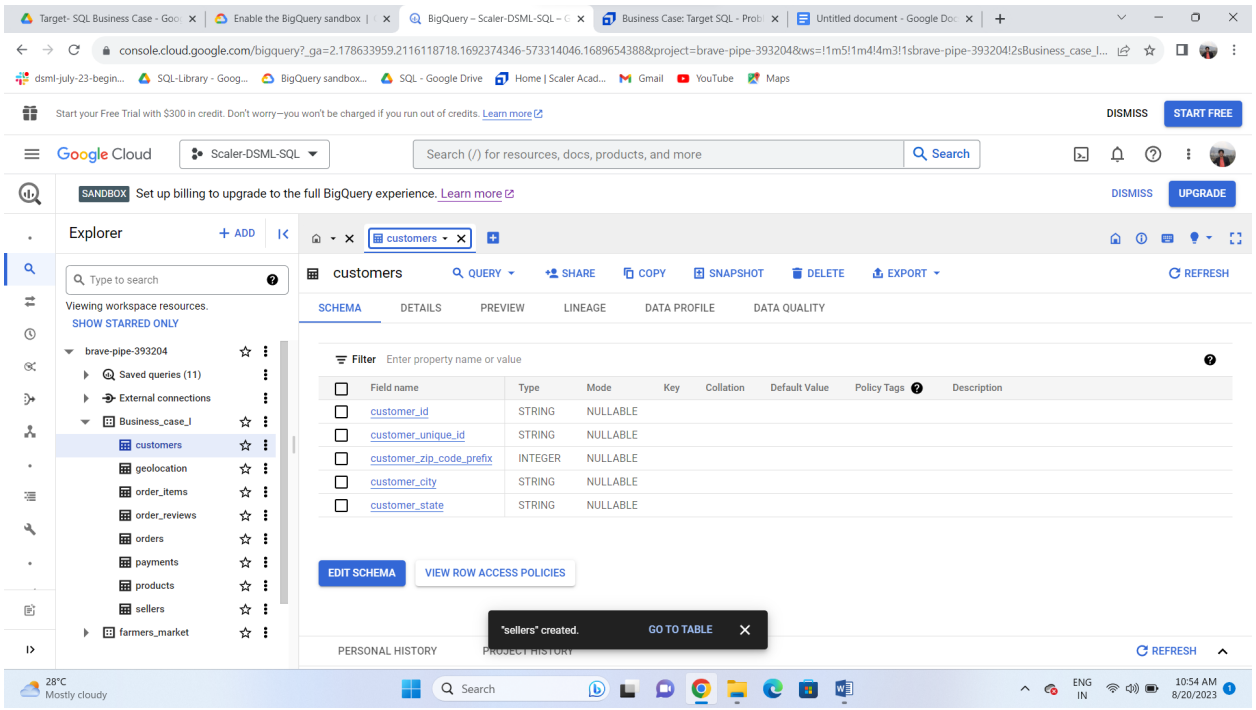
1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

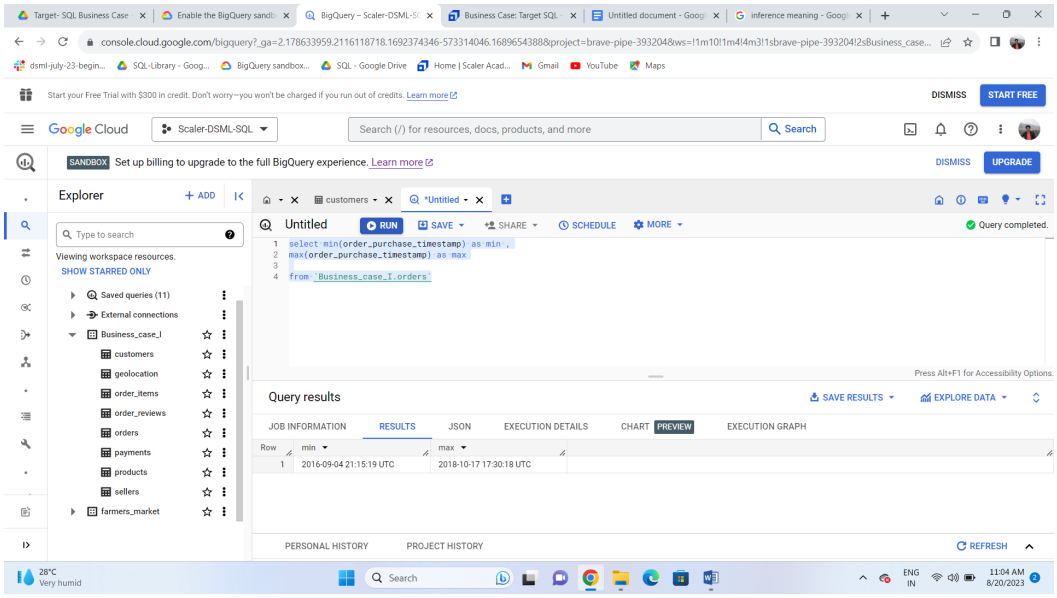
1. Data type of all columns in the "customers" table.



Inference - Data types of customers table are as follows
Customer_id - String
Customer_unique_id- String
Customer_zip_code_prefix-Integer
Customer_city- String
customer_state- String

2.Get the time range between which the orders were

```
select min(order_purchase_timestamp) as min ,
max(order_purchase_timestamp) as max

from `Business_case_I.orders`
```

