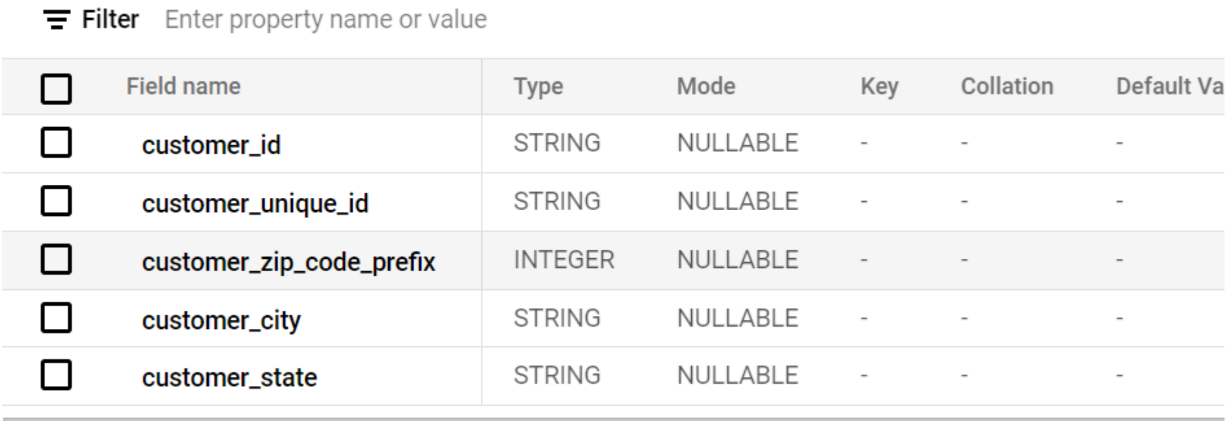
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1)

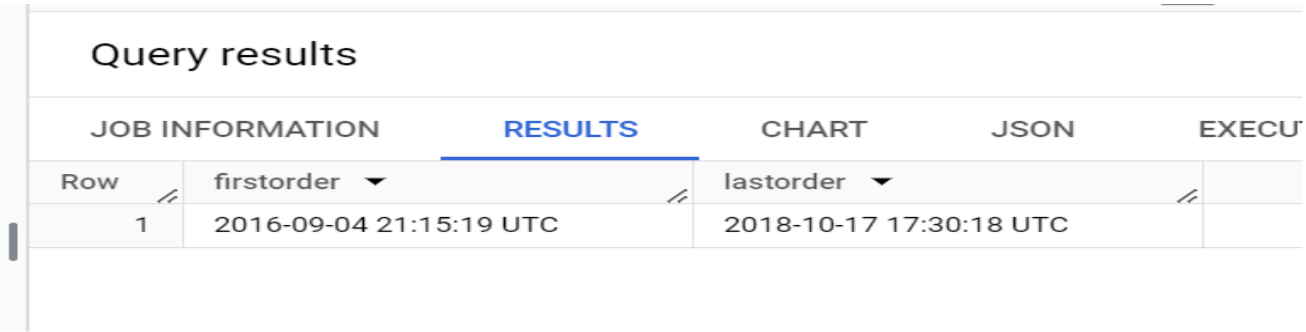


□	Field name	Type	Mode	Key	Collation	Default Va
□	customer_id	STRING	NULLABLE	-	-	-
□	customer_unique_id	STRING	NULLABLE	-	-	-
□	customer_zip_code_prefix	INTEGER	NULLABLE	-	-	-
□	customer_city	STRING	NULLABLE	-	-	-
□	customer_state	STRING	NULLABLE	-	-	-

Inference and recommendation: customer zip code prefix is Integer data type and rest all columns are of string data type.

2) Get the time range between which the orders were placed

```
select min(order_purchase_timestamp) as firstorder,max(order_purchase_timestamp) as
lastorder
 from `BCS1.orders`
```



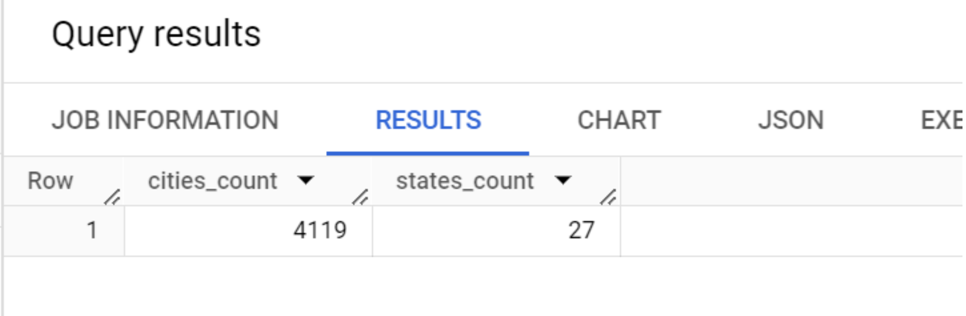
Query results

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

Inference and recommendation: If Purchase timestamp is the time when order was placed then first order was placed on 4th September 2016 at 21:15:19 UTC and last order was placed on 17th October 2018 at 17:30:18 UTC

3)Count the Cities & States of customers who ordered during the given period.

```
 select count(distinct customer_city) as cities_count, count(distinct customer_state)
as states_count
 from `BCS1.customers`
```



