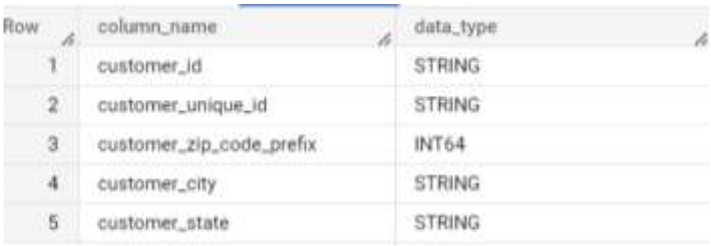
BUSINESS CASE :- Target SQL

1. **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 :-

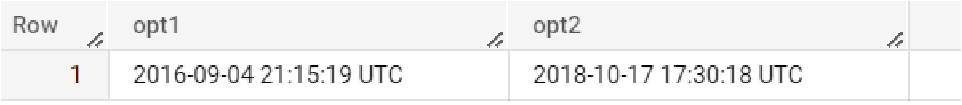
```
SELECT
  column_name, data_type
FROM
  model-bonsai-382608.retail_data.INFORMATION_SCHEMA.COLUMNS
  where table_name='customers'
```



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

2.Time period for which the data is given:-

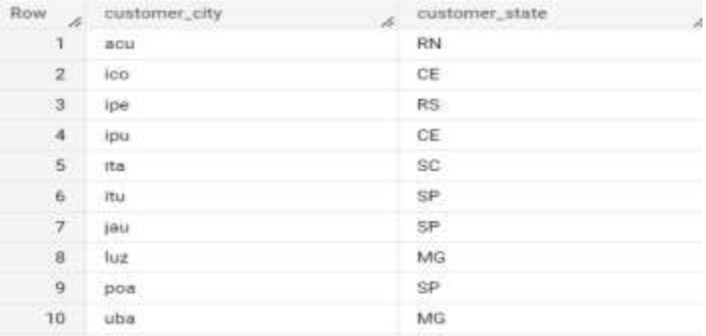
```
select min(order_purchase_timestamp) as opt1,

max(order_purchase_timestamp) as opt2 from `retail_data.orders`;
```



Row	opt1	opt2
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

3. Cities and States of customers ordered during the given period :-

```
select distinct customer_city,customer_state
from  `retail_data.customers`
```

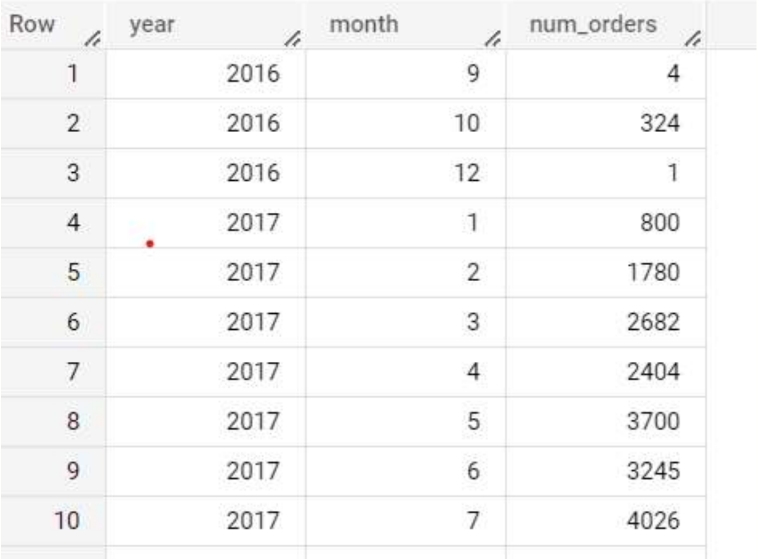


Row	customer_city	customer_state
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC
6	itu	SP
7	jau	SP
8	luz	MG
9	poa	SP
10	uba	MG

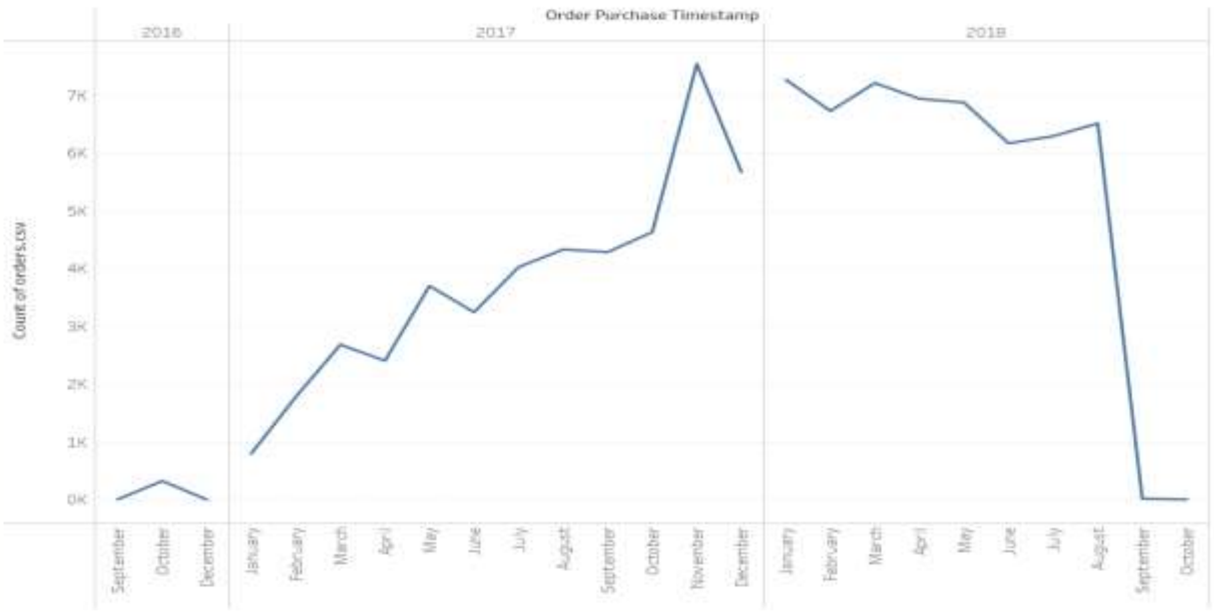
2. **In-depth Exploration**:

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?

```
SELECT
 EXTRACT (YEAR FROM order_purchase_timestamp) AS year,
 EXTRACT (MONTH FROM order_purchase_timestamp) AS month,
 COUNT(*) as num_orders
FROM `retail_data.orders`
GROUP BY year,month
ORDER BY year,month
```



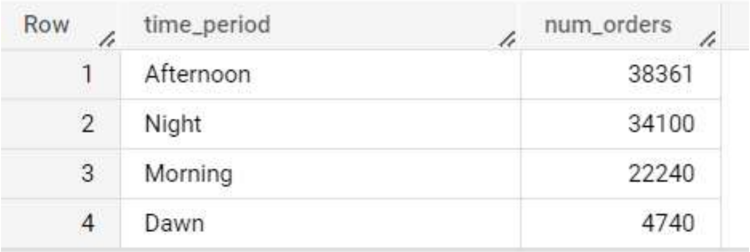
Row	year	month	num_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026



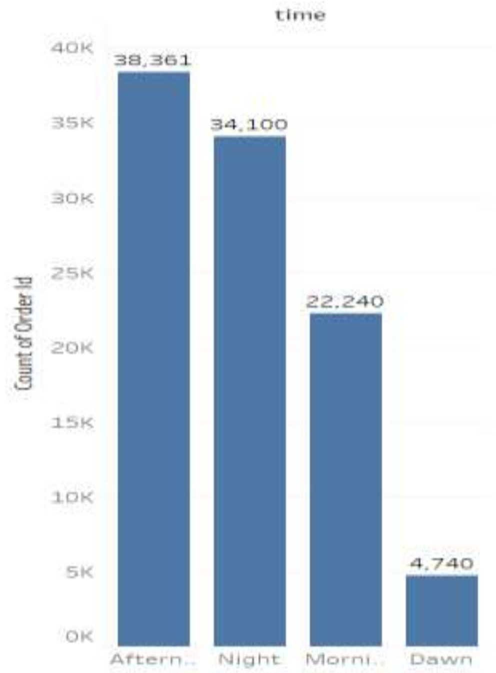
Yes, there is growing trend on e commerce in brazil.

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

```
SELECT

  CASE
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 0 AND EXTRACT(HOUR FROM
 order_purchase_timestamp) < 6 THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 6 AND EXTRACT(HOUR FROM
 order_purchase_timestamp) < 12 THEN 'Morning'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 12 AND EXTRACT(HOUR FRO
M order_purchase_timestamp) < 18 THEN 'Afternoon'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 18 AND EXTRACT(HOUR FRO
M order_purchase_timestamp) <= 23 THEN 'Night'
    ELSE 'Unknown'
         END AS time_period,
COUNT(*) AS num_orders
FROM `retail_data.orders`
GROUP BY time_period
ORDER BY num_orders DESC
```



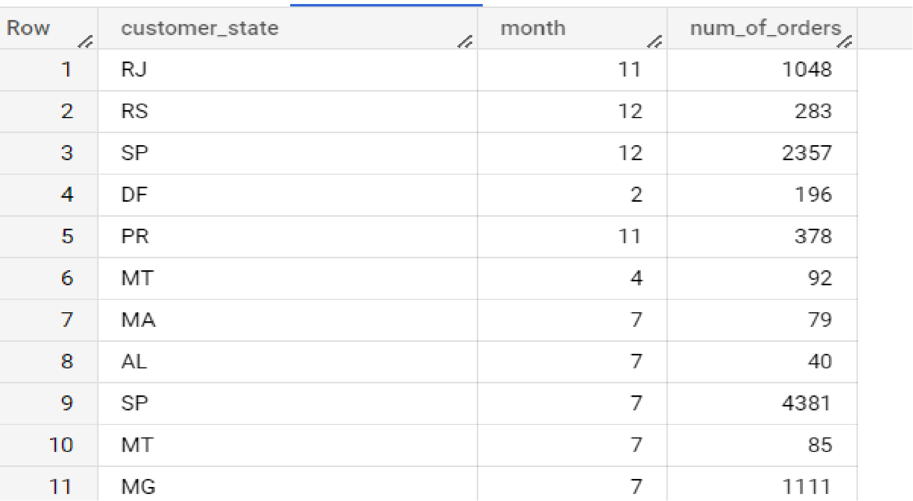
Row	time_period	num_orders
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740



