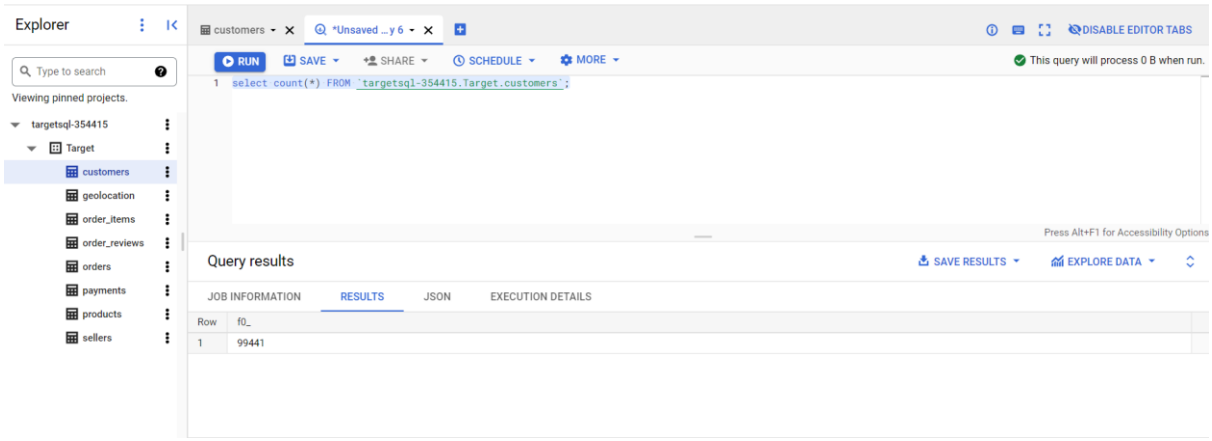
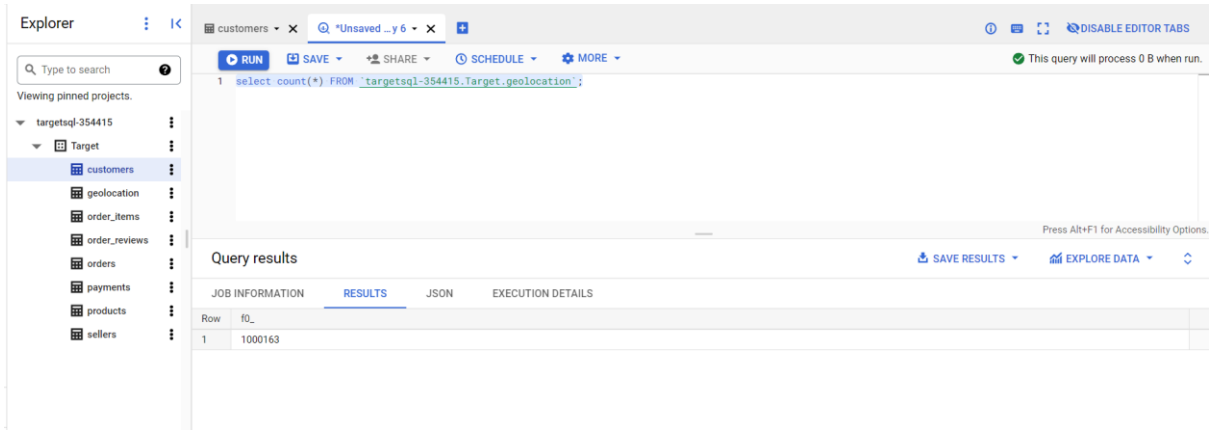
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