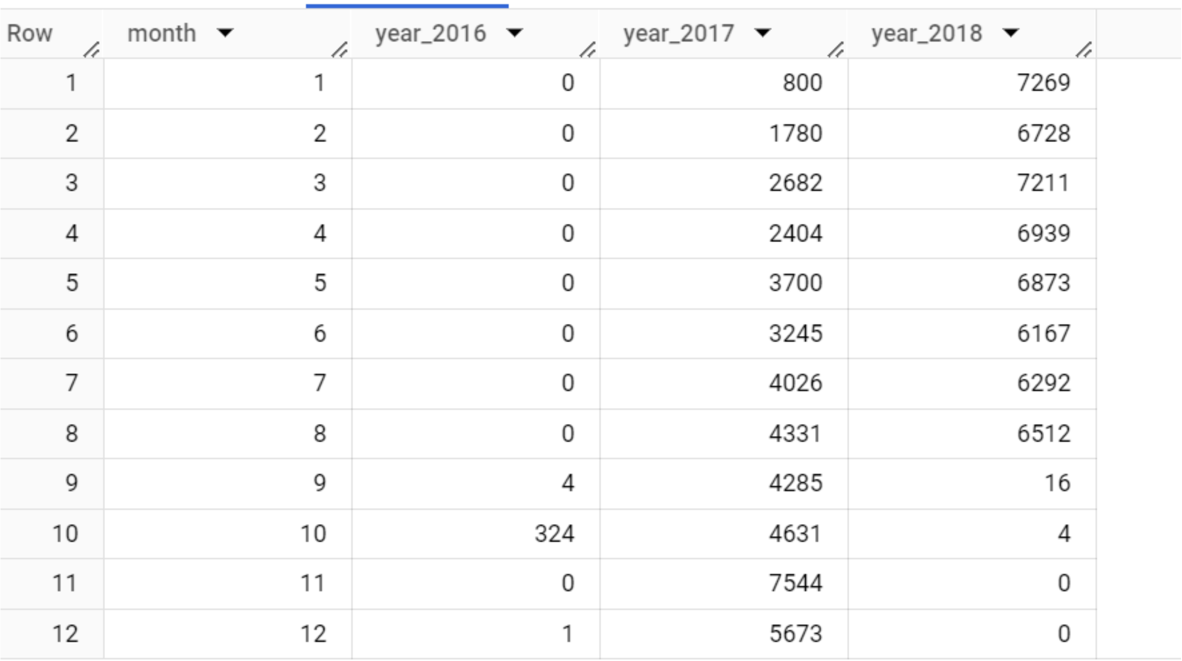
Row	cities_count	states_count
1	4119	27

Inference and recommendation: In total we have 4119 cities and 27 states in Brazil where orders were placed by the customers during the given time.

1. **In-depth Exploration:**

1) Is there a growing trend in the no. of orders placed over the past years?

```
select extract(month from order_purchase_timestamp) as month,
sum( case when  extract(year from order_purchase_timestamp)=2016 then 1 else 0 end) as
year_2016,
sum( case when  extract(year from order_purchase_timestamp)=2017 then 1 else 0 end) as
year_2017,
sum( case when  extract(year from order_purchase_timestamp)=2018 then 1 else 0 end) as
year_2018,
from `BCS1.orders`
group by extract(month from order_purchase_timestamp)
order by month
```



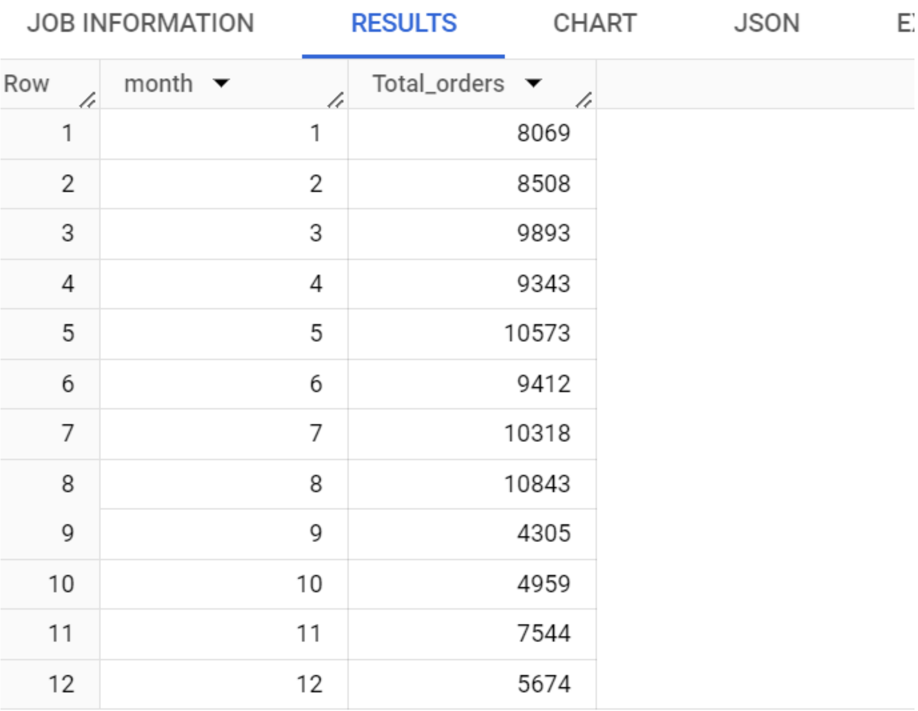
Row	month	year_2016	year_2017	year_2018
1	1	0	800	7269
2	2	0	1780	6728
3	3	0	2682	7211
4	4	0	2404	6939
5	5	0	3700	6873
6	6	0	3245	6167
7	7	0	4026	6292
8	8	0	4331	6512
9	9	4	4285	16
10	10	324	4631	4
11	11	0	7544	0
12	12	1	5673	0

Inference: We can see from above table month on month orders for each year.
In 2016, we had less orders. Mostly orders were in October.
In 2017, orders did increase every month and increased a lot in November.
In 2018, There were similar number of orders each month but those declined from September dill December. January and March had highest number of orders in this year.

Recommendation: Company can look for the reasons it led to decline in the orders for those months and that particular year and look if there are any major national/international events that led to decline or just the customer behavior.

2) Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
with cte as(
select *, extract(Month from order_purchase_timestamp) as month from `BCS1.orders`
)
select month,count(month) as Total_orders
from cte
group by month
order by month
```



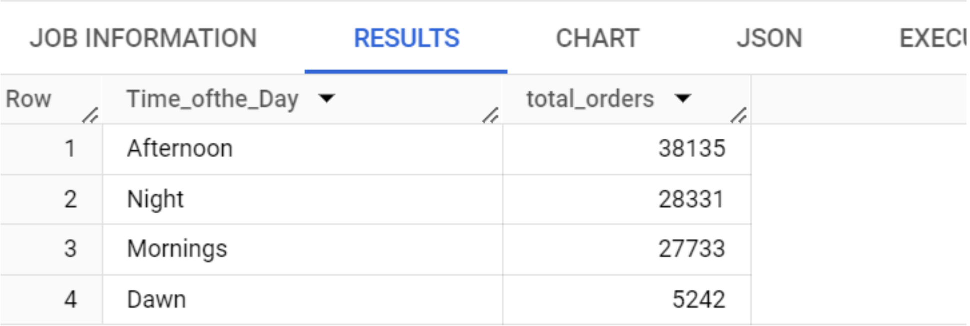
Row	month	Total_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

Inference: We have monthly total orders as above. It is quite clear that the number of orders were very less from 9th or September month till December compared to orders in other months. Also, Orders were highest in May, July and August
Recommendation: Company can introduce some offers from September to December to increase the sales. Also, pricing model can be changed for May, July and August to increase revenue as during those times the number of orders placed would be high

3) During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

```
with cte as (
select order_purchase_timestamp,
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 'Mornings'
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_ofthe_Day

 from `BCS1.orders` join `BCS1.customers` using(customer_id)
)
select Time_ofthe_Day,count(Time_ofthe_Day) as total_orders
from cte
group by Time_ofthe_Day
order by total_orders desc
```



Row	Time_ofthe_Day	total_orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Inference: It is clearly visible, Brazilian customers ordered most during the Afternoon.
Recommendations: Company can introduce some limited time offers for Dawn period so that sales can increase during that time also

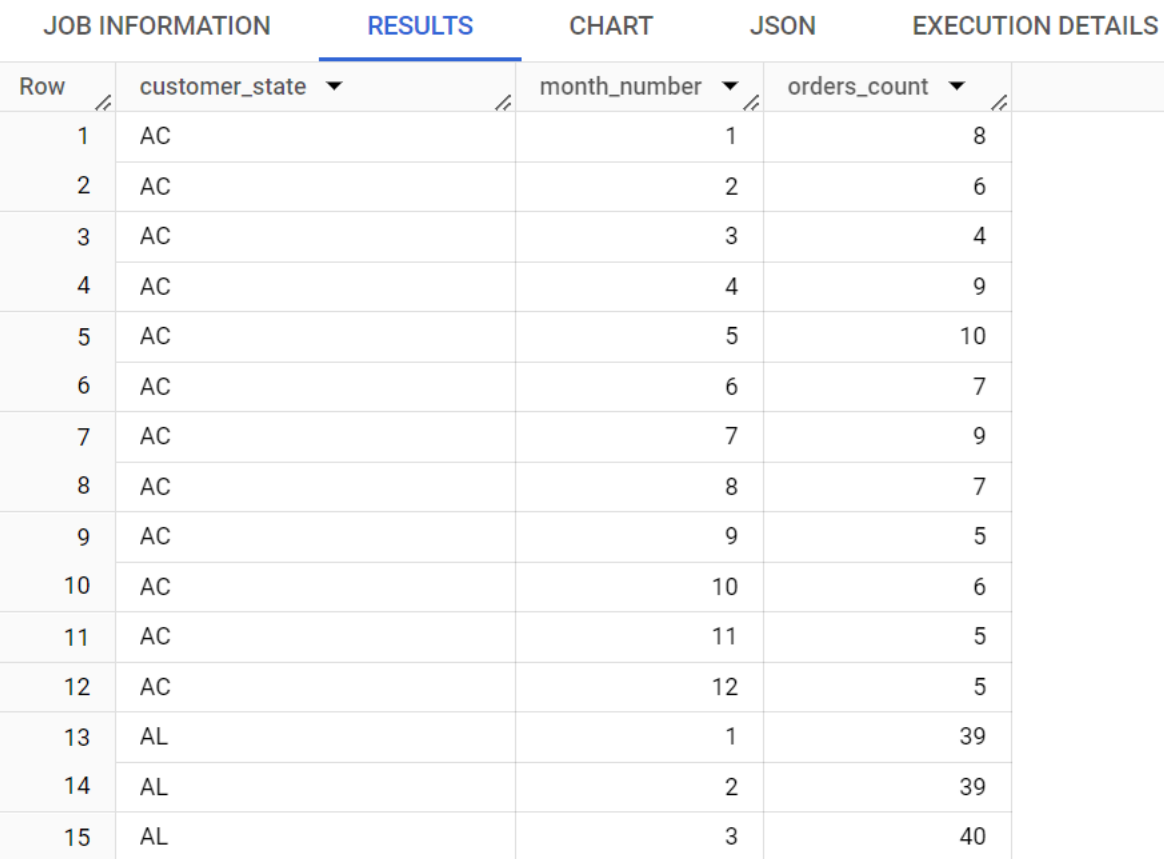
Evolution of E-commerce orders in the Brazil region:
1) Get the month on month no. of orders placed in each state.

```
select customer_state,extract(month from order_purchase_timestamp) as
month_number,count(order_id) as orders_count

 from  `BCS1.customers` left join `BCS1.orders` using(customer_id)

 group by customer_state,extract(month from order_purchase_timestamp)

