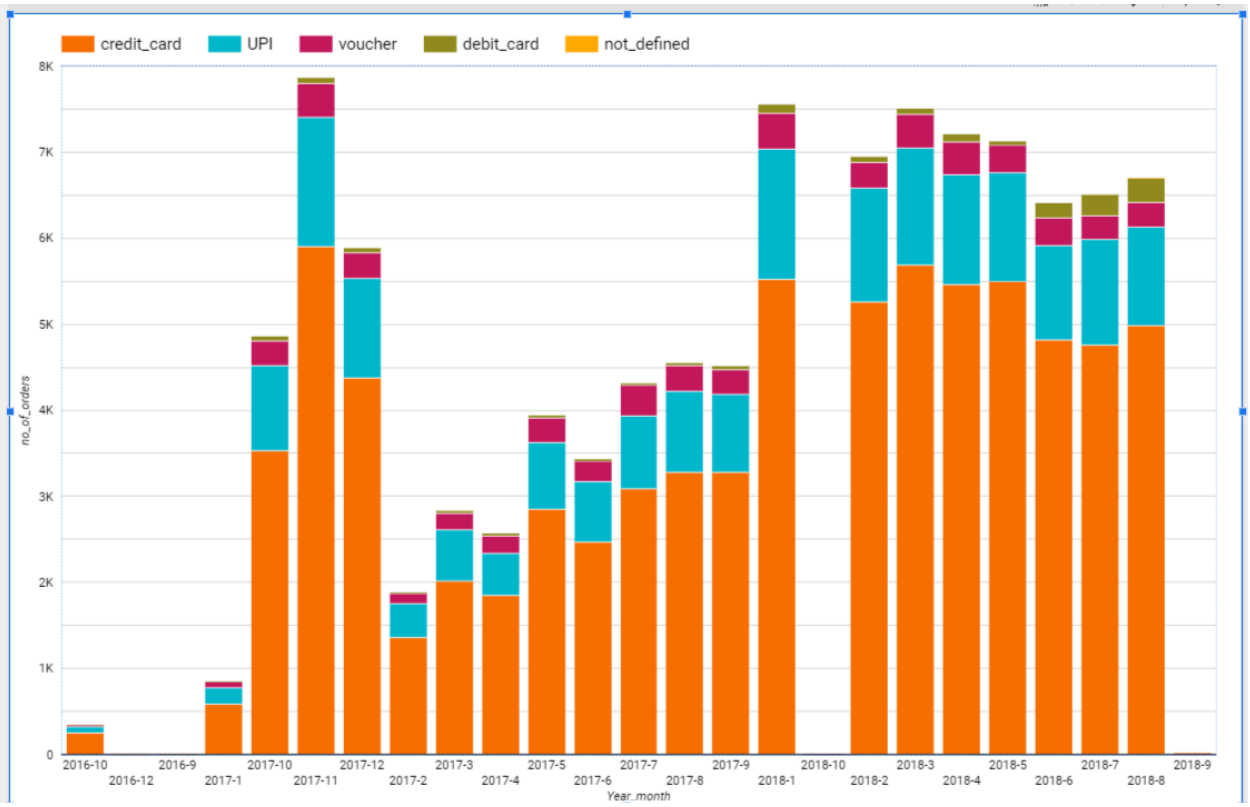
1. Month over Month count of orders for different payment types

Ans)

```
select concat(extract(year from order_purchase_timestamp) ,"-",
extract(month from order_purchase_timestamp) ) as Year_month, payment_type ,
count(o.order_id) as no_of_orders
from `sales.orders` as o
join `sales.payments` as p on o.order_id = p.order_id
group by Year_month, payment_type
order by Year_month ;
```



Row	Year_month	payment_type	no_of_orders
1	2016-10	credit_card	254
2	2016-10	UPI	63
3	2016-10	voucher	23
4	2016-10	debit_card	2
5	2016-12	credit_card	1
6	2016-9	credit_card	3
7	2017-1	credit_card	583
8	2017-1	UPI	197
9	2017-1	voucher	61

