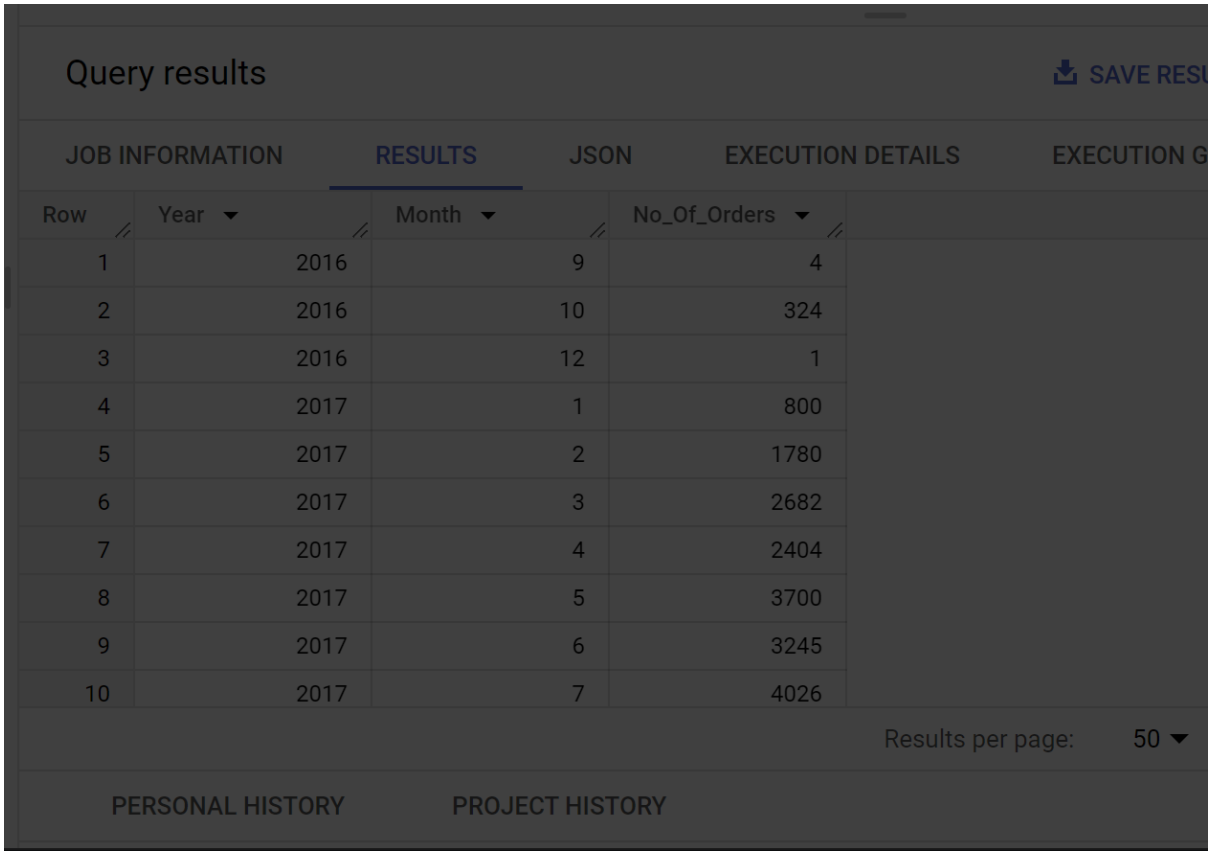
1. **In-depth Exploration:**

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

QUERY:
```
select *  from (select EXTRACT(year from order_purchase_timestamp) Year,
EXTRACT(month from order_purchase_timestamp) Month, count(order_id) No_Of_Orders

from `Scaler_DSML_Target_Project.orders`  group by EXTRACT(year from
order_purchase_timestamp), EXTRACT(month from order_purchase_timestamp)
) t order by Year , Month ASC;
```

Screenshot:



Query results

SAVE RESU

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION G

Row	Year	Month	No_Of_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

Results per page: 50

PERSONAL HISTORY | PROJECT HISTORY

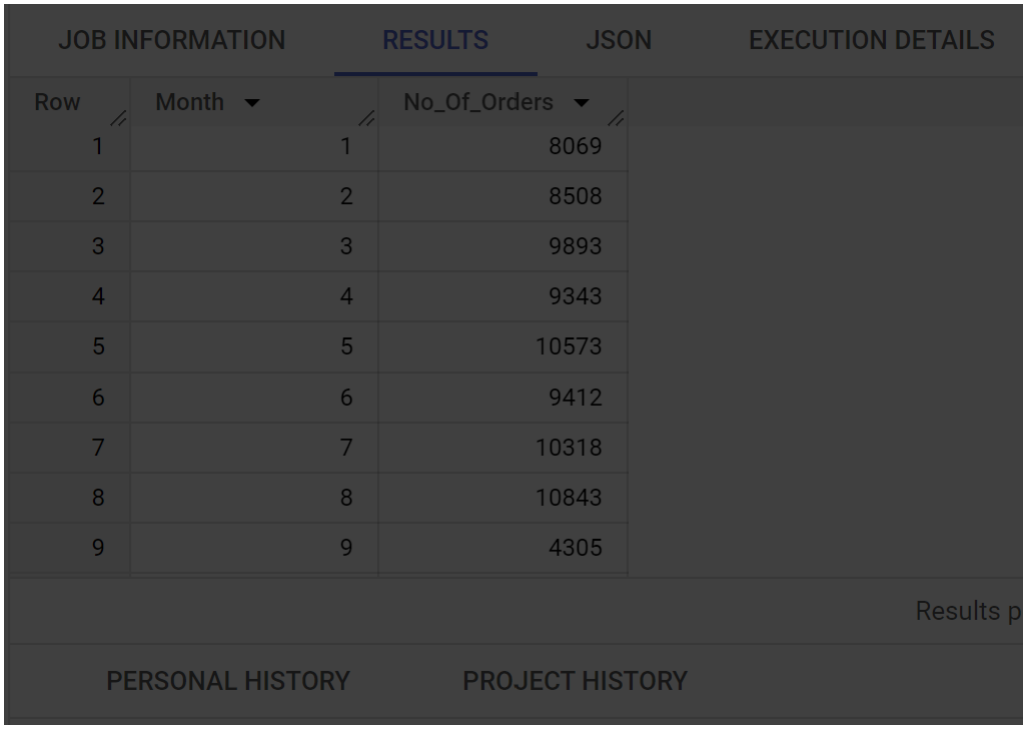
Inference: Looking out the data, it is cleared that there is gradual and steady growth from 2017 till mid of 2018. In 2016 there were less order(inconsistent) which might be because of start of business in brazil. Whereas in the 9th and 10th month data seems to be insufficient to infer anything. Decline towards the end of 2018 might be because of festive season in brazil.

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

Query:
```
select *  from (select --EXTRACT(year from order_purchase_timestamp) Year,

EXTRACT(month from order_purchase_timestamp) Month, count(order_id) No_Of_Orders
from `Scaler_DSML_Target_Project.orders`
group by EXTRACT(month from order_purchase_timestamp)
) t order by  Month ASC;
```

Screenshot:



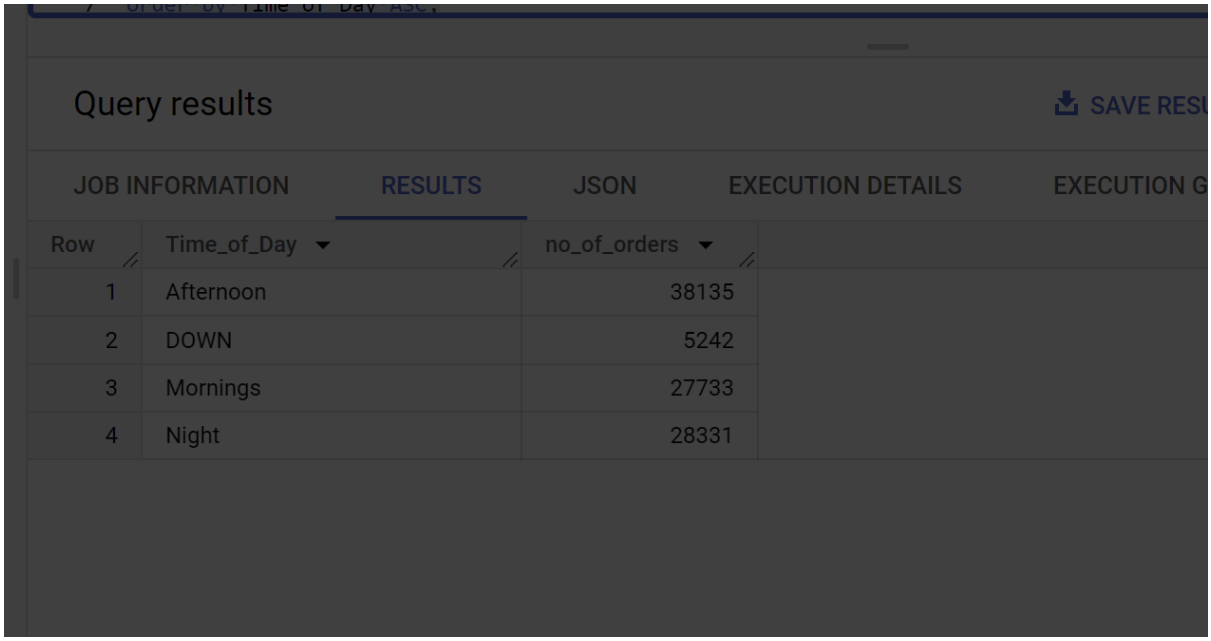
Inference: Tough there is no much difference on monthly basis but we see a significant increase in no of orders in the starting of winter season(in between may to august). Also we see a slight decrease in the spring and starting of summer.

3. 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

Query:
```
select Time_of_Day, count(Order_id) no_of_orders  from (select CASE WHEN
EXTRACT(Hour from order_purchase_timestamp) <=6 THEN 'DOWN'

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
Time_of_Day , order_id
from `Scaler_DSML_Target_Project.orders`
) t group by time_of_Day
order by Time_of_Day ASC;
```

ScreenShot:



Row	Time_of_Day	no_of_orders
1	Afternoon	38135
2	DOWN	5242
3	Mornings	27733
4	Night	28331

Inference: In the Afternoon time only Brazilians used to place most orders. But there are significant amount of order placed in Mornings as well as Night. And very low amount of orders placed in Down time.