order by customer_state,month_number
```

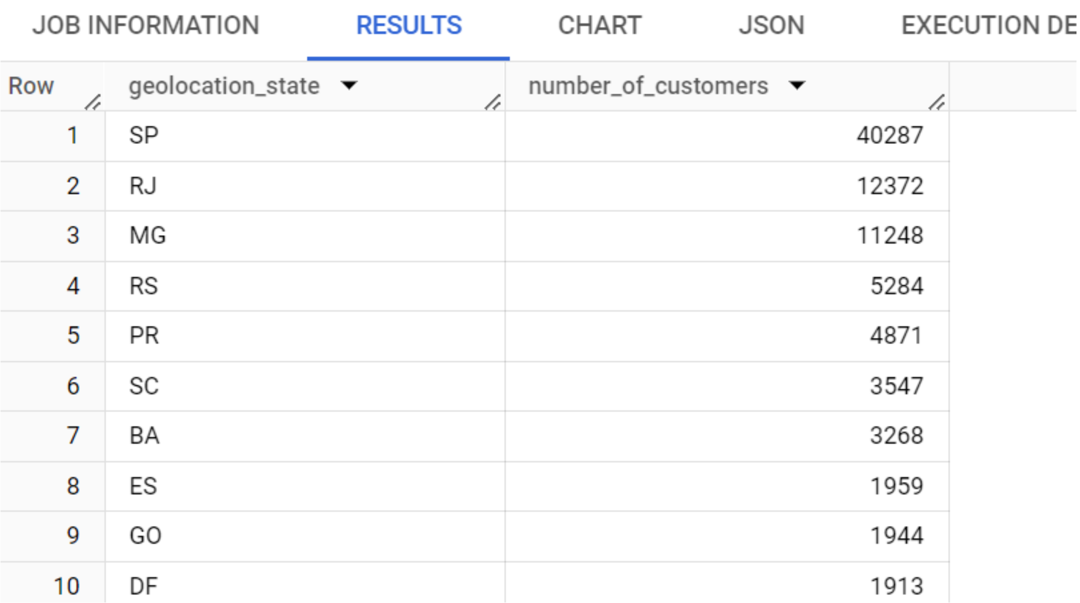


Row	customer_state	month_number	orders_count
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5
12	AC	12	5
13	AL	1	39
14	AL	2	39
15	AL	3	40

Inference: Above table shows Month on month Orders for each state sorted in ascending order of state and then months (1 being January and 12 being December)
Recommendation: We can find out for any state for example AC has least order in March and a smaller number of orders in this state as compared to other states. So, we can target our improvements in such states and months if needed.

2) How are the customers distributed across all the states?

```
select geolocation_state,count(distinct customer_Unique_id) as number_of_customers
from  `BCS1.geolocation`  left join `BCS1.customers`   on  geolocation_zip_code_prefix
=  customer_zip_code_prefix
group by geolocation_state
order by number_of_customers desc
```



Row	geolocation_state	number_of_customers
1	SP	40287
2	RJ	12372
3	MG	11248
4	RS	5284
5	PR	4871
6	SC	3547
7	BA	3268
8	ES	1959
9	GO	1944
10	DF	1913

Inference: With above sorted table we can see the highest number of customers are from SP (Sao Paulo) state. Similarly, if we scroll to the bottom, we can find the state with least customers.
Recommendation: State RR has least number of customers (45) and other states like AP, AC , AM etc. has a huge scope of customer onboarding . These states can be targeted with better offerings to increase customer base.

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

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

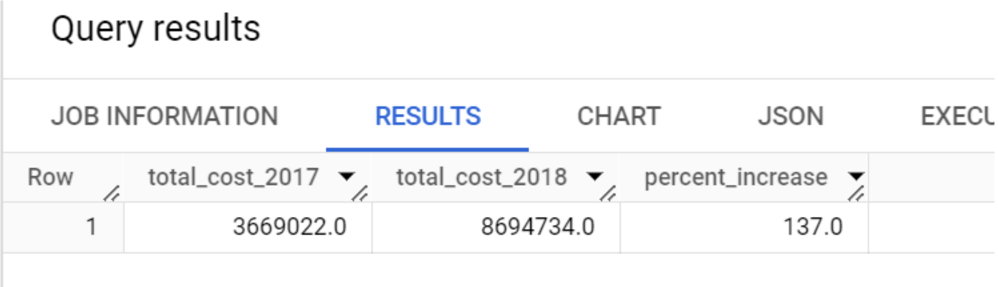
```
with cte as(
select extract(year from order_purchase_timestamp) as year,Round(sum(payment_value))
as total_cost
from `BCS1.orders` join `BCS1.payments` using(order_id)

where extract(year from order_purchase_timestamp) between 2017 and 2018 and
extract(month from order_purchase_timestamp) between 01 and 08

group by extract(year from order_purchase_timestamp)

),
t2 as (
select total_cost as total_cost_2018 from cte where year=2018
),
t3 as
(select total_cost as total_cost_2017 from cte where year=2017)

select total_cost_2017 ,total_cost_2018,Round(((total_cost_2018-
total_cost_2017)/total_cost_2017) * 100) as percent_increase from t2,t3
```



Row	total_cost_2017	total_cost_2018	percent_increase
1	3669022.0	8694734.0	137.0

Inference: with Above table we can see the total cost of orders from year 2017 as well as 2018 (only including months from Jan to Aug in both years) we can see there was 137 % increase in the cost of orders in 2018

Recommendation: This is great increase in total cost of orders placed as number of orders placed were higher in next year. Company can slightly increase the price and see the significant profit increase.

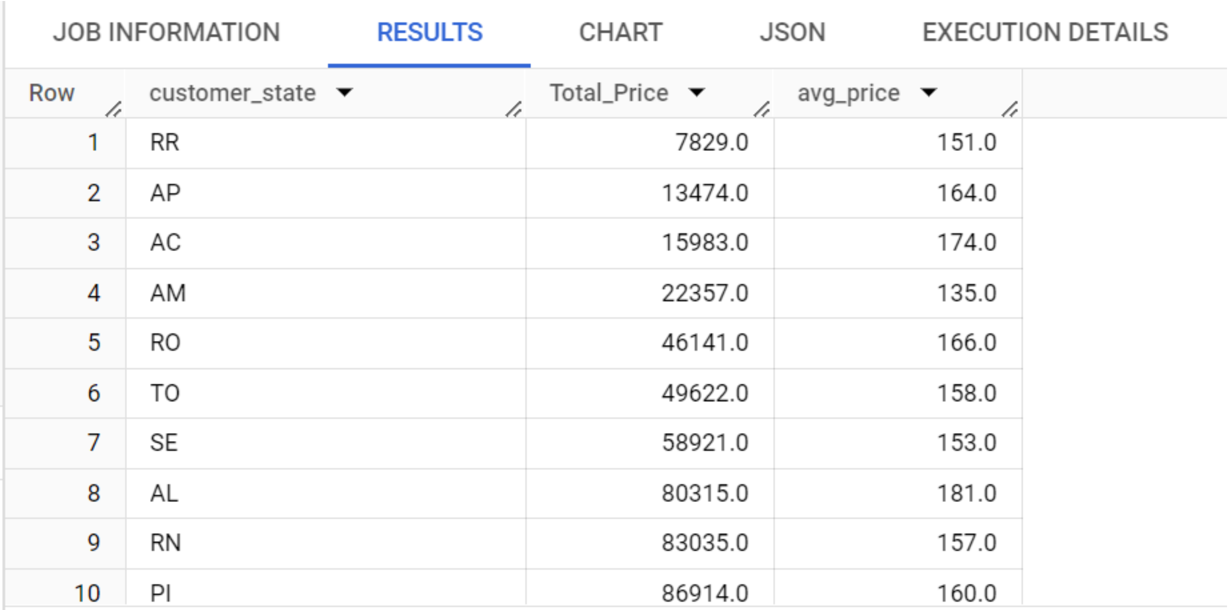
2) Calculate the Total & Average value of order price for each state