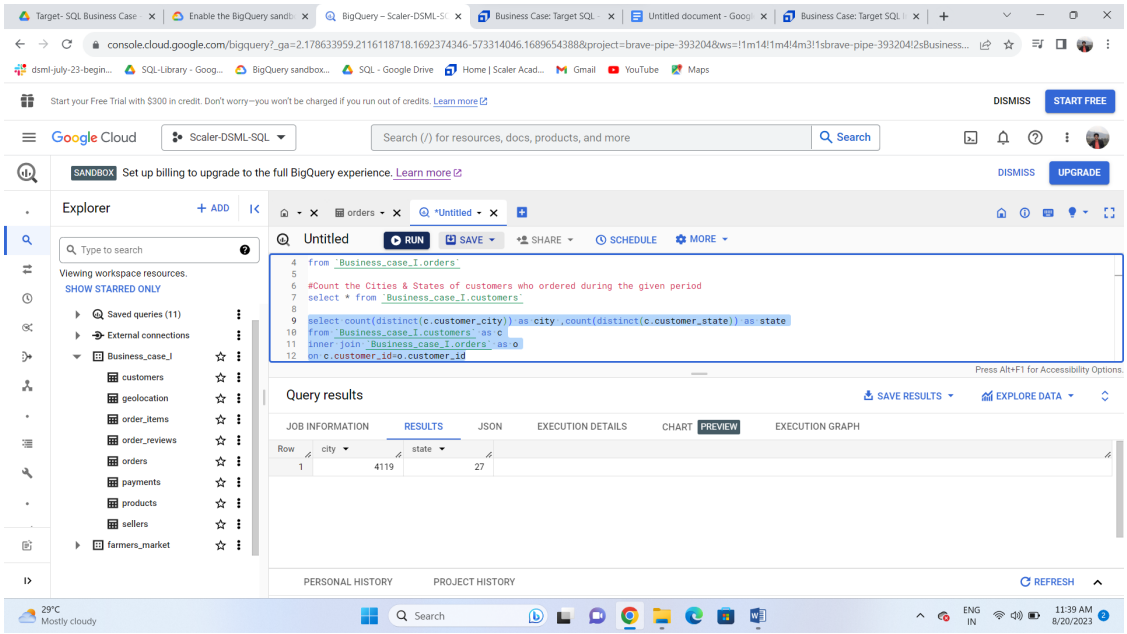


placed

Inference- orders when started getting placed are from 2016-09-04 21:15:19 UTC to 2018-10-17 17:30:18 UTC

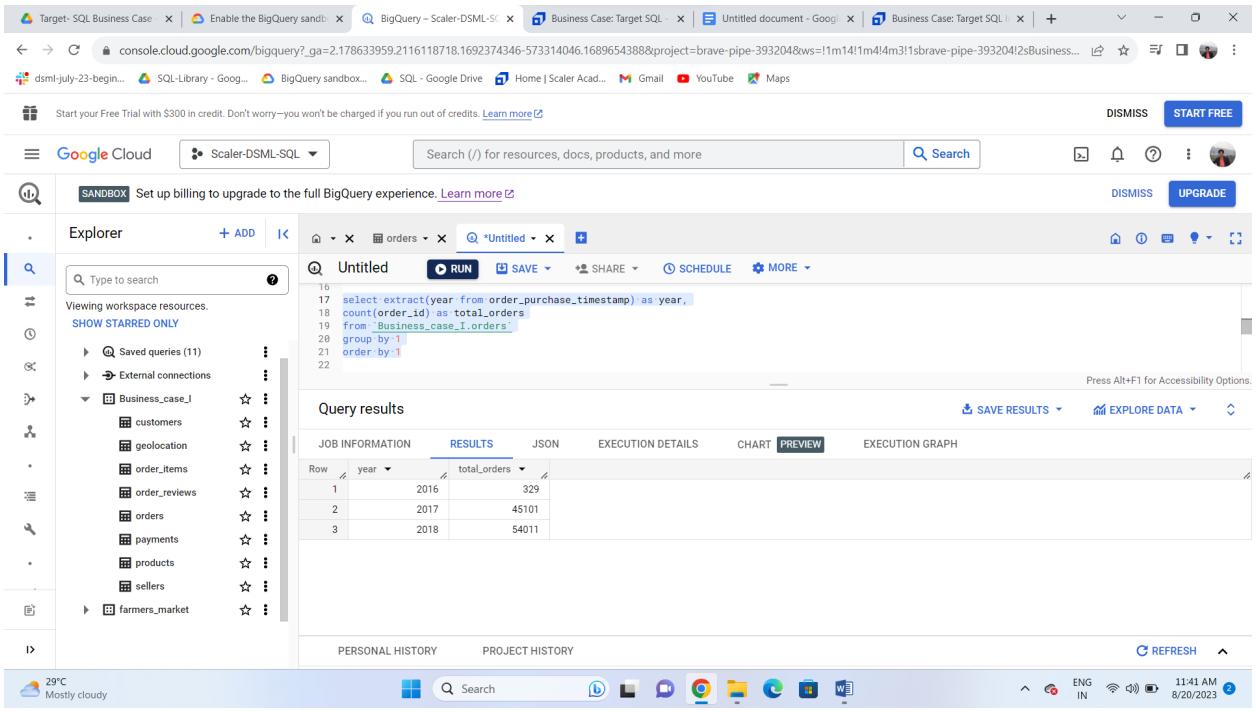
3.Count the Cities & States of customers who ordered during the given period

```
select count(distinct(c.customer_city)) as city ,count(distinct(c.customer_state)) as state
from `Business_case_I.customers` as c
inner join `Business_case_I.orders` as o
on c.customer_id=o.customer_id
```



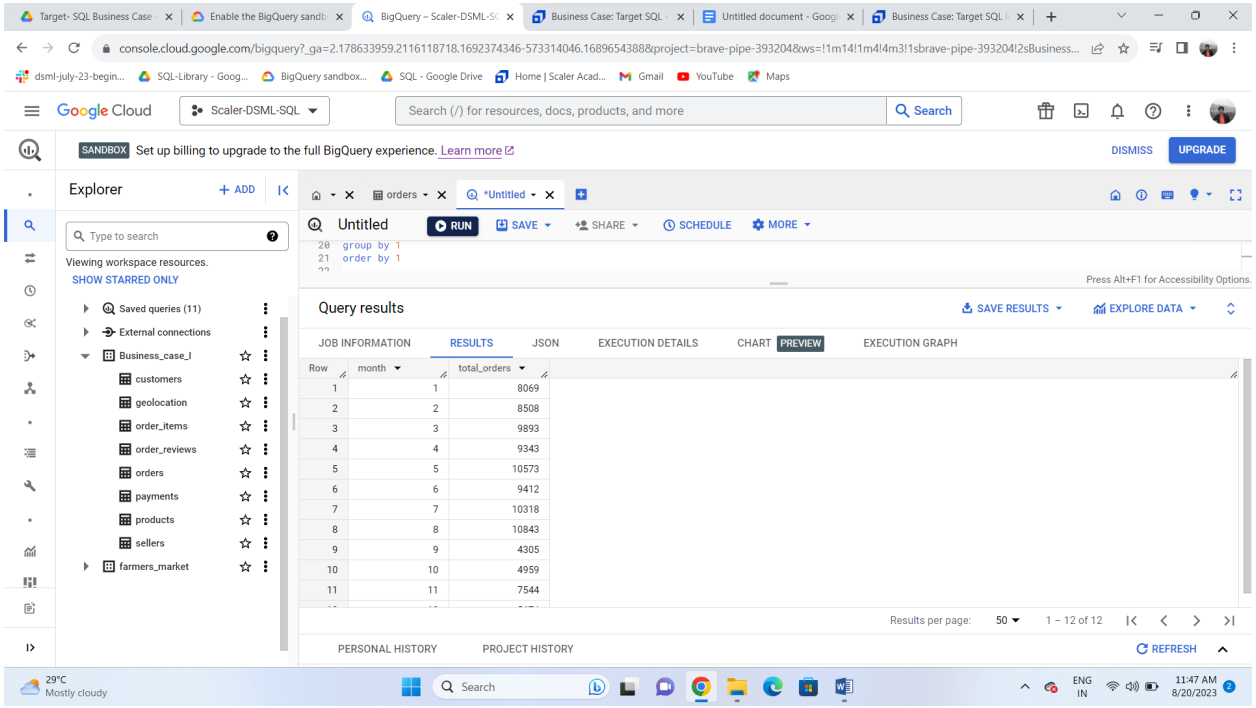
1. Is there a growing trend in the no. of orders placed over the past years?

```
select extract(year from order_purchase_timestamp) as year,
count(order_id) as total_orders
from `Business_case_I.orders`
group by 1
order by 1
```