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

1. Get month on month orders by states:-

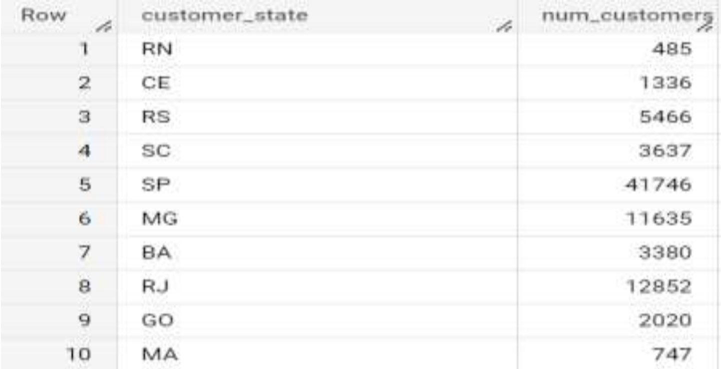
```
SELECT
    c.customer_state,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    COUNT(*) AS num_of_orders
FROM
    `retail_data.orders` AS o
    JOIN `retail_data.customers` AS c ON c.customer_id = o.customer_id
GROUP BY
    c.customer_state,
    month
```



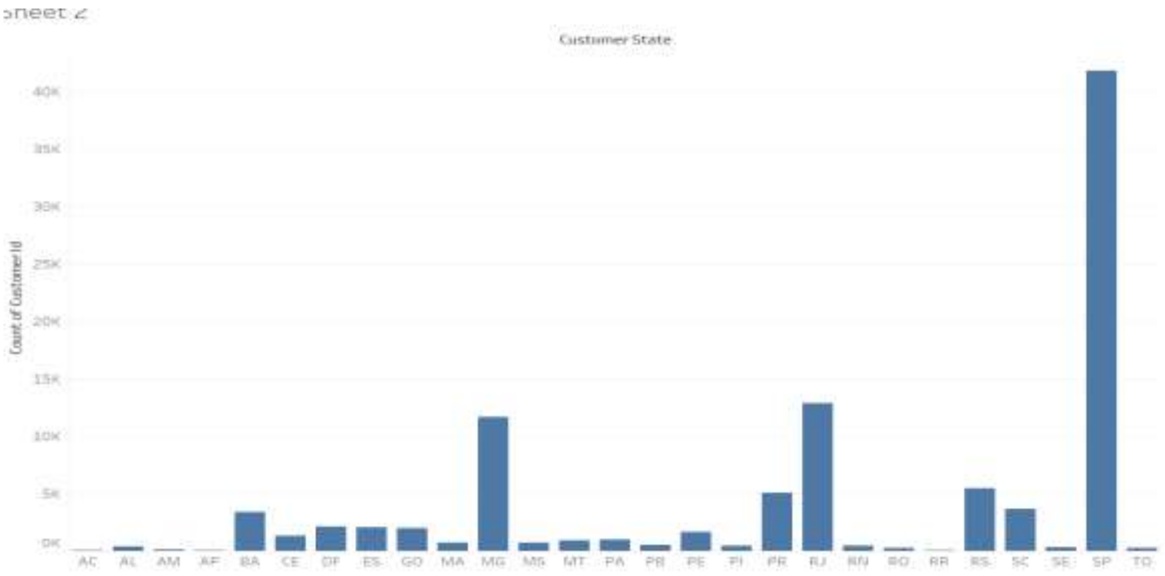
Row	customer_state	month	num_of_orders
1	RJ	11	1048
2	RS	12	283
3	SP	12	2357
4	DF	2	196
5	PR	11	378
6	MT	4	92
7	MA	7	79
8	AL	7	40
9	SP	7	4381
10	MT	7	85
11	MG	7	1111

2. Distribution of customers across the states in Brazil:-

```
select t1.customer_state,t1.num_customers,t2.num_orders
from

(select
customer_state,count(customer_id) as num_customers from `retail_data.cust
omers`
group by customer_state) as t1
join
(select c.customer_state,count(distinct o.order_id) as num_orders
from `retail_data.orders` o join `retail_data.customers` c on o.customer_
id=c.customer_id group by c.customer_state ) as t2

on t1.customer_state=t2.customer_state
```



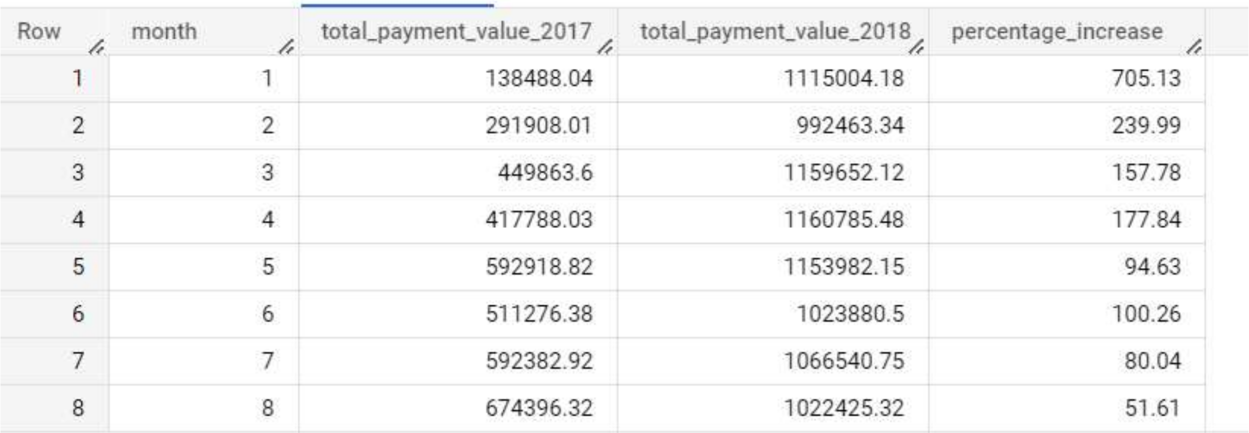
Row	customer_state	num_customers
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747



4.Impact on Economy: Analyze the money movement 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) - You can use "payment_value" column in payments table

