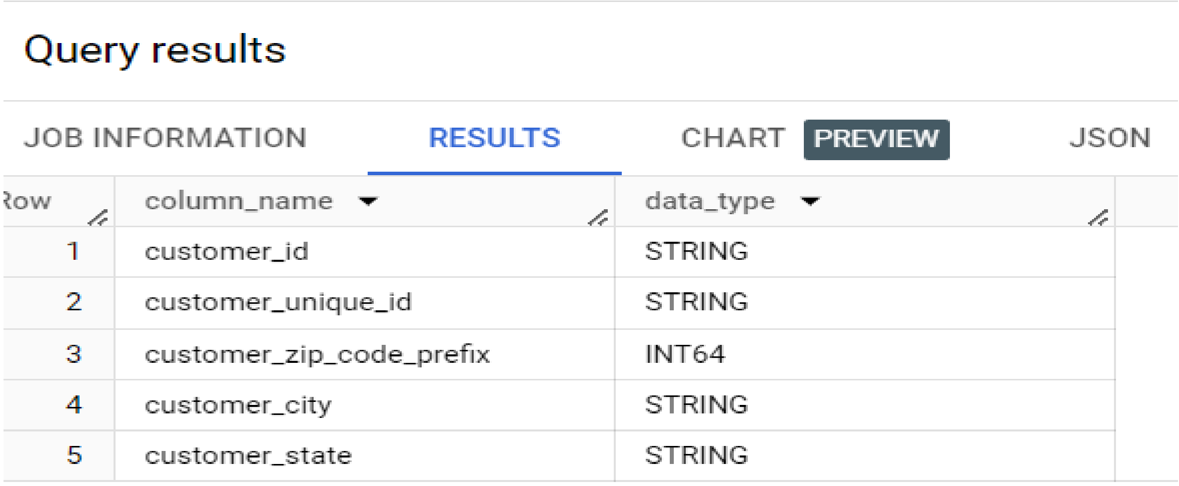
I. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

A. Data type of all columns in the “customers” table.

Ans A .

Query :

```
select column_name,data_type
from steady-burner-403317.store.INFORMATION_SCHEMA.COLUMNS
where table_name='customers';
```



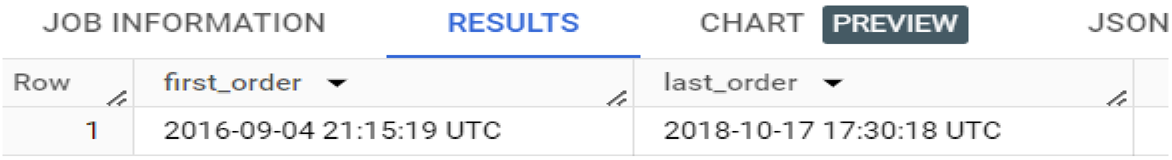
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Insights : There are four string type data types and On integer

B.Get the time range between which the orders were placed.

Ans

```
select min(order_purchase_timestamp) as first_order,max(order_purchase_timestamp) as
last_order
from `store.orders`
```

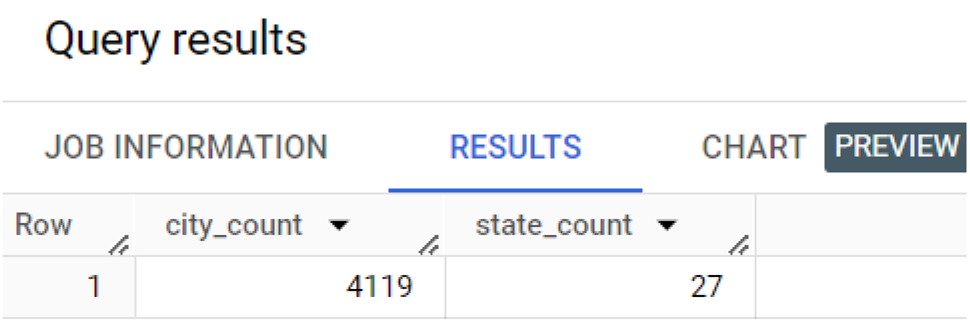


Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights : first order was placed in 2016 and last order was in 2018

C.Count the Cities & States of customers who ordered during the given period.
Ans :

```
select count(distinct customer_city) as city_count,count(distinct customer_state) as
state_count
from `store.customers` c
join `store.orders` o
on o.customer_id=c.customer_id
```



