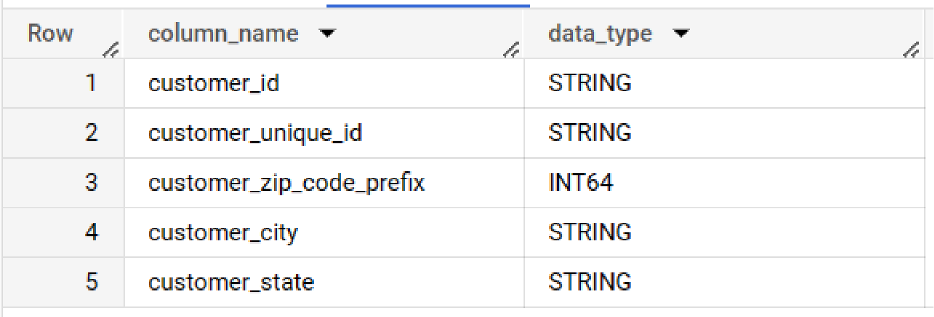
Question 1:

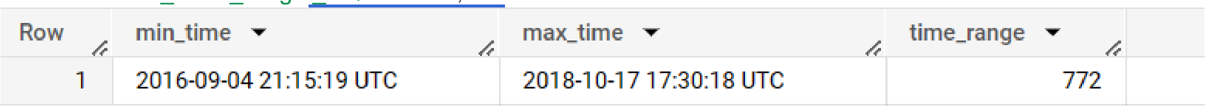
/*Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
Data type of all columns in the "customers" table.
Get the time range between which the orders were placed.
Count the Cities & States of customers who ordered during the given period.*/

```
select column_name, data_type
from Business_Case_Target_SQL.INFORMATION_SCHEMA.COLUMNS
where table_name= 'customers';
```



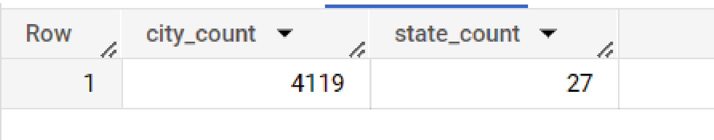
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

```
select min(order_purchase_timestamp) as min_time,
   max(order_purchase_timestamp) as max_time,
   date_diff(max(order_purchase_timestamp), min(order_purchase_timestamp),day) as time_range
from `Business_Case_Target_SQL.orders`;
```



Row	min_time	max_time	time_range
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

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

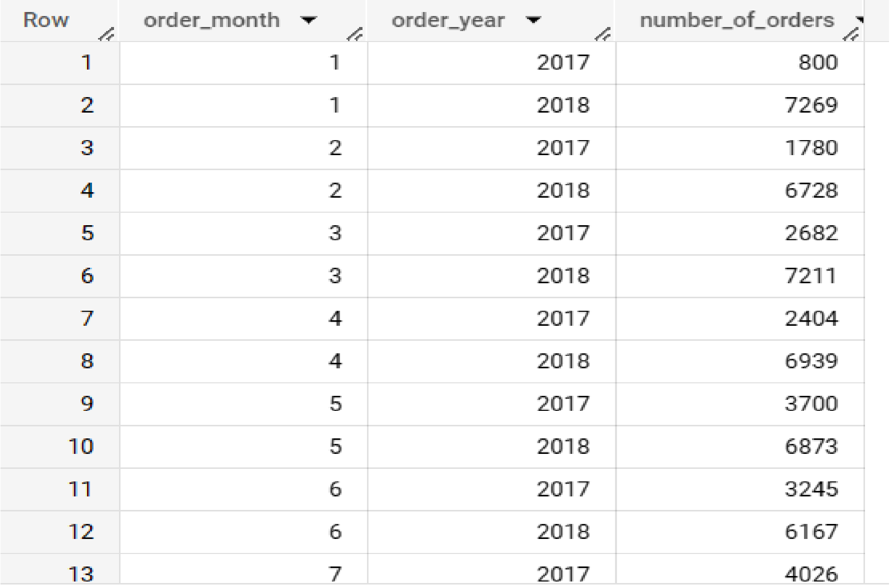


Row	city_count	state_count
1	4119	27

Question 2:

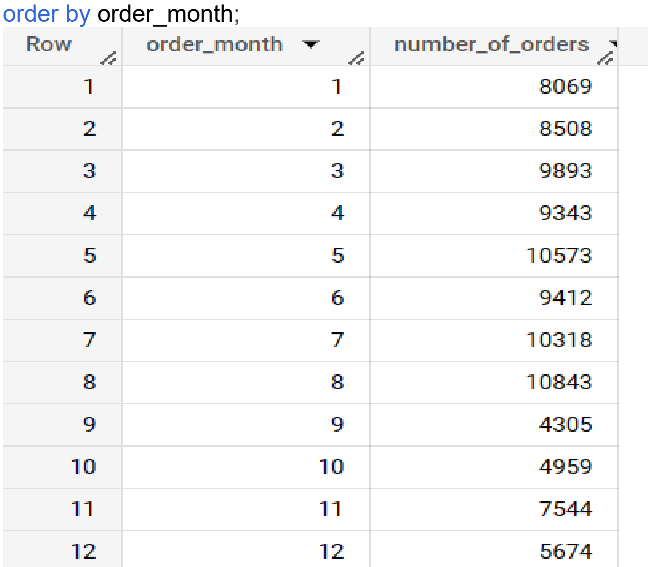
/*In-depth Exploration:

```
Is there a growing trend in the no. of orders placed over the past years?
Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
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*/

select extract(month from order_purchase_timestamp) as order_month, extract(year from
order_purchase_timestamp) as order_year ,count(order_id) as number_of_orders
from `Business_Case_Target_SQL.orders`
group by order_month, order_year
order by order_month,order_year;
```



Row	order_month	order_year	number_of_orders
1	1	2017	800
2	1	2018	7269
3	2	2017	1780
4	2	2018	6728
5	3	2017	2682
6	3	2018	7211
7	4	2017	2404
8	4	2018	6939
9	5	2017	3700
10	5	2018	6873
11	6	2017	3245
12	6	2018	6167
13	7	2017	4026

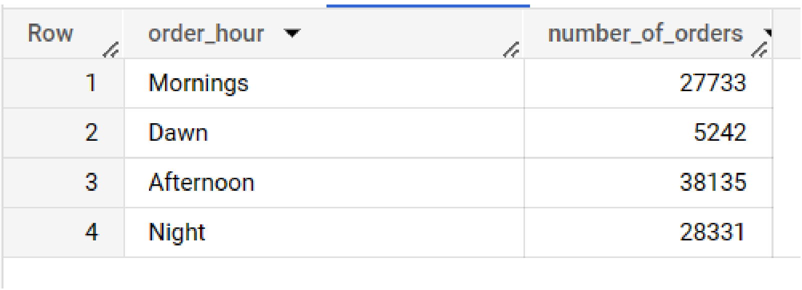
```
select extract(month from order_purchase_timestamp) as order_month,
count(order_id) as number_of_orders
from `Business_Case_Target_SQL.orders`
group by order_month
order by order_month;
```



Row	order_month	number_of_orders
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

```
select
case
```