A. Get number of rows in the data

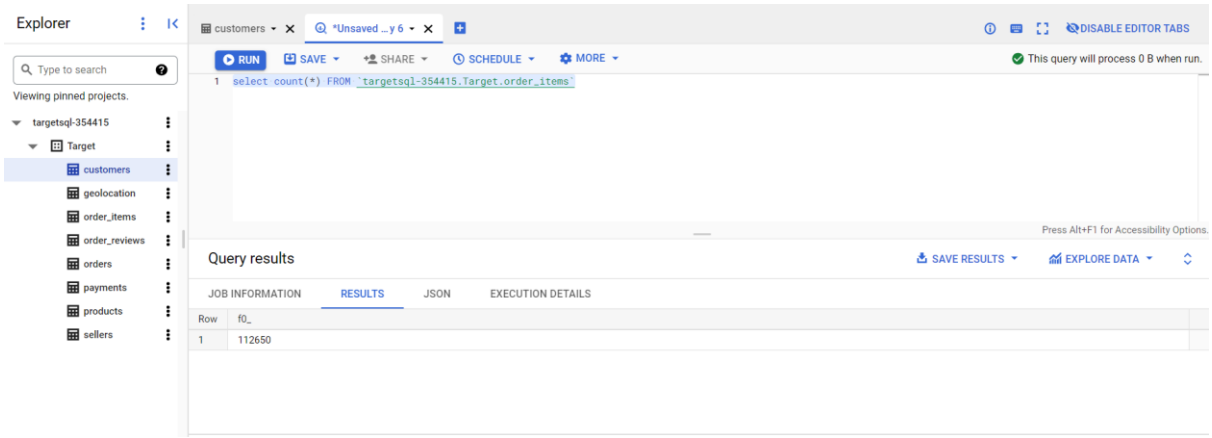
- Customers



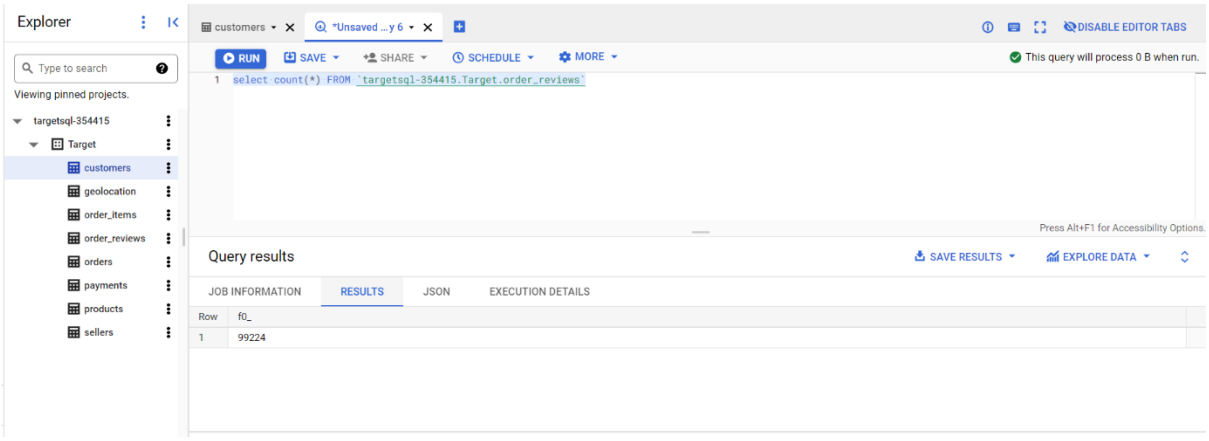
- Geolocations



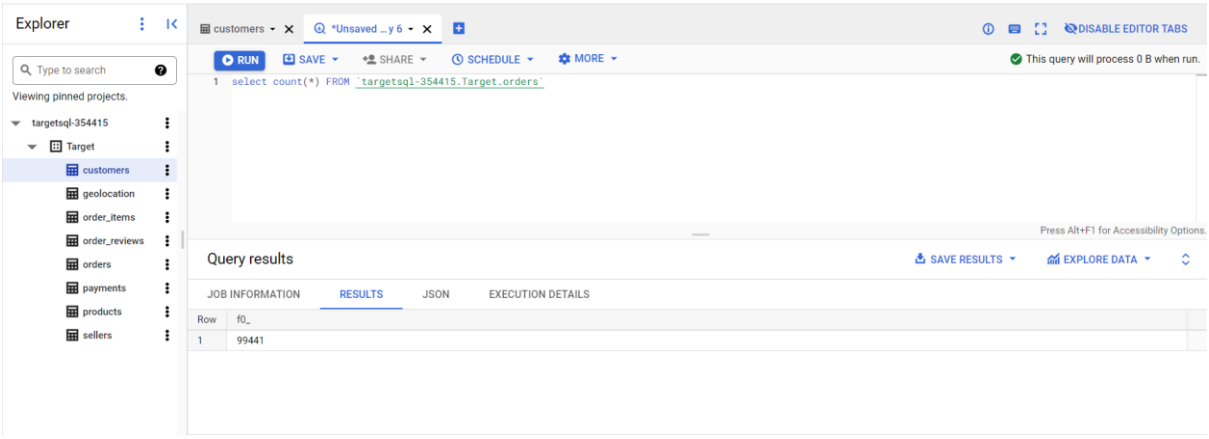
- Order_items



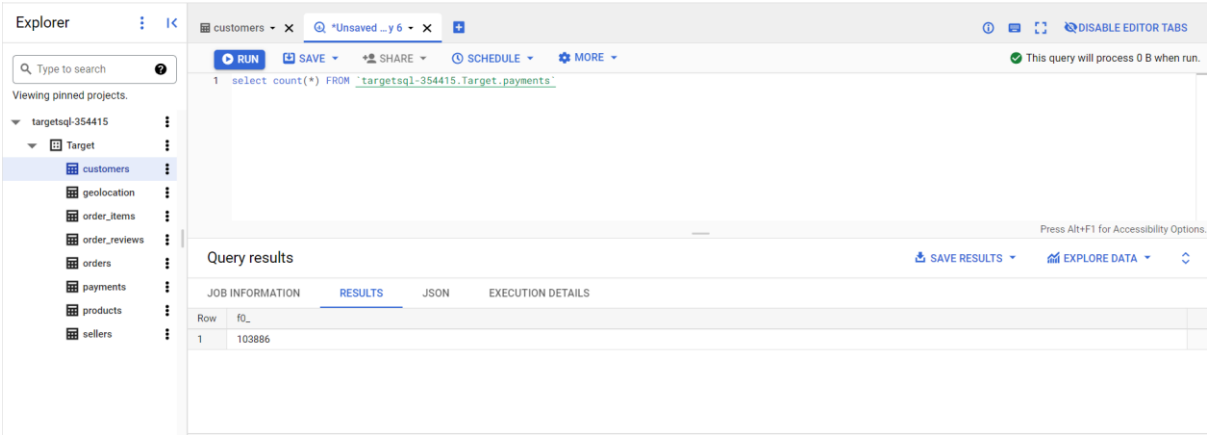
- Order_reviews



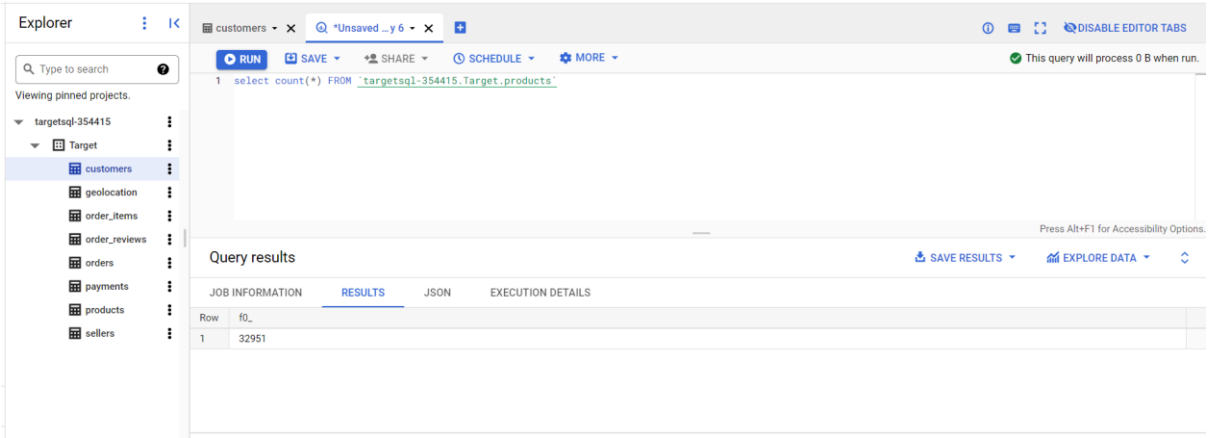
- orders



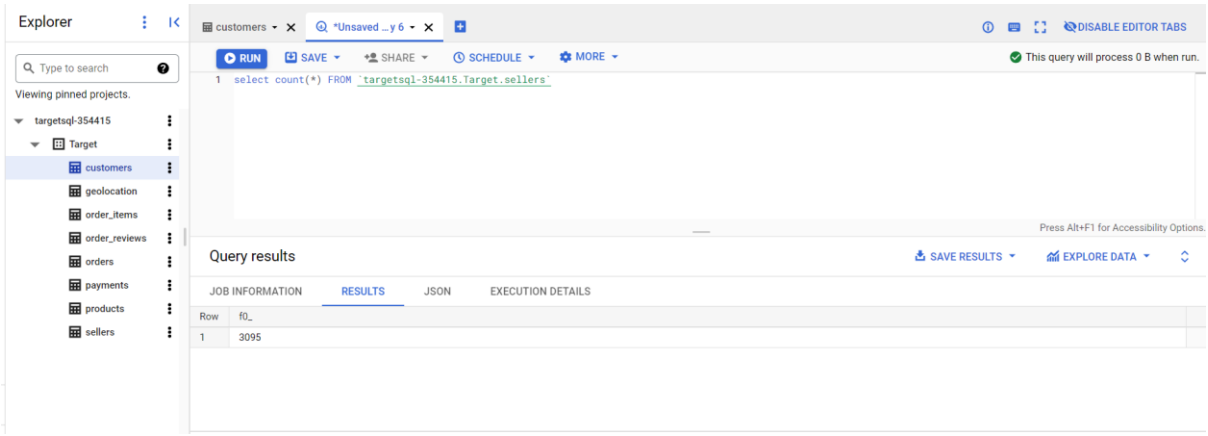
- payments



- products

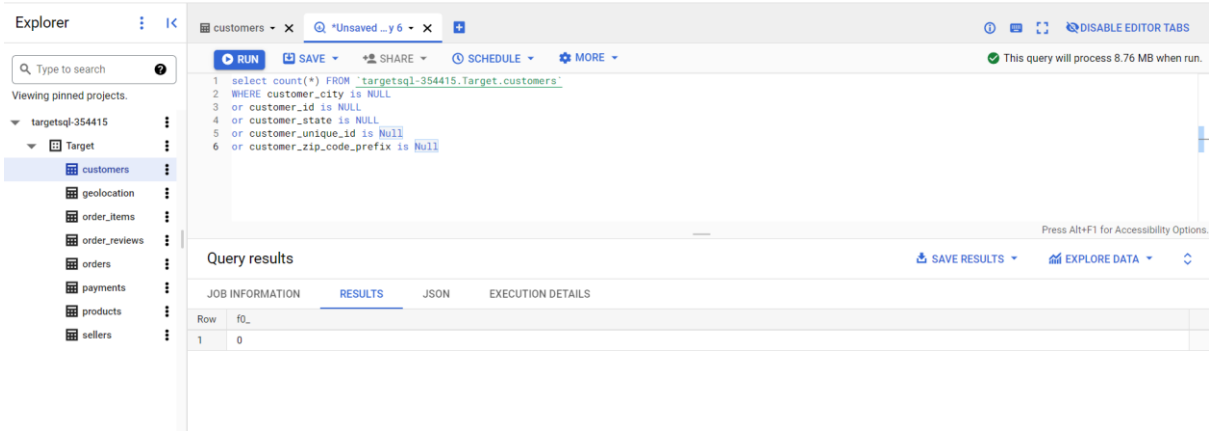


- sellers