```
select customer_state,Round(sum(price)) as Total_Price,Round(Avg(price)) as avg_price

from `BCS1.customers` join `BCS1.orders` using(customer_id)  join `BCS1.order_items` using(order_id)

group by customer_state

order by Total_price , avg_price
```



Row	customer_state	Total_Price	avg_price
1	RR	7829.0	151.0
2	AP	13474.0	164.0
3	AC	15983.0	174.0
4	AM	22357.0	135.0
5	RO	46141.0	166.0
6	TO	49622.0	158.0
7	SE	58921.0	153.0
8	AL	80315.0	181.0
9	RN	83035.0	157.0
10	PI	86914.0	160.0

Inference: With Above table, we can clearly see the state RR has the least total order value with almost similar average order price as compared to other states

Recommendation: Need to target this RR state along with AP, AC and AM to increase the count of the number of orders placed

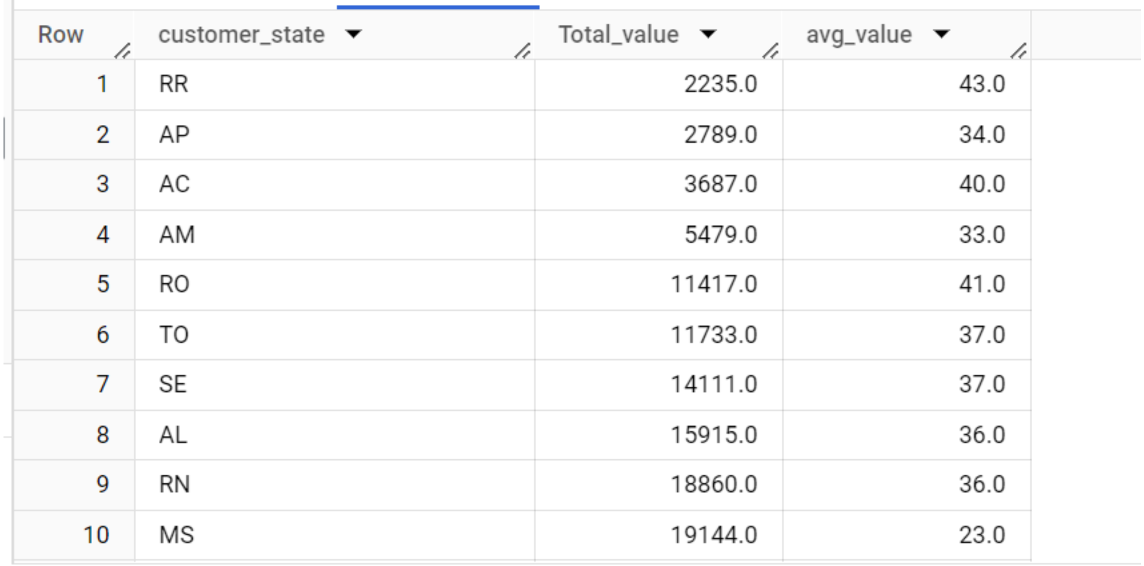
3) Calculate the Total & Average value of order freight for each state.

```
select customer_state,Round(sum(freight_value)) as
Total_value,Round(Avg(freight_value)) as avg_value

from `BCS1.customers`  join `BCS1.orders` using(customer_id)  join `BCS1.order_items`
using(order_id)

group by customer_state

order by Total_value , avg_value
```



Row	customer_state	Total_value	avg_value
1	RR	2235.0	43.0
2	AP	2789.0	34.0
3	AC	3687.0	40.0
4	AM	5479.0	33.0
5	RO	11417.0	41.0
6	TO	11733.0	37.0
7	SE	14111.0	37.0
8	AL	15915.0	36.0
9	RN	18860.0	36.0
10	MS	19144.0	23.0

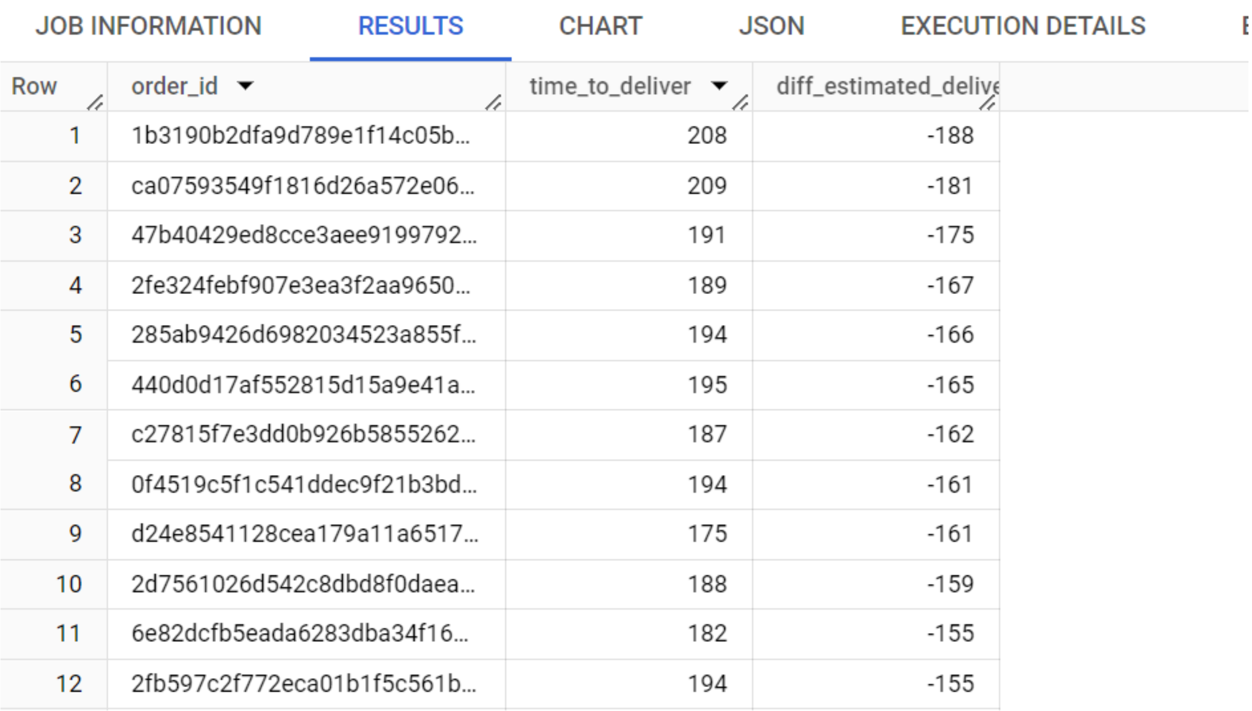
Inference: we can see with above table, RR has least total freight value and AP, AC, AM close to that.

Recommendation: States like RR, AP, AC and AM have considerable scope of improvement in terms of freight value

2.Analysis based on sales, freight, and delivery time.

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.
Do this in a single query.