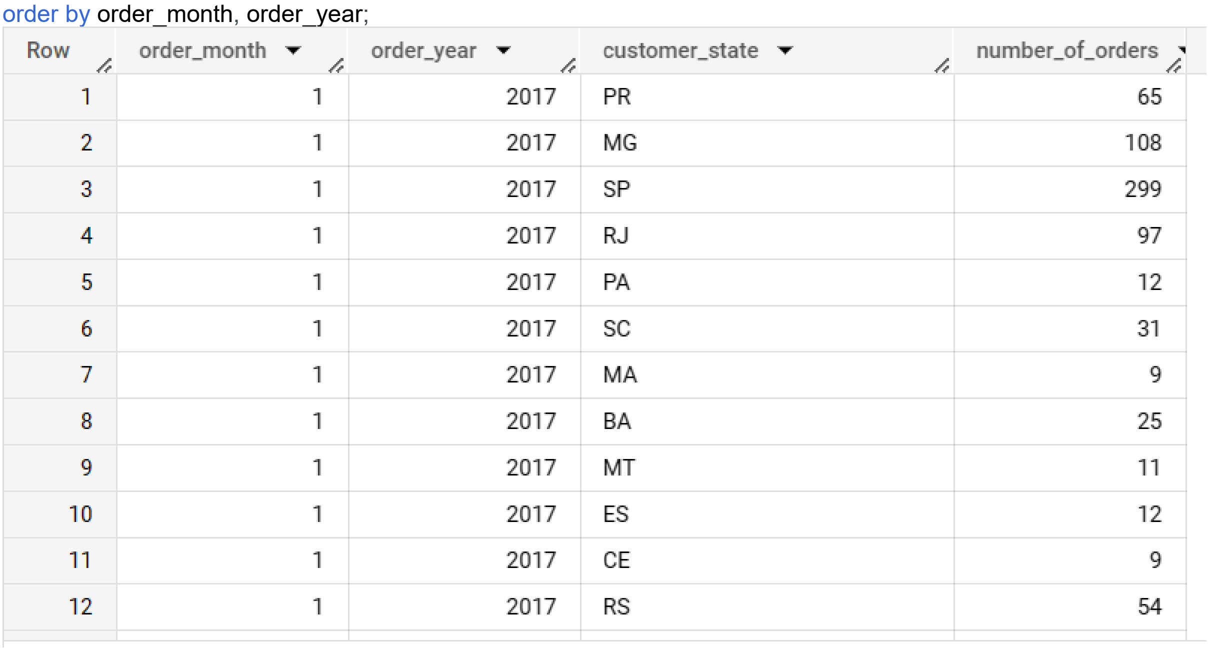
```
  when extract(hour from o.order_purchase_timestamp) between 0 and 6 then 'Dawn'
  when extract(hour from o.order_purchase_timestamp) between 7 and 12 then 'Mornings'
  when extract(hour from o.order_purchase_timestamp) between 13 and 18 then 'Afternoon'
  when extract(hour from o.order_purchase_timestamp) between 19 and 23 then 'Night'
end as order_hour,
count(o.order_id) as number_of_orders
from `Business_Case_Target_SQL.orders` as o
group by order_hour;
```



Row	order_hour	number_of_orders
1	Mornings	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

Question 3:

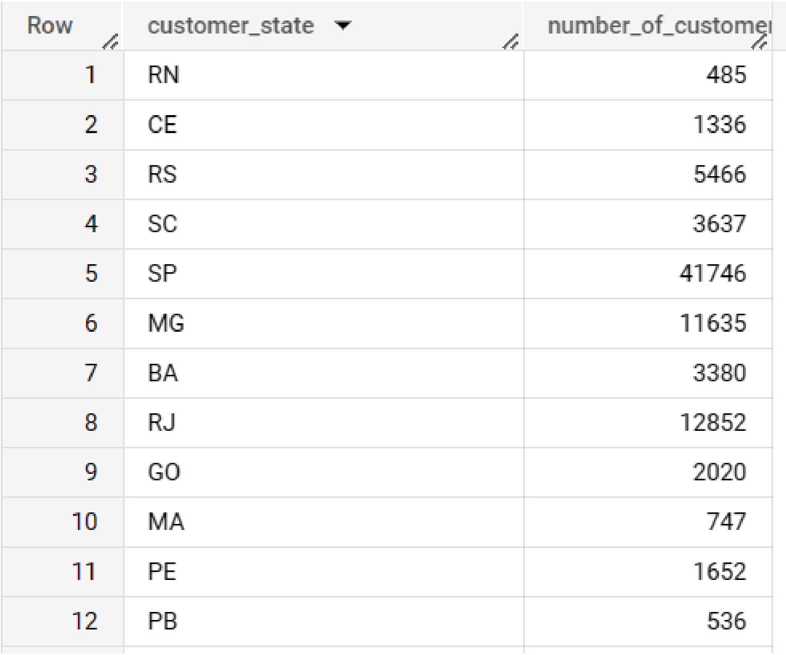
```
/*Evolution of E-commerce orders in the Brazil region:
Get the month on month no. of orders placed in each state.
How are the customers distributed across all the states?*/

select extract(month from o.order_purchase_timestamp) as order_month,
extract(year from o.order_purchase_timestamp) as order_year,
c.customer_state,
count(o.order_id) as number_of_orders
from `Business_Case_Target_SQL.orders` as o
inner join
`Business_Case_Target_SQL.customers` as c
on o.customer_id= c.customer_id
group by order_month, order_year, c.customer_state
order by order_month, order_year;
```