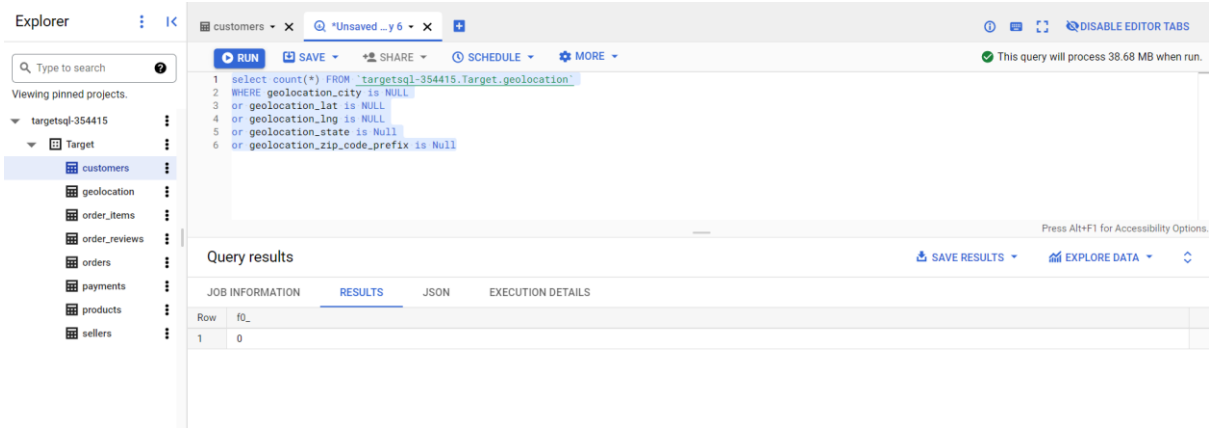


B. Number of Null or Missing values in columns.

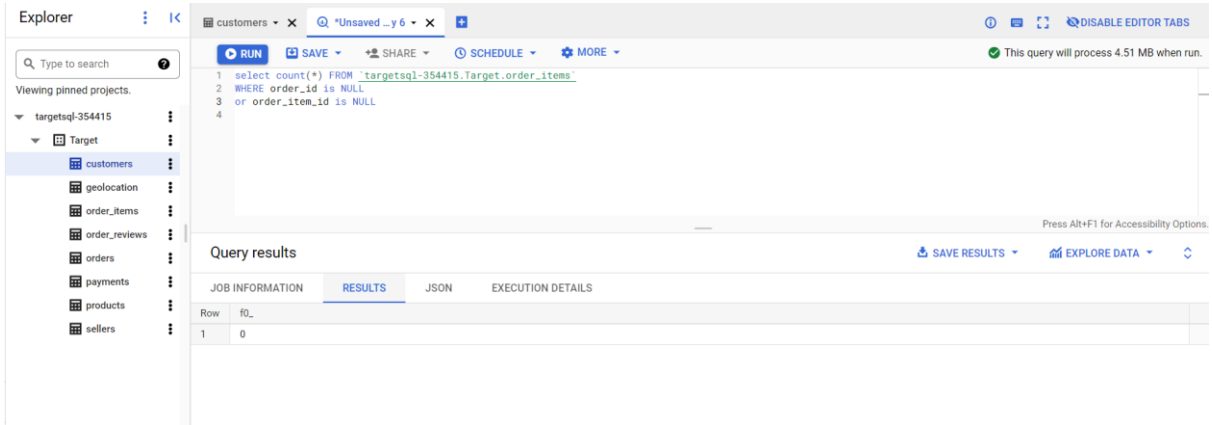
- customers



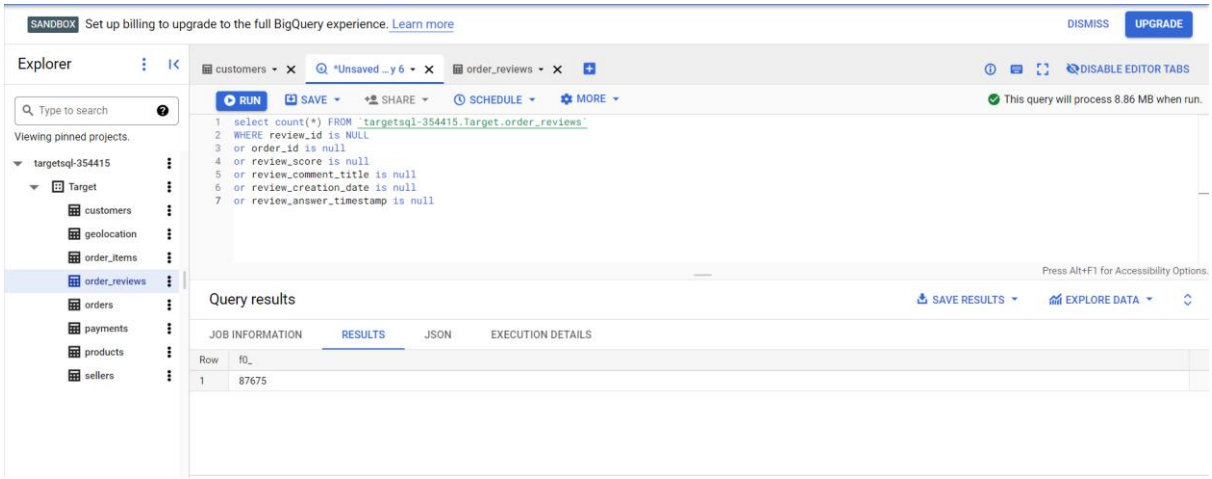
- geolocations



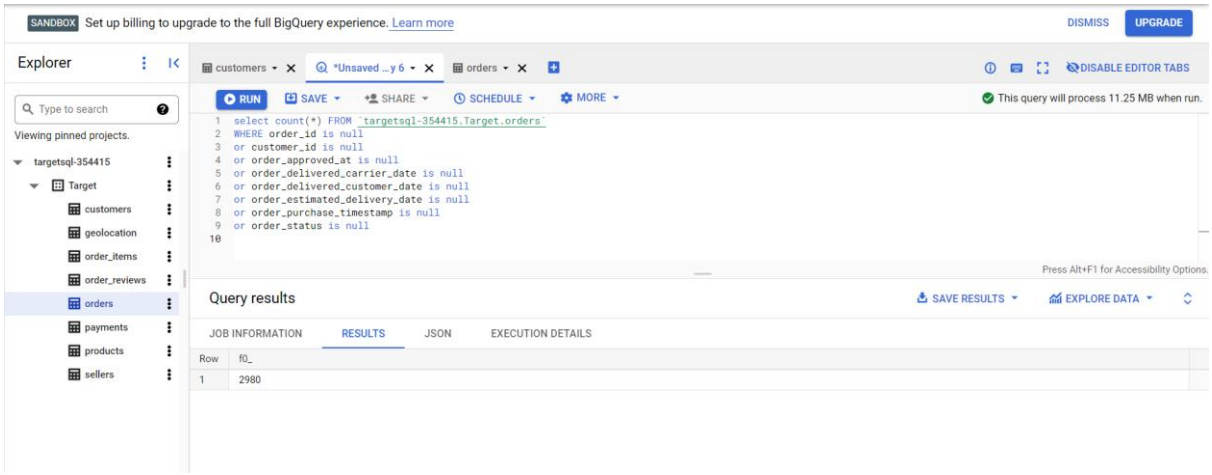
- order_items



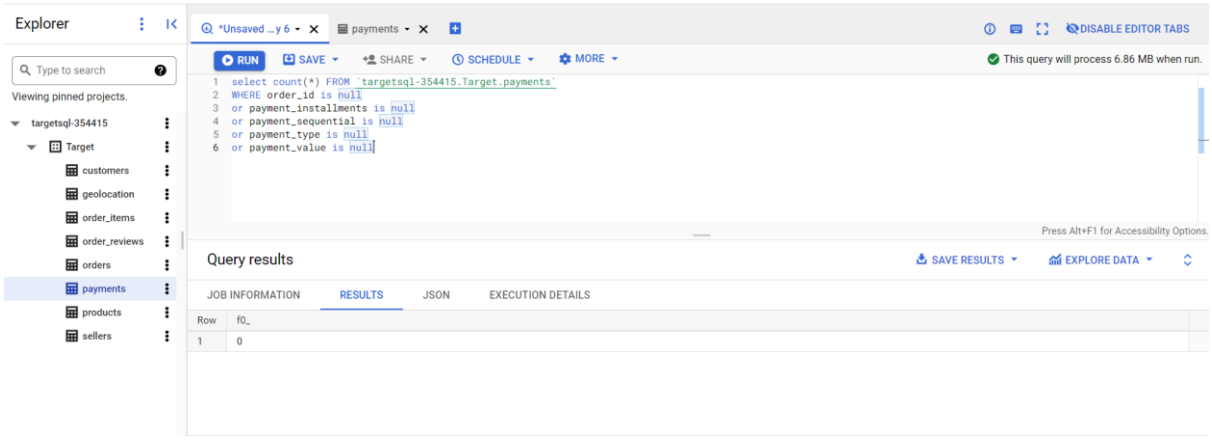
- order_reviews



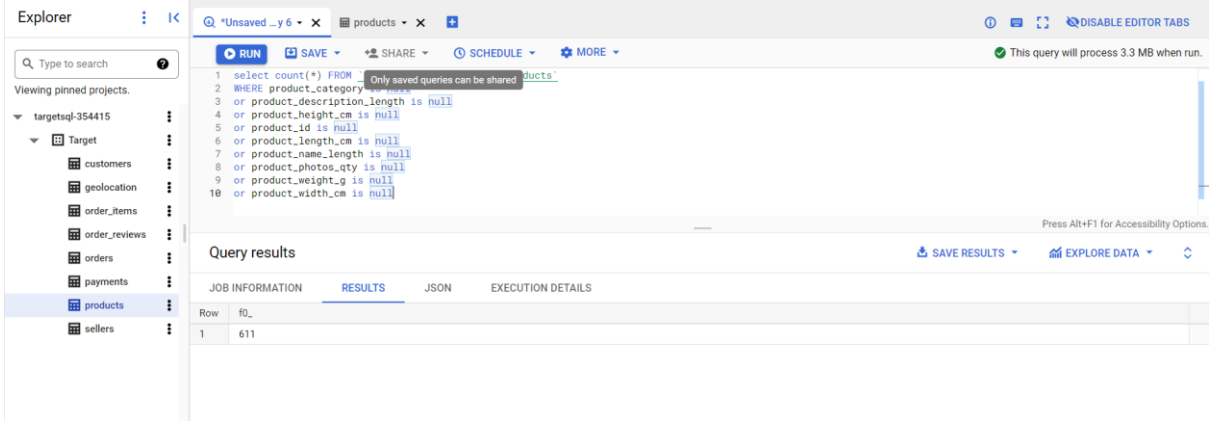
- orders



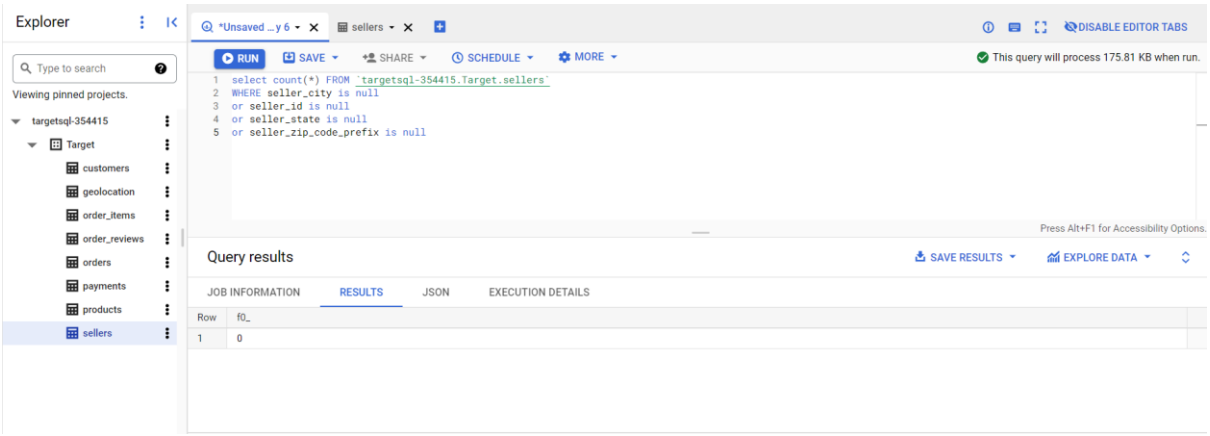
- payment



- product

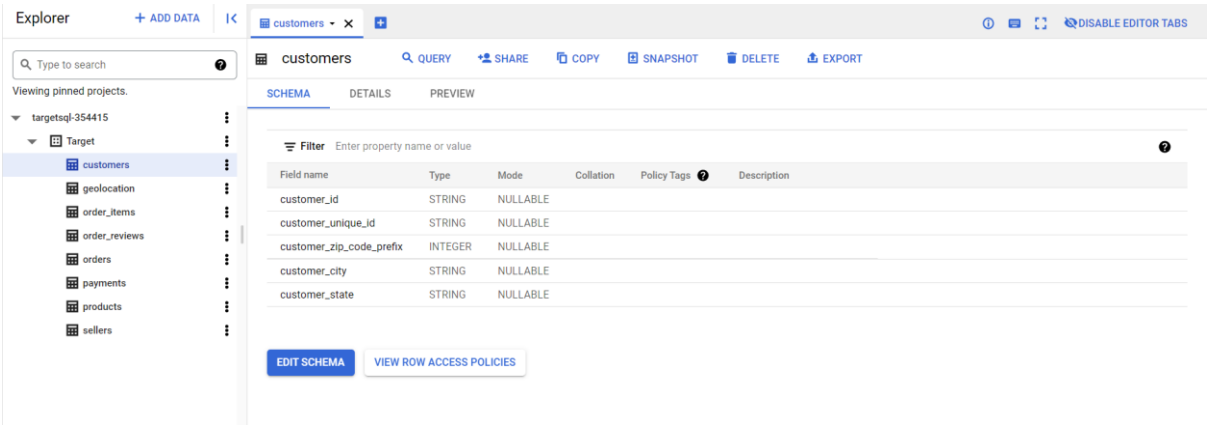


