1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
    1. Data type of all columns in the "customers" table.

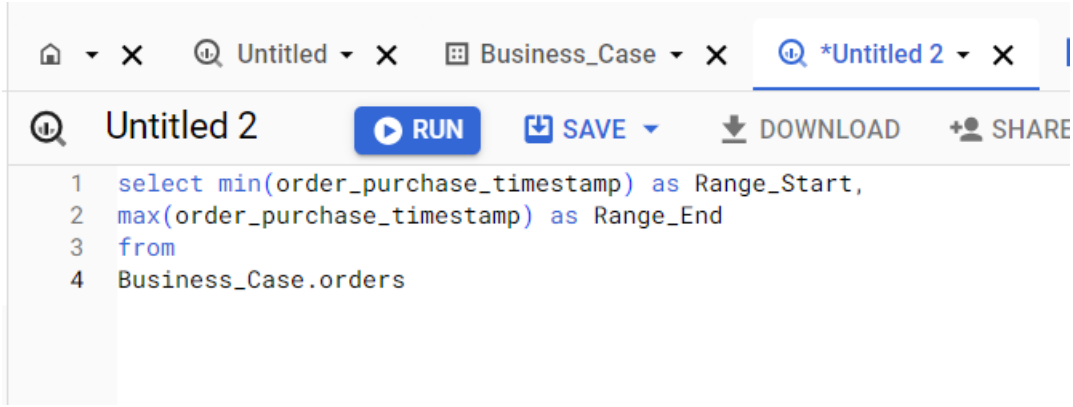
```
SELECT column_name, data_type
FROM peaceful-tome-406306.Business_Case.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers'
```

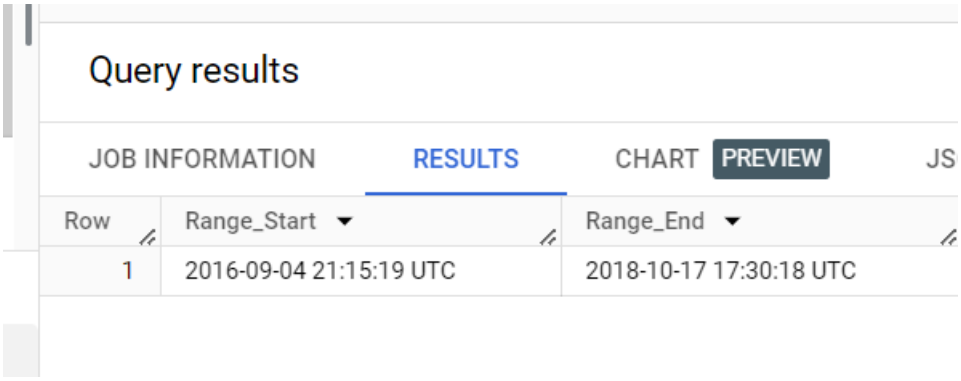
Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

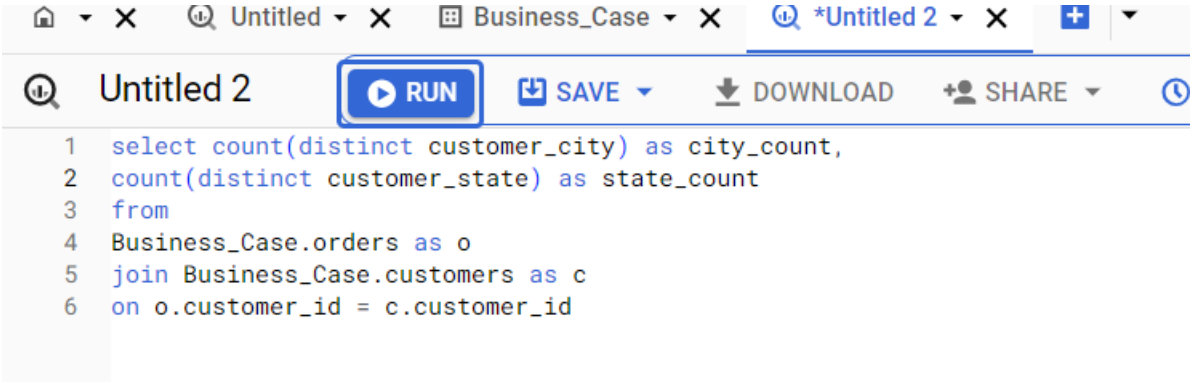
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