Inference- As you can observe that is there is growing trend in the no. of orders from 2016 to 2018

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

select extract(month from order_purchase_timestamp) as month,
count(order_id) as total_orders
from `Business_case_I.orders`
group by 1
order by 1
```



Inference- according to monthly seasonality no. of orders being placed in the month of may,july and august are maximum

2.In-depth Exploration:

During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

0-6 hrs : Dawn

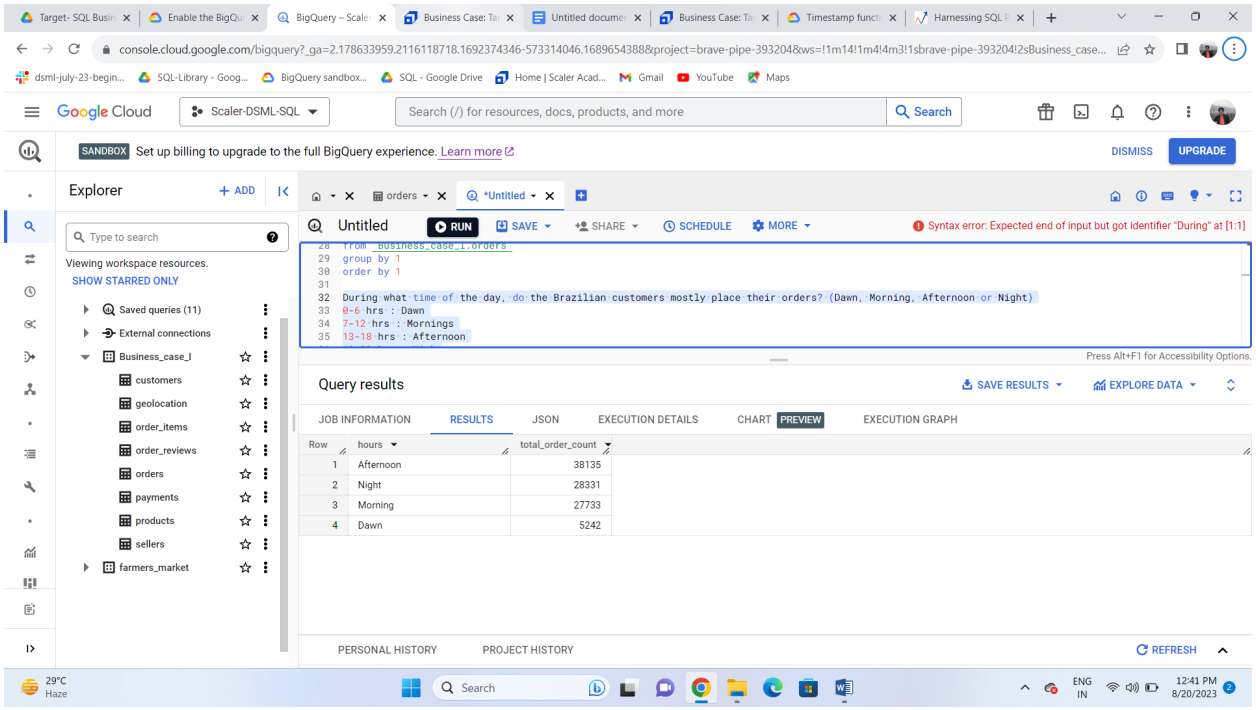
7-12 hrs : Mornings

13-18 hrs : Afternoon

19-23 hrs : Night

```
select
  case
```

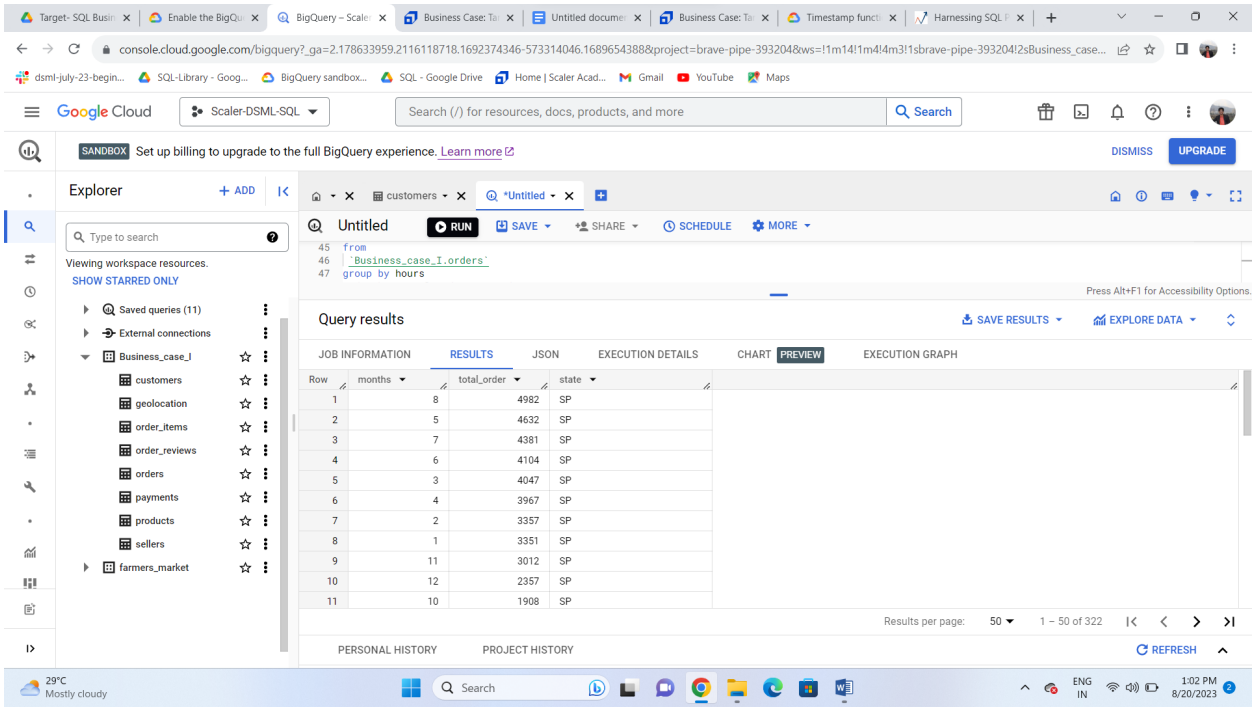
```
    when extract(HOUR FROM order_purchase_timestamp) between 0 and 6 THEN 'Dawn'
     when  extract(HOUR FROM order_purchase_timestamp) between 7 and 12 THEN 'Morning'
     when  extract(HOUR FROM order_purchase_timestamp) between 13 and 18 THEN
'Afternoon'
     when  extract(HOUR FROM order_purchase_timestamp) between 19 and 23 THEN 'Night'
  end as hours,
  COUNT(order_id) AS total_order_count
from
 `Business_case_I.orders`
group by hours
order by  total_order_count desc
```



Inference-maximum orders were placed in afternoon followed by night,morning and dawn

1. Get the month on month no. of orders placed in each state.