Row	order_month	order_year	customer_state	number_of_orders
1	1	2017	PR	65
2	1	2017	MG	108
3	1	2017	SP	299
4	1	2017	RJ	97
5	1	2017	PA	12
6	1	2017	SC	31
7	1	2017	MA	9
8	1	2017	BA	25
9	1	2017	MT	11
10	1	2017	ES	12
11	1	2017	CE	9
12	1	2017	RS	54

```
select customer_state, count(customer_id) as number_of_customers
```

```
from `Business_Case_Target_SQL.customers`
group by customer_state;
```

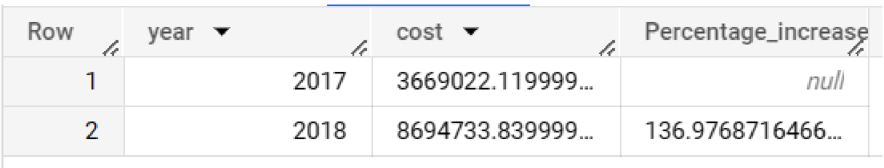


Row	customer_state	number_of_custome
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747
11	PE	1652
12	PB	536

Question 4:

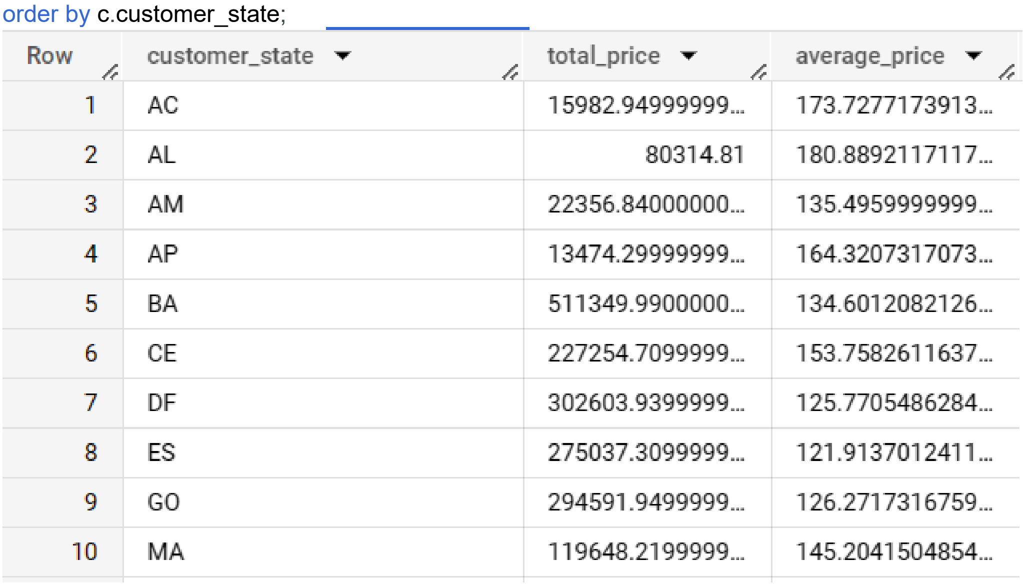
/*Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
You can use the "payment_value" column in the payments table to get the cost of orders.
Calculate the Total & Average value of order price for each state.
Calculate the Total & Average value of order freight for each state.*/

```
with cte1 as(
  select extract(year from o.order_purchase_timestamp) as year, extract(month from o.
order_purchase_timestamp) as month, p.payment_value
  from `Business_Case_Target_SQL.orders` as o
  inner join
  `Business_Case_Target_SQL.payments` as p
  on o.order_id=p.order_id
  where extract(year from o.order_purchase_timestamp) between 2017 and 2018 and
  extract(month from o. order_purchase_timestamp) between 01 and 08
),
cte2 as (
  select cte1.year,sum(payment_value) as cost
  from cte1
  group by cte1.year
)
select *,
  (cost- lag(cost, 1)over(order by cte2.year))*100/lag(cost, 1)over(order by cte2.year) as Percentage_increase
from cte2
order by cte2.year;
```



