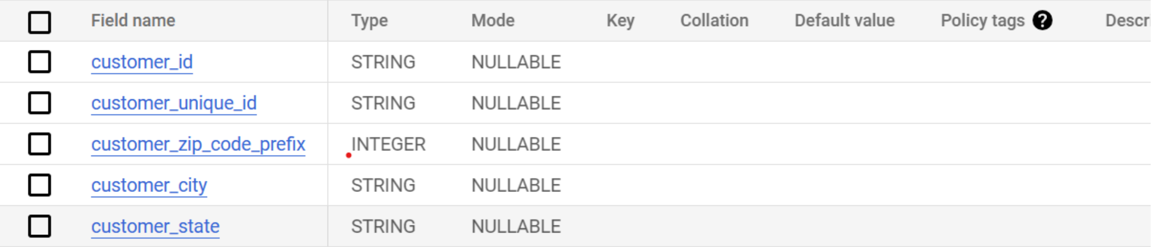
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
    1. Data type of all columns in the "customers" table.
    2. Get the time range between which the orders were placed.
    3. Count the Cities & States of customers who ordered during the given period.

Ques.no-1.1 Ans.

```
select *
from `BCTS.customers`
```



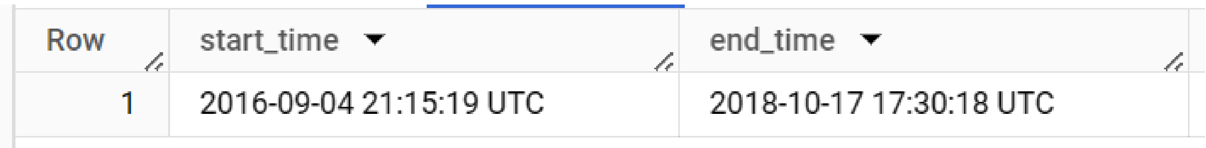
□	Field name	Type	Mode	Key	Collation	Default value	Policy tags ?	Descr
□	customer_id	STRING	NULLABLE					
□	customer_unique_id	STRING	NULLABLE					
□	customer_zip_code_prefix	INTEGER	NULLABLE					
□	customer_city	STRING	NULLABLE					
□	customer_state	STRING	NULLABLE					

Insights:-

There are total of five columns in which 4 columns having string data type and one col having integer data type.

Ques.no-1.2 Ans.

```
select
      min(order_purchase_timestamp) as start_time,
      max(order_purchase_timestamp) as end_time
from `BCTS.orders`
```



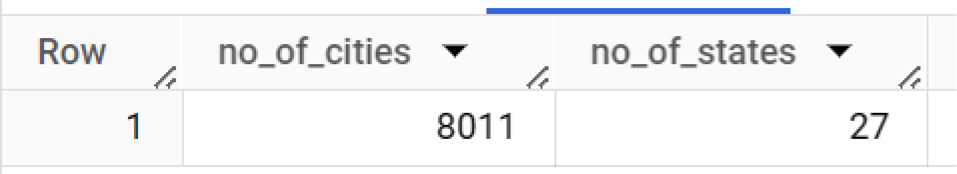
Row	start_time	end_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights:-

Earliest and Latest Order Timestamps: The min(order_purchase_timestamp) retrieves the earliest timestamp when an order was placed. It indicates the first order placed in the data set.
Conversely max(order_purchase_timestamp) provides the latest timestamp.

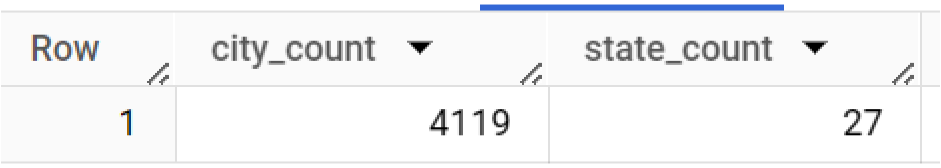
Ques.no-1.3 Ans.

```
select
      count(distinct geolocation_city) as no_of_cities,
      count(distinct geolocation_state) as no_of_states
from `BCTS.geolocation`
```



Row	no_of_cities	no_of_states
1	8011	27

```
select
      count(distinct customer_city) as city_count,
      count(distinct customer_state) as state_count
from `BCTS.customers`
```



Row	city_count	state_count
1	4119	27

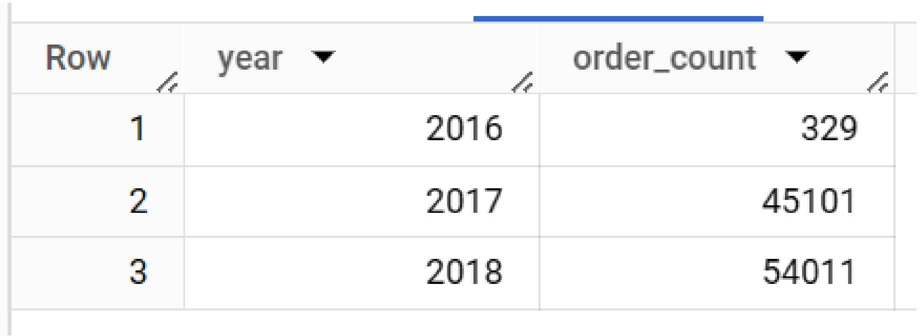
Insights:-
This query provides a breakdown of the number of orders placed by customers in different cities and states during the specified period.

2 . In-depth Exploration:

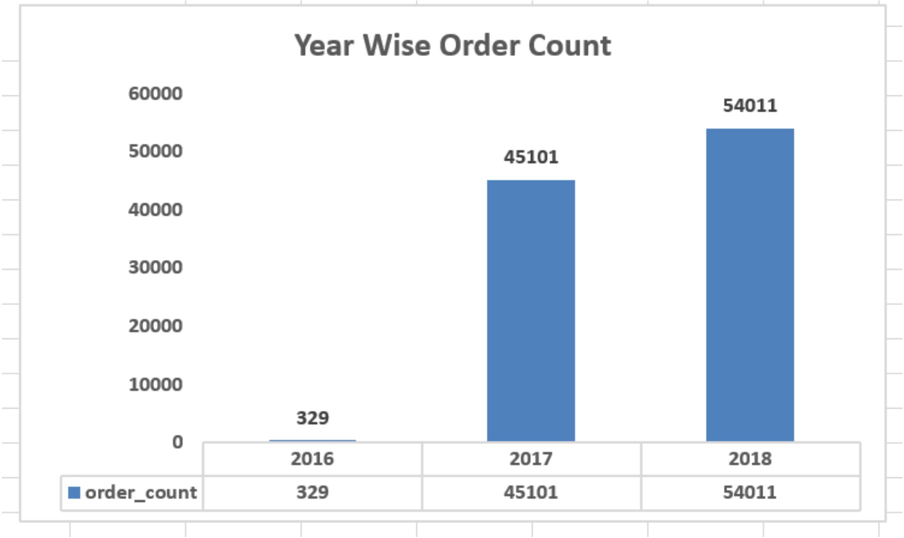
1. Is there a growing trend in the no. of orders placed over the past years?
2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
   - 0-6 hrs : Dawn
   - 7-12 hrs : Mornings
   - 13-18 hrs : Afternoon
   - 19-23 hrs : Night

Ques.no-2.1 Ans.

```
select
     extract(year from order_purchase_timestamp) as year,
     count(order_id) as order_count
from `BCTS.orders`
group by year
order by year asc;
```