```
select extract(month from o.order_purchase_timestamp) as months,
count(o.order_id) as total_order,
(c.customer_state) as state
from `Business_case_I.orders` as o
inner join `Business_case_I.customers` as c
on c.customer_id= o.customer_id
group by c.customer_state,months
order by total_order desc
```

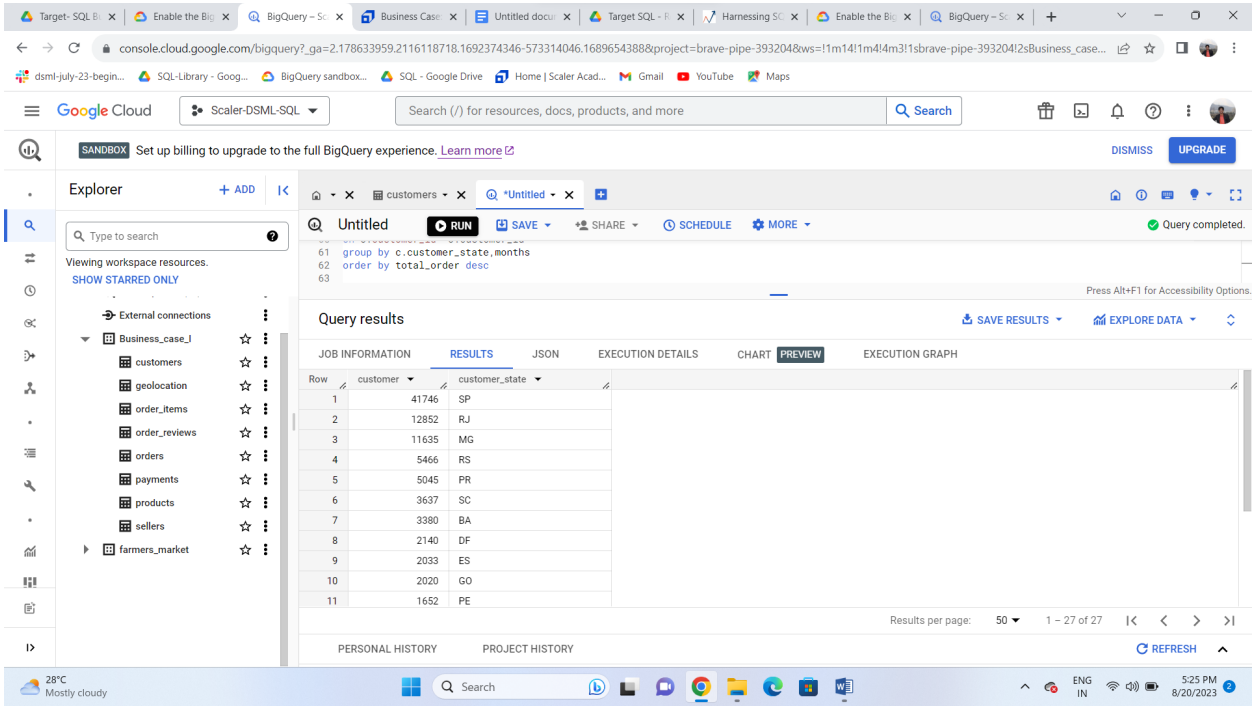


Row	months	total_order	state
1	8	4982	SP
2	5	4632	SP
3	7	4381	SP
4	6	4104	SP
5	3	4047	SP
6	4	3967	SP
7	2	3357	SP
8	1	3351	SP
9	11	3012	SP
10	12	2357	SP
11	10	1908	SP

Inference-As we can observed SP state has highest no. of orders in the month of august

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