```
select order_id, timestamp_diff
(order_delivered_customer_date,order_purchase_timestamp,Day) as time_to_deliver,
timestamp_diff( order_estimated_delivery_date,order_delivered_customer_date,Day) as
diff_estimated_delivery
from `BCS1.orders`
where order_delivered_customer_date is not null
order by diff_estimated_delivery
```



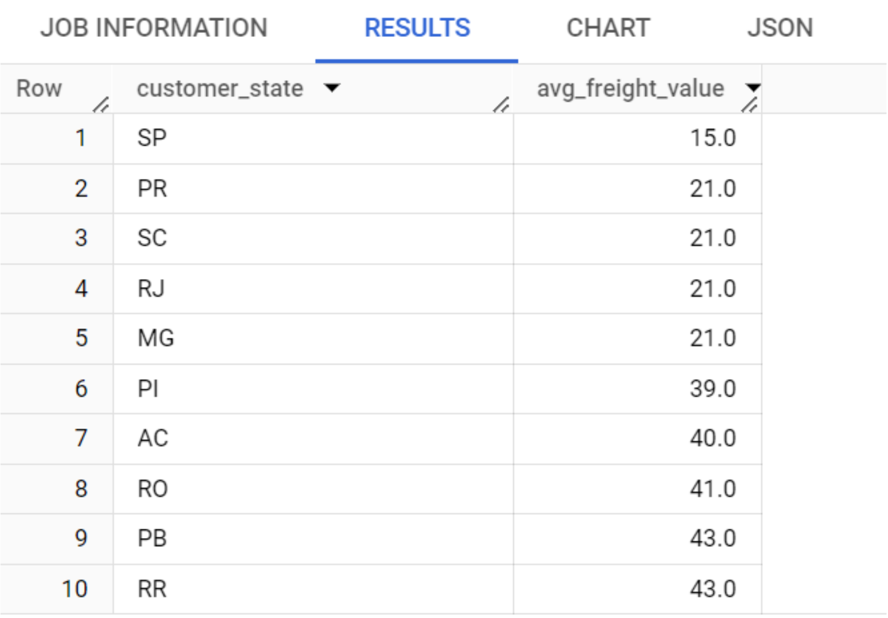
Row	order_id	time_to_deliver	diff_estimated_delive
1	1b3190b2dfa9d789e1f14c05b...	208	-188
2	ca07593549f1816d26a572e06...	209	-181
3	47b40429ed8cce3aee9199792...	191	-175
4	2fe324febf907e3ea3f2aa9650...	189	-167
5	285ab9426d6982034523a855f...	194	-166
6	440d0d17af552815d15a9e41a...	195	-165
7	c27815f7e3dd0b926b5855262...	187	-162
8	0f4519c5f1c541ddec9f21b3bd...	194	-161
9	d24e8541128cea179a11a6517...	175	-161
10	2d7561026d542c8dbd8f0daea...	188	-159
11	6e82dcfb5eada6283dba34f16...	182	-155
12	2fb597c2f772eca01b1f5c561b...	194	-155

Inference: Against each order we have the number of days it took to deliver the order as well as the time difference in estimated and actual delivery. The first record tells us the order with -188 difference, meaning it took 188 days extra to actually deliver this order than the estimated delivery date.

Recommendation: Company can start intervening for such orders with their delivery partners and find out the root cause and improve on those aspects which led to the delay of such orders and also provide some offers or coupons to compensate the customer for such delays so that it does not lose the customers.

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