Row	year	order_count
1	2016	329
2	2017	45101
3	2018	54011



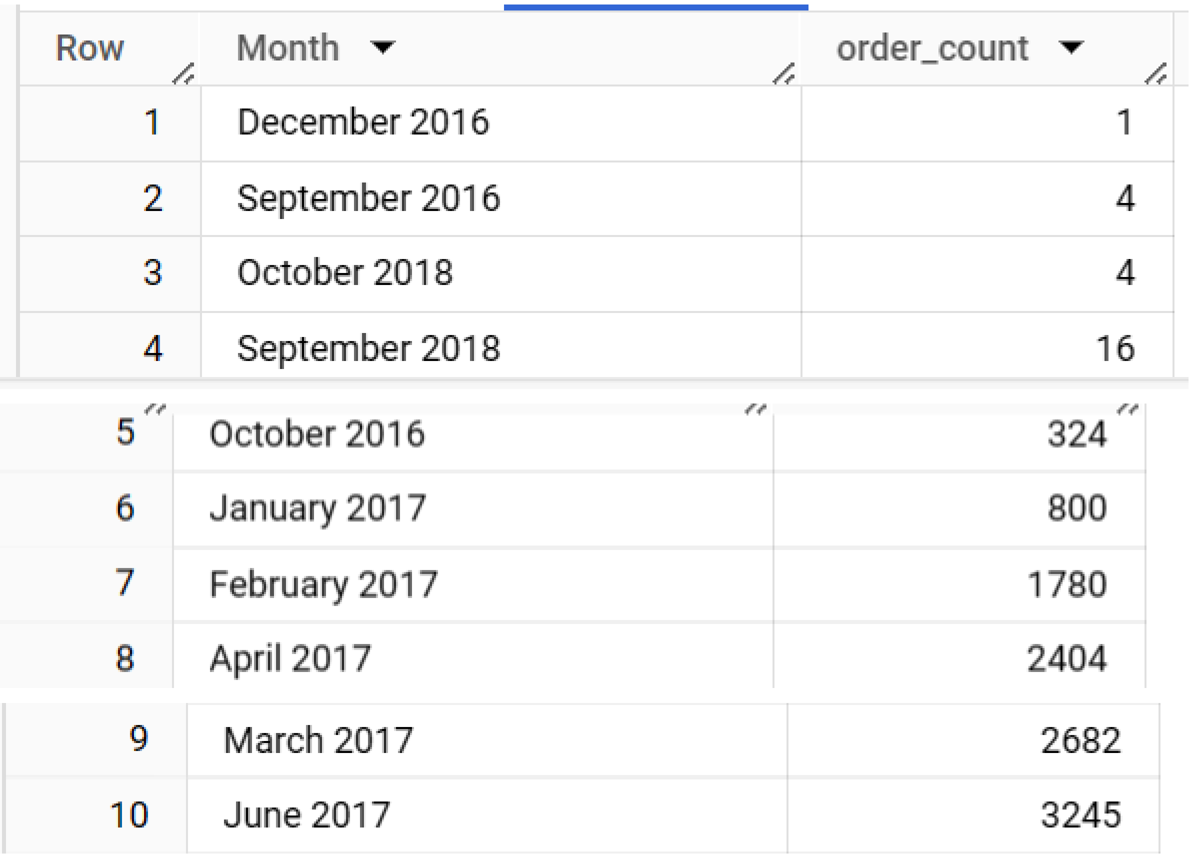
Insights:-

Business Growth Analysis Year Wise: As per above past year order trend, order is continuously improving year by year. This trend is indicative of business expansion and increasing customer demand.

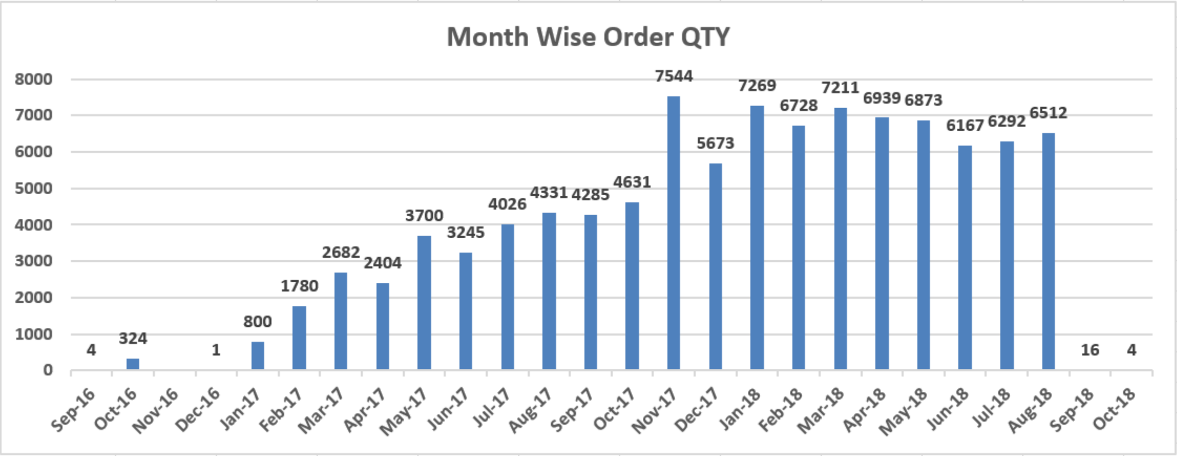
A growing trend in orders is typically a positive sign for a business. It suggests that the products or services are in demand, and the business is effectively reaching and engaging its target audience.

Ques.no-2.2 Ans.

```
select
      format_date('%B %Y',order_purchase_timestamp) as Month,
      count(order_id) as order_count
from `BCTS.orders`
group by month
order by order_count asc;
```



Row	Month	order_count
1	December 2016	1
2	September 2016	4
3	October 2018	4
4	September 2018	16
5	October 2016	324
6	January 2017	800
7	February 2017	1780
8	April 2017	2404
9	March 2017	2682
10	June 2017	3245



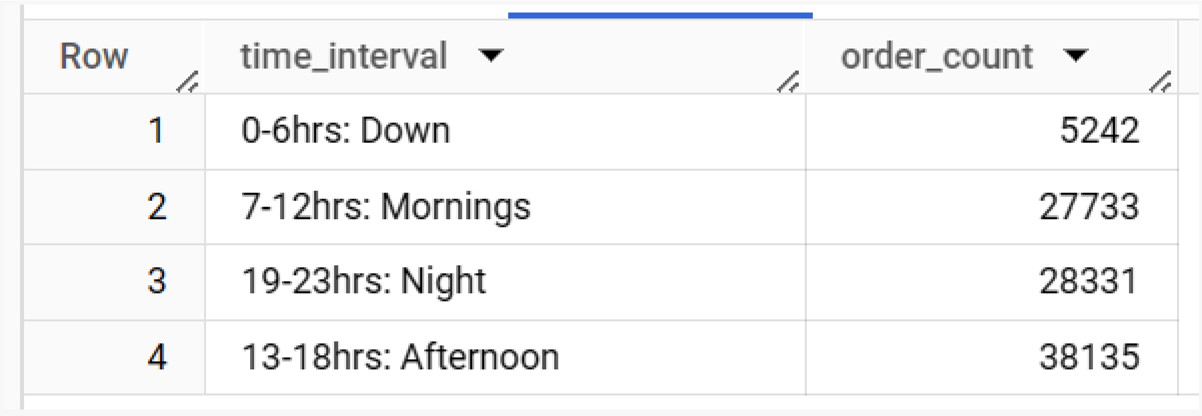
Insights:-

Business Growth Analysis Month Wise: As per above past months order trend, order is continuously improving month by month. This trend is indicative of business expansion and increasing customer demand.