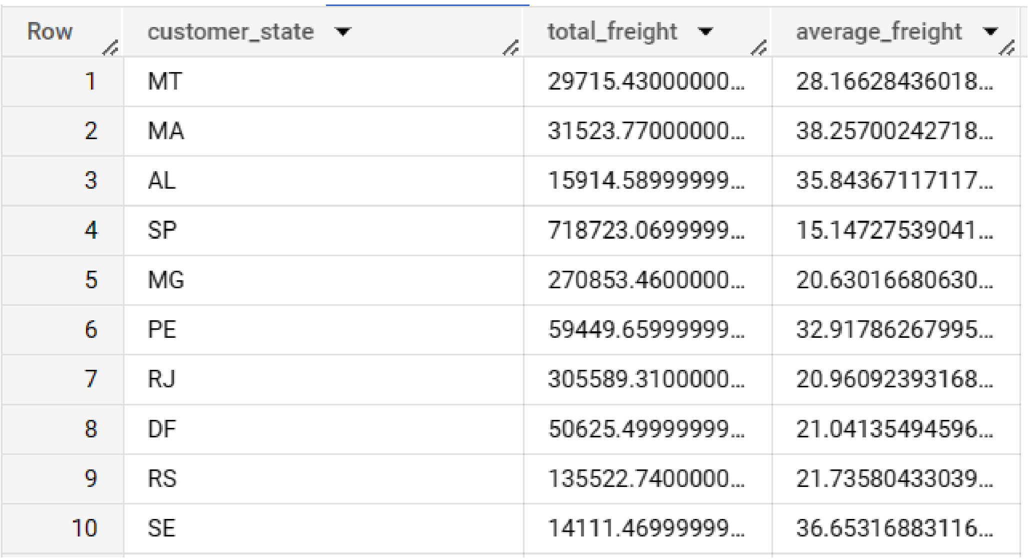
Row	year	cost	Percentage_increase
1	2017	3669022.119999...	null
2	2018	8694733.839999...	136.9768716466...

```
select c.customer_state, sum(oi.price) as total_price,avg(oi.price) as average_price
from `Business_Case_Target_SQL.customers` as c
inner join
`Business_Case_Target_SQL.orders` as o
on c.customer_id=o.customer_id
inner join
`Business_Case_Target_SQL.order_items` as oi
on o.order_id= oi.order_id
group by c.customer_state
order by c.customer_state;
```



Row	customer_state	total_price	average_price
1	AC	15982.94999999...	173.7277173913...
2	AL	80314.81	180.8892117117...
3	AM	22356.84000000...	135.4959999999...
4	AP	13474.29999999...	164.3207317073...
5	BA	511349.9900000...	134.6012082126...
6	CE	227254.7099999...	153.7582611637...
7	DF	302603.9399999...	125.7705486284...
8	ES	275037.3099999...	121.9137012411...
9	GO	294591.9499999...	126.2717316759...
10	MA	119648.2199999...	145.2041504854...

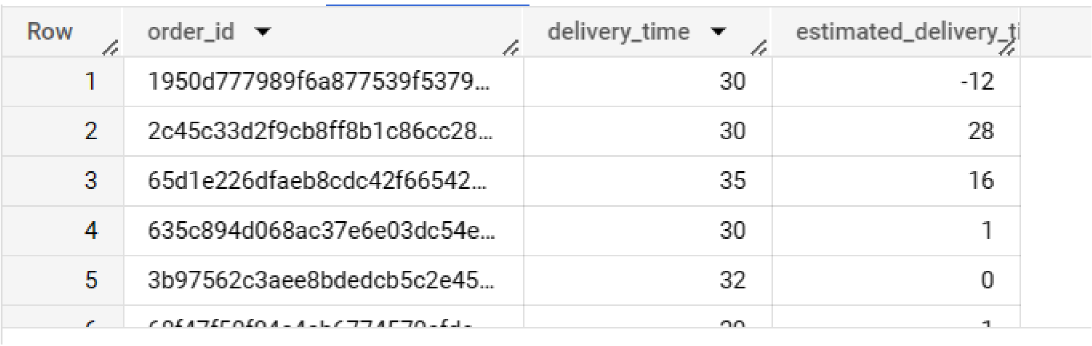
```
select c.customer_state, sum(oi.freight_value) as total_freight,avg(oi.freight_value) as average_freight
from `Business_Case_Target_SQL.customers` as c
inner join
`Business_Case_Target_SQL.orders` as o
on c.customer_id=o.customer_id
inner join
`Business_Case_Target_SQL.order_items` as oi
on o.order_id= oi.order_id
group by c.customer_state;
```