Row	city_count	state_count
1	4119	27

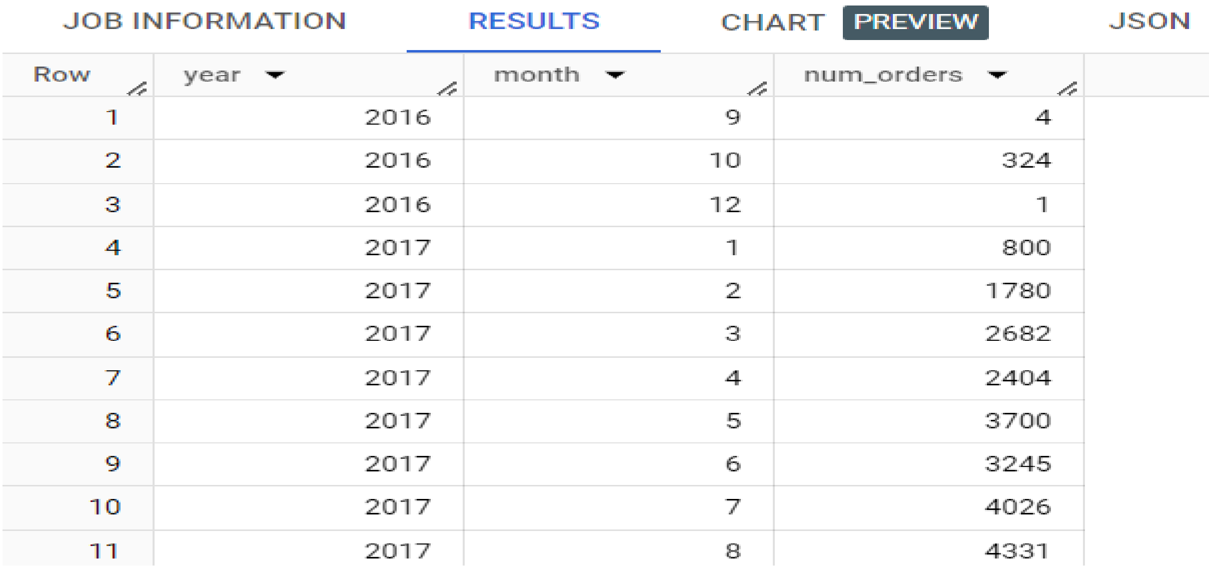
Insights : total cities are 4119 and 27 states

2. In-depth Exploration:

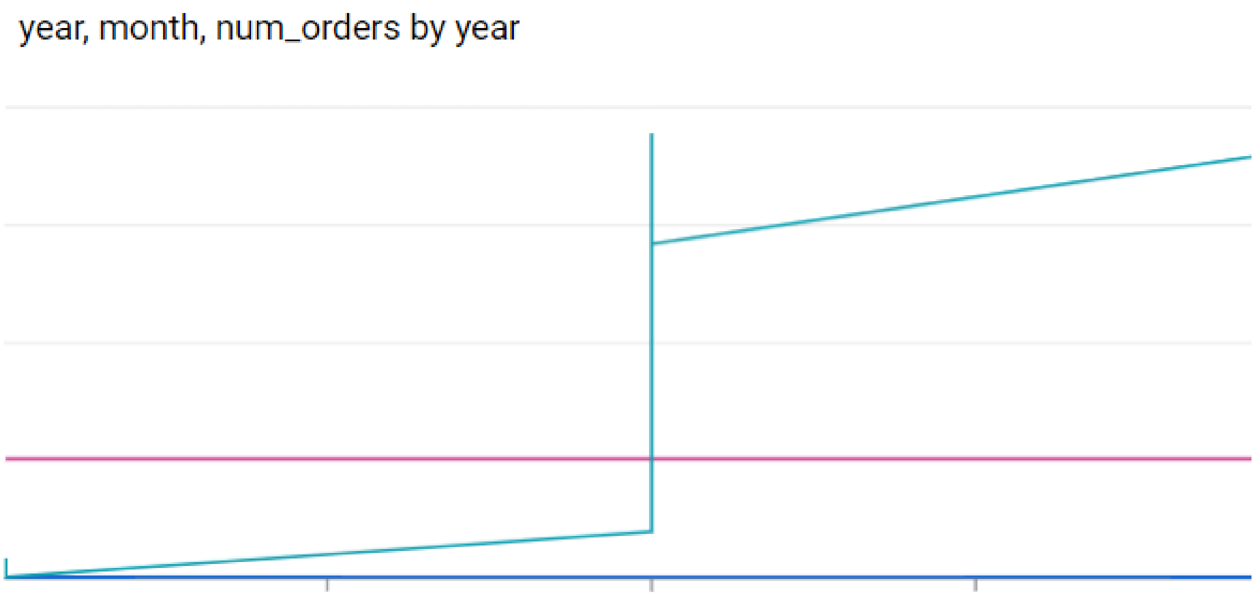
A. Is there a growing trend in the no. of orders placed over the past years?