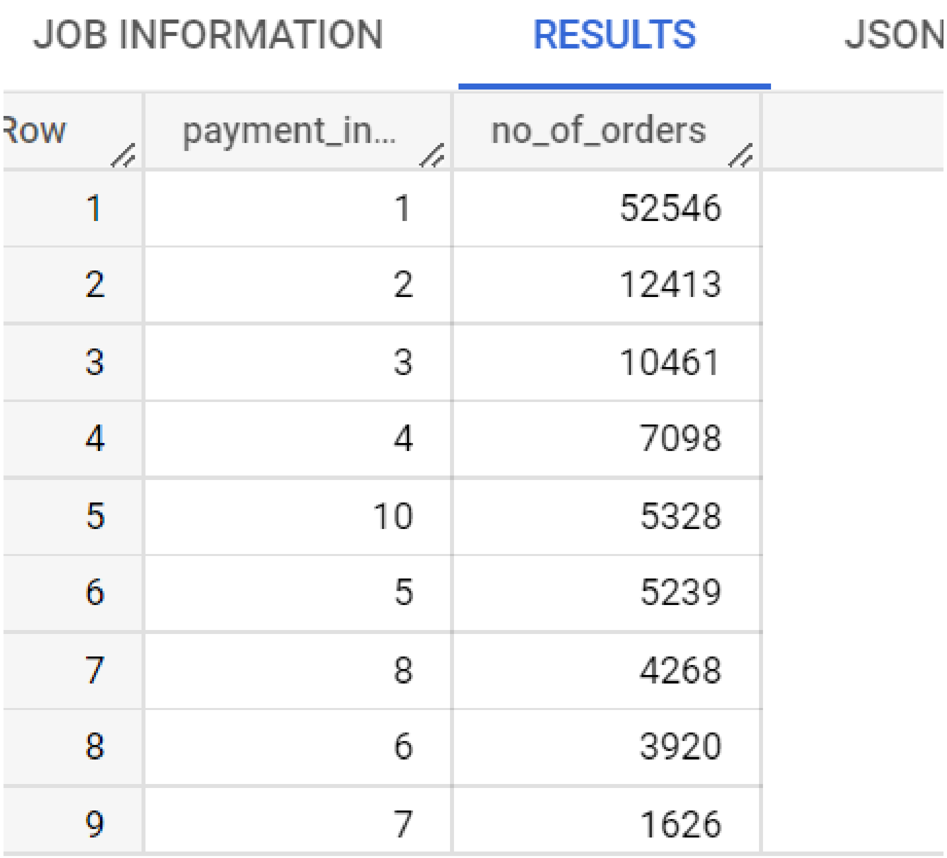


2. Distribution of payment installments and count of orders

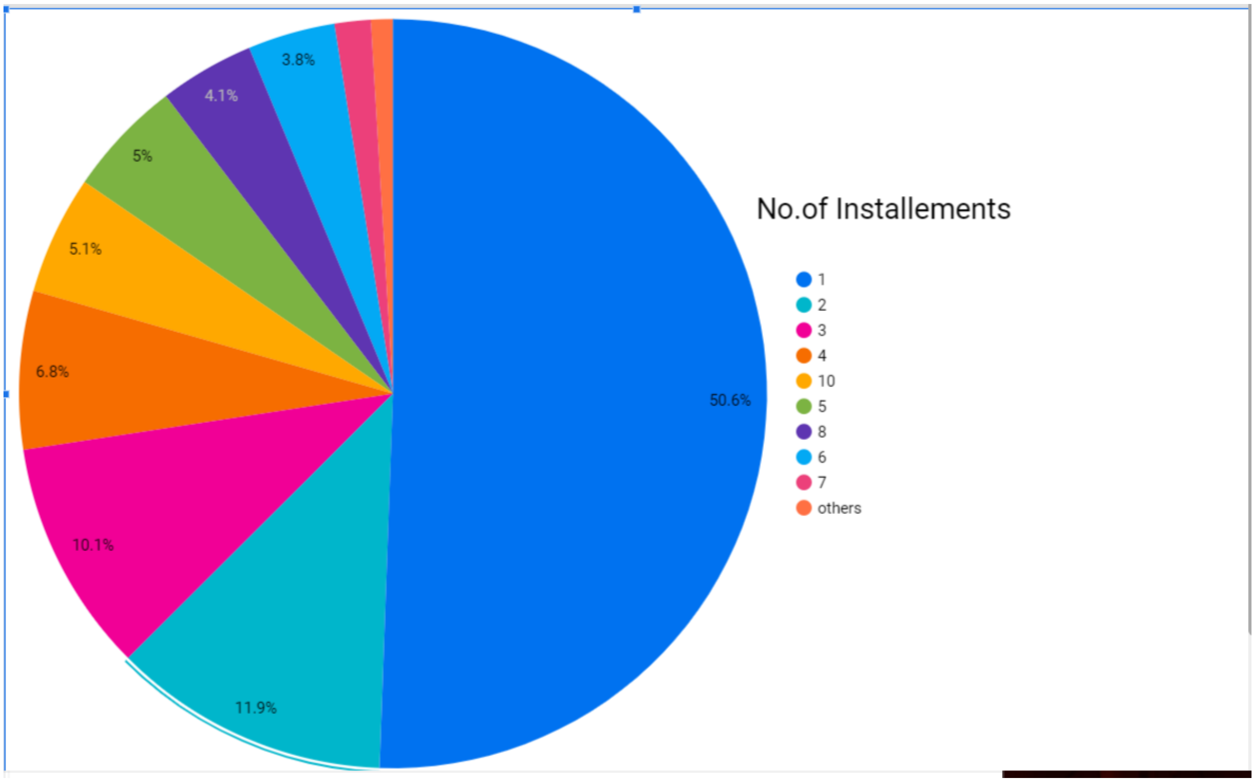
Ans)

```
select payment_installments , count(order_id) as no_of_orders
```

```
from `sales.payments`

group by payment_installments

order by no_of_orders desc;
```



Row	payment_in...	no_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626



★ Actionable Insights & Recommendations -

★ The given data data types Integer,float, string ,Timestamp and all the columns are nullable which means they allow nulls and some rows have been missing in date columns, the data contains the orders of e-commerce business in Brazil along with customer and sellers information with different locations and times with a detailed division of cost of orders as price & freight value.

★ The data is provided for the Orders that are placed between 4th September 2016 to 17th October 2018 which can be considered as a two year data .

★ There is a growing trend on e-commerce in Brazil as per the data and we can also the increasing number of order by year , we can also see the peak number of orders are seen in months of November, January , March which are the time periods of Thanksgiving , Christmas , New-Year and Good Friday , it would be a great profit in sales if the Organization puts up more offers, restock the best selled items and also to launch new products during these months.

★ Compared to 2017,in 2018 we can see a standard number of orders every month .

★ Brazilian customers tend to buy mostly during Afternoon and Night , also very less interested in buying during Dawn.

- 42% of the customers are from state SP which is almost near to half of the orders , it, great that a single state is producing such a high number but the organization can concentrate on other states as well to increase the Revenue.
- There is a cost increase of 1.215% in orders from 2017 to 2018 between months January to August which shows us an increased chance of Revenue and profit from year to year , this shows the positive impact of E-commerce on Economy as the increase of cost is less when compared to inflation rate.
- The total cost and Total price of orders is high in state SP but mean cost and mean price are high in state PB which indicates that people of SP are purchasing more number of products but people of PB are purchasing more expensive products, even the number of orders is less in PB the average cost of products is high , so there is a opportunity to increase the sales in PB as they tend to buy expensive products .
- Mean freight cost is high in RR,PB,RO,AC,PI in these places providing a free delivery above a certain order value could increase the orders from that region along with revenue
- States SP,PR,MG,DF have the lowest Freight cost and delivery time which indicates the warehouse and sellers might be mostly of these regions , So spreading the warehouses across the country and trying to add the local state sellers might increase the orders very effectively
- Payments are mostly done using Credit card , so having ties with credit card companies to provide seasonal offers may result in increased revenue and second most used payment method UPI has higher chances of increasing in future , having ties with UPI companies also helps for future.
- Most of the installments for payments are only one Installments but encouraging the EMI process and creating swift process with awareness of using Credit cards for EMI process can have good results.