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

Query:
```
select * from (select g.geolocation_state, EXTRACT(Month from o.order_purchase_timestamp) Month ,count(o.order_id) No_of_orders from `Scaler_DSML_Target_Project.orders` o join `Scaler_DSML_Target_Project.customers` c on o.customer_id = c.customer_id join `Scaler_DSML_Target_Project.geolocation` g on g.geolocation_zip_code_prefix = c.customer_zip_code_prefix group by g.geolocation_state , EXTRACT(Month from o.order_purchase_timestamp)

) t order by geolocation_state , Month
```

Screenshot:

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

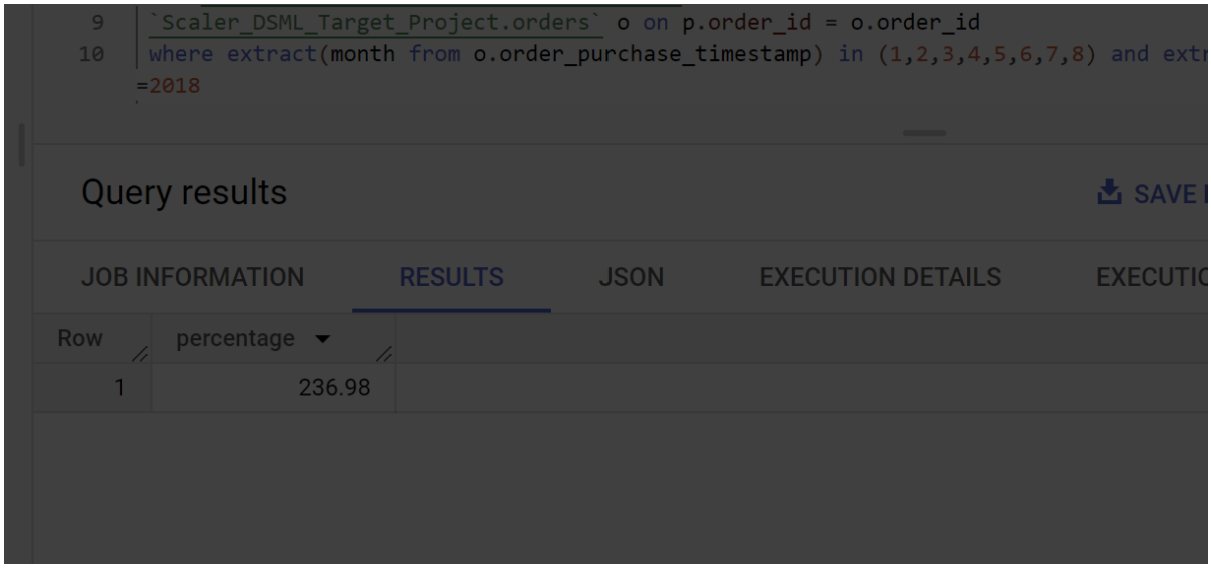
Inference: We can see that the customers resides in SP high shoppers followed by RJ and MG. Whereas It seems like people resides in RN, PI , AC , AL , AM , SE , TO , RR states orders very less.

3. **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).
       You can use the "payment_value" column in the payments table to get the cost of orders.

Query: `select  round((paymentsum2018/paymentsum2017) * 100,2) as percentage from (select`

```
(select sum(payment_value) paymentSum
from `Scaler_DSML_Target_Project.payments` p inner join
 `Scaler_DSML_Target_Project.orders` o on p.order_id = o.order_id
 where extract(month from o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8) and
extract(year from o.order_purchase_timestamp) = 2017
 group by EXTRACT(year from o.order_purchase_timestamp)) paymentsum2017 ,
 (select sum(payment_value) paymentSum
from `Scaler_DSML_Target_Project.payments` p inner join
 `Scaler_DSML_Target_Project.orders` o on p.order_id = o.order_id
 where extract(month from o.order_purchase_timestamp) in (1,2,3,4,5,6,7,8) and
extract(year from o.order_purchase_timestamp) =2018
 group by EXTRACT(year from o.order_purchase_timestamp)) paymentSum2018 ) t
```

Screenshot:



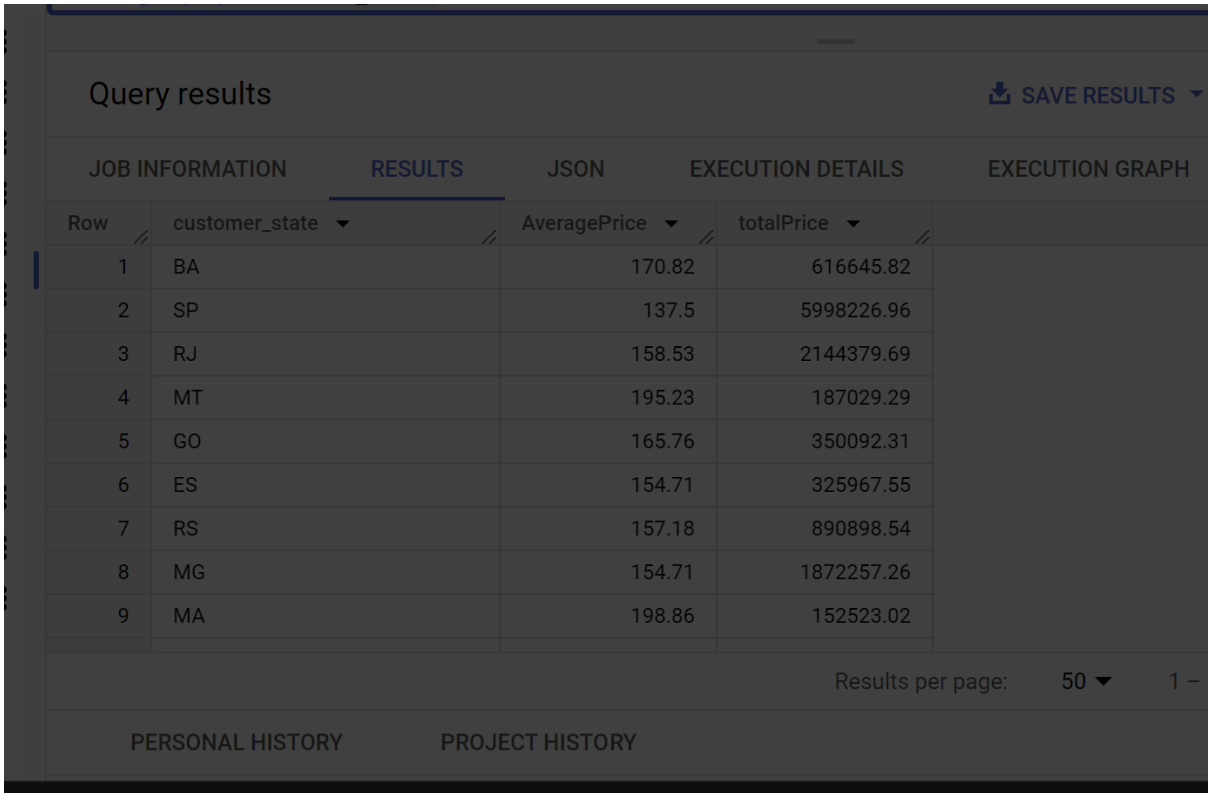
Analysis: We have seen a 2x increase in orders revenue in 2018 compared 2017.

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

Query:
```
select c.customer_state , Round(AVG(p.payment_value), 2) AveragePrice,
Round(Sum(p.payment_value), 2) totalPrice from
`Scaler_DSML_Target_Project.payments` p join `Scaler_DSML_Target_Project.orders` o
on o.order_id = p.order_id

left join `Scaler_DSML_Target_Project.customers` c on c.customer_id = o.customer_id
group by c.customer_state;
```

Screenshot:



Analysis: SP state is having the lower average price whereas PB state has the higher average price. For Total Price , SP leads among all whereas RR lags behind all.

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

Query:
```
select c.customer_state , Round(AVG(p.freight_value), 2) AveragePrice,
Round(Sum(p.freight_value), 2) totalPrice from
`Scaler_DSML_Target_Project.order_items` p join `Scaler_DSML_Target_Project.orders`
o on o.order_id = p.order_id

left join `Scaler_DSML_Target_Project.customers` c on c.customer_id = o.customer_id
group by c.customer_state;
```

Screenshot:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EX

Row	customer_state	AveragePrice	totalPrice
1	SP	15.15	718723.07
2	RJ	20.96	305589.31
3	PR	20.53	117851.68
4	SC	21.47	89660.26
5	DF	21.04	50625.5
6	MG	20.63	270853.46
7	PA	35.83	38699.3
8	BA	26.36	100156.68
9	GO	22.77	53114.98

Results per page:

PERSONAL HISTORY | PROJECT HISTORY

Analysis: The Average Freight price of RR is highest among all states whereas state SP is the lowest and vice versa in Total Freight price.

4. **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.

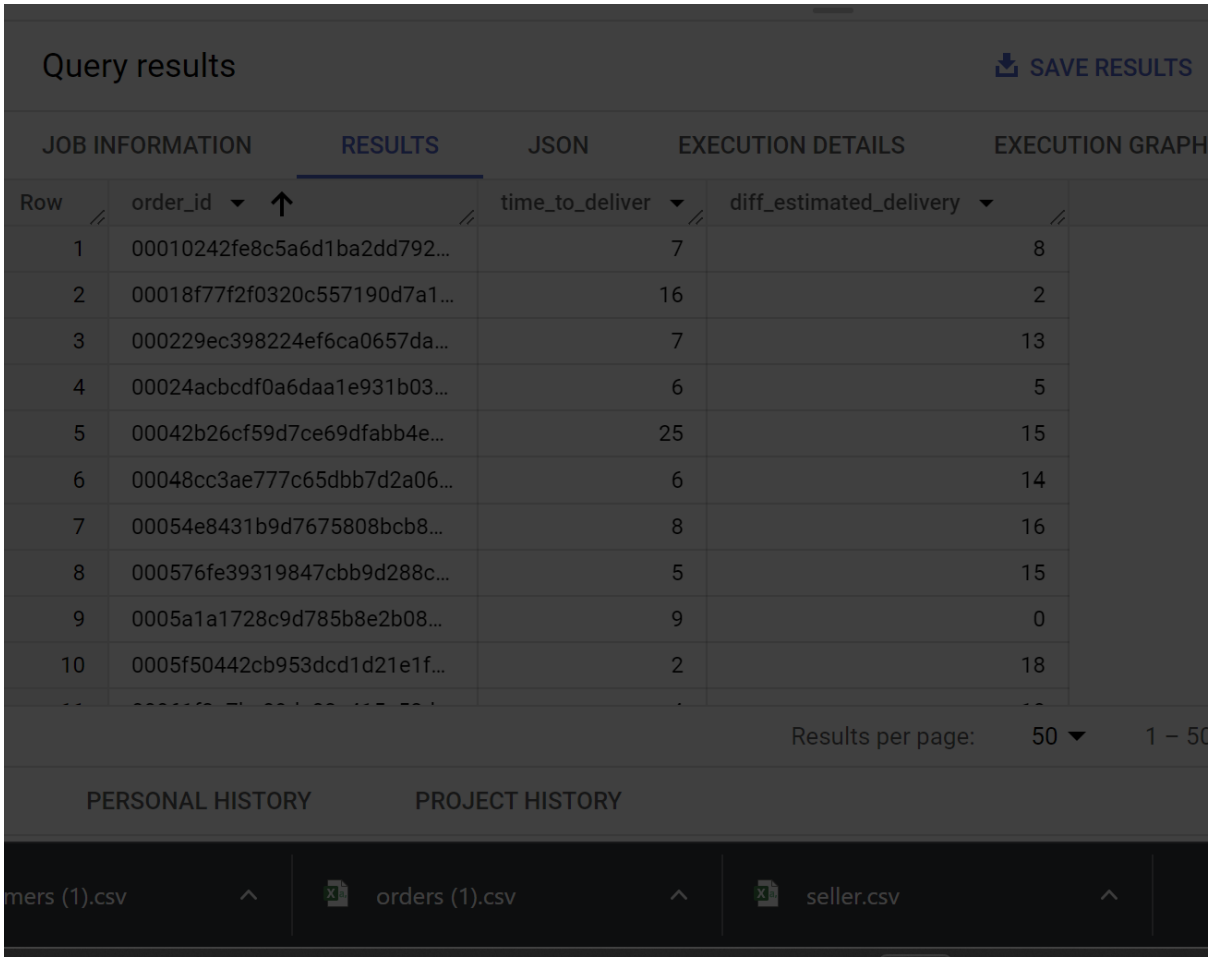
    You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