Ans

```
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(1) as num_orders
from `store.orders`
group by year,month
order by year,month
```

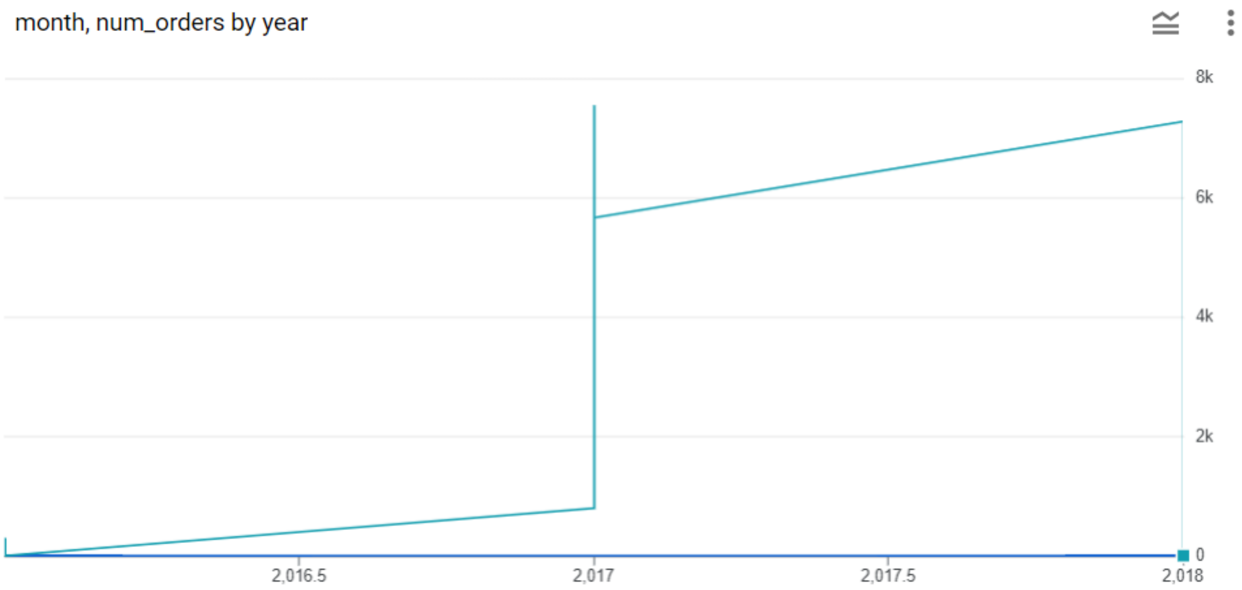


Row	year	month	num_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331



Insights: number on orders on 2016 month september is 4 and 2017 is last with orders 4331

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

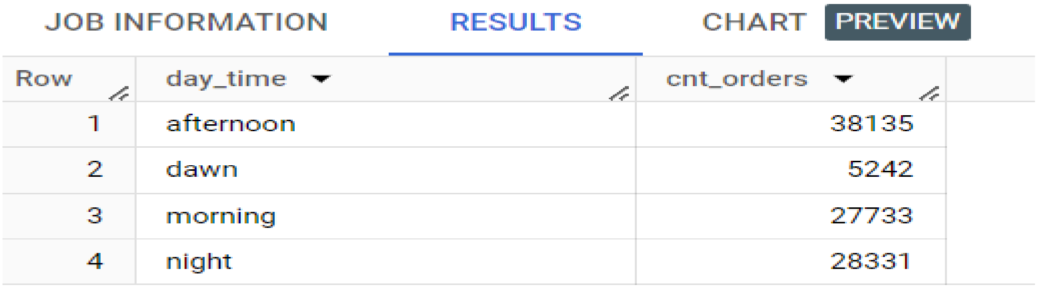


c.During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

Ans

```
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 day_time,
count(order_id) as cnt_orders
from `store.orders`
group by 1
order by 1
```



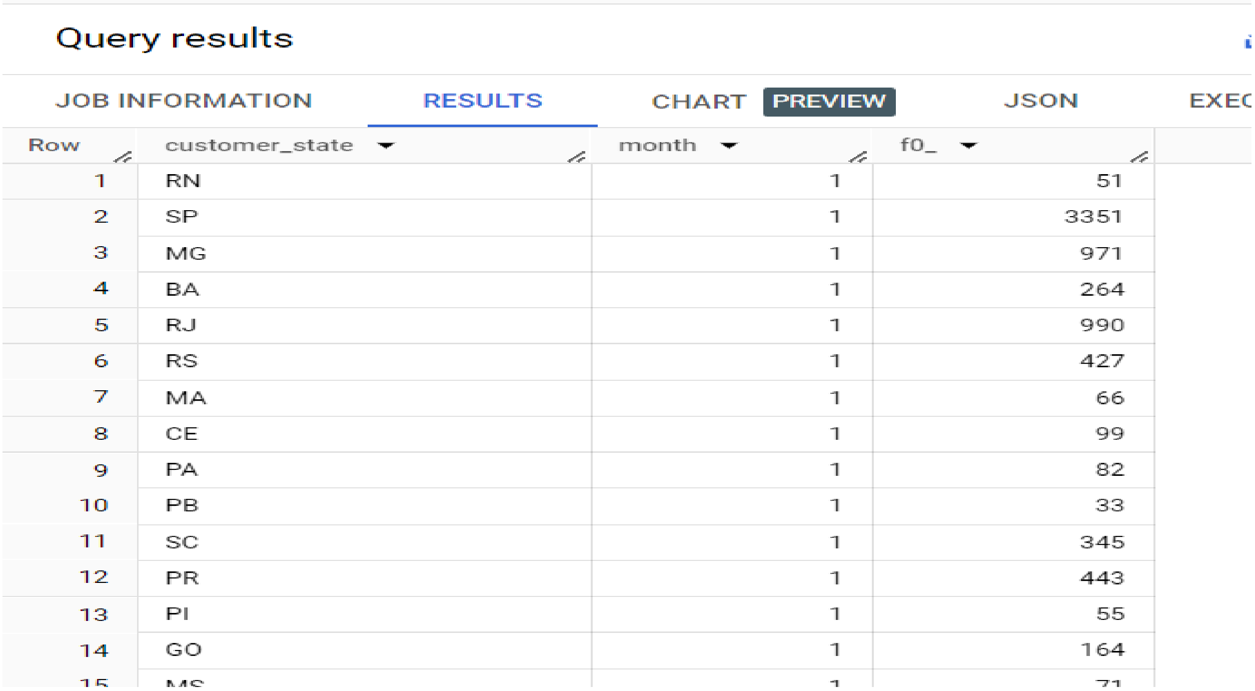
Row	day_time	cnt_orders
1	afternoon	38135
2	dawn	5242
3	morning	27733
4	night	28331