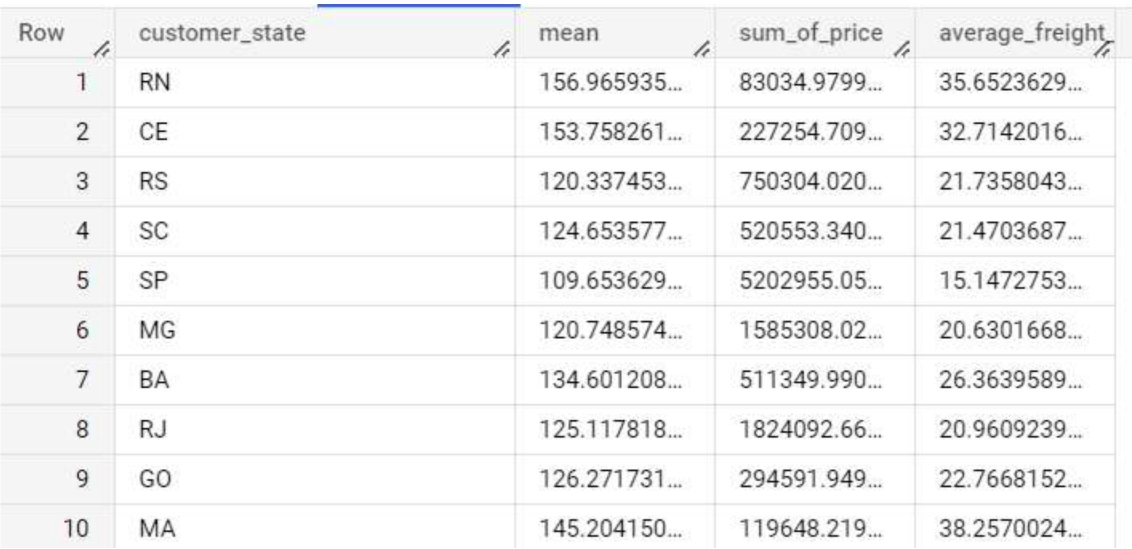
```
WITH orders_2017 AS (
  SELECT
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    round(SUM(p.payment_value),2) AS total_payment_value_2017
  FROM `retail_data.orders` o
  INNER JOIN `retail_data.payments` p ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
  GROUP BY EXTRACT(MONTH FROM o.order_purchase_timestamp)
),
orders_2018 AS (
  SELECT
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    round(SUM(p.payment_value),2) AS total_payment_value_2018
  FROM `retail_data.orders` o
  INNER JOIN `retail_data.payments` p ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018
    AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
  GROUP BY EXTRACT(MONTH FROM o.order_purchase_timestamp )
)
SELECT
  o1.month,
  o1.total_payment_value_2017,
  o2.total_payment_value_2018,
  round(((o2.total_payment_value_2018 - o1.total_payment_value_2017) / o1
.total_payment_value_2017) * 100,2) AS percentage_increase
FROM orders_2017 as o1
INNER JOIN orders_2018 as o2 ON o1.month = o2.month
order by o1.MONTH;
```



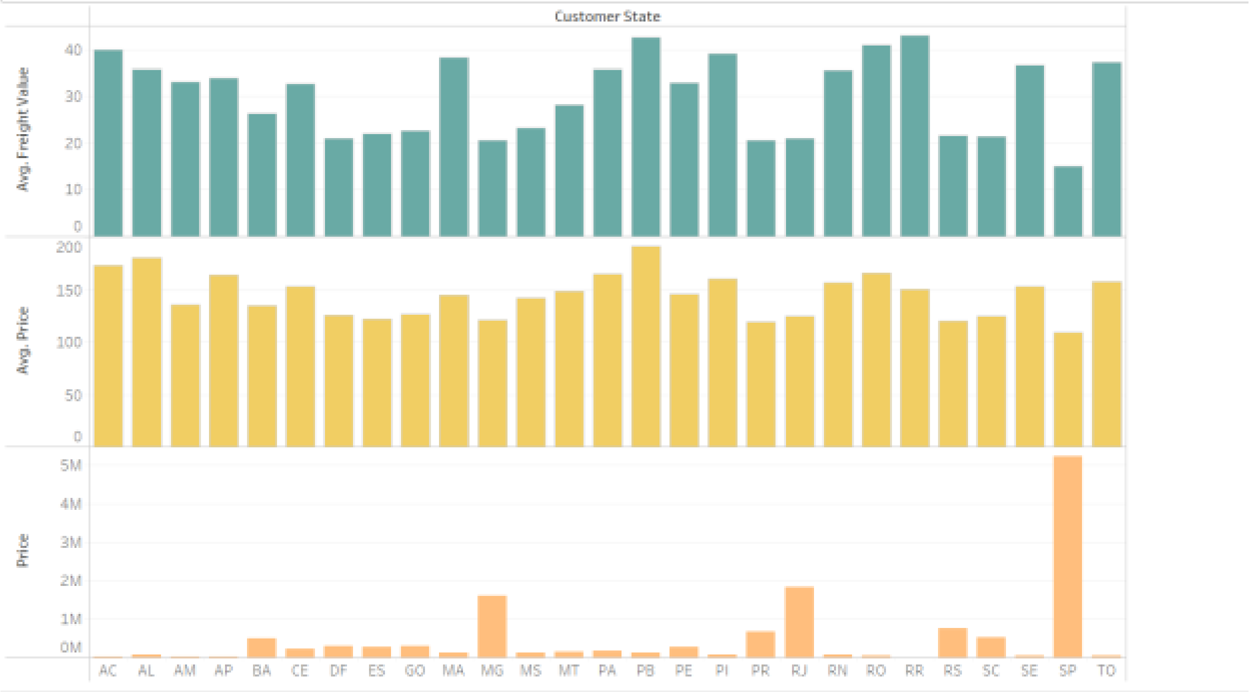
Row	month	total_payment_value_2017	total_payment_value_2018	percentage_increase
1	1	138488.04	1115004.18	705.13
2	2	291908.01	992463.34	239.99
3	3	449863.6	1159652.12	157.78
4	4	417788.03	1160785.48	177.84
5	5	592918.82	1153982.15	94.63
6	6	511276.38	1023880.5	100.26
7	7	592382.92	1066540.75	80.04
8	8	674396.32	1022425.32	51.61