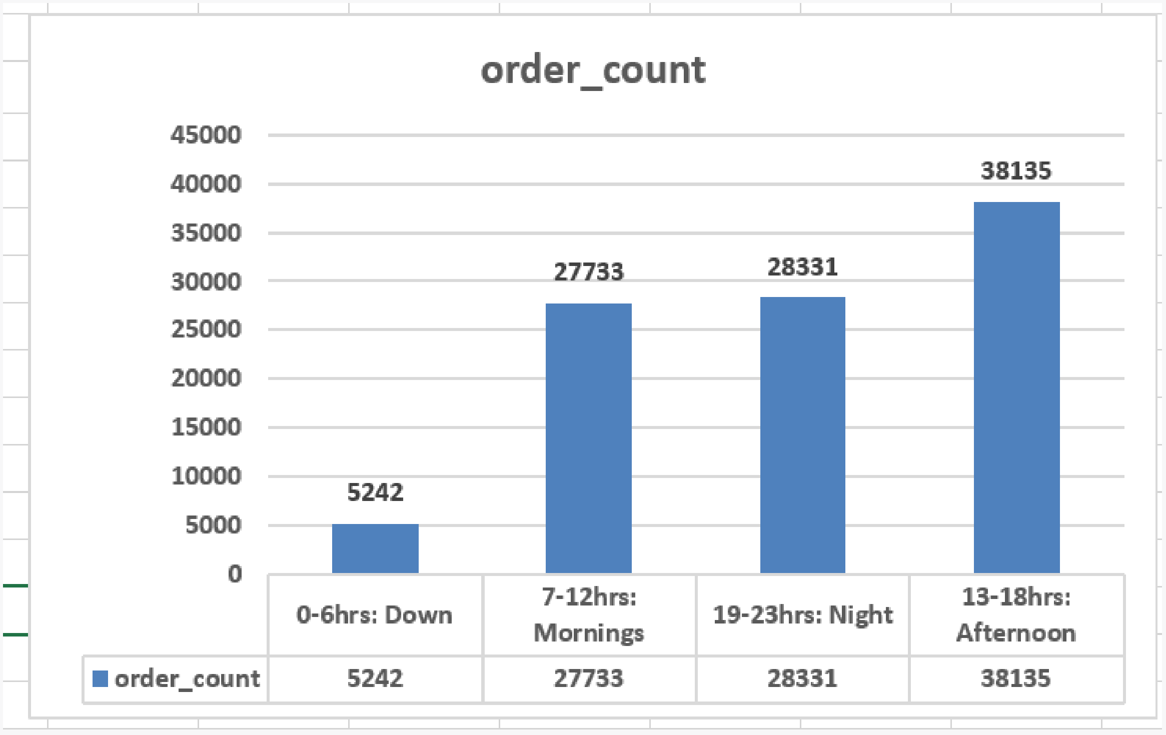
There is growth in sell month of Nov’2017-7544, Dec’2017-5673Jan’2018-7269
So we should keep inventory high in these months to fulfil the customer demand.

Ques.no-2.3 Ans.

```
select
      case
      when extract(hour from order_purchase_timestamp) between 0 and 6 then '0-6hrs: Down'
      when extract(hour from order_purchase_timestamp) between 7 and 12 then '7-12hrs: Mornings'
      when extract(hour from order_purchase_timestamp) between 13 and 18 then '13-18hrs: Afternoon'
      when extract(hour from order_purchase_timestamp) between 19 and 23 then '19-23hrs: Night'
      end as time_interval,
      count(order_id) as order_count
from `BCTS.orders`
group by time_interval
order by order_count asc;
```



Row	time_interval	order_count
1	0-6hrs: Down	5242
2	7-12hrs: Mornings	27733
3	19-23hrs: Night	28331
4	13-18hrs: Afternoon	38135



Insights:-

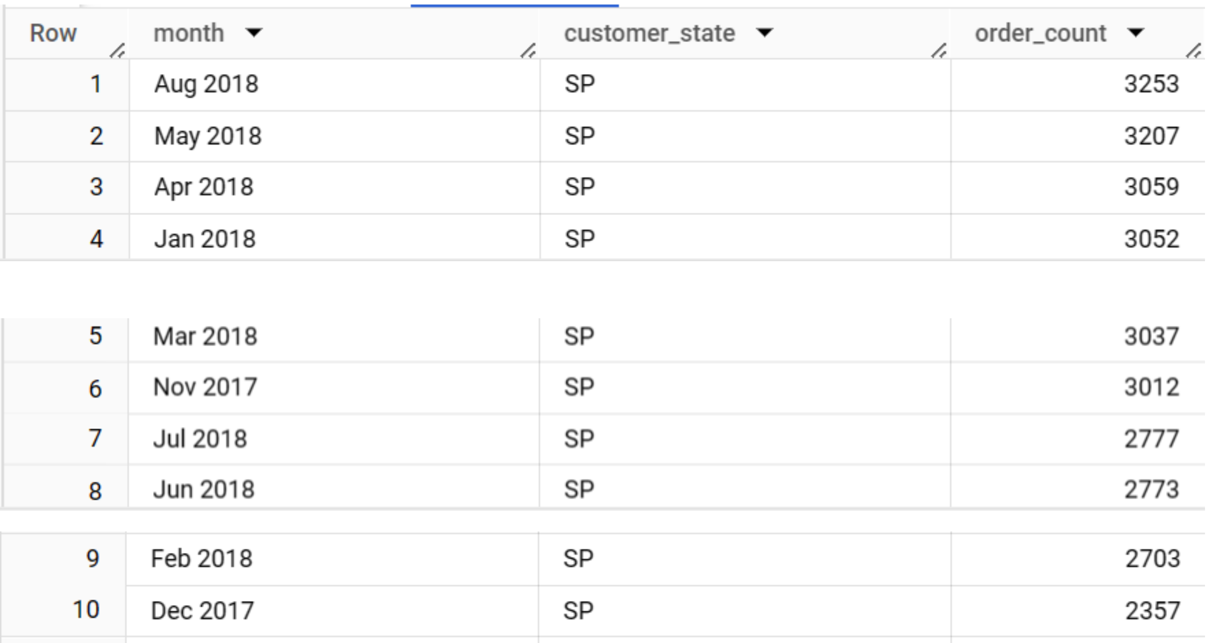
Order Placing Analysis Time Wise: As per above data Brazilian customers were placing most orders in 13-18hrs:Afternoon-38135 and then night hrs 19-23hrs-28331.
So during this time period we need to ensure inventory ,staff availability and network for better processing.

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

1. Get the month on month no. of orders placed in each state.
2. How are the customers distributed across all the states?