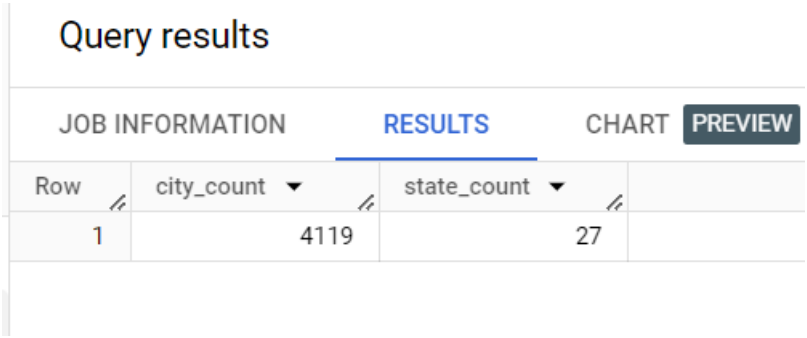
2. Get the time range between which the orders were placed.





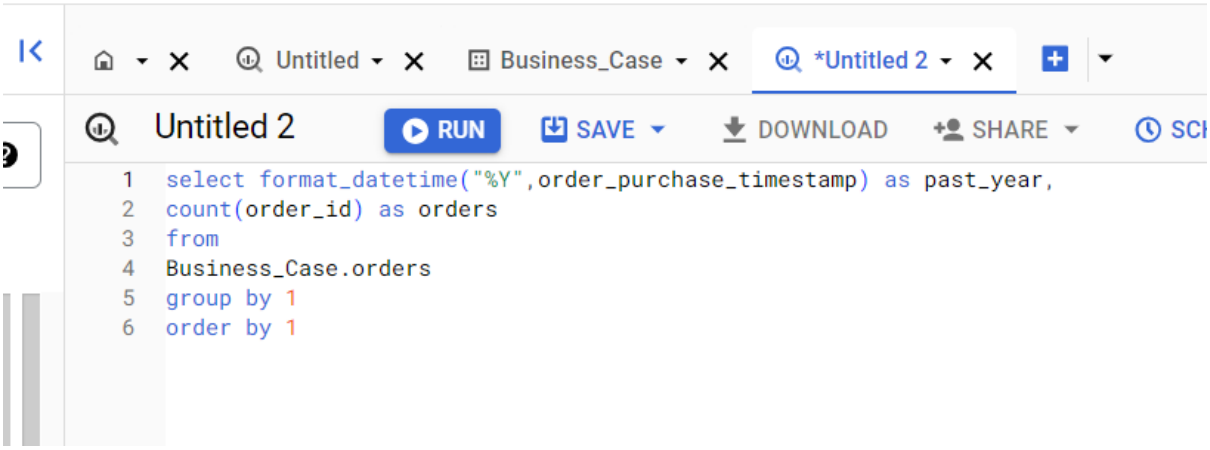
3. Count the Cities & States of customers who ordered during the given period.





2. **In-depth Exploration:**

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



Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	past_year	orders
1	2016	329
2	2017	45101
3	2018	54011

- Yes, there is growing trend in past year.

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

Untitled | Business_Case | *Untitled 2

Untitled 2 — RUN | SAVE | DOWNLOAD | SHARE

```
1  select format_datetime("%Y-%m",order_purchase_timestamp) as monthly,
2  count(order_id) as orders
3  from
4  Business_Case.orders
5  group by 1
6  order by 1
```

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	monthly	orders
1	2016-09	4
2	2016-10	324
3	2016-12	1
4	2017-01	800
5	2017-02	1780
6	2017-03	2682
7	2017-04	2404
8	2017-05	3700
9	2017-06	3245
10	2017-07	4026
11	2017-08	4331
12	2017-09	4285
13	2017-10	4631