Row	customer_state	total_freight	average_freight
1	MT	29715.43000000...	28.16628436018...
2	MA	31523.77000000...	38.25700242718...
3	AL	15914.58999999...	35.84367117117...
4	SP	718723.0699999...	15.14727539041...
5	MG	270853.4600000...	20.63016680630...
6	PE	59449.65999999...	32.91786267995...
7	RJ	305589.3100000...	20.96092393168...
8	DF	50625.49999999...	21.04135494596...
9	RS	135522.7400000...	21.73580433039...
10	SE	14111.46999999...	36.65316883116...

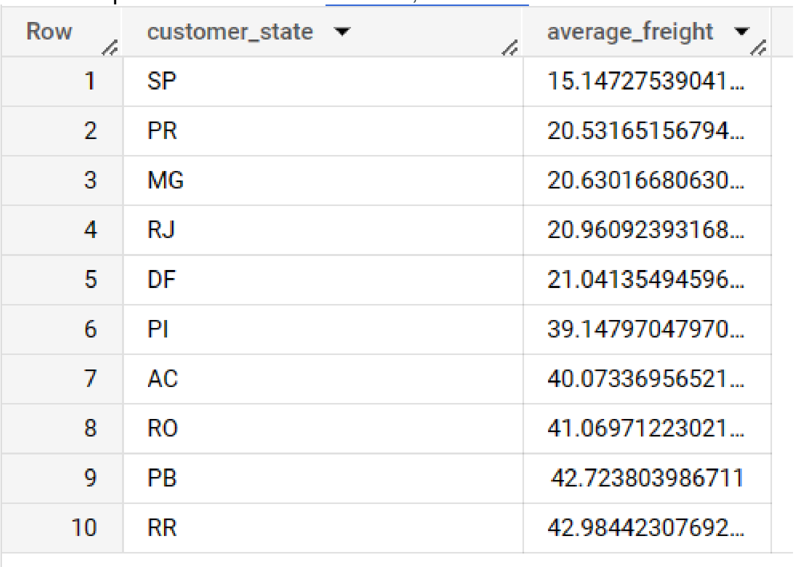
Question 5:

```
/*Analysis based on sales, freight and delivery time.
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
Find out the top 5 states with the highest & lowest average freight value.
Find out the top 5 states with the highest & lowest average delivery time.
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.*/

select order_id, date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as delivery_time,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as estimated_delivery_time
from `Business_Case_Target_SQL.orders`;
```



Row	order_id	delivery_time	estimated_delivery_ti
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	[illegible]	30	1

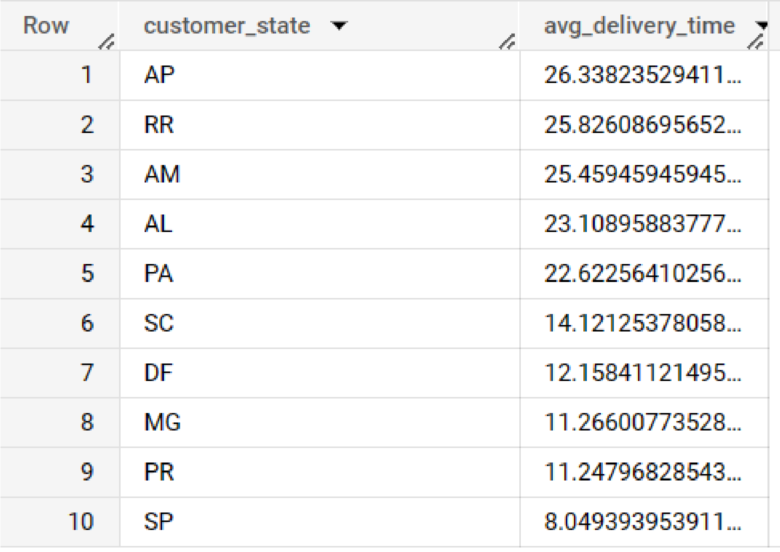
```
select a.customer_state,a.average_freight
from
(select c.customer_state,sum(oi.freight_value)/count(oi.order_id) as average_freight, row_number()over(order by sum(oi.freight_value)/count(oi.order_id) desc) as topfive,
row_number()over(order by sum(oi.freight_value)/count(oi.order_id)) as bottomfive
from `Business_Case_Target_SQL.order_items` as oi
inner join
`Business_Case_Target_SQL.orders` as o
on oi.order_id=o.order_id
inner join
`Business_Case_Target_SQL.customers` as c
on o.customer_id= c.customer_id
group by c.customer_state) as a
where a.topfive<=5 or a.bottomfive<=5;
```