Ques.no-3.1 Ans.

```
sselect
       format_date('%b %Y', o.order_purchase_timestamp) as month,
       c.customer_state,
       count(o.order_id) as order_count
from `BCTS.orders` o
left join `BCTS.customers` c
on c.customer_id = o.customer_id
group by c.customer_state, month
order by order_count desc;
```



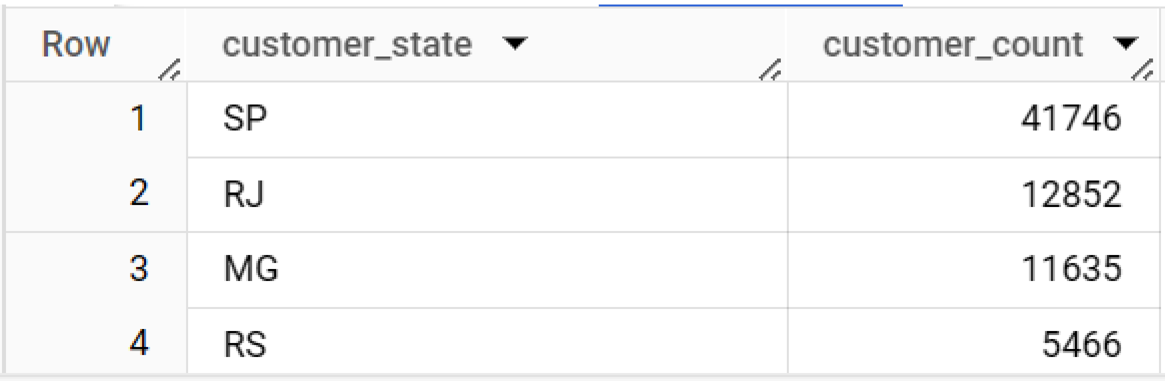
Row	month	customer_state	order_count
1	Aug 2018	SP	3253
2	May 2018	SP	3207
3	Apr 2018	SP	3059
4	Jan 2018	SP	3052
5	Mar 2018	SP	3037
6	Nov 2017	SP	3012
7	Jul 2018	SP	2777
8	Jun 2018	SP	2773
9	Feb 2018	SP	2703
10	Dec 2017	SP	2357

<u>Insights:-</u>

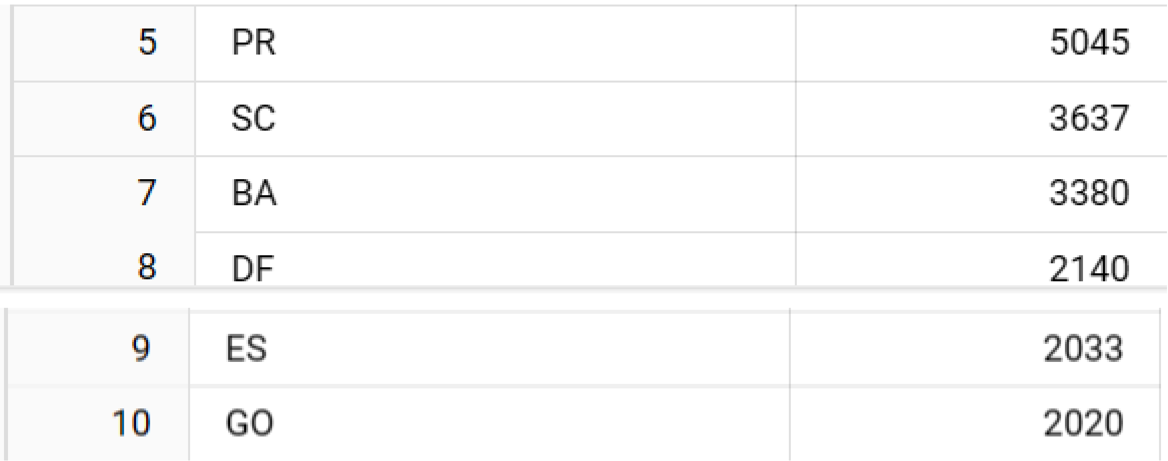
State Wise Order Analysis: There states SP, RJ and MG having consistently higher order numbers compared to others states so need to focus on these states.

<u>Ques.no-3.2 Ans.</u>

```
select c.customer_state,
       count(c.customer_id) as customer_count
from `BCTS.orders` o
left join `BCTS.customers` c
on c.customer_id = o.customer_id
group by c.customer_state
order by customer_count desc;
```



Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466



5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

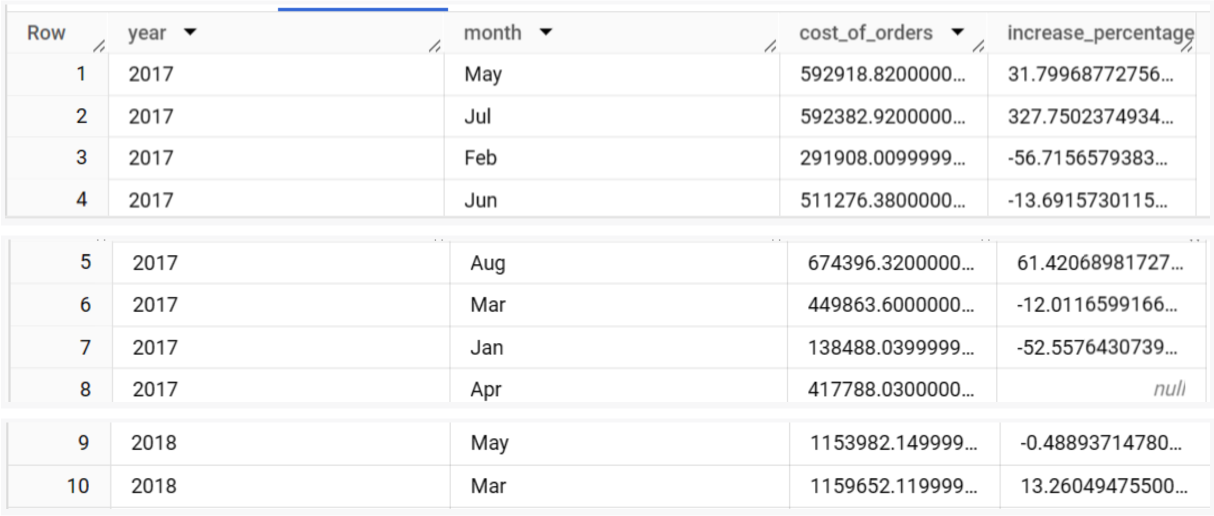
Insights:-

State Wise Customer Analysis: There states SP, RJ and MG having consistently higher customers compared to others states so need to focus on these states.
For getting more orders from these states we can start promotions in those areas.

4. **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.
    2. Calculate the Total & Average value of order price for each state.
    3. Calculate the Total & Average value of order freight for each state.

Ques.no-4.1 Ans.

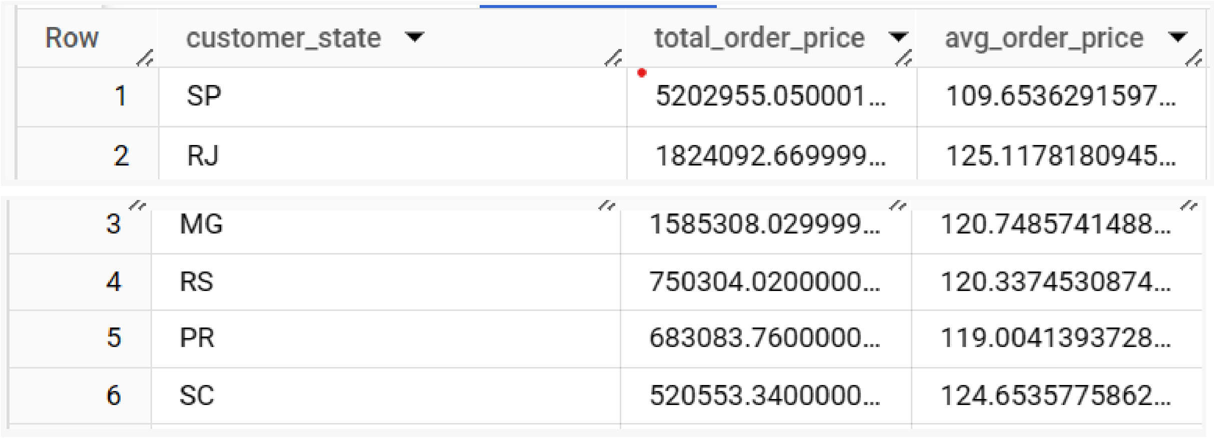
```
with final as(
select
      format_date('%Y', o.order_purchase_timestamp) as year,
      format_date('%b', o.order_purchase_timestamp) as month,
      sum(p.payment_value) as cost_of_orders
from `BCTS.orders` o
left join `BCTS.payments` p
on o.order_id = p.order_id
group by 1,2
having year in ('2018','2017') and month in ('Jan', 'Feb', 'Mar', 'Apr', 'May',
'Jun', 'Jul', 'Aug')
)
select *,
        100 * (cost_of_orders -
        lag(cost_of_orders) over(order by year, month))/
        lag(cost_of_orders) over(order by year, month) as increase_percentage
from final
order by 1;
```