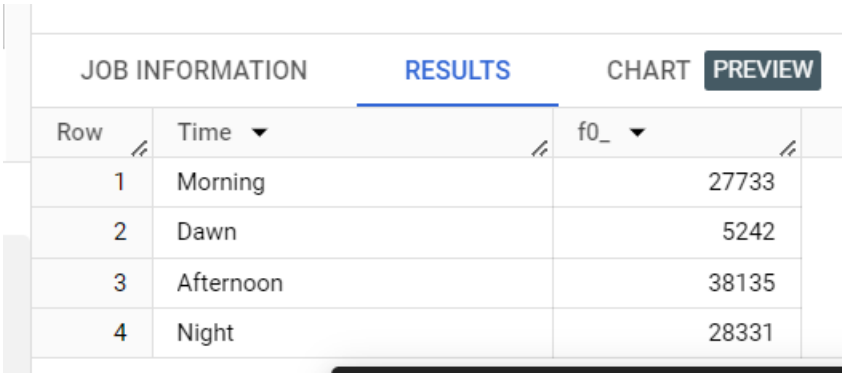
➢ Yes, In July, August, September & October orders are increased.

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

Untitled | Business_Case | *Untitled 2

Untitled 2 — RUN | SAVE | DOWNLOAD | SHARE | SCHEDULE

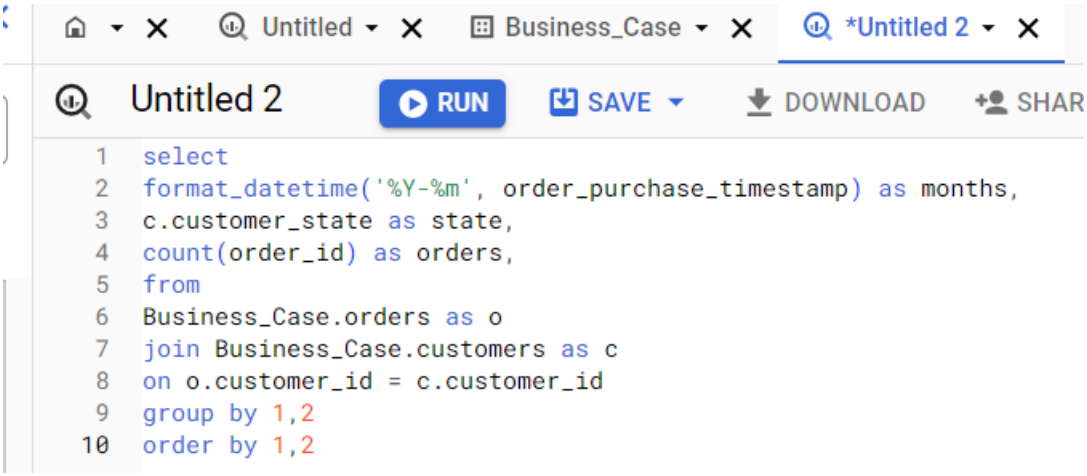
```
1  select
2  case
3  when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
4  when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Morning'
5  when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
6  else 'Night'
7  end as Time,
8  count(order_id)
9  from
10 Business_Case.orders
11 group by 1
```



Row	Time	f0_
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

- During Afternoon, Brazillian placed order mostly.