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

```
select
c.customer_state,
avg(oi.price) as mean,
sum(oi.price) as sum_of_price,
avg(oi.freight_value) as average_freight_value
from `retail_data.customers`as c  join `retail_data.orders`
as o on o.customer_id = c.customer_id
join `retail_data.order_items` as oi on oi.order_id=o.order_id
group by c.customer_state
```



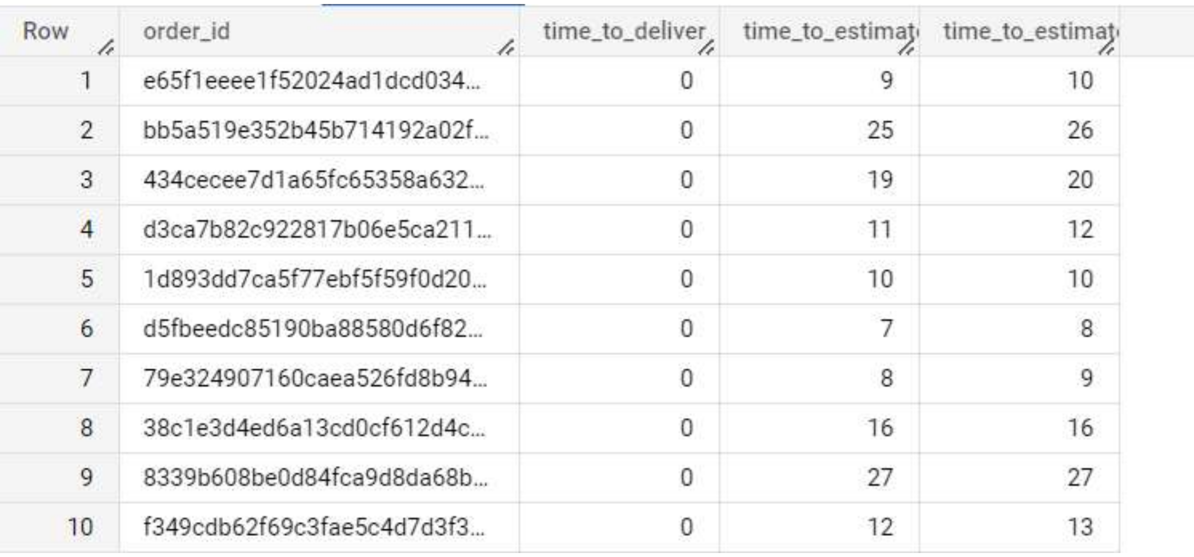
Row	customer_state	mean	sum_of_price	average_freight_
1	RN	156.965935...	83034.9799...	35.6523629...
2	CE	153.758261...	227254.709...	32.7142016...
3	RS	120.337453...	750304.020...	21.7358043...
4	SC	124.653577...	520553.340...	21.4703687...
5	SP	109.653629...	5202955.05...	15.1472753...
6	MG	120.748574...	1585308.02...	20.6301668...
7	BA	134.601208...	511349.990...	26.3639589...
8	RJ	125.117818...	1824092.66...	20.9609239...
9	GO	126.271731...	294591.949...	22.7668152...
10	MA	145.204150...	119648.219...	38.2570024...



5. Analysis on sales, freight and delivery time:-

1. Calculate days between purchasing, delivering and estimated delivery:-

```
select
order_id,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY
) AS time_to_estimate_deliver,
DATE_DIFF(order_estimated_delivery_date,order_delivered_carrier_date,DAY)
 AS time_to_estimate_deliver
from `retail_data.orders`
where order_delivered_customer_date is not null
order by time_to_deliver
```

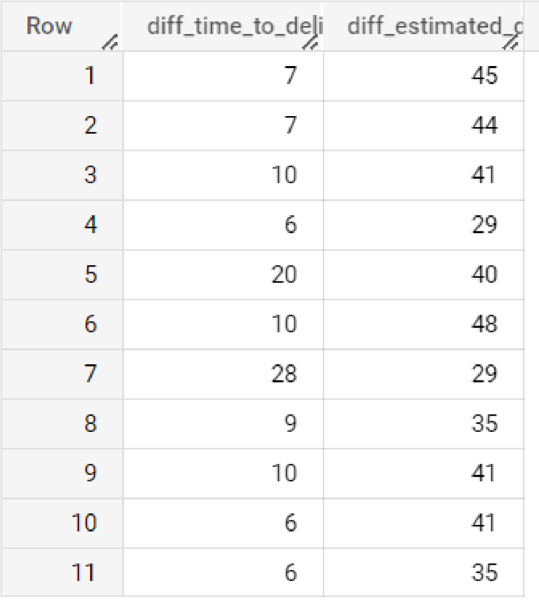


Row	order_id	time_to_deliver	time_to_estimat	time_to_estimat
1	e65f1eeee1f52024ad1dcd034...	0	9	10
2	bb5a519e352b45b714192a02f...	0	25	26
3	434cecee7d1a65fc65358a632...	0	19	20
4	d3ca7b82c922817b06e5ca211...	0	11	12
5	1d893dd7ca5f77ebf5f59f0d20...	0	10	10
6	d5fbeedc85190ba88580d6f82...	0	7	8
7	79e324907160caea526fd8b94...	0	8	9
8	38c1e3d4ed6a13cd0cf612d4c...	0	16	16
9	8339b608be0d84fca9d8da68b...	0	27	27
10	f349cdb62f69c3fae5c4d7d3f3...	0	12	13