Row	customer_state	average_freight
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...
6	PI	39.14797047970...
7	AC	40.07336956521...
8	RO	41.06971223021...
9	PB	42.723803986711
10	RR	42.98442307692...

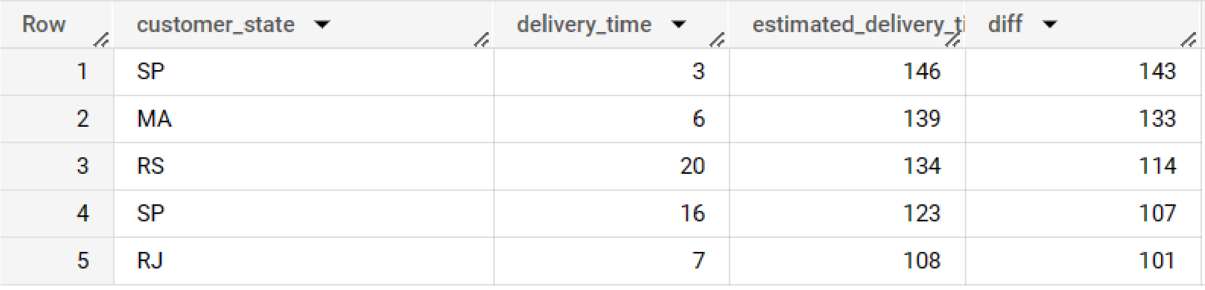
```
select a.customer_state,a.avg_delivery_time
from
(select c.customer_state, sum(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))/count(order_id) as avg_delivery_time,
row_number()over(order by sum(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))/count(order_id)) as topfive,
row_number()over(order by sum(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))/count(order_id) desc) as bottomfive
from `Business_Case_Target_SQL.orders` as o
inner join
`Business_Case_Target_SQL.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state) as a
```

```
where a.topfive<=5 or a.bottomfive<=5;
```



Row	customer_state	avg_delivery_time
1	AP	26.33823529411...
2	RR	25.82608695652...
3	AM	25.45945945945...
4	AL	23.10895883777...
5	PA	22.62256410256...
6	SC	14.12125378058...
7	DF	12.15841121495...
8	MG	11.26600773528...
9	PR	11.24796828543...
10	SP	8.049393953911...

```
select *, (a.estimated_delivery_time-a.delivery_time) as diff
from
(select c.customer_state, date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
delivery_time, date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as
estimated_delivery_time
from `Business_Case_Target_SQL.orders` as o
inner join
`Business_Case_Target_SQL.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state,order_delivered_customer_date,
order_purchase_timestamp,order_estimated_delivery_date) as a
where a.delivery_time<a.estimated_delivery_time
order by diff desc
limit 5;
```