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



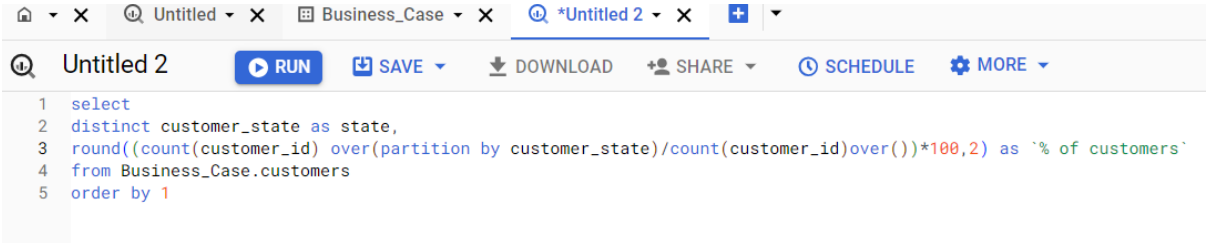

```sql
select
format_datetime('%Y-%m', order_purchase_timestamp) as months,
c.customer_state as state,
count(order_id) as orders,
from
Business_Case.orders as o
join Business_Case.customers as c
on o.customer_id = c.customer_id
group by 1,2
order by 1,2
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DE

Row	months	state	orders
1	2016-09	RR	1
2	2016-09	RS	1
3	2016-09	SP	2
4	2016-10	AL	2
5	2016-10	BA	4
6	2016-10	CE	8
7	2016-10	DF	6
8	2016-10	ES	4
9	2016-10	GO	9
10	2016-10	MA	4
11	2016-10	MG	40
12	2016-10	MT	3
13	2016-10	PA	4

Load more

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



Query results

JOB INFORMATION | RESULTS | CHART PREVIEW

Row	state	% of customers
1	AC	0.08
2	AL	0.42
3	AM	0.15
4	AP	0.07
5	BA	3.4
6	CE	1.34
7	DF	2.15
8	ES	2.04
9	GO	2.03
10	MA	0.75
11	MG	11.7
12	MS	0.72
13	MT	0.91

Load more

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.

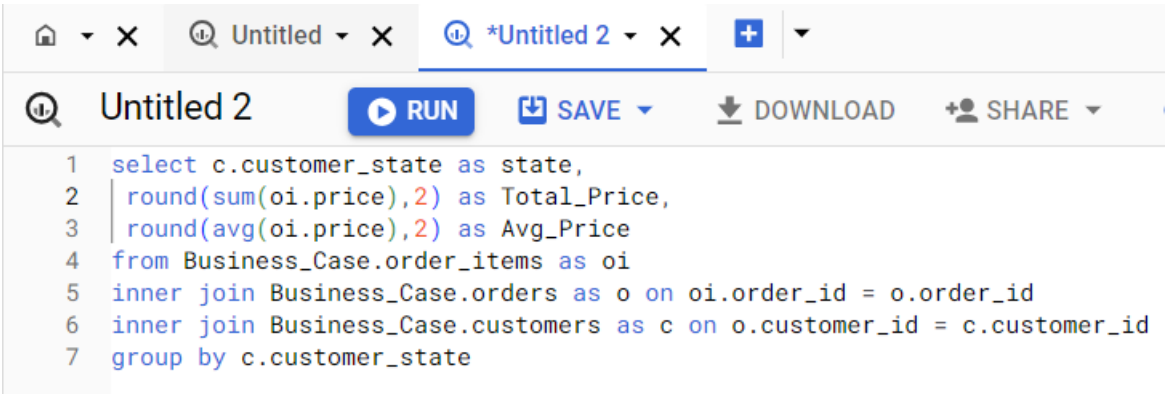
```
With CTE as (