- seller

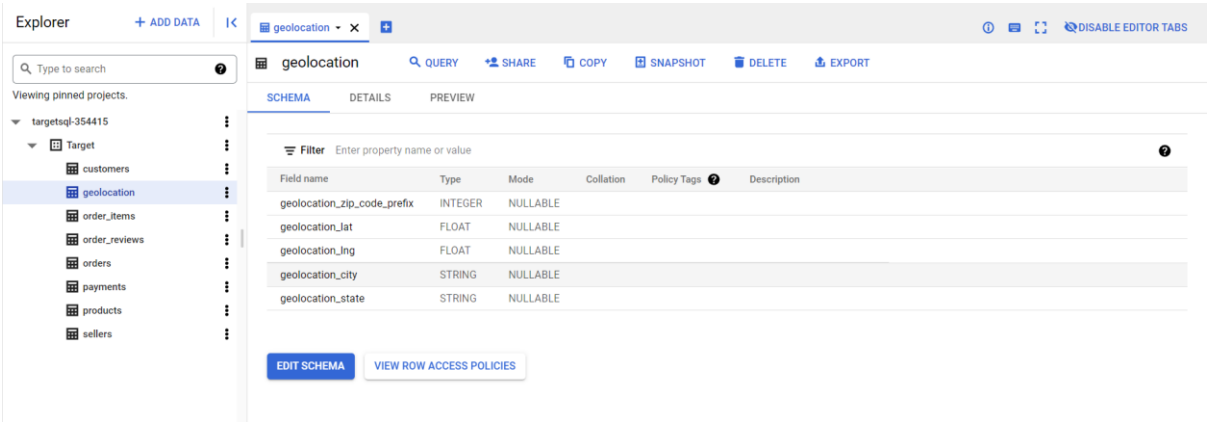


C. Datatypes of columns in the table

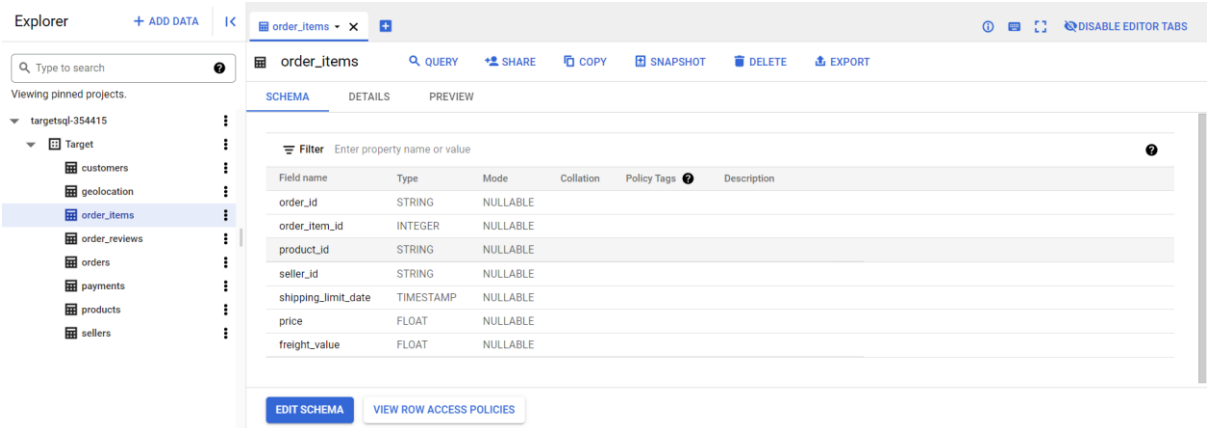
- customers



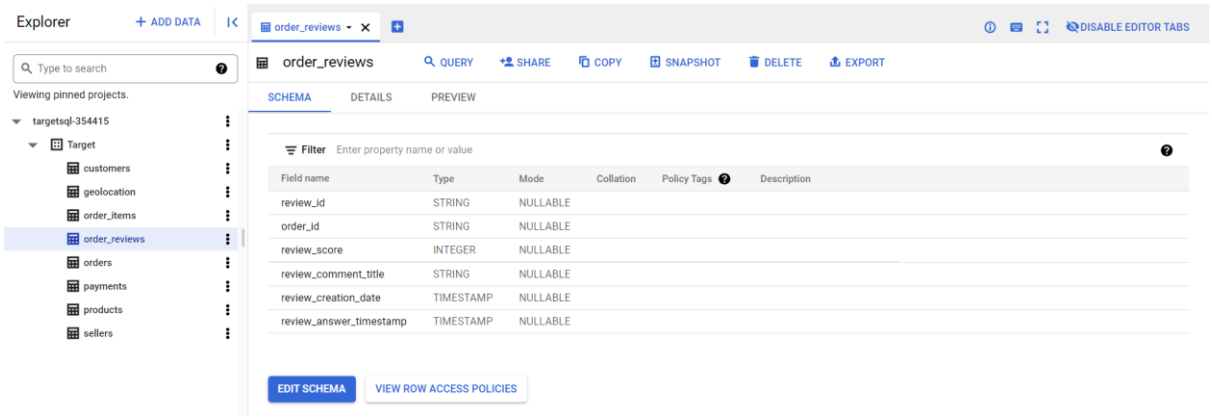
- geolocations



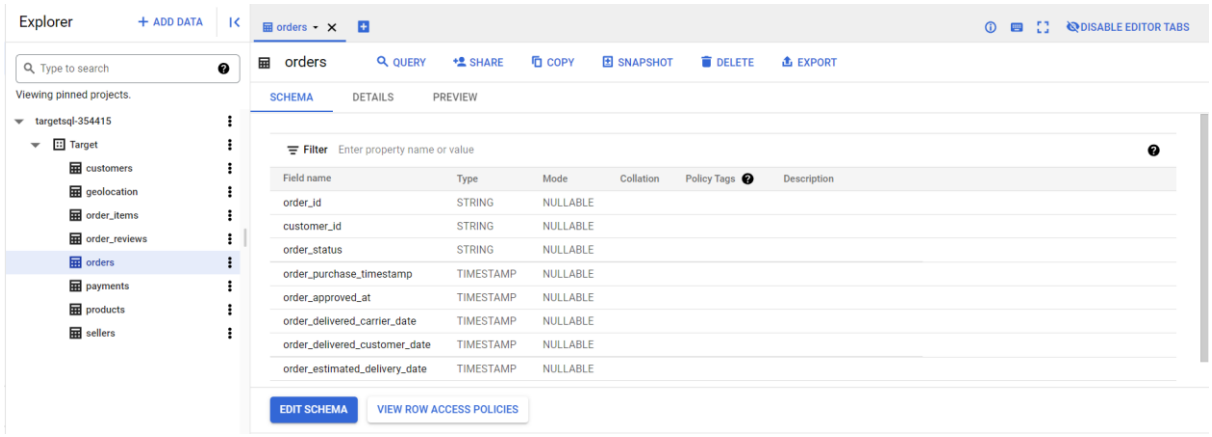
- order_items



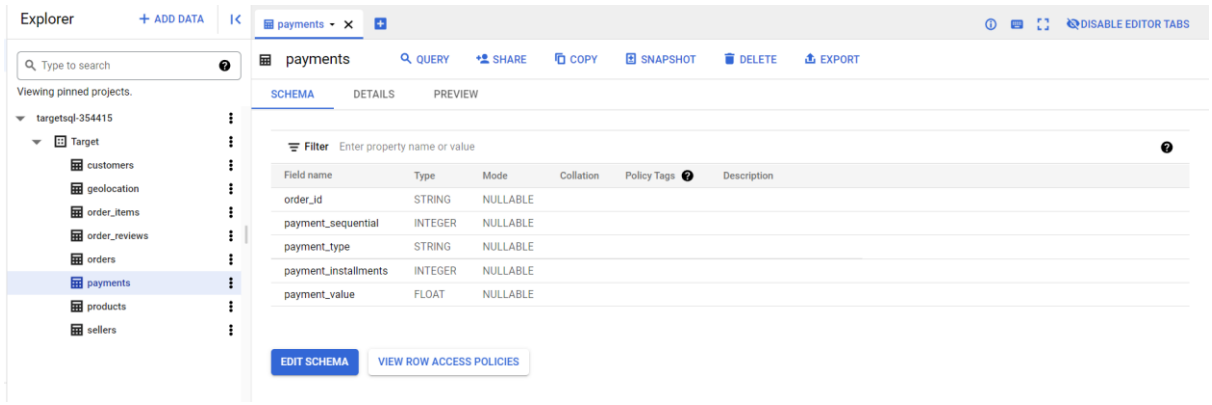
- order_reviews



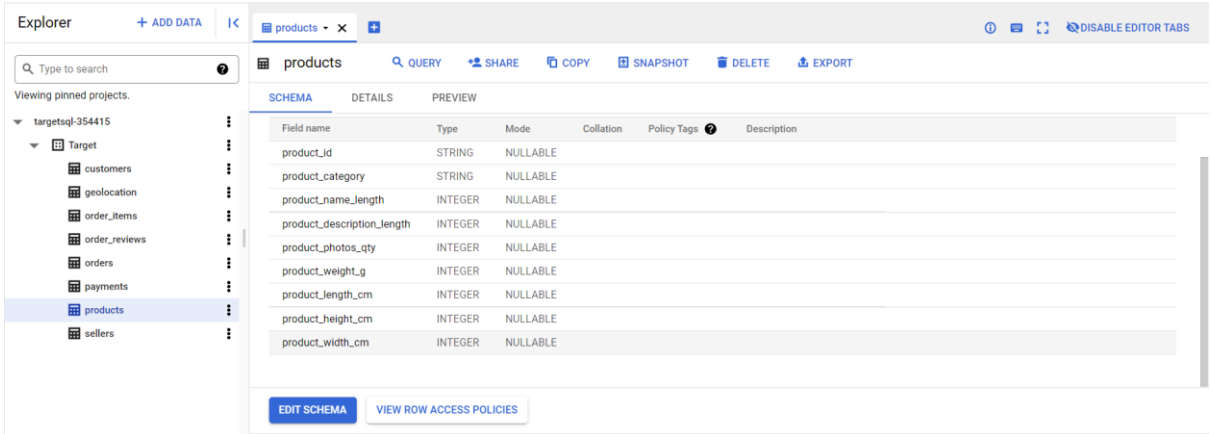
- orders



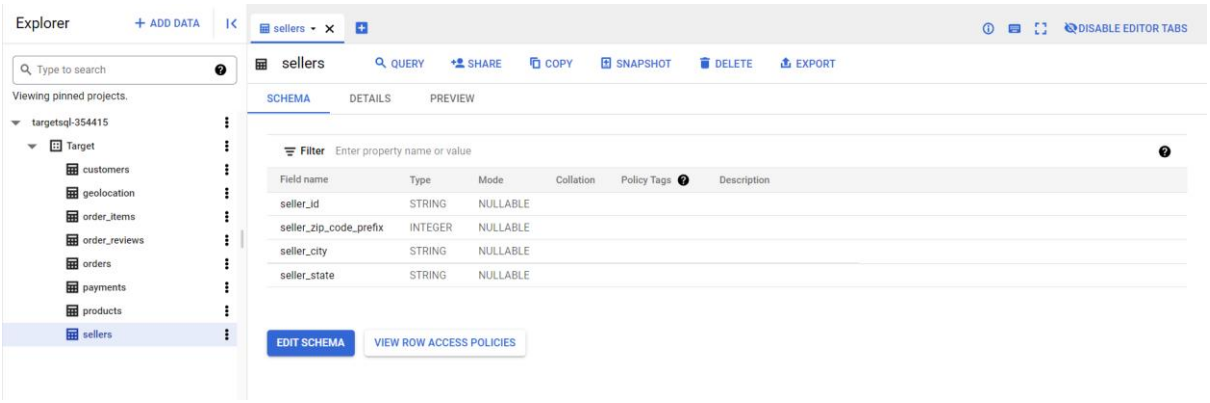
- payment