Insights : we have known the total orders placed in afternoon ,dawn,morning and night

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

A. Get the month on month no. of orders placed in each state.

Ans
```
select customer_state,extract(month from order_purchase_timestamp) as
month,count(order_id)
from `store.customers` c
join `store.orders` o
on o.customer_id=c.customer_id
group by 1,2
order by month
```



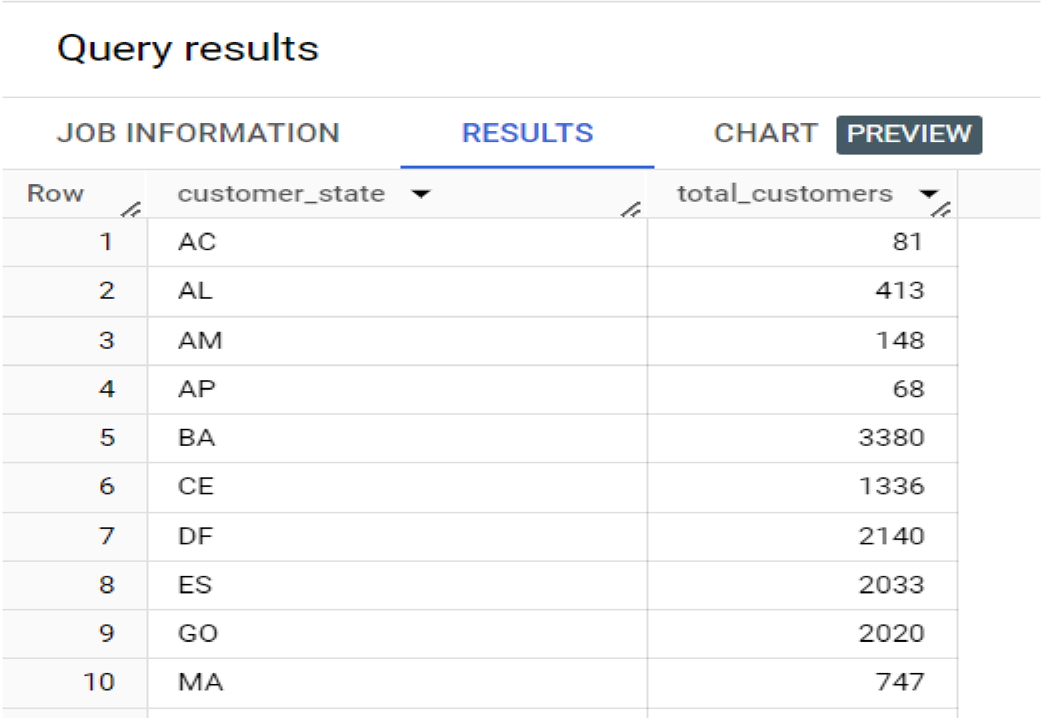
Row	customer_state	month	f0_
1	RN	1	51
2	SP	1	3351
3	MG	1	971
4	BA	1	264
5	RJ	1	990
6	RS	1	427
7	MA	1	66
8	CE	1	99
9	PA	1	82
10	PB	1	33
11	SC	1	345
12	PR	1	443
13	PI	1	55
14	GO	1	164
15	MS	1	71

Insights: each month the number orders palced

B. How are the customers distributed across all the states?

Ans

```
  select customer_state,count(customer_id) as total_customers
from `store.customers`
group by 1
order by 1
```



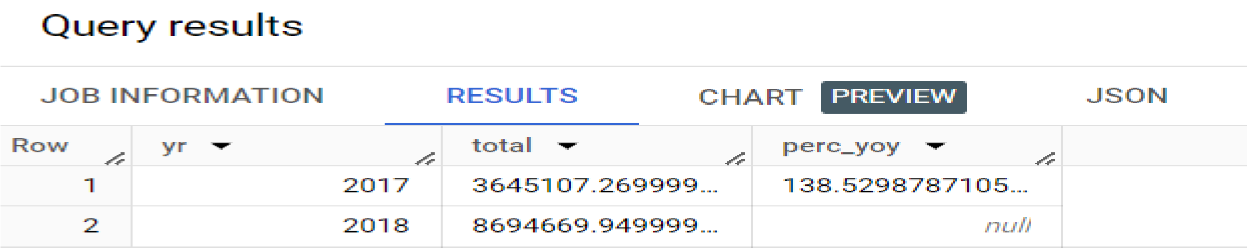
Row	customer_state	total_customers
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

```
Insights : Customers distributed across states
```

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

A. Get the % increase in the cost of orders from year 2017 to 2018 *(include months between Jan to Aug only)*.
You can use the “payment_value” column in the payments table to get the cost of orders.