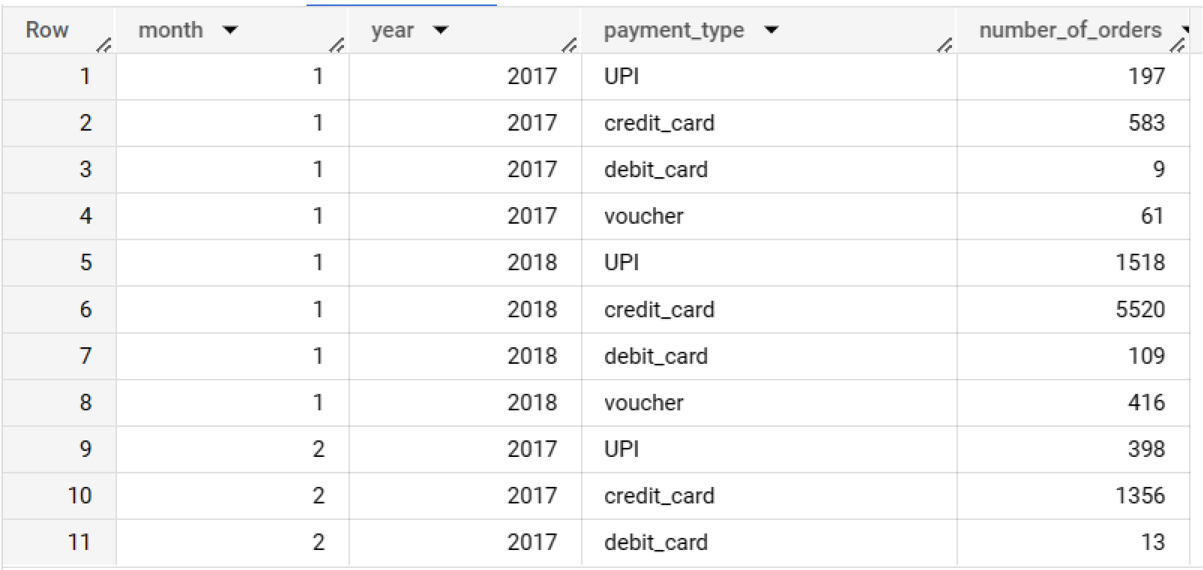


Row	customer_state	delivery_time	estimated_delivery_ti	diff
1	SP	3	146	143
2	MA	6	139	133
3	RS	20	134	114
4	SP	16	123	107
5	RJ	7	108	101

Question 6:

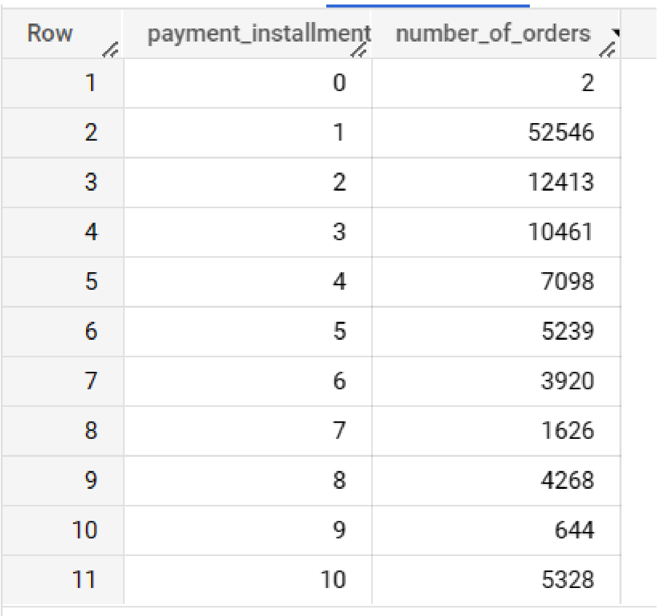
```
/*Analysis based on the payments:
Find the month on month no. of orders placed using different payment types.
Find the no. of orders placed on the basis of the payment installments that have been paid.*/

select extract(month from o.order_purchase_timestamp) as month,extract(year from
o.order_purchase_timestamp) as year, p.payment_type, count(o.order_id) as number_of_orders
from `Business_Case_Target_SQL.payments` as p
inner join
`Business_Case_Target_SQL.orders` as o
```

```
on p.order_id=o.order_id
group by month,year, p.payment_type
order by month, year, p.payment_type;
```



Row	month	year	payment_type	number_of_orders
1	1	2017	UPI	197
2	1	2017	credit_card	583
3	1	2017	debit_card	9
4	1	2017	voucher	61
5	1	2018	UPI	1518
6	1	2018	credit_card	5520
7	1	2018	debit_card	109
8	1	2018	voucher	416
9	2	2017	UPI	398
10	2	2017	credit_card	1356
11	2	2017	debit_card	13

```
select payment_installments, count(order_id) as number_of_orders
from `Business_Case_Target_SQL.payments`
group by payment_installments
order by payment_installments;
```



Row	payment_installment	number_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