select Extract(YEAR from o.order_purchase_timestamp) as Year
,sum(payment_value) as payment
from Business_Case.payments as p
inner join Business_Case.orders as o on p.order_id = o.order_id
where Extract(MONTH from o.order_purchase_timestamp) between 1 and 8
group by 1),
CTE2 as (
select Year, round(100*((payment/lead(payment) over (order by YEAR desc))-1),2) as
increase
from CTE
order by Year Desc)
Select * from CTE2 where Year = 2018
```

JOB INFORMATION | RESULTS | CHART

Row	Year	increase
1	2018	136.98

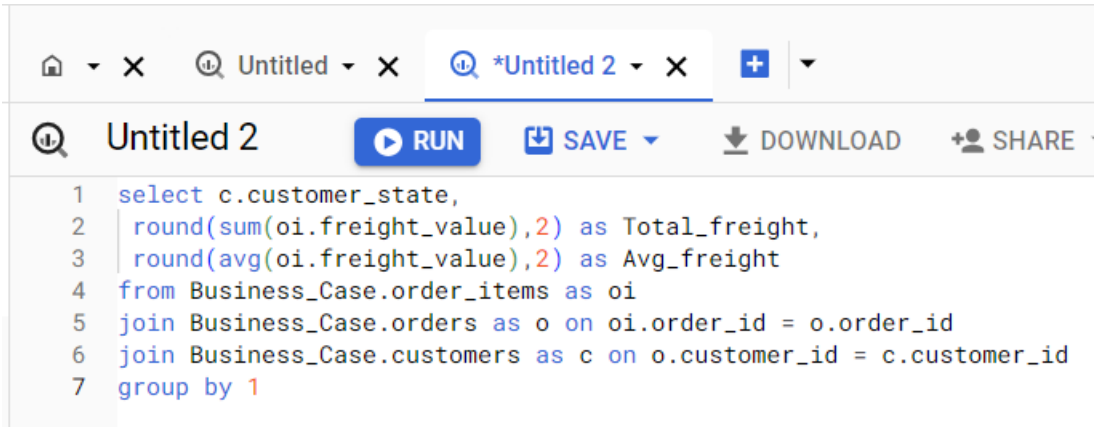
2. Calculate the Total & Average value of order price for each state.



JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

Row	state	Total_Price	Avg_Price
1	MT	156453.53	148.3
2	MA	119648.22	145.2
3	AL	80314.81	180.89
4	SP	5202955.05	109.65
5	MG	1585308.03	120.75
6	PE	262788.03	145.51
7	RJ	1824092.67	125.12
8	DF	302603.94	125.77
9	RS	750304.02	120.34
10	SE	58920.85	153.04

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



Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EX

Row	customer_state	Total_freight	Avg_freight
1	MT	29715.43	28.17
2	MA	31523.77	38.26
3	AL	15914.59	35.84
4	SP	718723.07	15.15
5	MG	270853.46	20.63
6	PE	59449.66	32.92
7	RJ	305589.31	20.96
8	DF	50625.5	21.04
9	RS	135522.74	21.74
10	SE	14111.47	36.65

5. 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_delivered_customer_date - order_estimated_delivery_date

Untitled | *Untitled 2

Untitled 2 RUN SAVE DOWNLOAD SHARE SCHE

```
1  select order_id,
2  date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
3  time_for_delivery,
4  date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as
5  diff_estimated_delivery
6  from Business_Case.orders
7  where order_status = "delivered"
```

```
select order_id,
```

```
date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
time_for_delivery,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as
diff_estimated_delivery
from Business_Case.orders
where order_status = "delivered"
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXE

Row	order_id	time_for_delivery	diff_estimated_delive
1	635c894d068ac37e6e03dc54e...	30	1
2	3b97562c3aee8bdedcb5c2e45...	32	0
3	68f47f50f04c4cb6774570cfde...	29	1
4	276e9ec344d3bf029ff83a161c...	43	-4
5	54e1a3c2b97fb0809da548a59...	40	-4
6	fd04fa4105ee8045f6a0139ca5...	37	-1
7	302bb8109d097a9fc6e9cefc5...	33	-5
8	66057d37308e787052a32828...	38	-6
9	19135c945c554eebfd7576c73...	36	-2
10	4493e45e7ca1084efcd38ddeb...	34	0

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

```
With Avg_frieght as (
select c.customer_state,
avg(oi.freight_value) as Avg_freight
from Business_Case.order_items as oi
inner join Business_Case.orders as o on oi.order_id = o.order_id
inner join Business_Case.customers as c on o.customer_id = c.customer_id
group by c.customer_state),
Frieght_Rank as (
select customer_state,
dense_rank() over (order by Avg_freight) as lh_rank,
dense_rank() over (order by Avg_freight desc) as hl_rank
from Avg_frieght)
select customer_state, "Bottom 5 States" as top_bottom
from Frieght_Rank where lh_rank <= 5
union all
select customer_state, "Top 5 States" as top_bottom
from Frieght_Rank where hl_rank <= 5
```