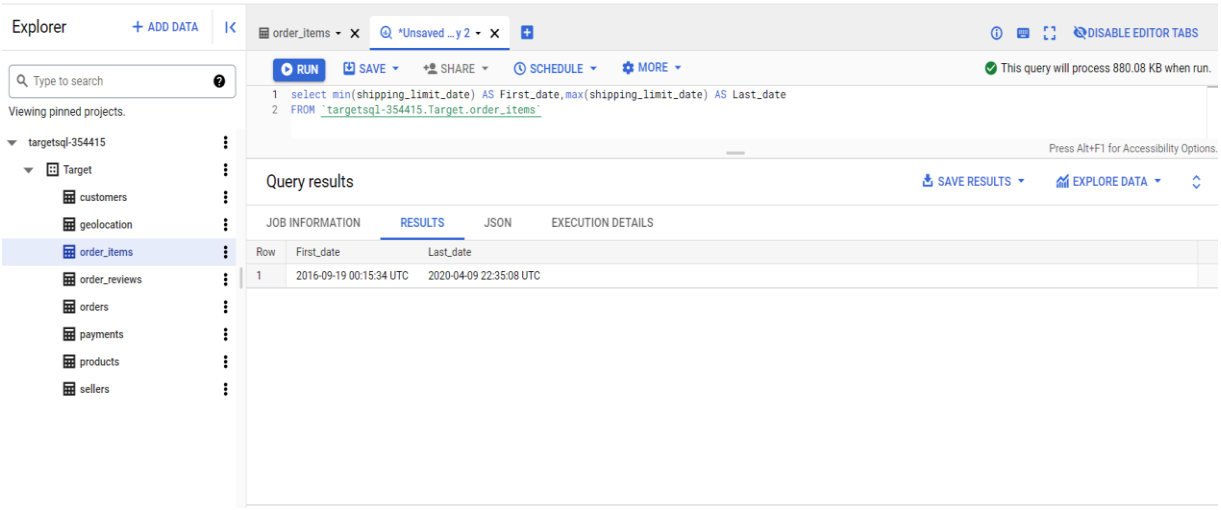
- product



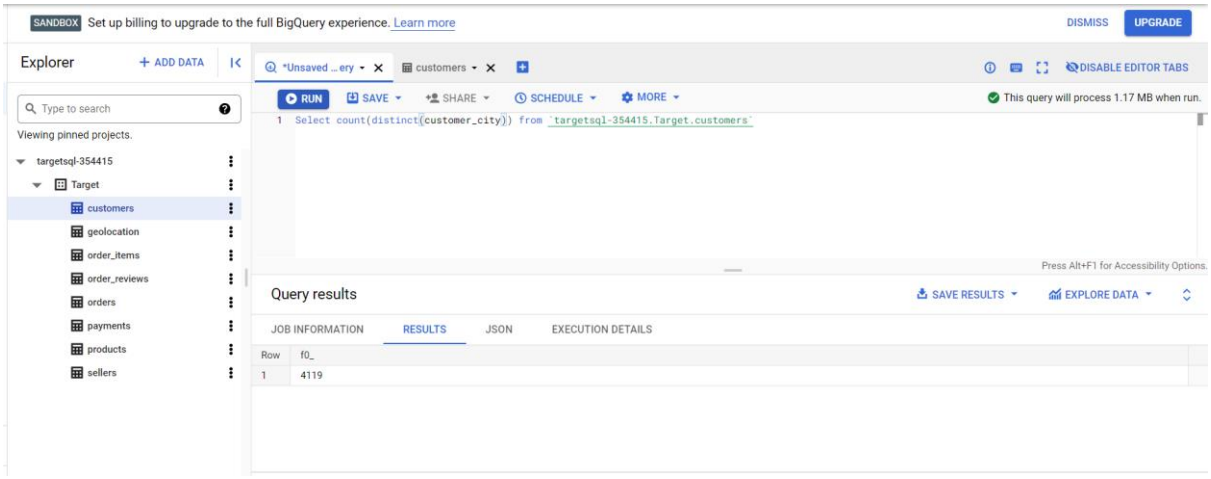
- seller



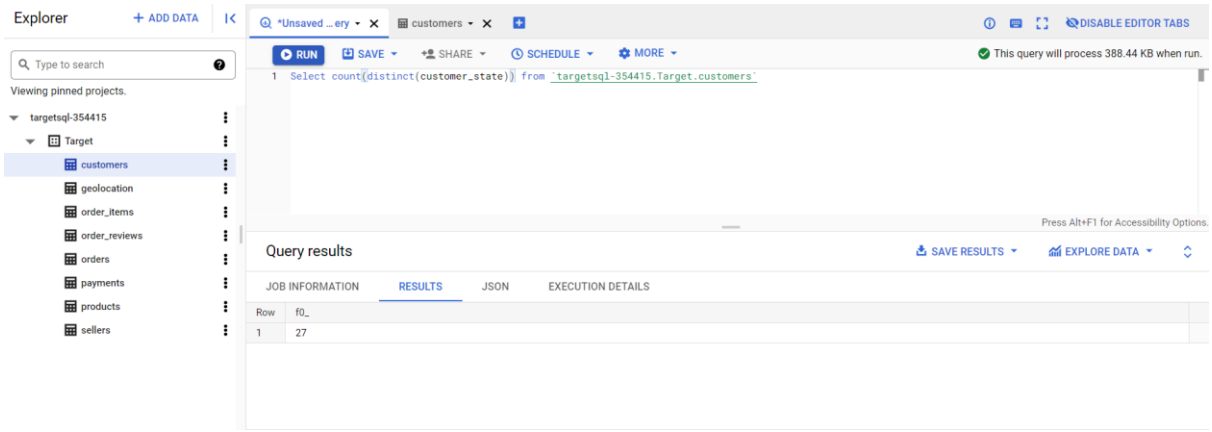
D. Get the time period for the which the data is given



E. Number of cities in our dataset

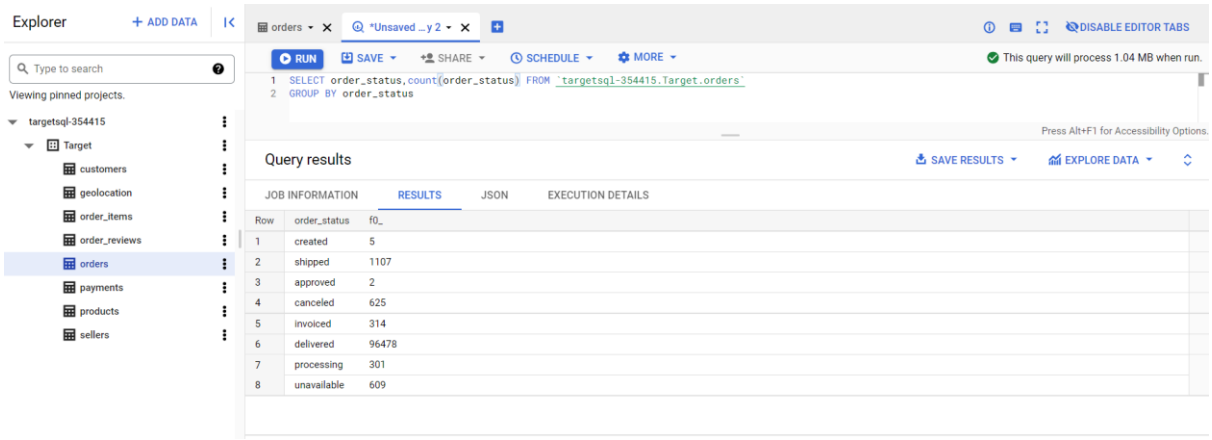


F. Number of states in out dataset



2. In-Depth Exploration

A. How many order do we have for each order status?



B. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?