```
select count(customer_id) as customer,customer_state
from `Business_case_I.customers`
group by 2
order by customer desc
```



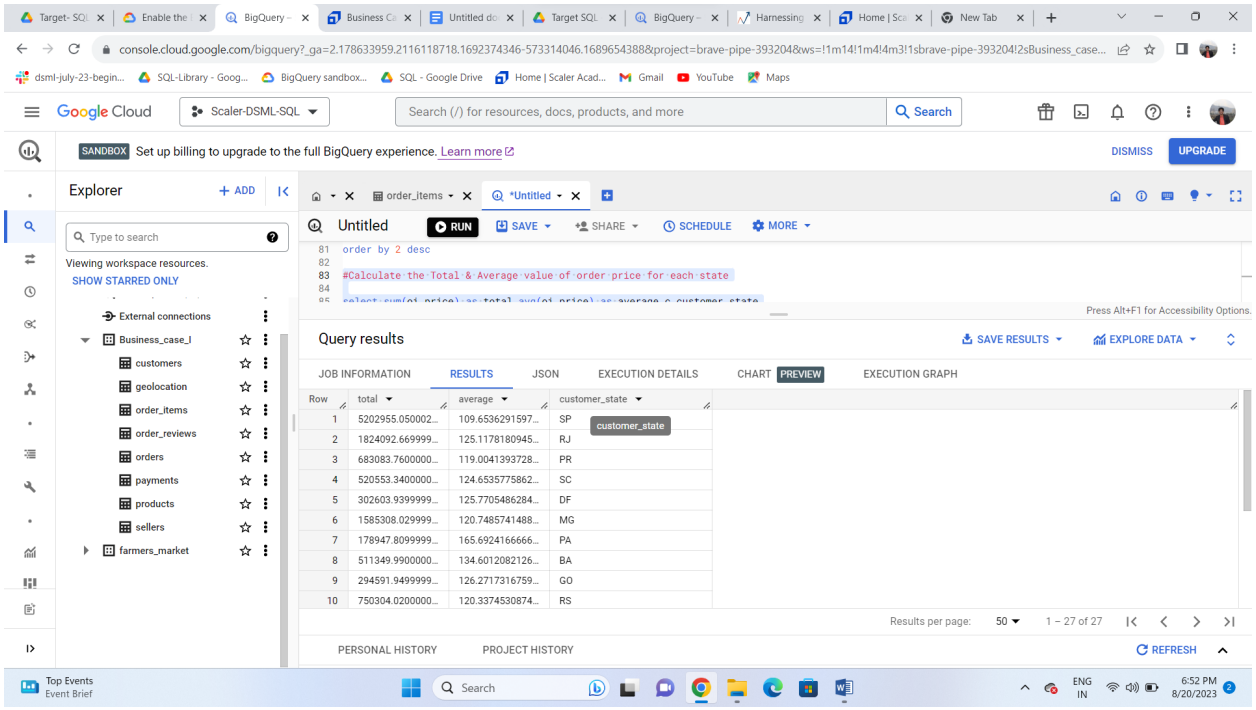
Inference- data show us that SP state has highest no. of customers followed by RJ,MG, RS etc

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).

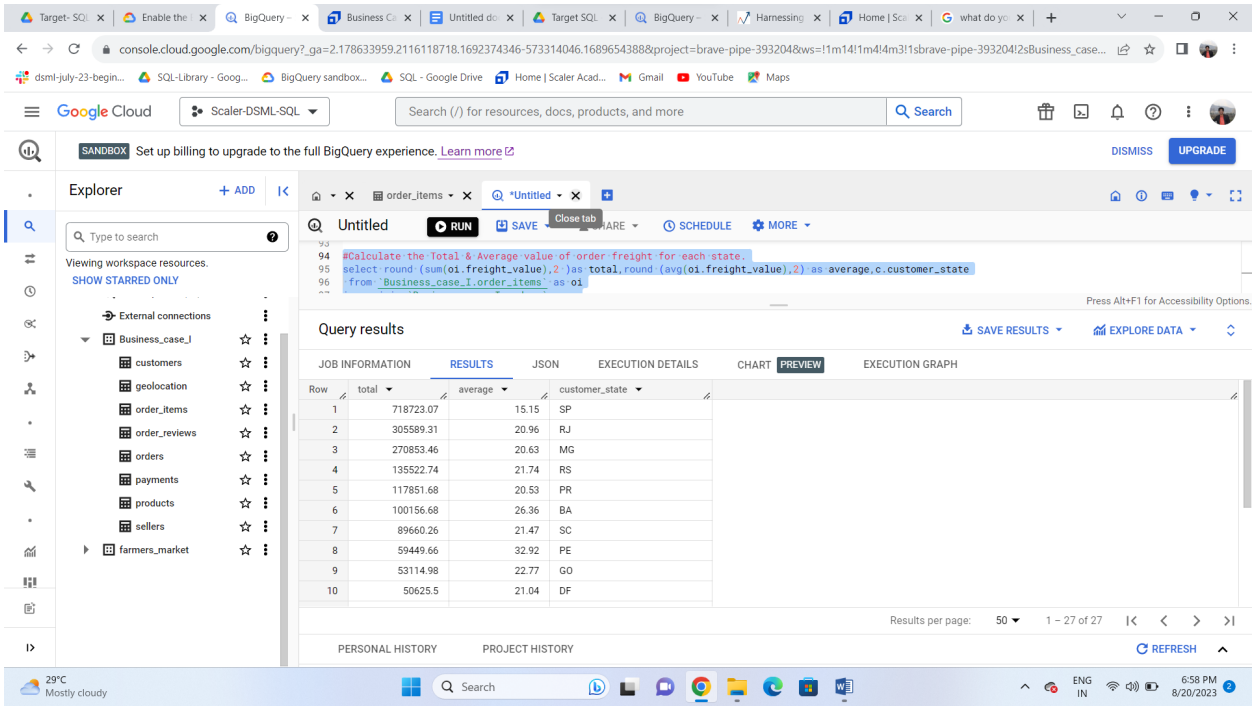
2. Calculate the Total & Average value of order price for each state

```
select sum(oi.price) as total,avg(oi.price) as average,c.customer_state
 from `Business_case_I.order_items` as oi
inner join `Business_case_I.orders` as o
on oi.order_id=o.order_id
inner join `Business_case_I.customers` as c
on o.customer_id=c.customer_id
group by 3
```



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