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON

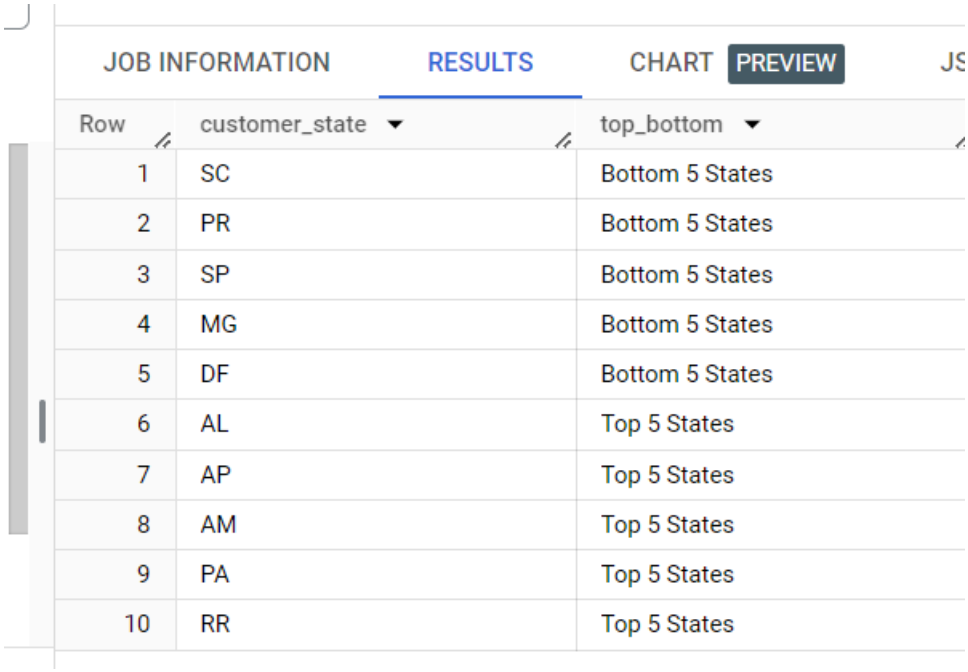
Row	customer_state	top_bottom
1	AC	Top 5 States
2	RO	Top 5 States
3	PB	Top 5 States
4	PI	Top 5 States
5	RR	Top 5 States
6	RJ	Bottom 5 States
7	MG	Bottom 5 States
8	PR	Bottom 5 States
9	DF	Bottom 5 States
10	SP	Bottom 5 States

3. Find out the top 5 states with the highest & 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

```
with del_time as (
select c.customer_id,
c.customer_state,
o.order_id,
date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day) as
delivery_time
from Business_Case.orders as o
inner join Business_Case.customers as c on o.customer_id = c.customer_id
where order_status = "delivered"),
avg_del_time as (
select customer_state
,round(avg(delivery_time),2) as avg_time
from del_time
group by customer_state),
rank_del_time as (
select customer_state,
dense_rank() over(order by avg_time) as lh_del_time,
dense_rank() over(order by avg_time desc) as hl_del_time
from avg_del_time)
select customer_state,
```

```
"Top 5 States" as top_bottom
from rank_del_time where hl_del_time<=5
union all
select customer_state,
"Bottom 5 States" as top_bottom
from rank_del_time where lh_del_time<=5
```



JOB INFORMATION | RESULTS | CHART PREVIEW | JS

Row	customer_state	top_bottom
1	SC	Bottom 5 States
2	PR	Bottom 5 States
3	SP	Bottom 5 States
4	MG	Bottom 5 States
5	DF	Bottom 5 States
6	AL	Top 5 States
7	AP	Top 5 States
8	AM	Top 5 States
9	PA	Top 5 States
10	RR	Top 5 States

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

```
select format_datetime("%Y-%m",order_purchase_timestamp) as Year_Month,
 c.payment_type as state,
 count(o.order_id) as order_count
from Business_Case.orders as o
inner join Business_Case.payments as c on o.order_id = c.order_id
group by 1,2
order by 1,2
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DE

Row	Year_Month	state	order_count
1	2016-09	credit_card	3
2	2016-10	UPI	63
3	2016-10	credit_card	254
4	2016-10	debit_card	2
5	2016-10	voucher	23
6	2016-12	credit_card	1
7	2017-01	UPI	197
8	2017-01	credit_card	583
9	2017-01	debit_card	9
10	2017-01	voucher	61

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

```
select count(distinct order_id)
from Business_Case.payments
where payment_installments > 1
```