Code :-

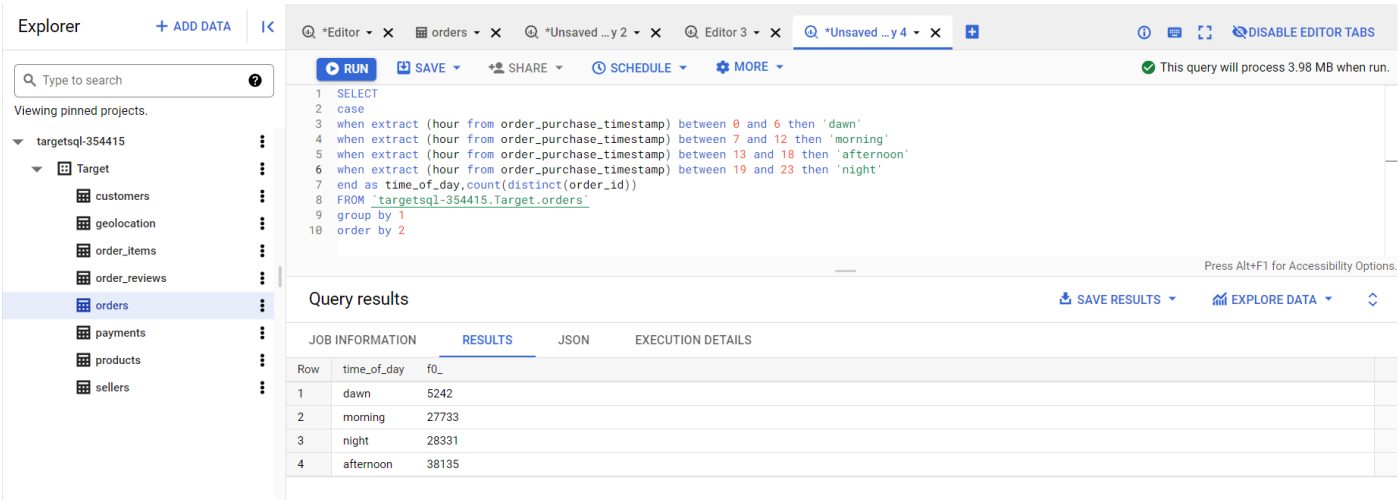
```
SELECT OrderYear,OrderMonth,Count(*) TotalOrdersInMonth

FROM

(SELECT EXTRACT(YEAR from order_purchase_timestamp) OrderYear, EXTRACT(MONTH from order_purchase_timestamp) OrderMonth

FROM `targetsql-354415.Target.orders`)
SUB
GROUP BY OrderYear,OrderMonth
ORDER BY OrderYear,OrderMonth
```

Row	OrderYear	OrderMonth	TotalOrdersInMonth
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

Row	OrderYear	OrderMonth	TotalOrdersInMonth
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728
18	2018	3	7211

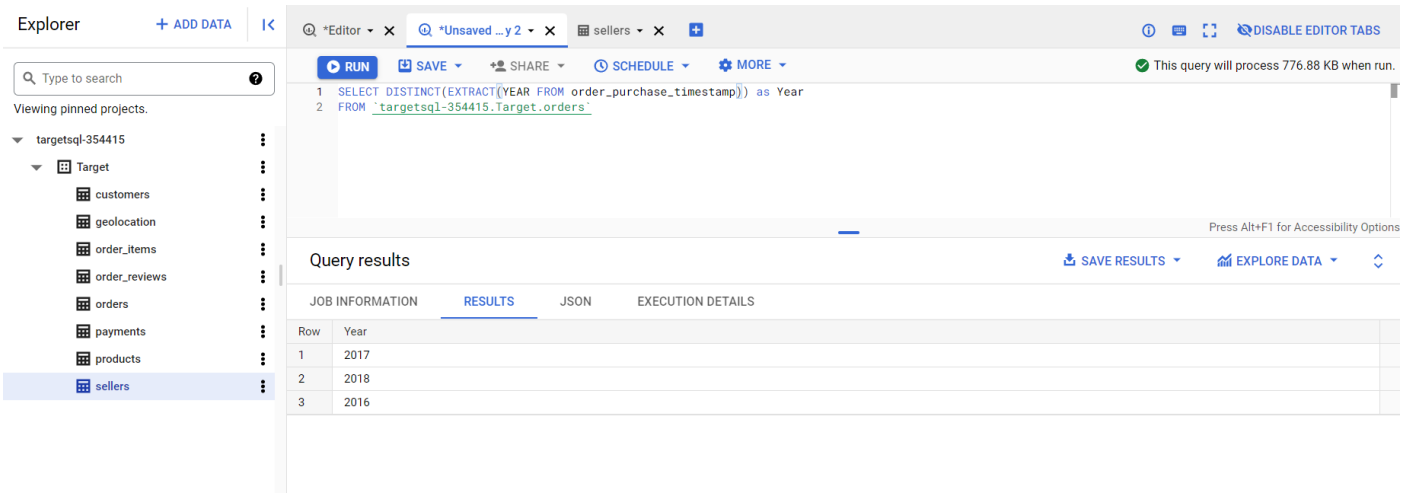
Row	OrderYear	OrderMonth	TotalOrdersInMonth
17	2018	2	6728
18	2018	3	7211
19	2018	4	6939
20	2018	5	6873
21	2018	6	6167
22	2018	7	6292
23	2018	8	6512
24	2018	9	16
25	2018	10	4