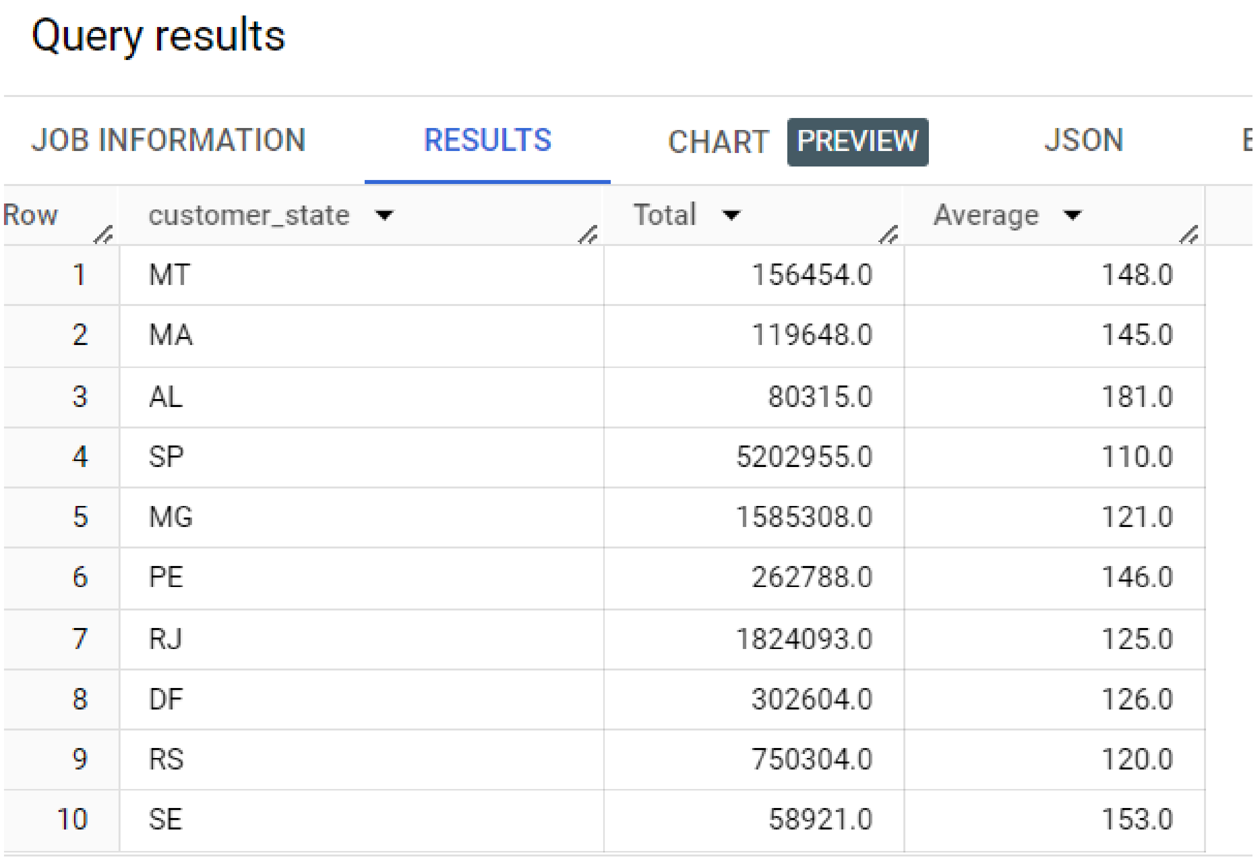
```
with final as(
Select extract(year from order_purchase_timestamp) as yr
,sum(payment_value) as total
from `store.orders` o
join `store.payments` p
on o.order_id=p.order_id
where order_purchase_timestamp between '2017-01-01' and '2017-08-31'
or order_purchase_timestamp between '2018-01-01' and '2018-08-31'
group by 1
)

Select *,100*(lead(total)over(order by yr) - total)/total as perc_yoy
from final
```



Row	yr	total	perc_yoy
1	2017	3645107.269999...	138.5298787105...
2	2018	8694669.949999...	null

B.Calculate the Total & Average value of order price for each state.

```
select customer_state,round(sum(price)) as Total,round(avg(price)) as Average
from `store.customers` c
inner join `store.orders` o
on c.customer_id=o.customer_id
inner join `store.order_items` ot
on o.order_id=ot.order_id
group by 1
```