```
select round (sum(oi.freight_value),2 )as total,round (avg(oi.freight_value),2) as
average,c.customer_state
 from `Business_case_I.order_items` as oi
inner join `Business_case_I.orders` as o
on oi.order_id=o.order_id
inner join `Business_case_I.customers` as c
on o.customer_id=c.customer_id
group by 3
order by total desc
```



Inference - from output we can understand that state SP has highest no. of total freight value

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.

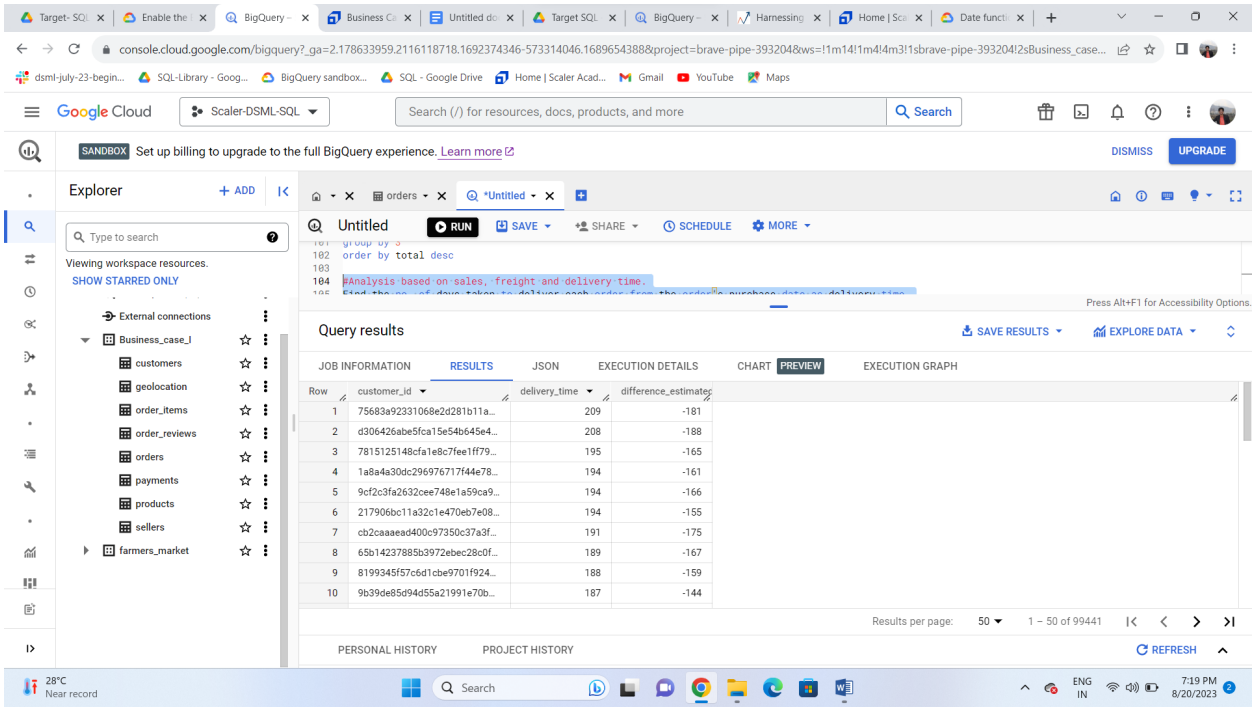
Do this in a single query.

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

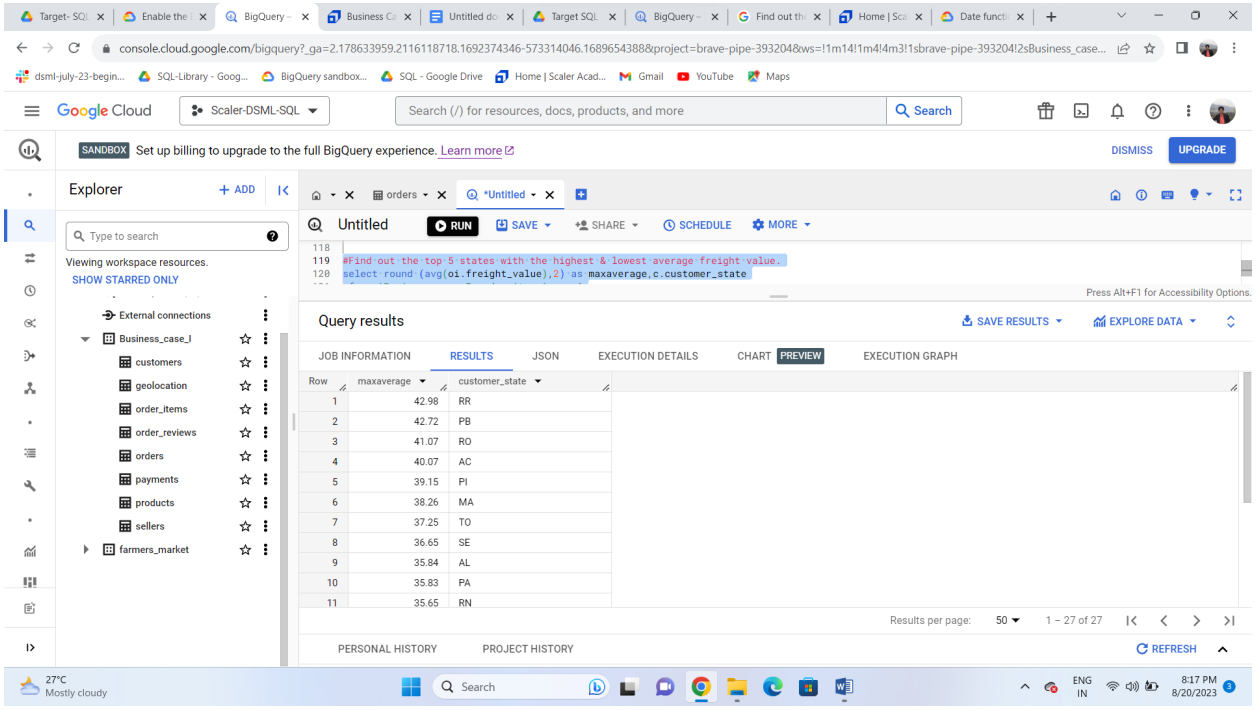
```
select customer_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS delivery_time,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS difference_estimated_delivery
 from `Business_case_I.orders`
 order by delivery_time desc
```



Inference - highest delivery time required to customer was 209 days and we need to improve in our delivery time to customers

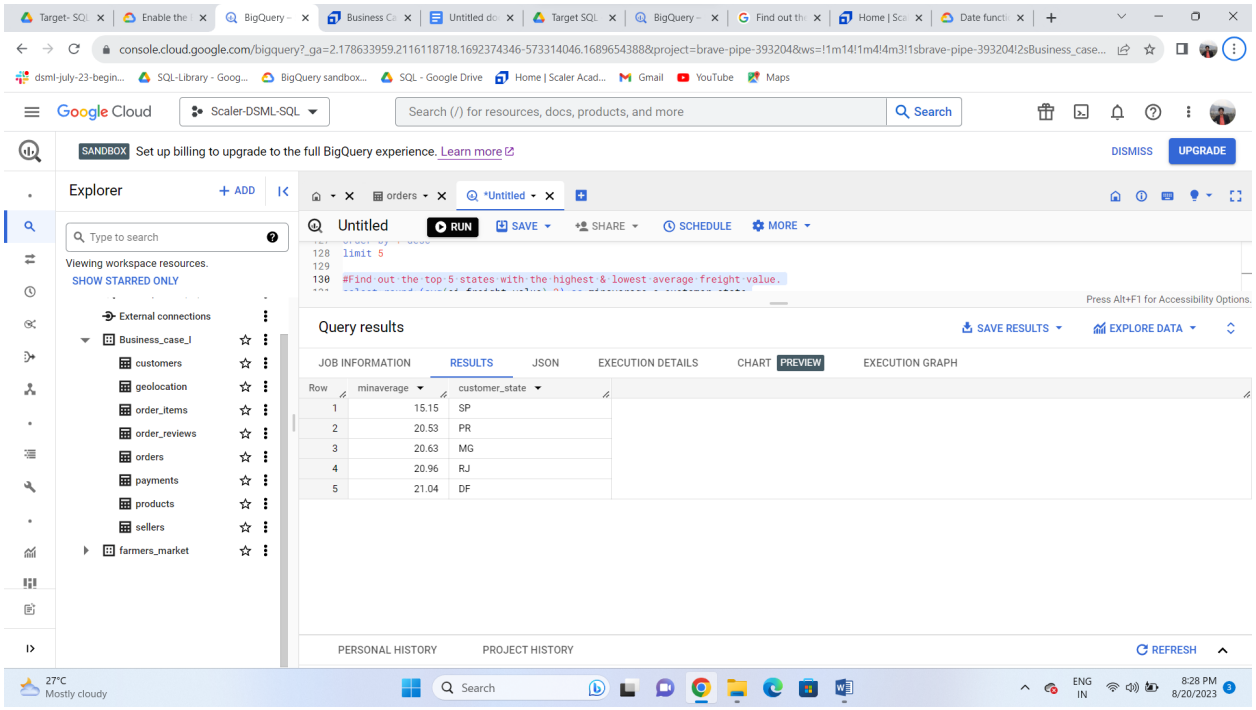
2.Find out the top 5 states with the highest & lowest average freight value.