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

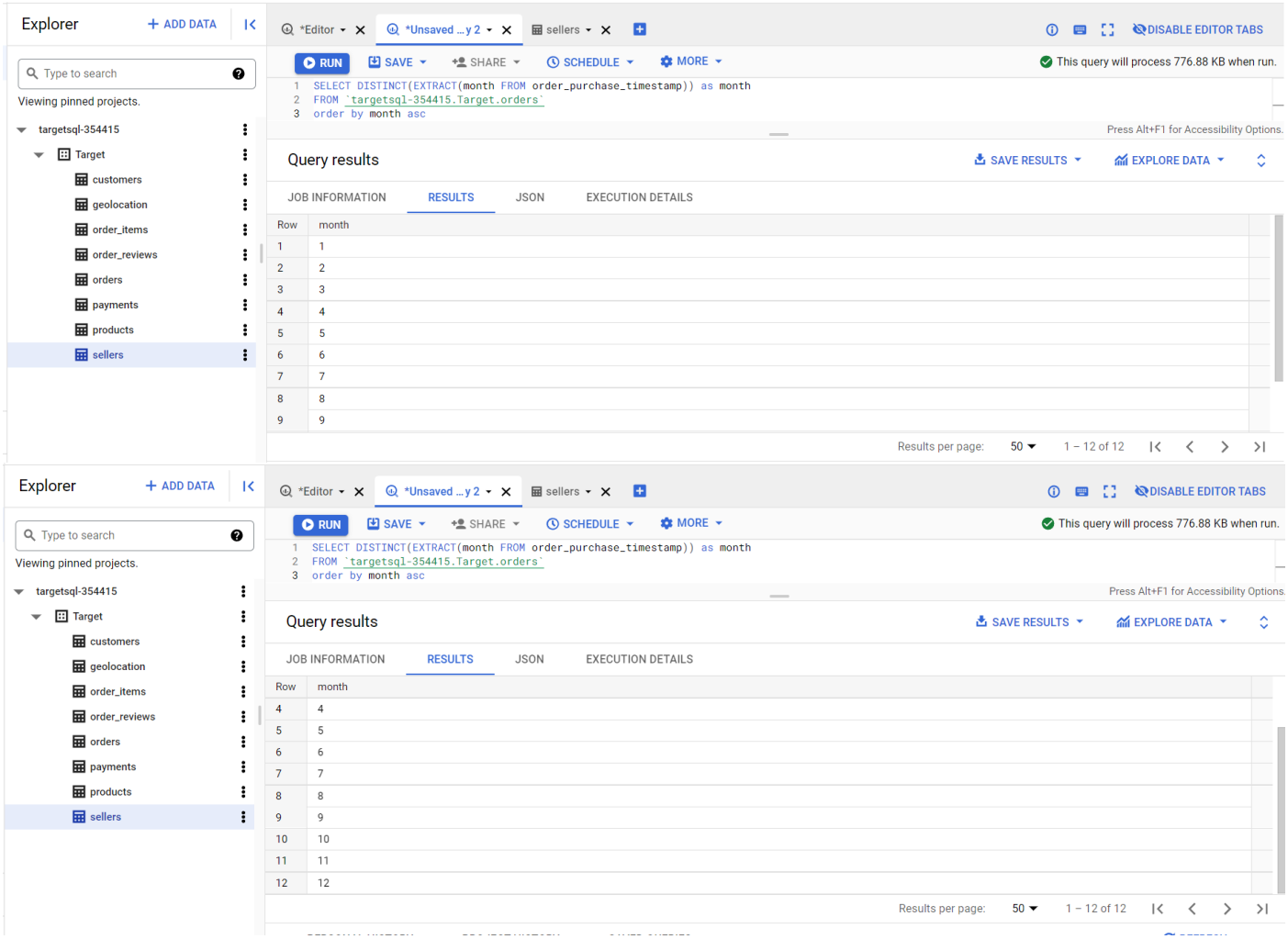


D. Feature Extraction: Through order_purchase_timestamp in “orders” dataset extract

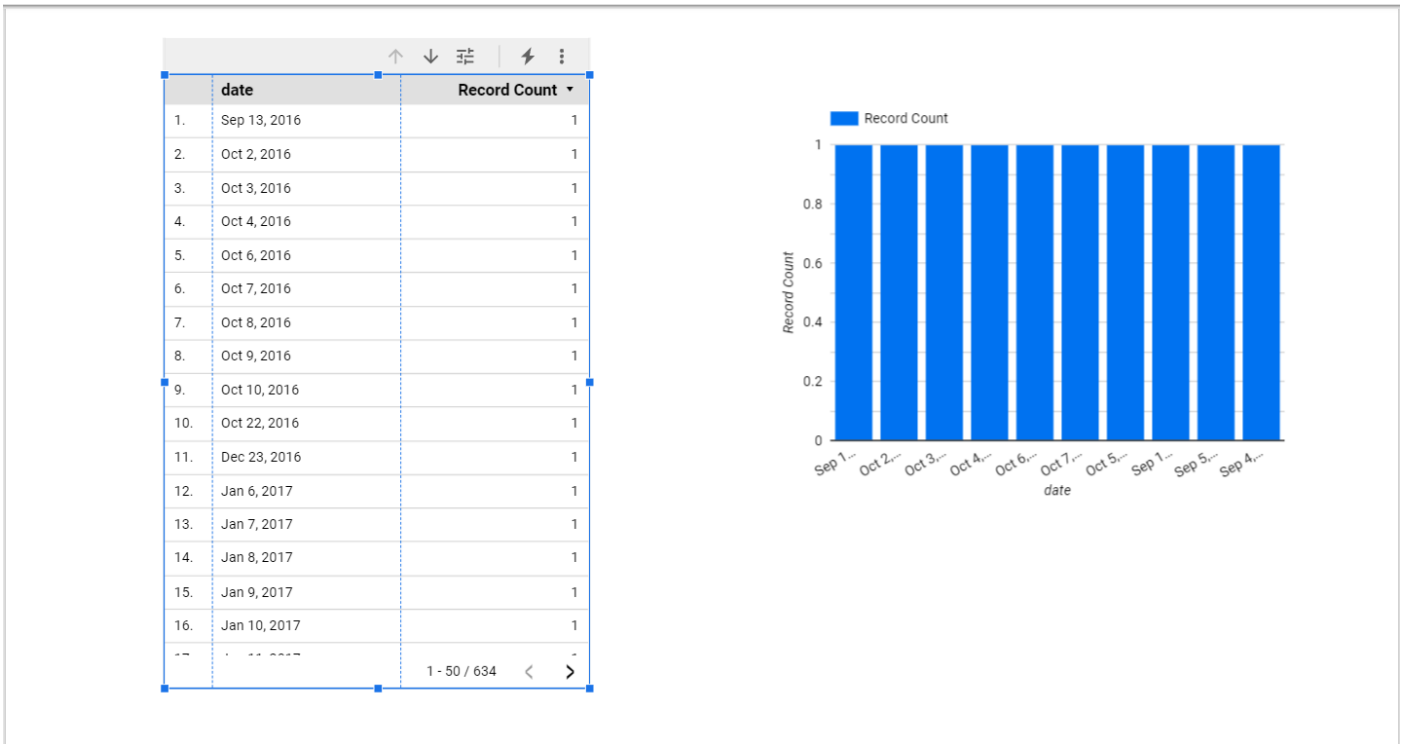
A. order_purchase_year



B. order_purchase_month



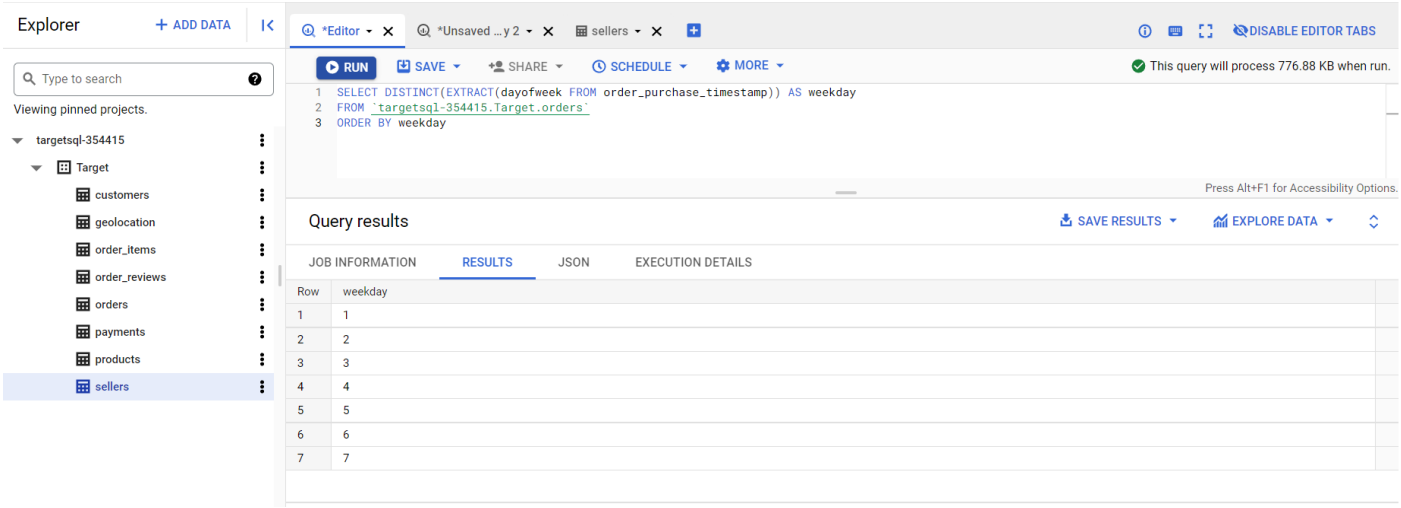
C. order_purchase_date



	date	Record Count
1.	Sep 13, 2016	1
2.	Oct 2, 2016	1
3.	Oct 3, 2016	1
4.	Oct 4, 2016	1
5.	Oct 6, 2016	1
6.	Oct 7, 2016	1
7.	Oct 8, 2016	1
8.	Oct 9, 2016	1
9.	Oct 10, 2016	1
10.	Oct 22, 2016	1
11.	Dec 23, 2016	1
12.	Jan 6, 2017	1
13.	Jan 7, 2017	1
14.	Jan 8, 2017	1
15.	Jan 9, 2017	1
16.	Jan 10, 2017	1

1 - 50 / 634

D. order_purchase_day

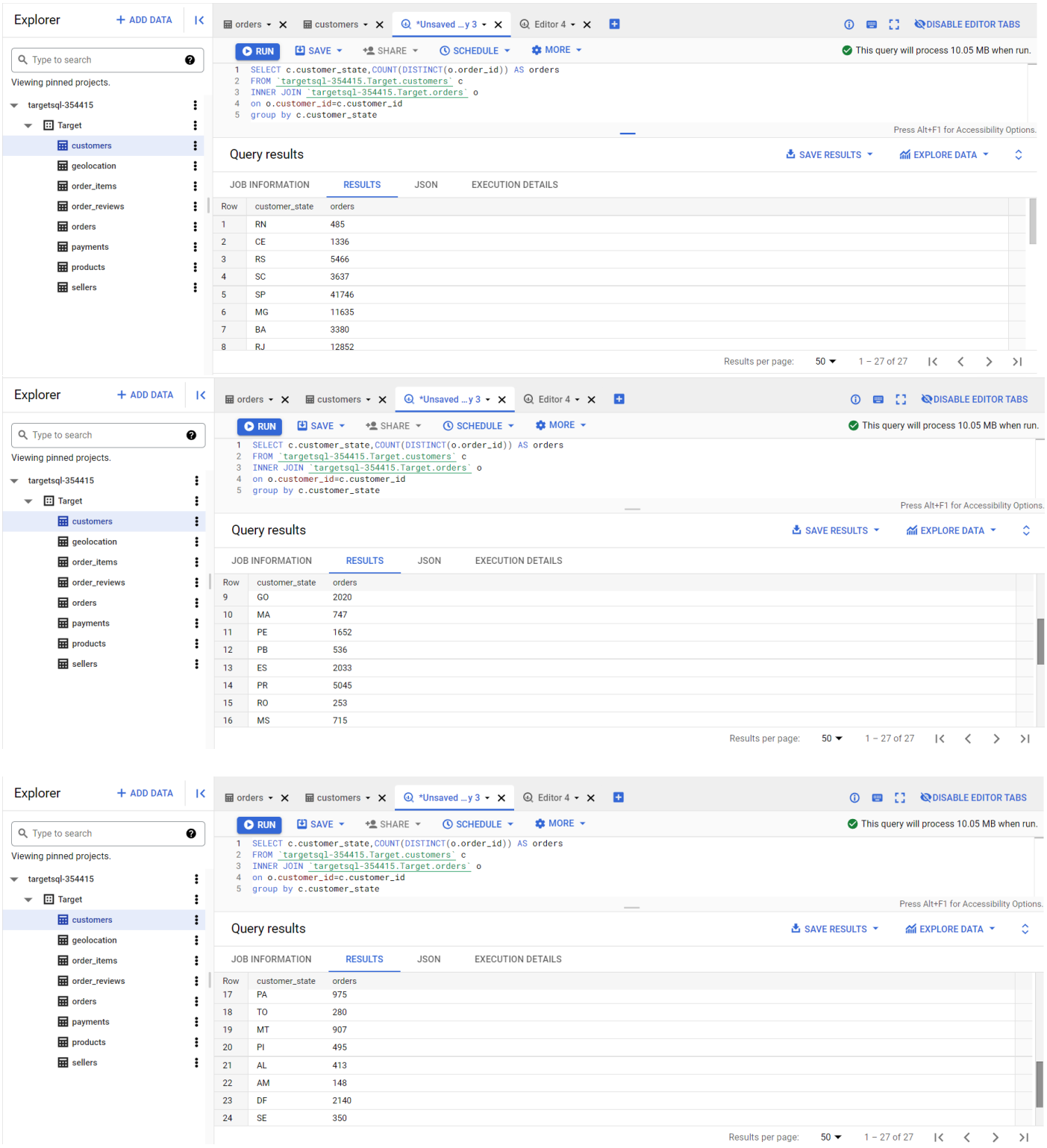


E. order_purchase_hour

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

1. Get month on month orders by region

2. Total of customer orders by state



Explorer

+ ADD DATA

Type to search

Viewing pinned projects.

- targetsql-354415
  - Target
    - customers
    - geolocation
    - order_items
    - order_reviews
    - orders
    - payments
    - products
    - sellers

orders | customers | *Unsaved ... y 3 | Editor 4

DISABLE EDITOR TABS

RUN | SAVE | SHARE | SCHEDULE | MORE

This query will process 10.05 MB when run.

```
1 SELECT c.customer_state,COUNT(DISTINCT(o.order_id)) AS orders
2 FROM `targetsql-354415.Target.customers` c
3 INNER JOIN `targetsql-354415.Target.orders` o
4 on o.customer_id=c.customer_id
5 group by c.customer_state
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	orders
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852

Results per page: 50 | 1 – 27 of 27

Explorer

+ ADD DATA

Type to search

Viewing pinned projects.

- targetsql-354415
  - Target
    - customers
    - geolocation
    - order_items
    - order_reviews
    - orders
    - payments
    - products
    - sellers

orders | customers | *Unsaved ... y 3 | Editor 4

DISABLE EDITOR TABS

RUN | SAVE | SHARE | SCHEDULE | MORE

This query will process 10.05 MB when run.

```
1 SELECT c.customer_state,COUNT(DISTINCT(o.order_id)) AS orders
2 FROM `targetsql-354415.Target.customers` c
3 INNER JOIN `targetsql-354415.Target.orders` o
4 on o.customer_id=c.customer_id
5 group by c.customer_state
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	orders
9	GO	2020
10	MA	747
11	PE	1652
12	PB	536
13	ES	2033
14	PR	5045
15	RO	253
16	MS	715

Results per page: 50 | 1 – 27 of 27

Explorer

+ ADD DATA

Type to search

Viewing pinned projects.