2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

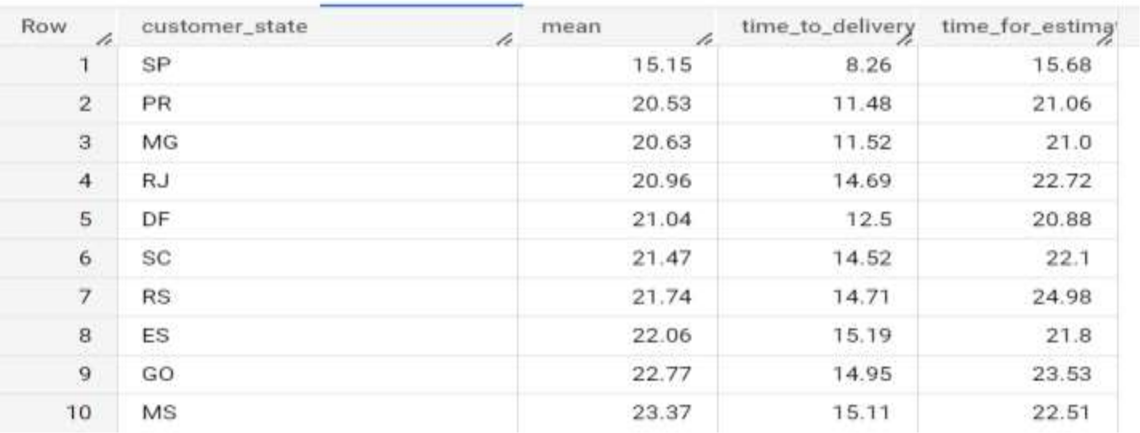
```
select
abs(DATE_DIFF(order_purchase_timestamp,order_delivered_customer_date,day)
) as time_to_delivery,
abs(date_diff(order_estimated_delivery_date,order_delivered_customer_date
,day)) as diff_estimated_delivery
from `retail_data.orders`
```



Row	diff_time_to_deli	diff_estimated_c
1	7	45
2	7	44
3	10	41
4	6	29
5	20	40
6	10	48
7	28	29
8	9	35
9	10	41
10	6	41
11	6	35

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

```
select
c.customer_state,
round(avg(oi.freight_value),2) as mean,
round(Avg(date_diff(o.order_delivered_customer_date,o.order_purcha
se_timestamp,day)),2) as time_to_delivery,
round(avg(date_diff(o.order_estimated_delivery_date,o.order_delive
red_carrier_date,day)),2) as time_for_estimated_delivery
from `retail_data.customers` as c
 join `retail_data.orders` as o on c.customer_id=o.customer_id
join `retail_data.order_items` as oi on oi.order_id=o.order_id
group by c.customer_state
order by mean
```

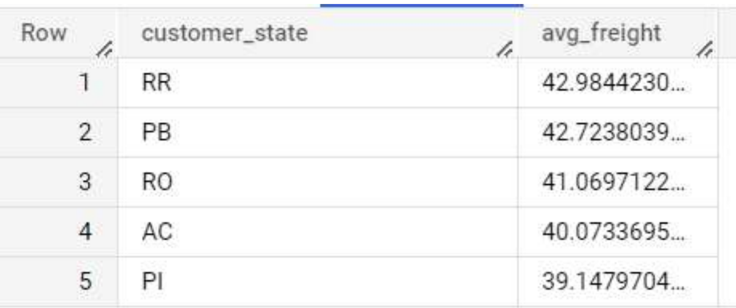


Row	customer_state	mean	time_to_delivery	time_for_estima
1	SP	15.15	8.26	15.68
2	PR	20.53	11.48	21.06
3	MG	20.63	11.52	21.0
4	RJ	20.96	14.69	22.72
5	DF	21.04	12.5	20.88
6	SC	21.47	14.52	22.1
7	RS	21.74	14.71	24.98
8	ES	22.06	15.19	21.8
9	GO	22.77	14.95	23.53
10	MS	23.37	15.11	22.51

4.Sort the data to get the following:

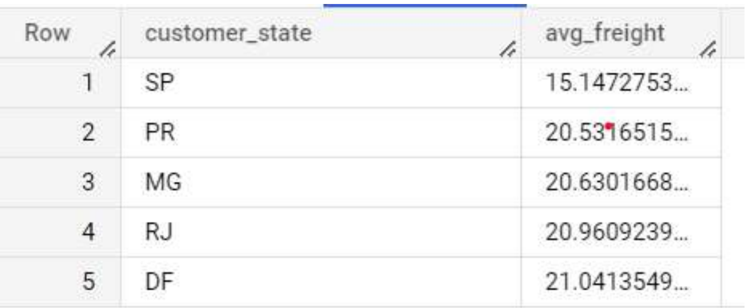
5.Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
HIGHEST AVERAGE FREIGHT VALUE

select c.customer_state,AVG(oi.freight_value) as avg_freight
from `retail_data.orders` as o
join `retail_data.customers` as c on c.customer_id=o.customer_id
join `retail_data.order_items` as oi on oi.order_id=o.order_id
group by c.customer_state
order by avg_freight desc
limit 5
```



Row	customer_state	avg_freight
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