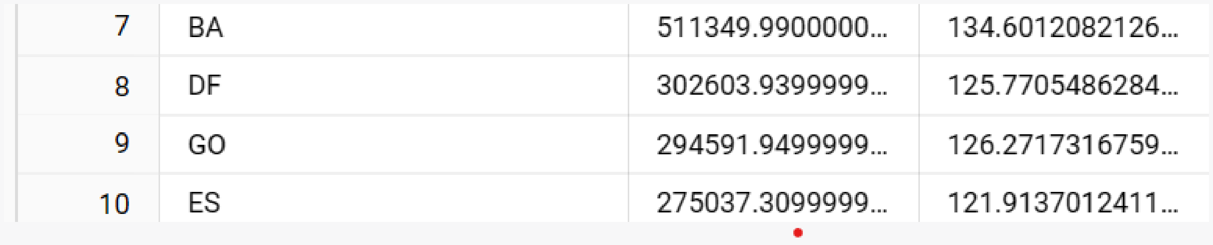
Row	year	month	cost_of_orders	increase_percentage
1	2017	May	592918.8200000...	31.79968772756...
2	2017	Jul	592382.9200000...	327.7502374934...
3	2017	Feb	291908.0099999...	-56.7156579383...
4	2017	Jun	511276.3800000...	-13.6915730115...
5	2017	Aug	674396.3200000...	61.42068981727...
6	2017	Mar	449863.6000000...	-12.0116599166...
7	2017	Jan	138488.0399999...	-52.5576430739...
8	2017	Apr	417788.0300000...	null
9	2018	May	1153982.149999...	-0.48893714780...
10	2018	Mar	1159652.119999...	13.26049475500...

Ques.no-4.2 Ans.

```
select c.customer_state,
       sum(oi.price) as total_order_price,
       avg(oi.price) as avg_order_price
from`BCTS.customers` c
left join `BCTS.orders` o
on c.customer_id = o.customer_id
left join `BCTS.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by total_order_price desc;
```



Row	customer_state	total_order_price	avg_order_price
1	SP	5202955.050001...	109.6536291597...
2	RJ	1824092.669999...	125.1178180945...
3	MG	1585308.029999...	120.7485741488...
4	RS	750304.0200000...	120.3374530874...
5	PR	683083.7600000...	119.0041393728...
6	SC	520553.3400000...	124.6535775862...



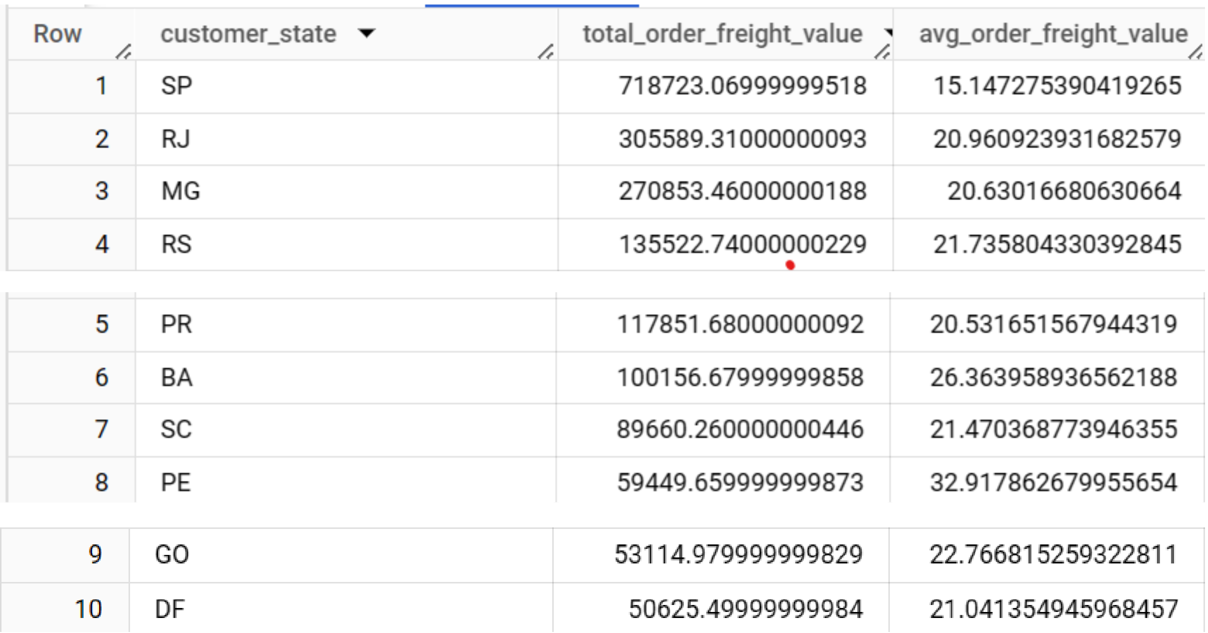
7	BA	511349.9900000...	134.6012082126...
8	DF	302603.9399999...	125.7705486284...
9	GO	294591.9499999...	126.2717316759...
10	ES	275037.3099999...	121.9137012411...

Insights:-

State wise total order price and avg. price analysis: As per above data, states having lower avg_order_ price those having higher selling so we need to focus on inventory of lower_avg_price material to complete customer demands.

Ques.no-4.3 Ans.

```
select c.customer_state,
       sum(oi.freight_value) as total_order_freight_value,
       avg(oi.freight_value) as avg_order_freight_value
from`BCTS.customers` c
left join `BCTS.orders` o
on c.customer_id = o.customer_id
left join `BCTS.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by total_order_freight_value desc;
```



Row	customer_state	total_order_freight_value	avg_order_freight_value
1	SP	718723.06999999518	15.147275390419265
2	RJ	305589.31000000093	20.960923931682579
3	MG	270853.46000000188	20.63016680630664
4	RS	135522.74000000229	21.735804330392845
5	PR	117851.68000000092	20.531651567944319
6	BA	100156.67999999858	26.363958936562188
7	SC	89660.260000000446	21.470368773946355
8	PE	59449.659999999873	32.917862679955654
9	GO	53114.979999999829	22.766815259322811
10	DF	50625.49999999984	21.041354945968457