Query:
```
Select order_id, date_diff(order_delivered_customer_date, order_purchase_timestamp, DAY) time_to_deliver,

date_diff( order_estimated_delivery_date , order_delivered_customer_date , day) diff_estimated_delivery
from `Scaler_DSML_Target_Project.orders`
```

Screenshot:



Inference: Null values specifies that order is yet to delivered. Now coming to POV of analysis, it seems like most of the orders delivered are delivered late only(as we have max no of diff in estimated delivery is in negative(-)).

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

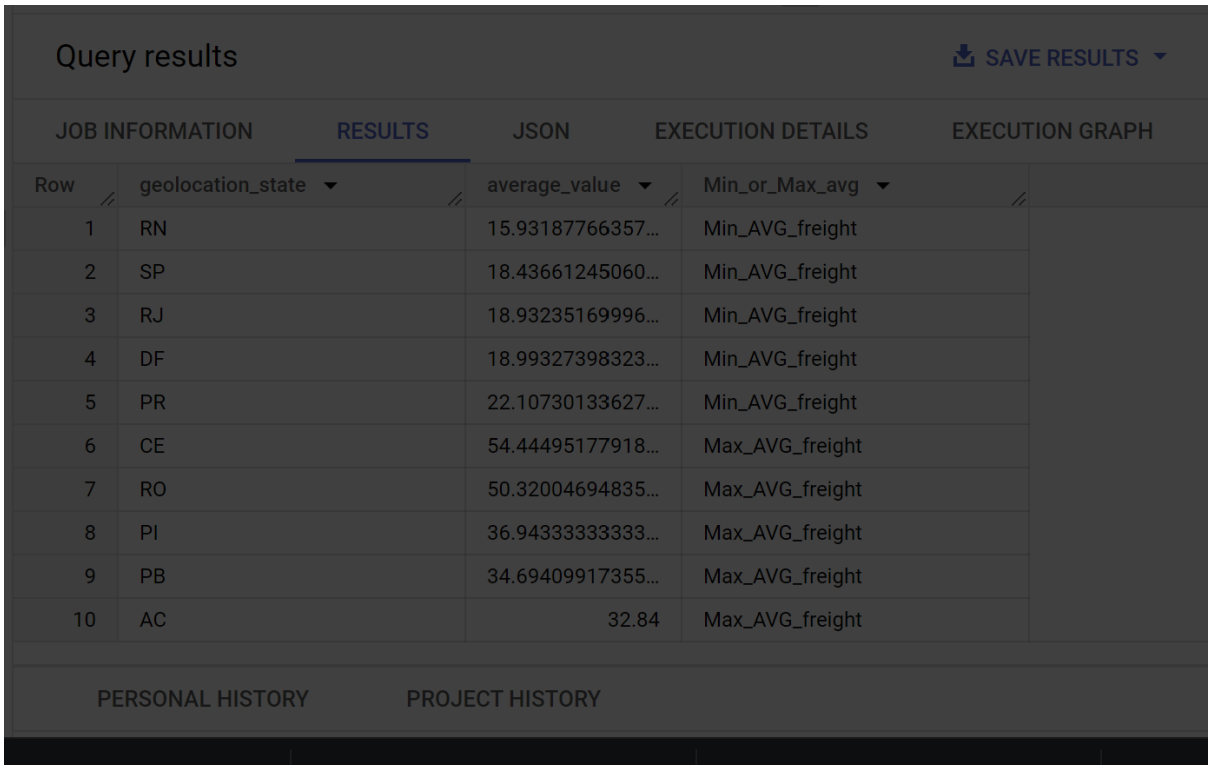
Query:
```
with avg_freight as

(Select g.geolocation_state,
AVG(freight_value) Avg_freight_value
From `Scaler_DSML_Target_Project.geolocation` g
left join `Scaler_DSML_Target_Project.sellers` s on s.seller_zip_code_prefix = g.geolocation_zip_code_prefix
left join `Scaler_DSML_Target_Project.order_items` oi on s.seller_id = oi.seller_id
```

```
where freight_value is not null
group by g.geolocation_state

)
Select * from (select geolocation_state, min(Avg_freight_value) average_value,
'Min_AVG_freight' Min_or_Max_avg from avg_freight group by geolocation_state order
by average_value asc limit 5 ) t1
union all
Select * from (select geolocation_state, max(Avg_freight_value) average_value,
'Max_AVG_freight' Min_or_Max_avg from avg_freight group by geolocation_state order
by average_value desc limit 5 ) t2
```