```sql
select round (avg(oi.freight_value),2) as maxaverage,c.customer_state
 from `Business_case_I.order_items` as oi
inner join `Business_case_I.orders` as o
on oi.order_id=o.order_id
inner join `Business_case_I.customers` as c
on o.customer_id=c.customer_id
group by 2
order by 1 desc
```

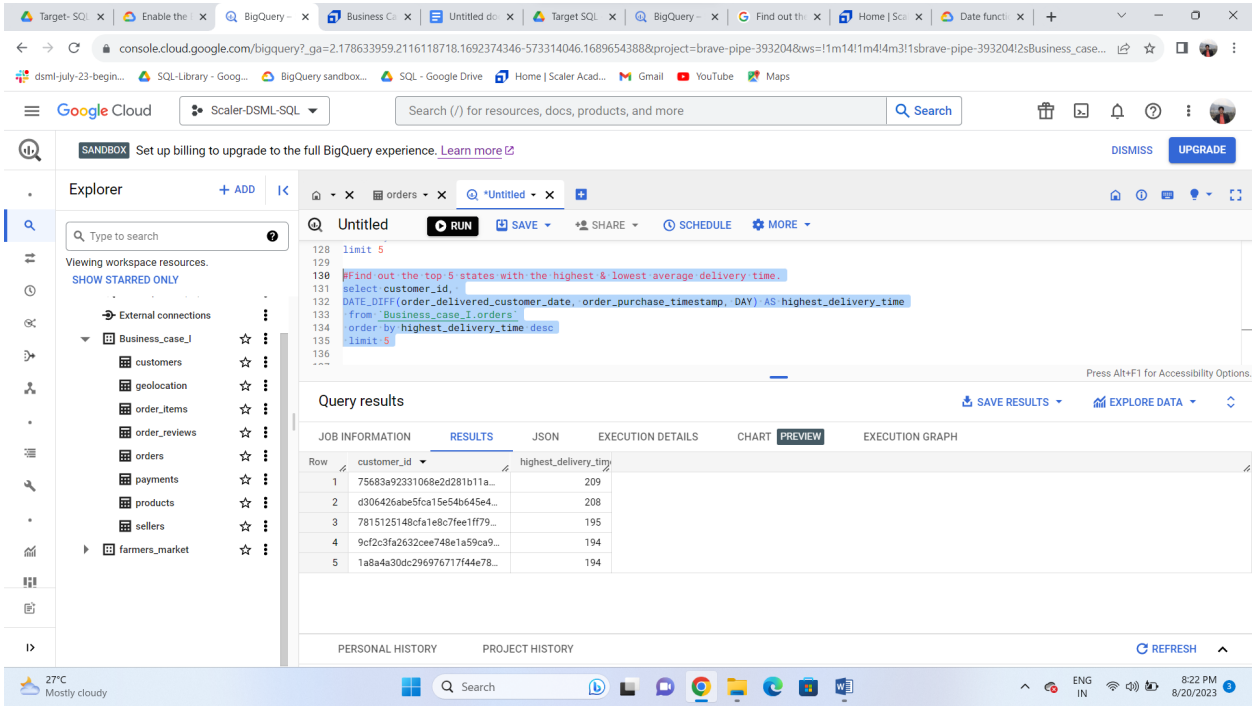


2.Find out the top 5 states with the highest & lowest average freight value.

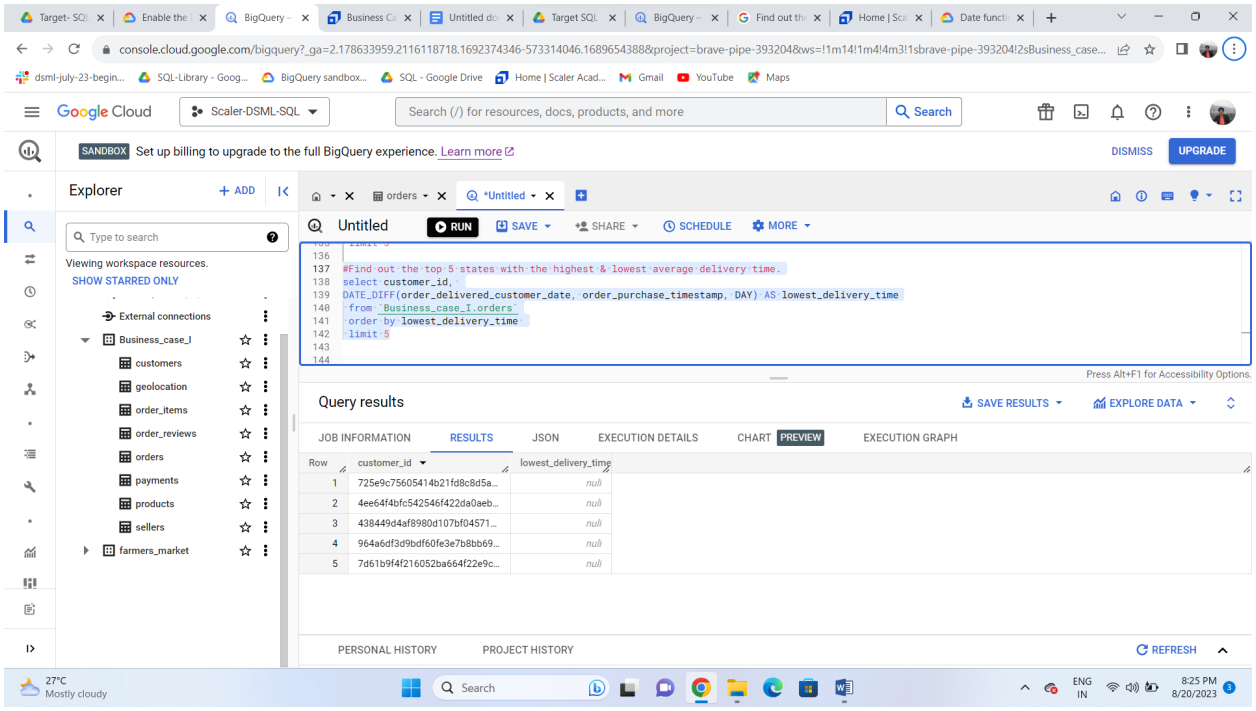
```sql
select round (avg(oi.freight_value),2) as minaverage,c.customer_state
 from `Business_case_I.order_items` as oi
inner join `Business_case_I.orders` as o
on oi.order_id=o.order_id
inner join `Business_case_I.customers` as c
on o.customer_id=c.customer_id
group by 2
order by 1
limit 5
```