Insights:-

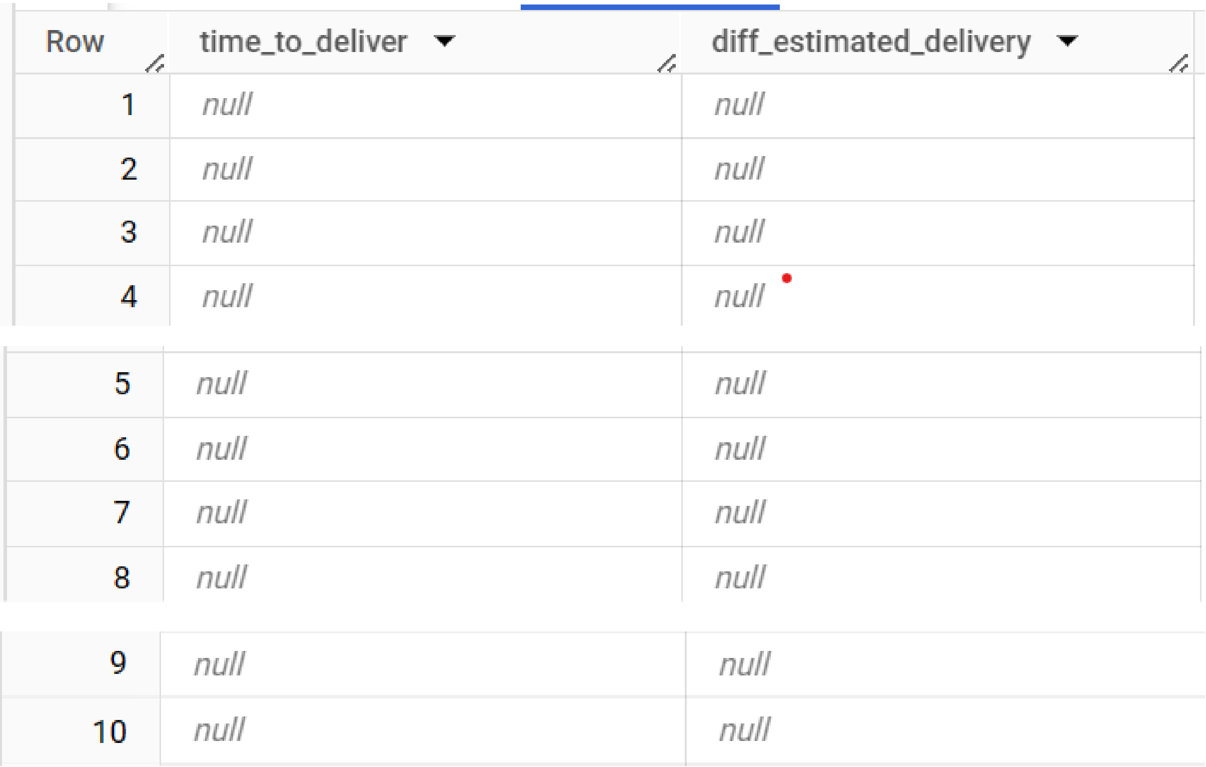
State wise total order freight value and avg. order freight value analysis: As per above data, states having lower avg_order_ freight_value those having higher selling so we need to focus on inventory of lower_avg_freight_value material to complete customer demands.

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

Ques.no- 5.1 Ans.

```
select
       order_purchase_timestamp - order_delivered_customer_date as
       time_to_deliver,
       order_estimated_delivery_date - order_delivered_customer_date as
       diff_estimated_delivery
from `BCTS.orders`
```

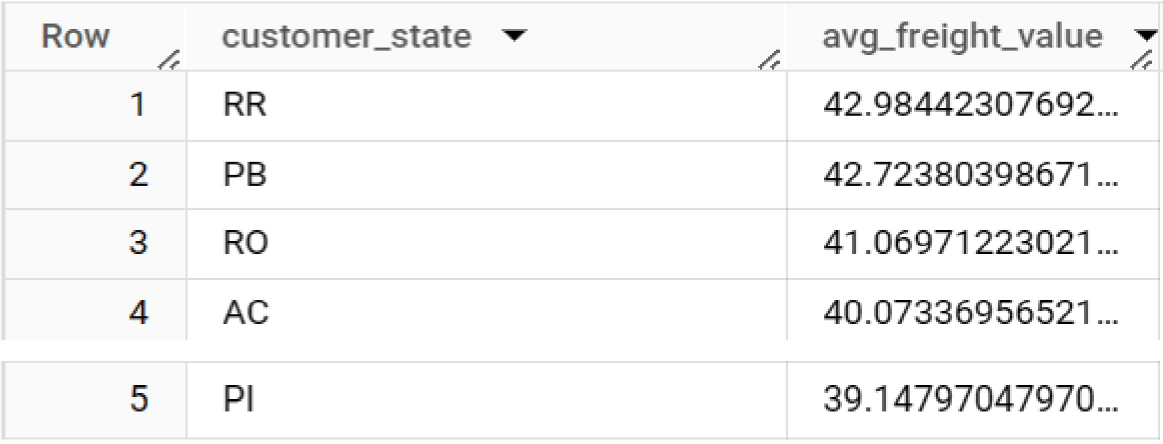


Row	time_to_deliver	diff_estimated_delivery
1	*null*	*null*
2	*null*	*null*
3	*null*	*null*
4	*null*	*null*
5	*null*	*null*
6	*null*	*null*
7	*null*	*null*
8	*null*	*null*
9	*null*	*null*
10	*null*	*null*

Insights:-

Ques.no- 5.2 Ans.

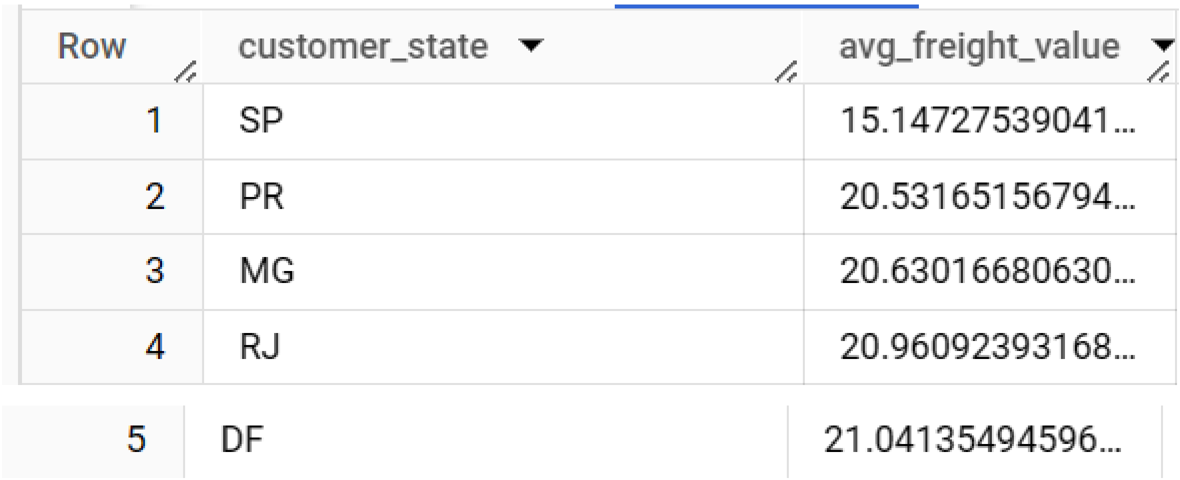
```
Max_avg_freight_value
select c.customer_state,
       avg(oi.freight_value) as avg_freight_value,
from`BCTS.customers` c
left join `BCTS.orders` o
on c.customer_id = o.customer_id
left join `BCTS.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by avg_freight_value desc;
```



Row	customer_state	avg_freight_value
1	RR	42.98442307692…
2	PB	42.72380398671…
3	RO	41.06971223021…
4	AC	40.07336956521…
5	PI	39.14797047970…

```
Min_avg_freight_value

select c.customer_state,
       avg(oi.freight_value) as avg_freight_value,
from`BCTS.customers` c
left join `BCTS.orders` o
on c.customer_id = o.customer_id
left join `BCTS.order_items` oi
on o.order_id = oi.order_id
group by c.customer_state
order by avg_freight_value asc;
```