```
LOWEST AVERAGE FREIGHT VALUE
select c.customer_state,AVG(oi.freight_value) as avg_freight
from `retail_data.orders` as o
join `retail_data.customers` as c on c.customer_id=o.customer_id
join `retail_data.order_items` as oi on oi.order_id=o.order_id
group by c.customer_state
order by avg_freight ASC
limit 5
```

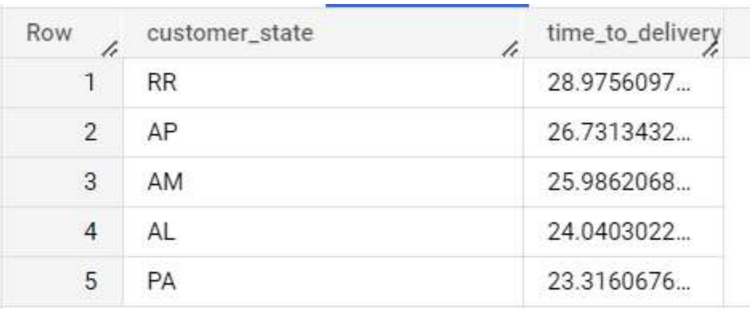


Row	customer_state	avg_freight
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

6. Top 5 states with highest/lowest average time to delivery

HIGHEST AVERAGE TIME FOR DELIVERY

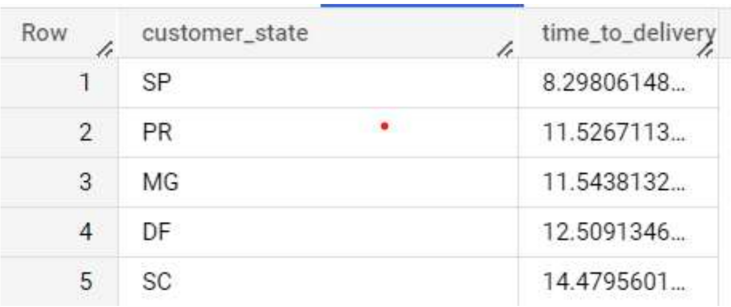
```
select c.customer_state,AVG(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)) as time_to_delivery
from `retail_data.orders` as o
join `retail_data.customers` as c on c.customer_id=o.customer_id
group by c.customer_state
order by time_to_delivery desc
limit 5
```



Row	customer_state	time_to_delivery
1	RR	28.9756097...
2	AP	26.7313432...
3	AM	25.9862068...
4	AL	24.0403022...
5	PA	23.3160676...

LOWEST AVERAGE TIME FOR DELIVERY

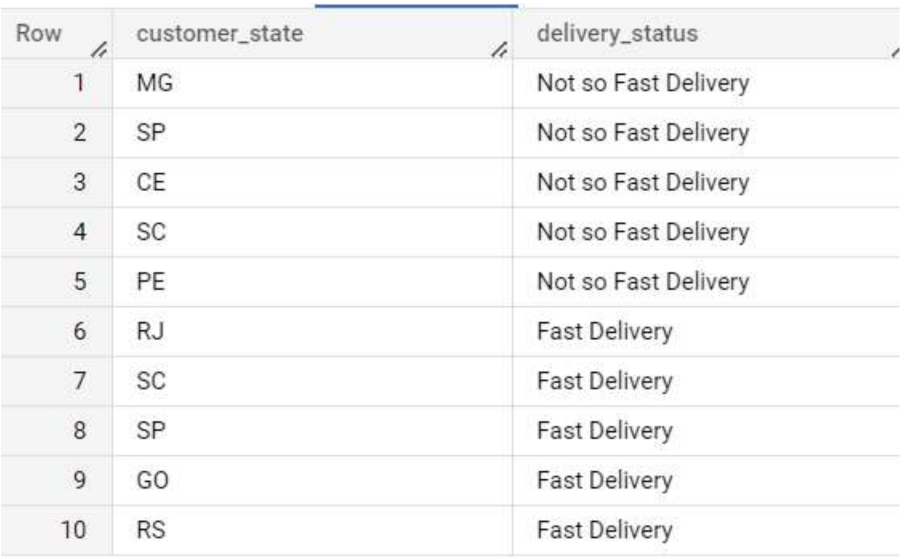
```
select c.customer_state,AVG(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day)) as time_to_delivery
from `retail_data.orders` as o
join `retail_data.customers` as c on c.customer_id=o.customer_id
group by c.customer_state
order by time_to_delivery desc
limit 5
```



Row	customer_state	time_to_delivery
1	SP	8.29806148...
2	PR	11.5267113...
3	MG	11.5438132...
4	DF	12.5091346...
5	SC	14.4795601...

7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