Query results

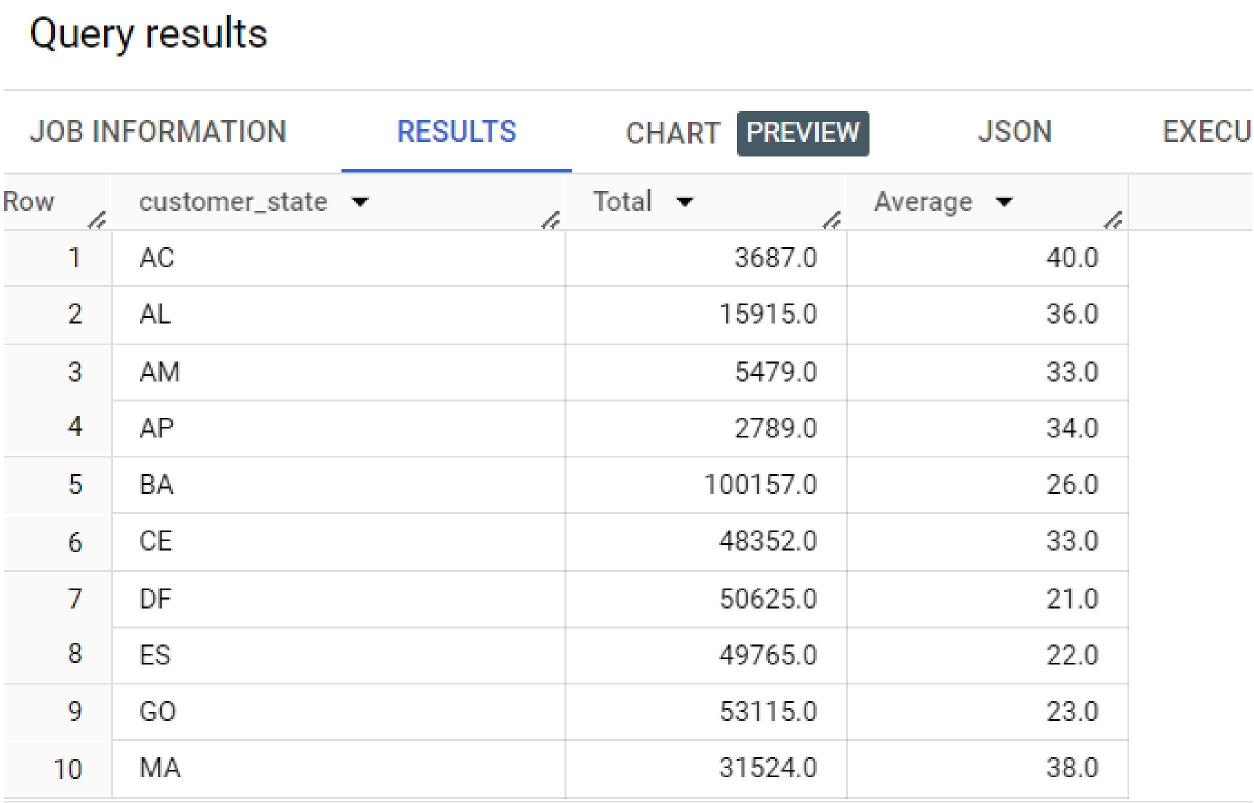
Row	customer_state	Total	Average
1	MT	156454.0	148.0
2	MA	119648.0	145.0
3	AL	80315.0	181.0
4	SP	5202955.0	110.0
5	MG	1585308.0	121.0
6	PE	262788.0	146.0
7	RJ	1824093.0	125.0
8	DF	302604.0	126.0
9	RS	750304.0	120.0
10	SE	58921.0	153.0

Insights average and total order price of each state

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

Ans

```
select customer_state,round(sum(freight_value)) as
Total,round(avg(freight_value)) as Average
from `store.customers` c
inner join `store.orders` o
on c.customer_id=o.customer_id
inner join `store.order_items` ot
on o.order_id=ot.order_id
group by 1
order by 1
```



Row	customer_state	Total	Average
1	AC	3687.0	40.0
2	AL	15915.0	36.0
3	AM	5479.0	33.0
4	AP	2789.0	34.0
5	BA	100157.0	26.0
6	CE	48352.0	33.0
7	DF	50625.0	21.0
8	ES	49765.0	22.0
9	GO	53115.0	23.0
10	MA	31524.0	38.0

```
Insights : average and total of freight value
```