```sql
#Find out the top 5 states with the highest & lowest average delivery time.
select customer_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
highest_delivery_time
 from `Business_case_I.orders`
 order by highest_delivery_time desc
 limit 5
```



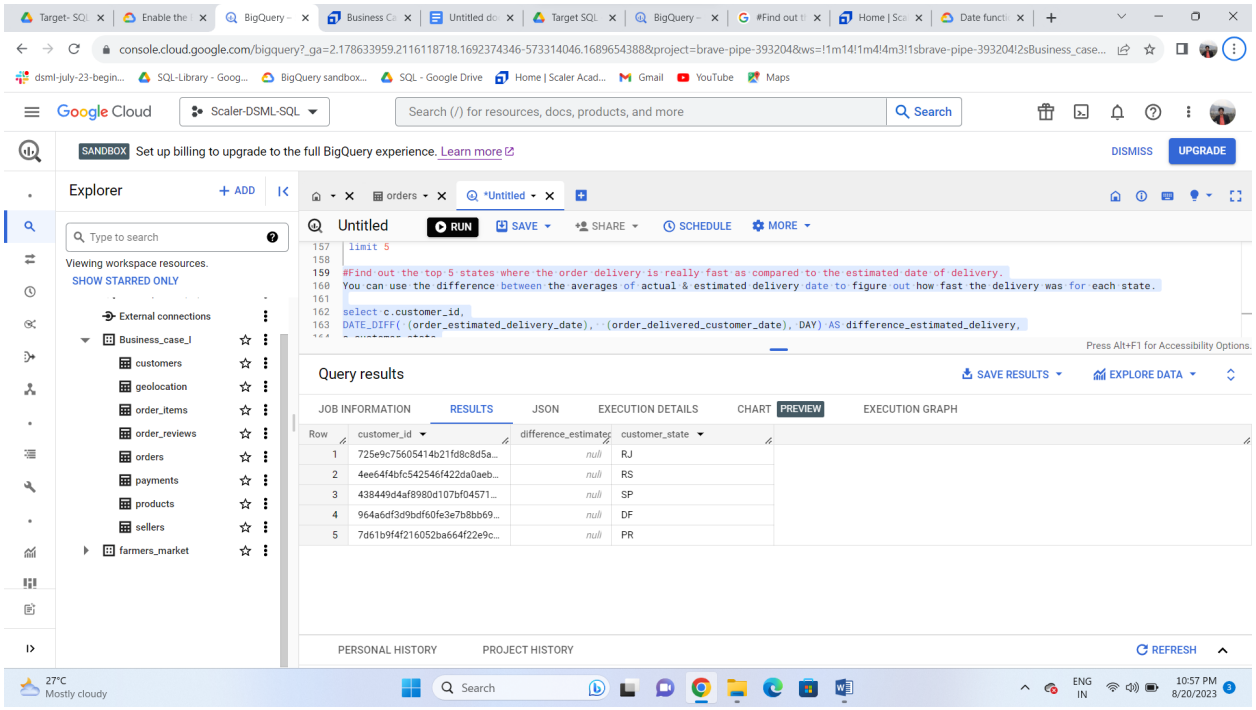
```sql
#Find out the top 5 states with the highest & lowest average delivery time.
select customer_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
lowest_delivery_time
 from `Business_case_I.orders`
 order by lowest_delivery_time
 limit 5
```



5.Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

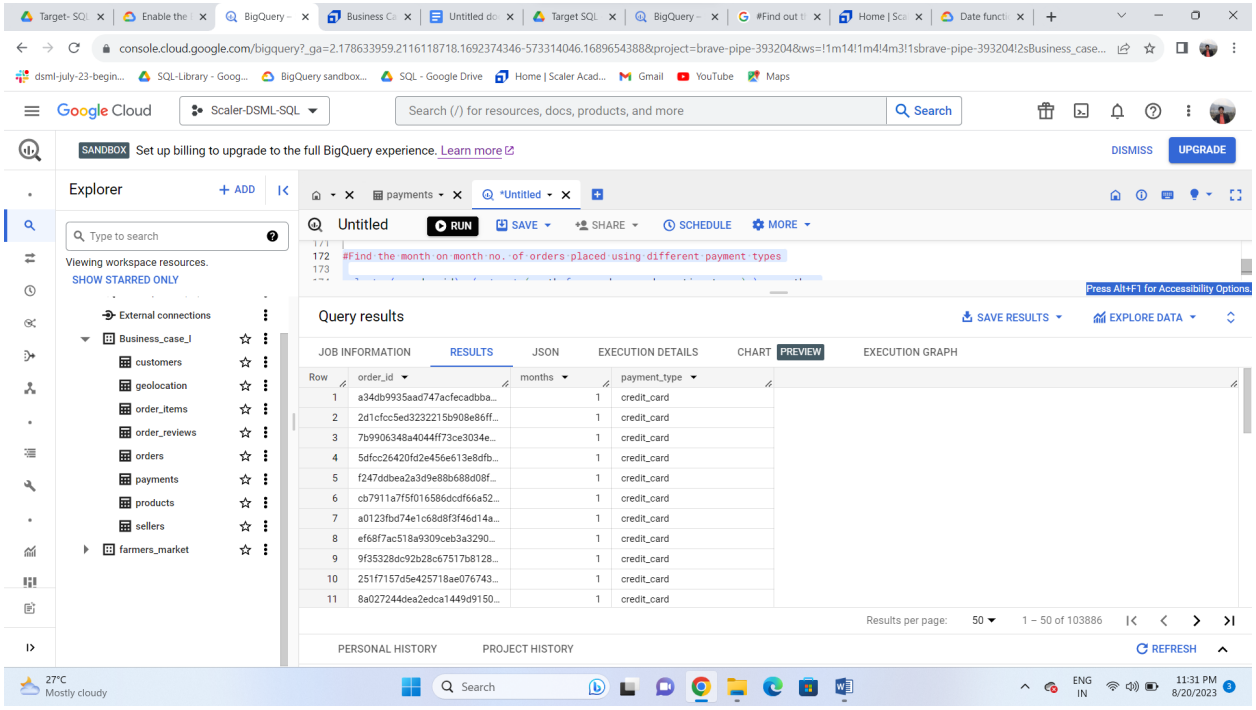
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
select c.customer_id,
DATE_DIFF( (order_estimated_delivery_date),  (order_delivered_customer_date), DAY) AS
difference_estimated_delivery,
c.customer_state
 from `Business_case_I.orders` as o
 inner join `Business_case_I.customers` as c
 on o.customer_id=c.customer_id
 order by difference_estimated_delivery

 limit 5
```



Inference

```
#Find the month on month no. of orders placed using different payment types

select  (o.order_id), (extract (month from order_purchase_timestamp ) )as months,
 payment_type
from `Business_case_I.orders` as o
inner join `Business_case_I.payments`as p
on o.order_id=p.order_id
order by extract (month from order_purchase_timestamp )
```



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

```
select * from `Business_case_I.payments`

select count(order_id) total_orders, payment_installments
from `Business_case_I.payments`
group by payment_installments
order by 2 desc
```