```
WITH delivery_time AS (

  SELECT
    c.customer_state,
    CASE
      WHEN o.order_delivered_customer_date <= o.order_estimated_delivery_date THEN 'Fast Delivery'
      ELSE 'Not so Fast Delivery'
    END AS delivery_status,
  FROM `retail_data.orders` as o join `retail_data.customers` as c
 on o.customer_id=c.customer_id
  WHERE o.order_delivered_customer_date IS NOT NULL
  GROUP BY c.customer_state, delivery_status
),
delivery_times_rank AS (
  SELECT
    customer_state,
    delivery_status,
    ROW_NUMBER() OVER (PARTITION BY delivery_status ) AS rank_of
  FROM delivery_time
)
SELECT
  customer_state,
  delivery_status,
FROM delivery_times_rank
WHERE rank_of <= 5
ORDER BY delivery_status desc
```

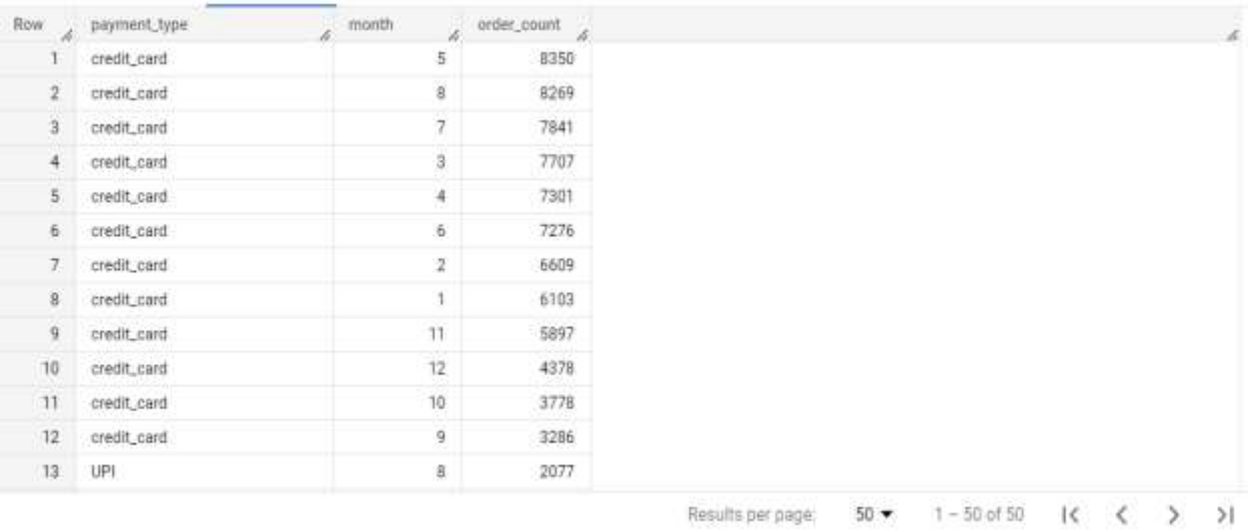


Row	customer_state	delivery_status
1	MG	Not so Fast Delivery
2	SP	Not so Fast Delivery
3	CE	Not so Fast Delivery
4	SC	Not so Fast Delivery
5	PE	Not so Fast Delivery
6	RJ	Fast Delivery
7	SC	Fast Delivery
8	SP	Fast Delivery
9	GO	Fast Delivery
10	RS	Fast Delivery

6. Payment type analysis:

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

```
SELECT
  p.payment_type,
  EXTRACT(MONTH from o.order_purchase_timestamp) AS month,
  count(*) AS order_count
FROM
  `retail_data.orders` as o
JOIN `retail_data.payments` p ON o.order_id = p.order_id
GROUP BY
  p.payment_type,
  month
ORDER BY
  order_count desc
```

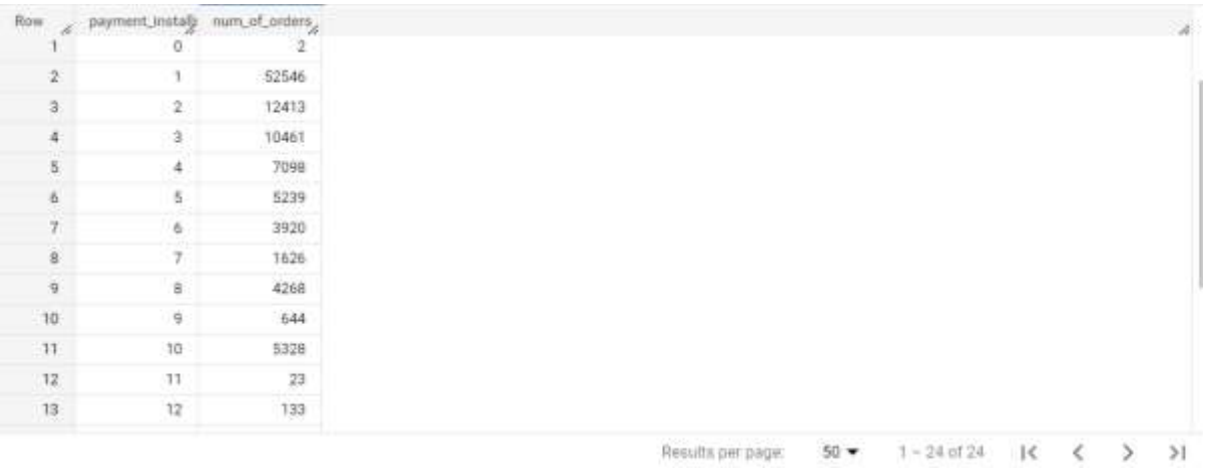


Row	payment_type	month	order_count
1	credit_card	5	8350
2	credit_card	8	8269
3	credit_card	7	7841
4	credit_card	3	7707
5	credit_card	4	7301
6	credit_card	6	7276
7	credit_card	2	6609
8	credit_card	1	6103
9	credit_card	11	5897
10	credit_card	12	4378
11	credit_card	10	3778
12	credit_card	9	3286
13	UPI	8	2077

We can see that orders are more ordered by payment type credit card

2. Count of orders based on the no. of payment installments

```
select payment_installments,count(order_id) as num_of_orders
from `retail_data.payments` as p
group by payment_installments
```



Row	payment_install	num_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328
12	11	23
13	12	133