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

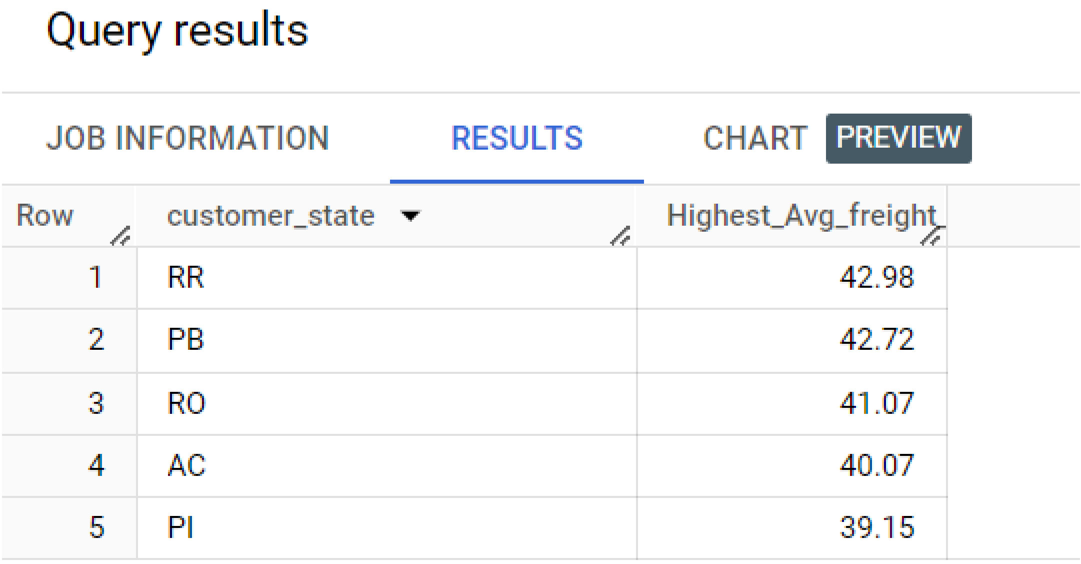
A .Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.

Also, calculate the difference (in days) between the estimated & actual delivery date of an order.

Do this in a single query.

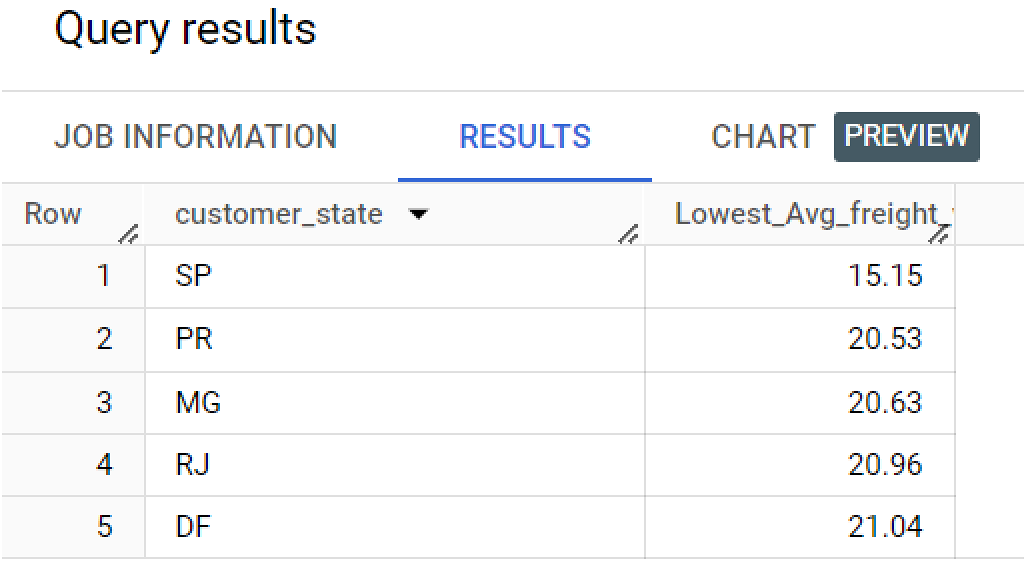
B.Find out the top 5 states with the highest & lowest average freight value

```
(SELECT customer_state, ROUND(AVG(freight_value),2) AS
Highest_Avg_freight_value
FROm `store.orders` AS o INNER JOIN `store.customers` AS c
ON o.customer_id = c.customer_id INNER JOIN `store.order_items` AS ord
ON o.order_id = ord.order_id
GROUP BY customer_state
ORDER BY Highest_Avg_freight_value desc
LIMIT 5)

(SELECT customer_state, ROUND(AVG(freight_value),2) AS Lowest_Avg_freight_value
FROM `store.orders` AS o INNER JOIN `store.customers` AS c
ON o.customer_id = c.customer_id INNER JOIN `store.order_items` AS ord
ON o.order_id = ord.order_id
GROUP BY customer_state
ORDER BY Lowest_Avg_freight_value asc
LIMIT 5)
```



Query results

Row	customer_state	Highest_Avg_freight_
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15



Query results

Row	customer_state	Lowest_Avg_freight_
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