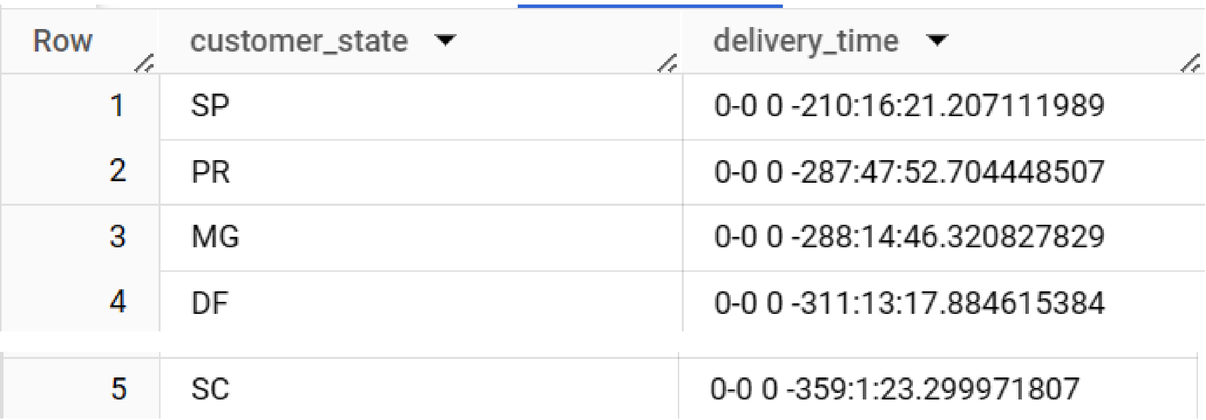
Row	customer_state	avg_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

Insights:-

Highest freight value state wise: as per above data highest freight value states are RR,PB, RO, AC, PI and lowest freight value states are SP, PR, MG, RJ,DF.

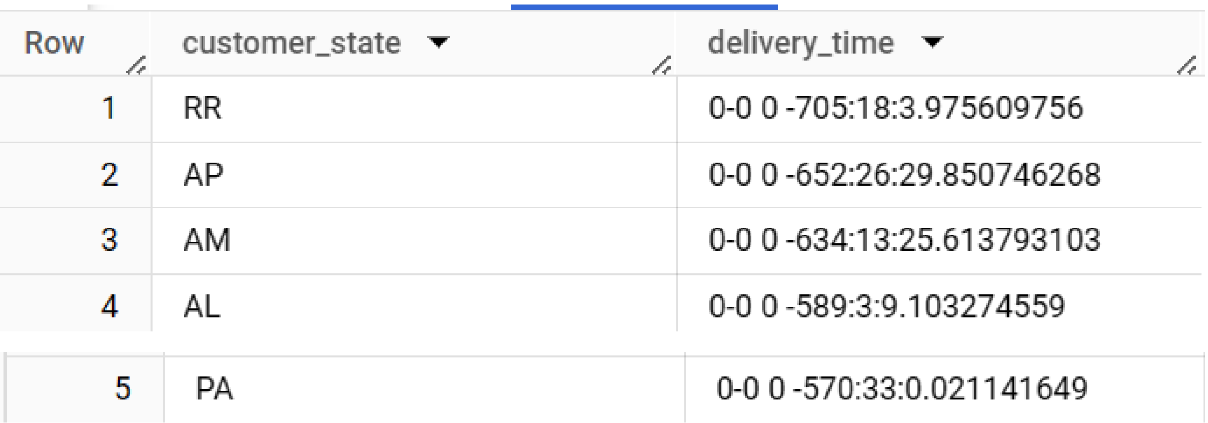
Ques.no- 5.3 Ans.

```
Max_avg_delivery_time
select c.customer_state,
       avg(o.order_purchase_timestamp - o.order_delivered_customer_date) as
       delivery_time
from`BCTS.customers` c
left join `BCTS.orders` o
on c.customer_id = o.customer_id
group by c.customer_state
order by delivery_time desc;
```



Row	customer_state	delivery_time
1	SP	0-0 0 -210:16:21.207111989
2	PR	0-0 0 -287:47:52.704448507
3	MG	0-0 0 -288:14:46.320827829
4	DF	0-0 0 -311:13:17.884615384
5	SC	0-0 0 -359:1:23.299971807

```
Max_avg_delivery_time
select c.customer_state,
       avg(o.order_purchase_timestamp - o.order_delivered_customer_date) as
       delivery_time
from`BCTS.customers` c
left join `BCTS.orders` o
on c.customer_id = o.customer_id
group by c.customer_state
order by delivery_time asc;
```



Row	customer_state	delivery_time
1	RR	0-0 0 -705:18:3.975609756
2	AP	0-0 0 -652:26:29.850746268
3	AM	0-0 0 -634:13:25.613793103
4	AL	0-0 0 -589:3:9.103274559
5	PA	0-0 0 -570:33:0.021141649

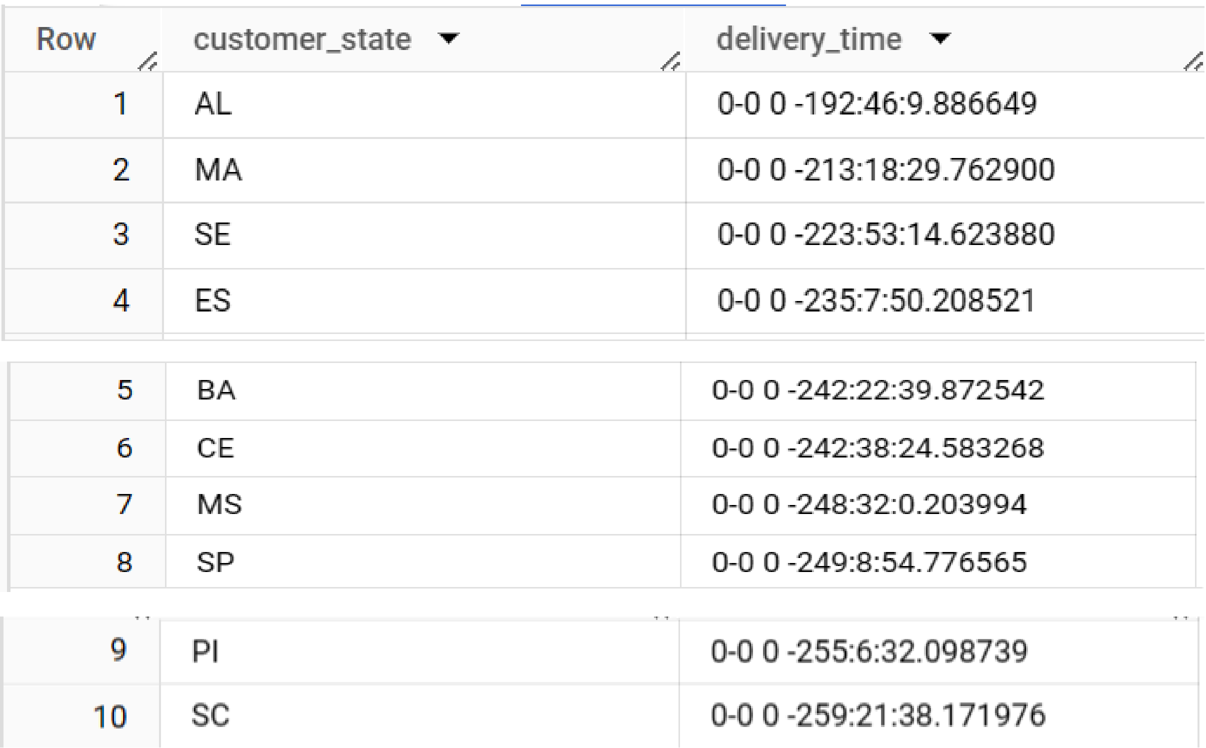
Insights:-

Stage wise highest and lowest average delivery time analysis: as per above data highest delivery time states are SP, PR, MG, DF,SC and lowest delivery time states are RR, AP, AM, AL, PA.