Screenshot:



Row	geolocation_state	average_value	Min_or_Max_avg
1	RN	15.93187766357...	Min_AVG_freight
2	SP	18.43661245060...	Min_AVG_freight
3	RJ	18.93235169996...	Min_AVG_freight
4	DF	18.99327398323...	Min_AVG_freight
5	PR	22.10730133627...	Min_AVG_freight
6	CE	54.44495177918...	Max_AVG_freight
7	RO	50.32004694835...	Max_AVG_freight
8	PI	36.94333333333...	Max_AVG_freight
9	PB	34.69409917355...	Max_AVG_freight
10	AC	32.84	Max_AVG_freight

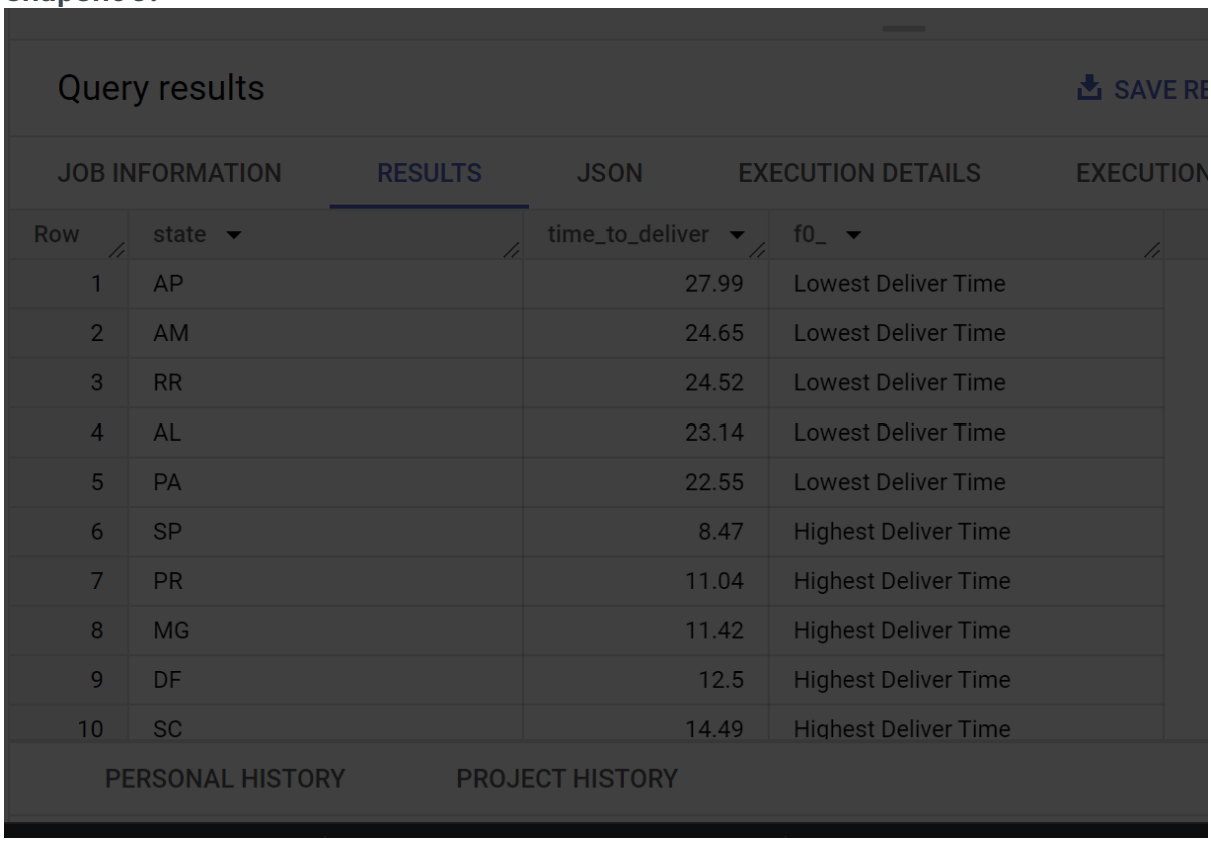
3. Find out the top 5 states with the highest & lowest average delivery time.