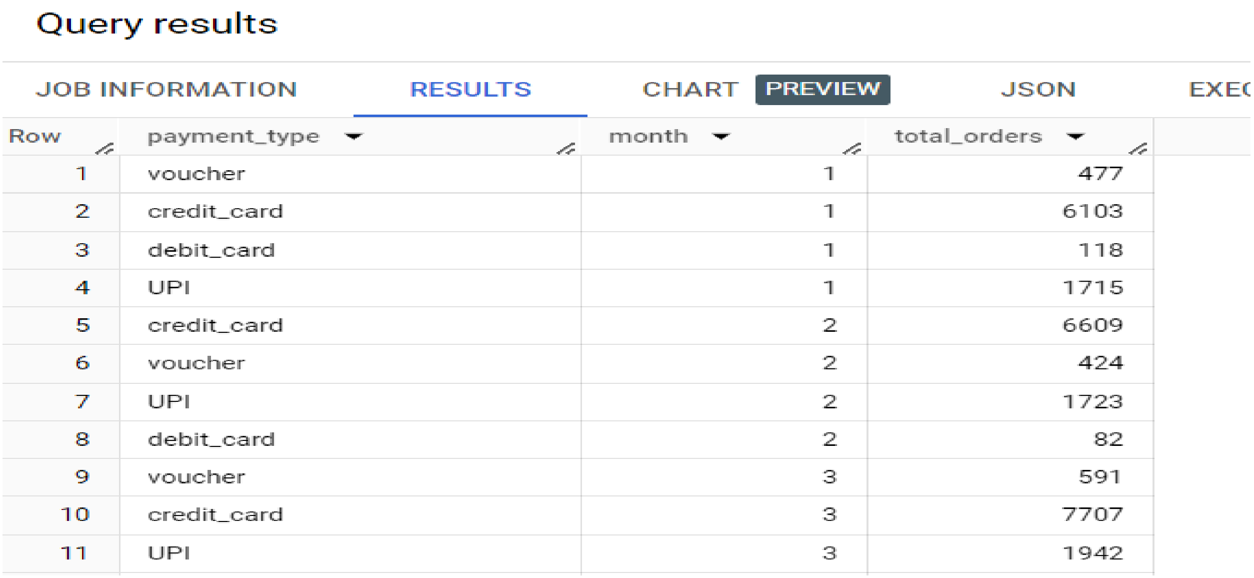
Insights : obtained average of freight of 5 highest and lowest states

6.Analysis based on the payments:

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

Ans

```
select payment_type,extract(month from order_purchase_timestamp) as
month,count(p.order_id) as total_orders
from `store.payments` p
join `store.orders` o
on o.order_id=p.order_id
group by 1,2
order by month
```

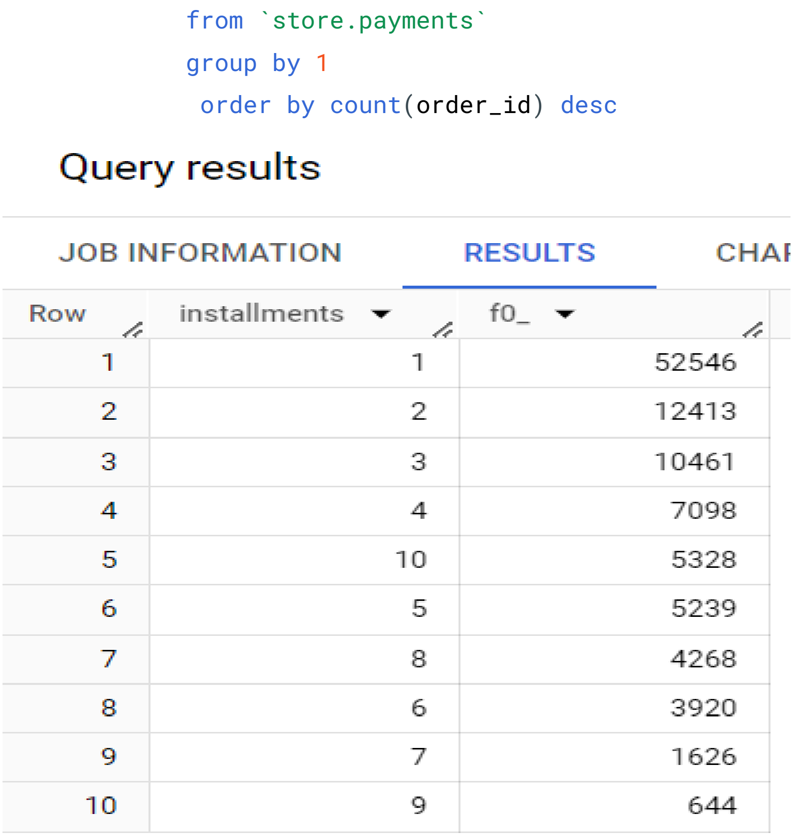


Row	payment_type	month	total_orders
1	voucher	1	477
2	credit_card	1	6103
3	debit_card	1	118
4	UPI	1	1715
5	credit_card	2	6609
6	voucher	2	424
7	UPI	2	1723
8	debit_card	2	82
9	voucher	3	591
10	credit_card	3	7707
11	UPI	3	1942

```
Insights : No of orders placed during months by payment type
```

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

ANS
```
select payment_installments as installments,
 count(order_id)
```



```
from `store.payments`
group by 1
 order by count(order_id) desc
```

Query results

Row	installments	f0_
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