Ques.no- 5.4 Ans.

```
select c.customer_state,
       avg(o.order_delivered_customer_date - o.order_estimated_delivery_date) as
delivery_time
from `BCTS.orders` o
left join `BCTS.customers` c
```

```
on o.customer_id = c.customer_id
group by c.customer_state
order by delivery_time asc;
```



Row	customer_state	delivery_time
1	AL	0-0 0 -192:46:9.886649
2	MA	0-0 0 -213:18:29.762900
3	SE	0-0 0 -223:53:14.623880
4	ES	0-0 0 -235:7:50.208521
5	BA	0-0 0 -242:22:39.872542
6	CE	0-0 0 -242:38:24.583268
7	MS	0-0 0 -248:32:0.203994
8	SP	0-0 0 -249:8:54.776565
9	PI	0-0 0 -255:6:32.098739
10	SC	0-0 0 -259:21:38.171976

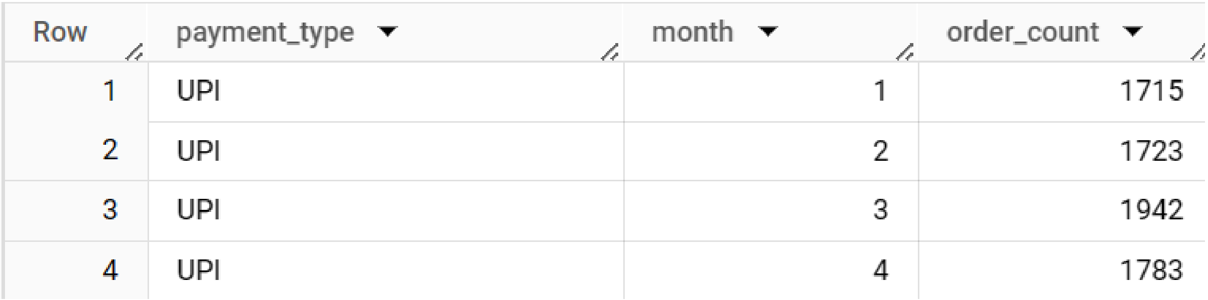
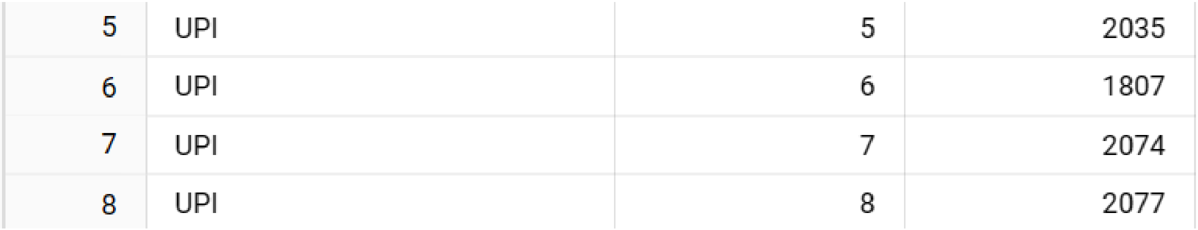
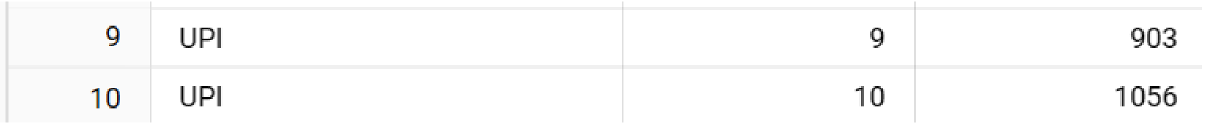
Insights:-

State wise delivery time analysis: The fastest delivery states are AL,MA,SC, ES and BA.

6. **Analysis based on the payments:**
    1. Find the month on month no. of orders placed using different payment types.
    2. Find the no. of orders placed on the basis of the payment instalments that have been paid.

Ques.no- 6.1 Ans.

```
select p.payment_type,
       extract(month from o.order_purchase_timestamp) as month,
       count(distinct o.order_id) as order_count
from `BCTS.orders` o
join `BCTS.payments` p
on o.order_id = p.order_id
group by p.payment_type, month
order by p.payment_type, month;
```

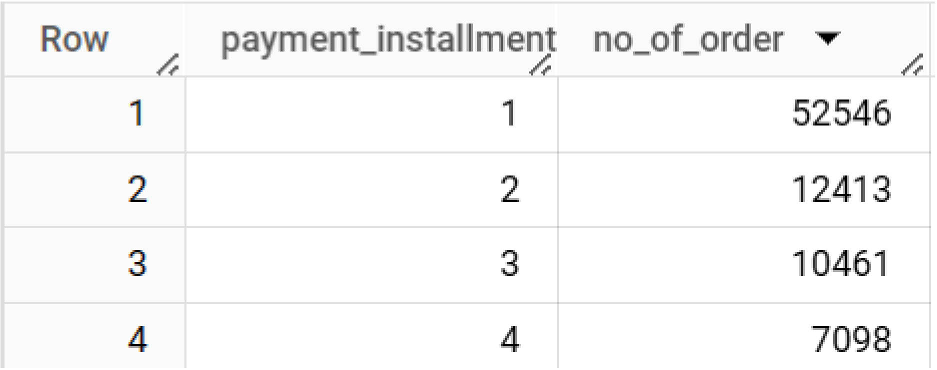
Row	payment_type	month	order_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056

Insights:-

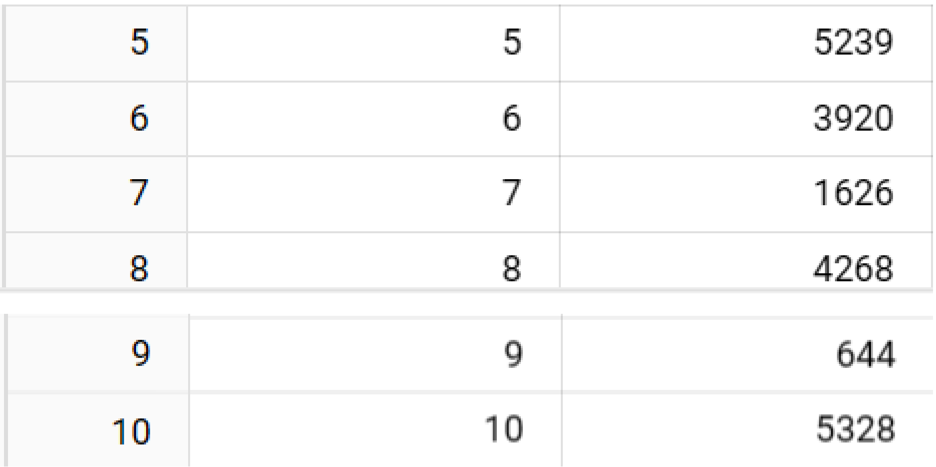
Payments type analysis: There are total 4 types of payment method i.e UPI, Credit_Card, Debit_Card and Voucher.

Ques.no- 6.2 Ans.

```
select payment_installments,
       count(order_id) as no_of_order
from `BCTS.payments`
where payment_installments > 0
group by payment_installments
order by payment_installments asc;
```



Row	payment_installment	no_of_order
1	1	52546
2	2	12413
3	3	10461
4	4	7098



5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

Insights:-

Payment installments analysis: There are a total 13 types of payment installment in which major payment installment types are 1, 2 and 3.