Query:

```
with lowest_deliverTime as
( Select g.geolocation_state state, AVG(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)) time_to_deliver
from `Scaler_DSML_Target_Project.geolocation` g left Join
`Scaler_DSML_Target_Project.customers` c on
c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
join  `Scaler_DSML_Target_Project.orders` o on o.customer_id = c.customer_id
group by g.geolocation_state
order by time_to_deliver desc limit 5
),
Highest_deliverTime as
( Select g.geolocation_state state, AVG(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)) time_to_deliver
```

```
from `Scaler_DSML_Target_Project.geolocation` g left Join
`Scaler_DSML_Target_Project.customers` c on
c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
join  `Scaler_DSML_Target_Project.orders` o on o.customer_id = c.customer_id
group by g.geolocation_state
order by time_to_deliver ASC limit 5
)
select lowest_deliverTime.state, round(lowest_deliverTime.time_to_deliver, 2)
time_to_deliver, 'Lowest Deliver Time' from lowest_deliverTime
union all
select Highest_deliverTime.state, round(Highest_deliverTime.time_to_deliver, 2)
time_to_deliver, 'Highest Deliver Time' from Highest_deliverTime;
```

Snapshot:



Row	state	time_to_deliver	f0_
1	AP	27.99	Lowest Deliver Time
2	AM	24.65	Lowest Deliver Time
3	RR	24.52	Lowest Deliver Time
4	AL	23.14	Lowest Deliver Time
5	PA	22.55	Lowest Deliver Time
6	SP	8.47	Highest Deliver Time
7	PR	11.04	Highest Deliver Time
8	MG	11.42	Highest Deliver Time
9	DF	12.5	Highest Deliver Time
10	SC	14.49	Highest Deliver Time

Analysis: AP state has the highest Average delivery time whereas SP has the lowest Average delivery time.

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
   You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

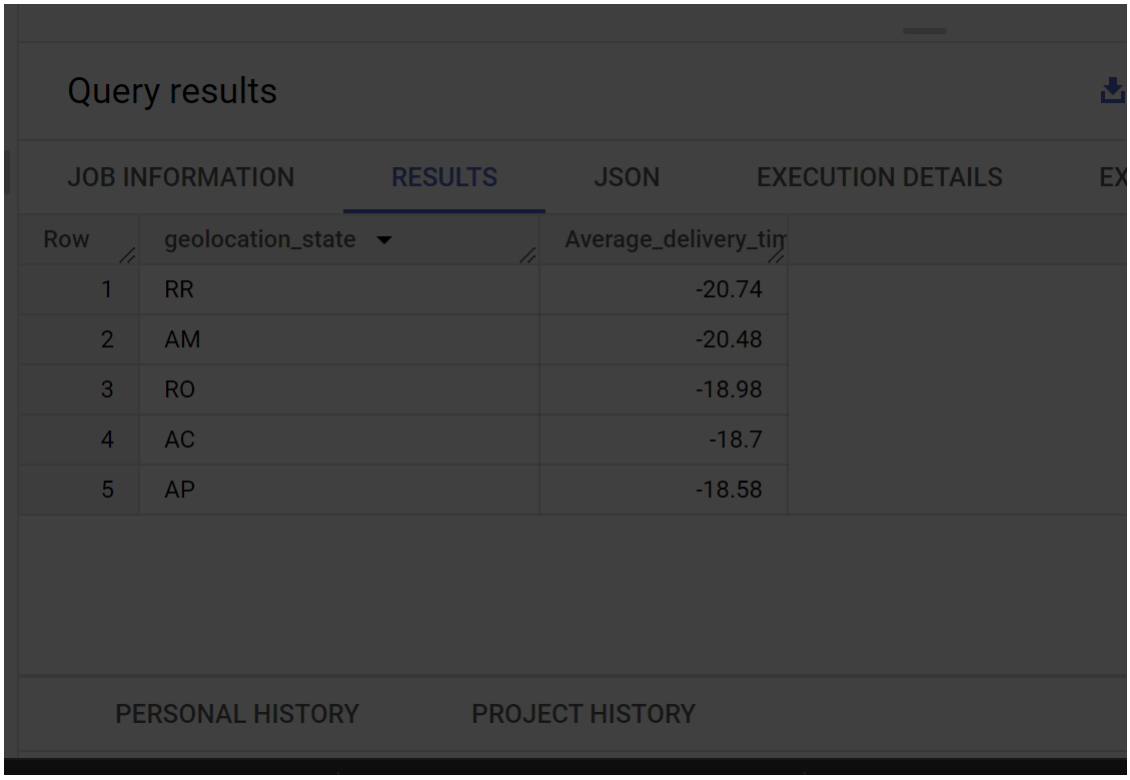
Query:

```
with delivery as

(Select g.geolocation_state  , DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY) Actual_Delivery_time ,
DATE_DIFF(o.order_estimated_delivery_date, o.order_purchase_timestamp, DAY)
Estimated_Delivery_time
```

```
from `Scaler_DSML_Target_Project.geolocation` g Left join
`Scaler_DSML_Target_Project.customers` c on g.geolocation_zip_code_prefix =
c.customer_zip_code_prefix
Join `Scaler_DSML_Target_Project.orders` o on o.customer_id = c.customer_id
)
select geolocation_state , Round(AVG(Actual_Delivery_time -
Estimated_Delivery_time),2) Average_delivery_time from delivery
group by geolocation_state
order by Average_delivery_time
limit 5;
```

Screenshot:



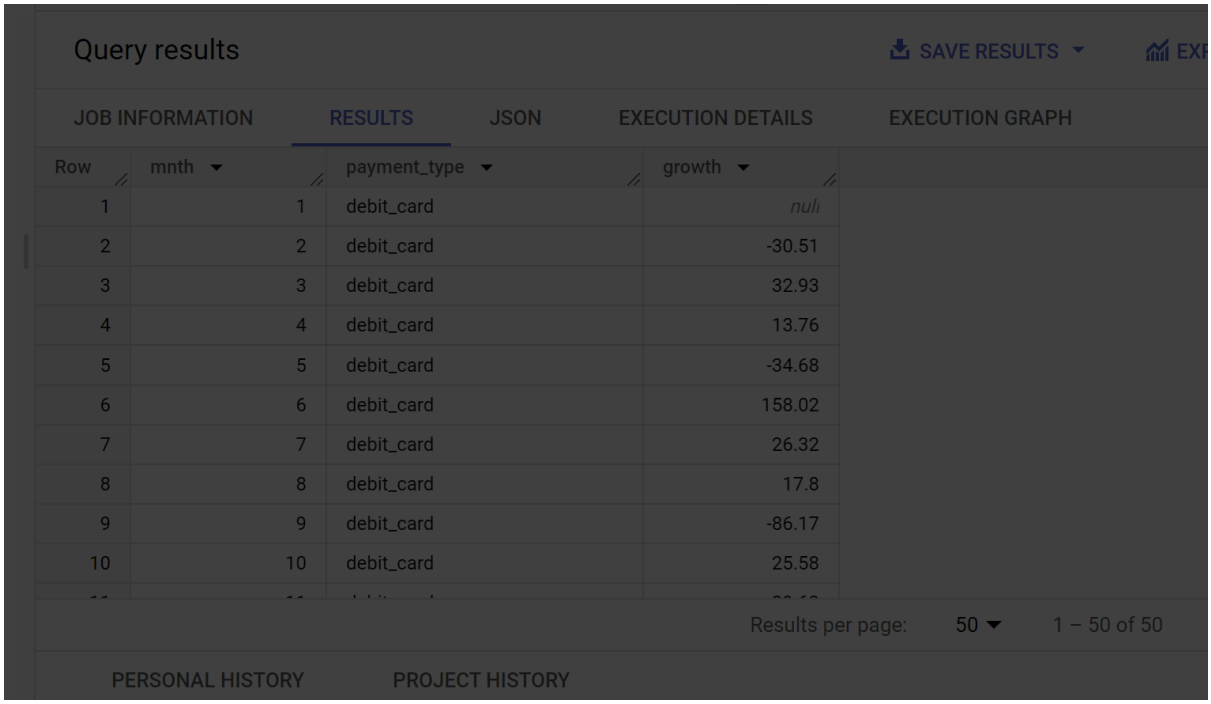
Analysis: RR has the faster average delivery time followed by AM, RO, AC, AP.

5. **Analysis based on the payments:**
    1. Find the month on month no. of orders placed using different payment types.

Query:

```
WITH PayType AS (
        SELECT o.order_id, Extract(month from o.order_purchase_timestamp) AS mnth ,
p.payment_type
          FROM `Scaler_DSML_Target_Project.orders` o join
`Scaler_DSML_Target_Project.payments` p
          on o.order_id = p.order_id
            )
SELECT mnth
     , payment_type
     , Round(100 * (COUNT(*) - LAG(COUNT(*)) OVER (PARTITION BY payment_type ORDER
BY mnth)) / LAG(COUNT(*)) OVER (PARTITION BY payment_type ORDER BY mnth),2) AS
growth_of_orders
  FROM PayType
 GROUP BY Payment_type, mnth;
```

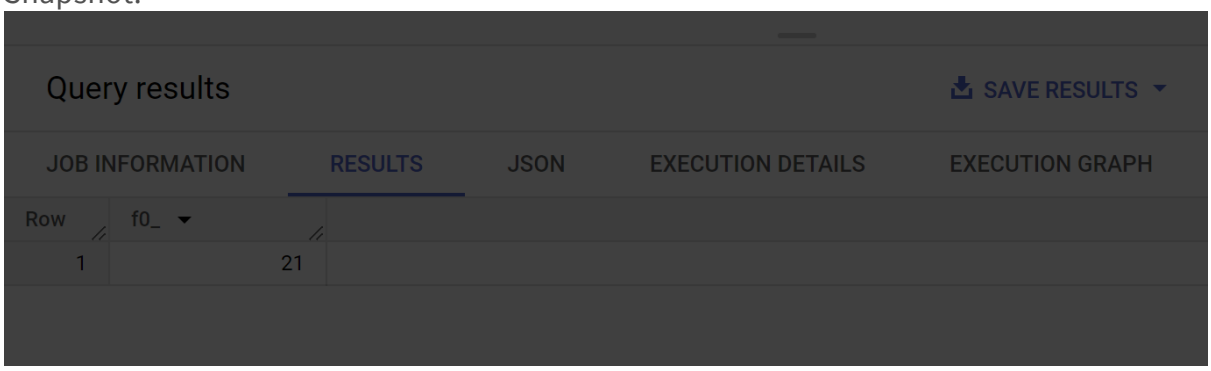
Screenshot:



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

Query:
```
Select count(Distinct ot.order_item_id)  from
`Scaler_DSML_Target_Project.payments` p
join `Scaler_DSML_Target_Project.order_items` ot on p.order_id = p.order_id
where p.payment_installments = 0
```

Snapshot:



Analysis: There are 21 orders which are purchased on instalments but have got paid.