```
with cte as (
select customer_state,Round(avg(freight_value)) as avg_freight_value
 from `BCS1.customers` join `BCS1.orders` using(customer_id) join `BCS1.order_items`
using(order_id)
 group by customer_state
)
(select customer_state,avg_freight_value from cte order by avg_freight_value desc
limit 5)
union all
(select customer_state,avg_freight_value from cte order by avg_freight_value limit 5)
order by avg_freight_value
```



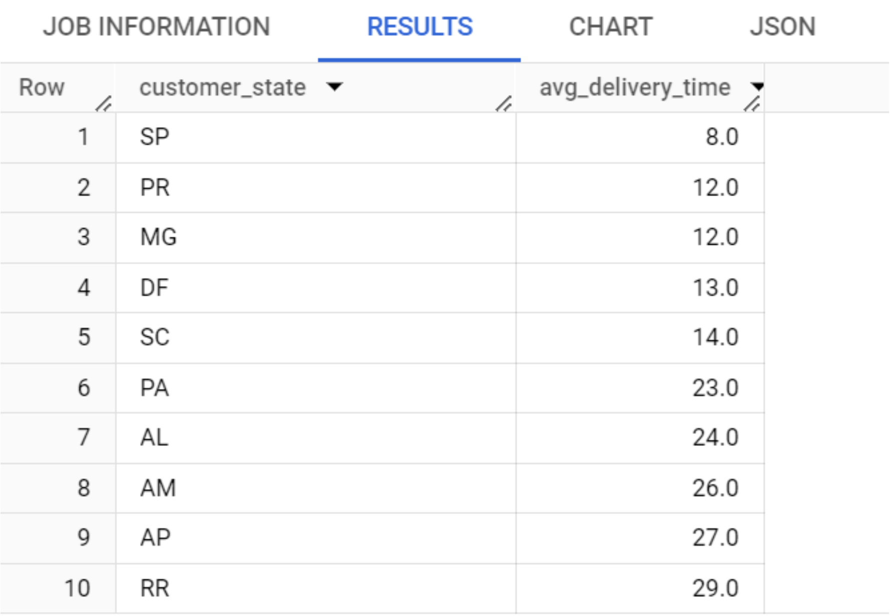
Row	customer_state	avg_freight_value
1	SP	15.0
2	PR	21.0
3	SC	21.0
4	RJ	21.0
5	MG	21.0
6	PI	39.0
7	AC	40.0
8	RO	41.0
9	PB	43.0
10	RR	43.0

Inference: Above table tells us the top 5 states with lowest average freight value followed by top 5 states with highest average freight value sorted in increasing order of average freight value. Clearly, there is a significant difference in lowest average freight value (15) and highest average freight value (43)

Recommendation: State PB and RR has the highest average freight value and similarly RO, AC and PI. Company can explore alternate cargo options to get better pricing and increase profits in these states.

3) Find out the top 5 states with the highest & lowest average delivery time.

```
with cte as (
select  timestamp_diff (order_delivered_customer_date,order_purchase_timestamp,Day) as
delivery_time,customer_state
 from `BCS1.customers` join `BCS1.orders` using(customer_id)
),
t2 as (
select customer_state, Round(avg(delivery_time)) as avg_delivery_time from cte
where delivery_time is not null
group by customer_state
)
(
select customer_state,avg_delivery_time from t2 order by avg_delivery_time desc limit
5)
union all
(
select customer_state,avg_delivery_time from t2 order by avg_delivery_time limit 5
)
order by avg_delivery_time
```



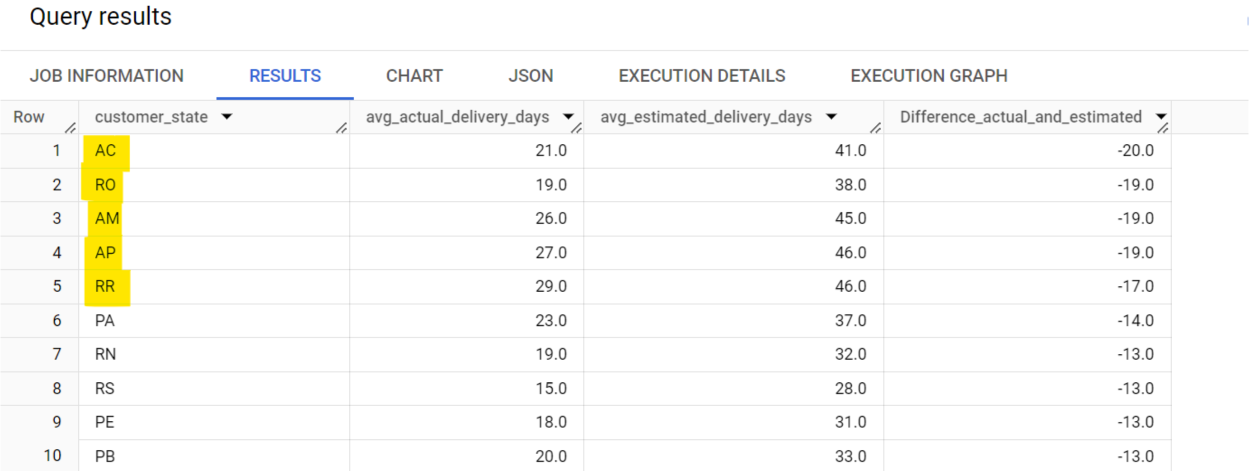
Row	customer_state	avg_delivery_time
1	SP	8.0
2	PR	12.0
3	MG	12.0
4	DF	13.0
5	SC	14.0
6	PA	23.0
7	AL	24.0
8	AM	26.0
9	AP	27.0
10	RR	29.0

Inference: Above table tells us the top 5 states with lowest average delivery time in Days followed by top 5 states with highest average delivery time in days sorted by average delivery time in increasing order
RR has highest average delivery time and SP has lowest average delivery time

Recommendation: Company needs to investigate states with highest average delivery time and find out at which junction these delays are happening and work on those areas. Could be with delivery partner or maybe the distance from which these orders are shipped.

4) Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```
with cte as(
select  customer_state,Round(avg(timestamp_diff
(order_delivered_customer_date,order_purchase_timestamp,Day))) as
avg_actual_delivery_days,
 Round(avg(timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,
Day))) as avg_estimated_delivery_days
 from `BCS1.customers` join `BCS1.orders` using(customer_id)
 where order_delivered_customer_date is not null
 group by customer_state
)
select *,cte.avg_actual_delivery_days - cte.avg_estimated_delivery_days as
Difference_actual_and_estimated  from cte
order by Difference_actual_and_estimated
```



Row	customer_state	avg_actual_delivery_days	avg_estimated_delivery_days	Difference_actual_and_estimated
1	AC	21.0	41.0	-20.0
2	RO	19.0	38.0	-19.0
3	AM	26.0	45.0	-19.0
4	AP	27.0	46.0	-19.0
5	RR	29.0	46.0	-17.0
6	PA	23.0	37.0	-14.0
7	RN	19.0	32.0	-13.0
8	RS	15.0	28.0	-13.0
9	PE	18.0	31.0	-13.0
10	PB	20.0	33.0	-13.0

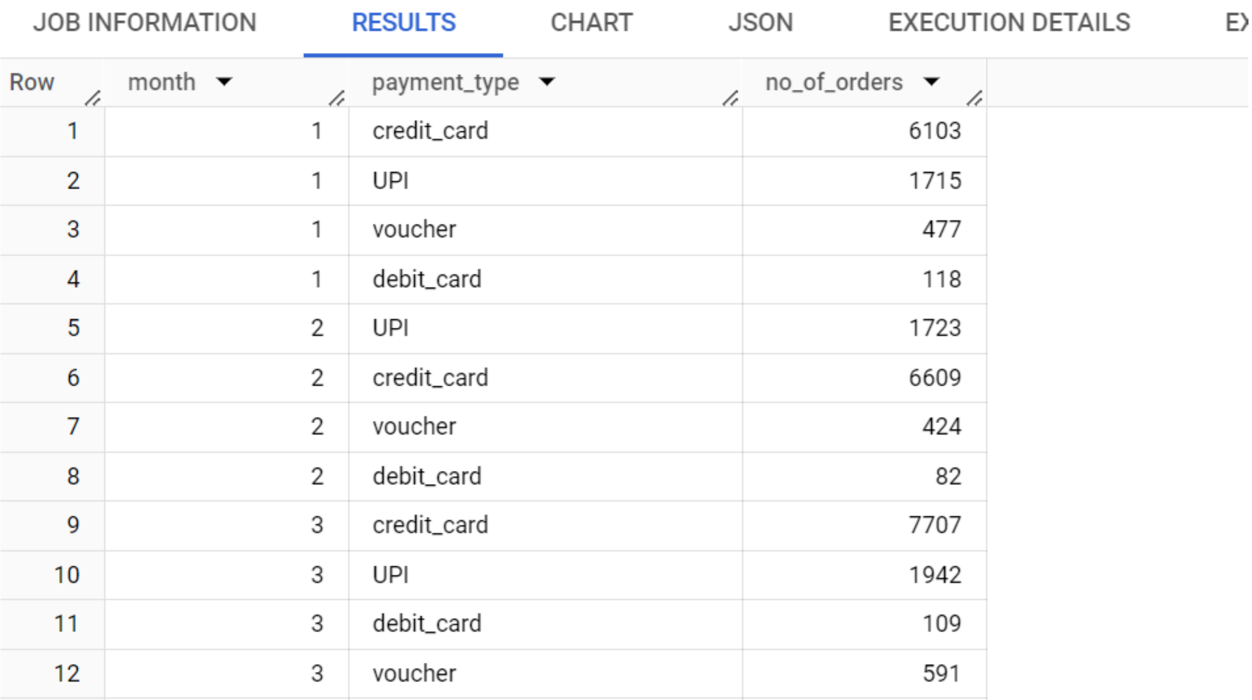
Inference: From the above table we can see the column Difference_actual_and_estimated. This column is the result of difference in average of actual delivery date and average of estimated delivery date for each state.
Negative sign is indicating that ordered was delivered before the estimated delivery day. Top 5 states with the fastest delivery as compared to estimated delivery are highlighted in yellow with AC state having the fastest actual delivery compared to estimated delivery followed by RO and so on till RR.

Recommendation: Company can value the reasons for so fast delivery in these states , improve its function in other states which has slower deliveries and compare the factors which led to faster and slower deliveries

2.Analysis based on the payments:

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

```
select extract(month from order_purchase_timestamp) as
month,payment_type,count(order_id) as no_of_orders from `BCS1.orders` join
`BCS1.payments` using(order_id)
group by extract(month from order_purchase_timestamp),payment_type
order by month
```



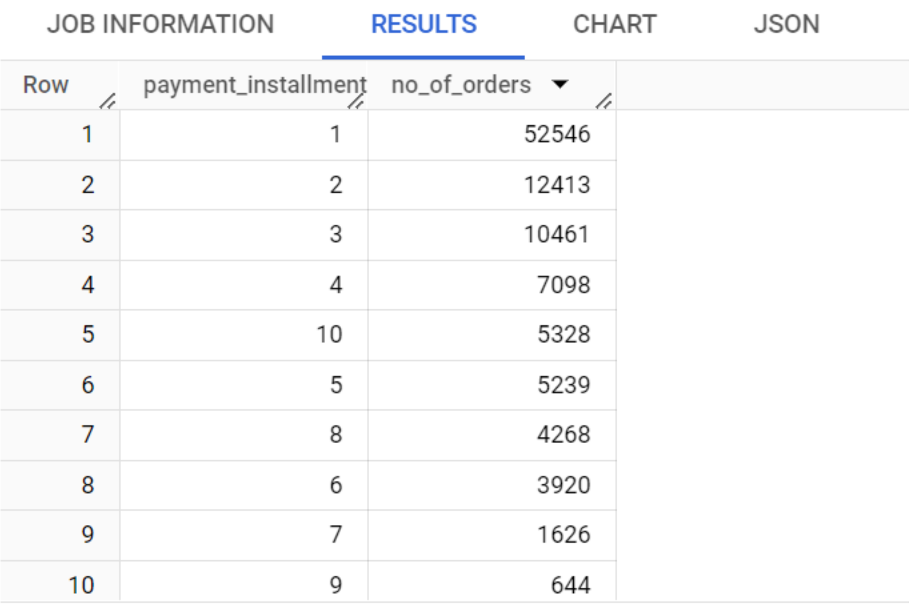
Row	month	payment_type	no_of_orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6609
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942
11	3	debit_card	109
12	3	voucher	591

Inference: With above table we can see the months 1(Jan), 2(Feb) and so on. Along with that the count of orders placed for each payment type in that month.
Clearly, credit card is most widely used for payments every month.

Recommendation: Company can partnership with some banks for their special credit cards, provide some discounts on those cards and this would further increase the number of orders in those categories.

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

```
select  payment_installments,count(order_id)as no_of_orders from `BCS1.orders` join
`BCS1.payments` using(order_id)
where payment_installments>=1
group by payment_installments
order by no_of_orders desc
```



Row	payment_installment	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
10	9	644

Inference: with above two columns we can see the number of orders placed based on number of payments installments made and sorted with number of orders in descending order
There are 52546 orders which are highest in number for which 1 installment has been paid in their EMI.
Clearly, there are lots of orders within less payment installments.
Recommendation: Company can increase the order price slightly as customers are able to pay for most of the orders in very less installments, this would generate more profit.