- targetsql-354415
  - Target
    - customers
    - geolocation
    - order_items
    - order_reviews
    - orders
    - payments
    - products
    - sellers

orders | customers | *Unsaved ... y 3 | Editor 4

DISABLE EDITOR TABS

RUN | SAVE | SHARE | SCHEDULE | MORE

This query will process 10.05 MB when run.

```
1 SELECT c.customer_state,COUNT(DISTINCT(o.order_id)) AS orders
2 FROM `targetsql-354415.Target.customers` c
3 INNER JOIN `targetsql-354415.Target.orders` o
4 on o.customer_id=c.customer_id
5 group by c.customer_state
```

Press Alt+F1 for Accessibility Options.

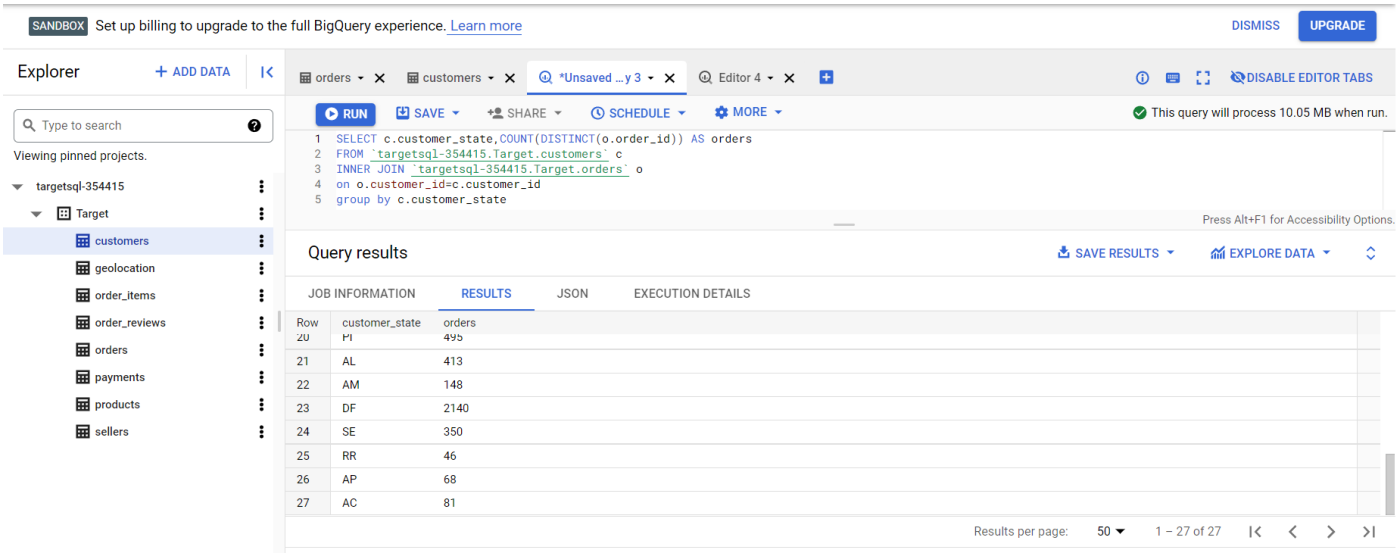
Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	orders
17	PA	975
18	TO	280
19	MT	907
20	PI	495
21	AL	413
22	AM	148
23	DF	2140
24	SE	350

Results per page: 50 | 1 – 27 of 27



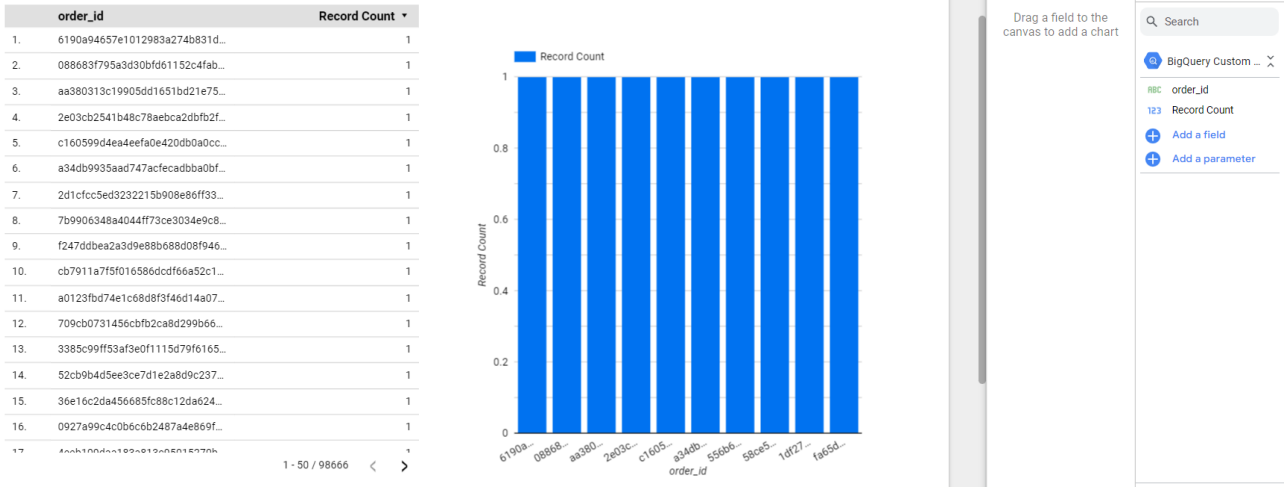
4. Impact on Economy: Analyze the money movemented by e-commerce by looking at order prices, freight and others.

Step 1: Using CTE

1. Group data by year and month, aggregation count(order_id), sum(price), sum(freight_value)

count(order_id)

```
select orders.order_id
FROM Target.orders
INNER JOIN Target.order_items
ON orders.order_id = order_items.order_id
GROUP BY orders.order_id
order by count(orders.order_id);
```

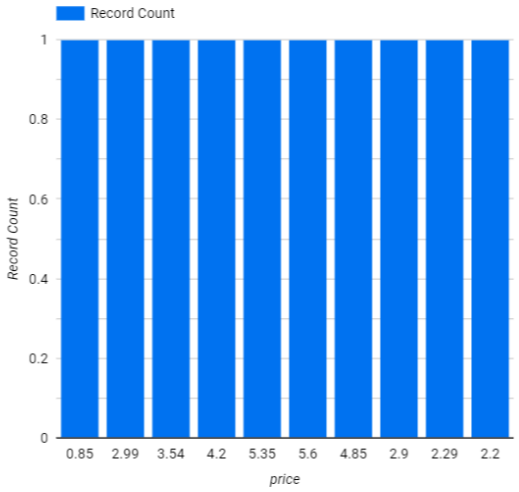


Sum(price)

```
select order_items.price
FROM Target.orders
INNER JOIN Target.order_items
ON orders.order_id =order_items.order_id
GROUP BY order_items.price
order by sum(order_items.price);
```

	price	Record Count
3.	3.54	1
4.	4.2	1
5.	5.35	1
6.	5.6	1
7.	5.7	1
8.	5.73	1
9.	5.95	1
10.	3	1
11.	6.27	1
12.	6.37	1
13.	7.09	1
14.	7.12	1
15.	7.48	1
16.	7.49	1

1 - 50 / 5968

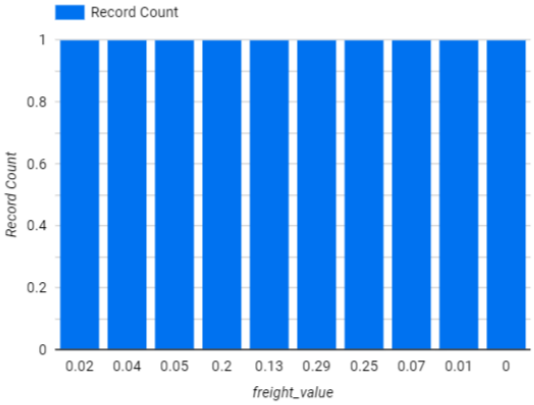


sum(freight_value)

```
select order_items.freight_value
FROM Target.orders
INNER JOIN Target.order_items
ON orders.order_id =order_items.order_id
GROUP BY order_items.freight_value
order by sum(order_items.freight_value);
```

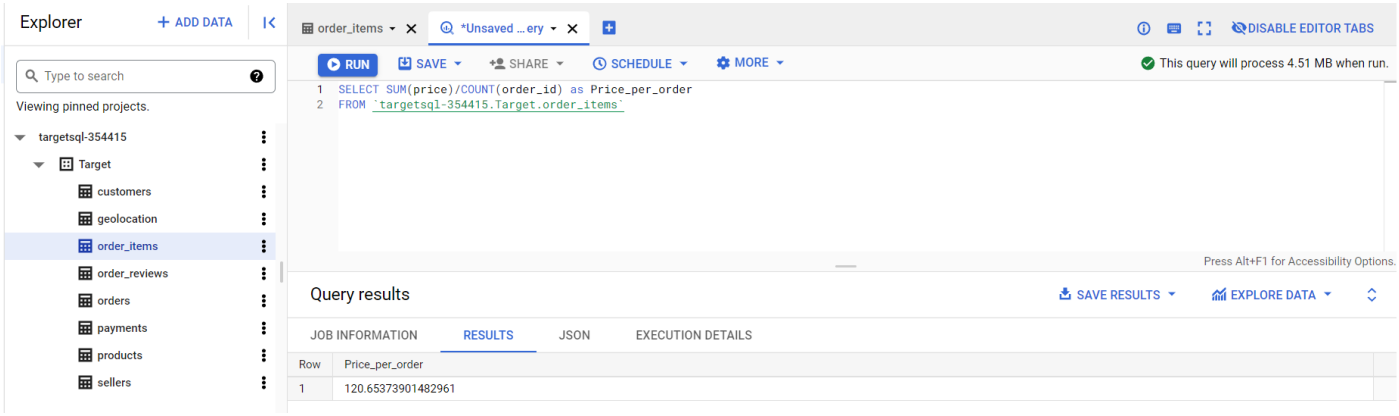
	freight_value	Record Count
1.	0.02	1
2.	0.04	1
3.	0.05	1
4.	0.2	1
5.	0.13	1
6.	0.29	1
7.	0.1	1
8.	0.31	1
9.	0.11	1
10.	0.34	1
11.	0.12	1

1 - 50 / 6999

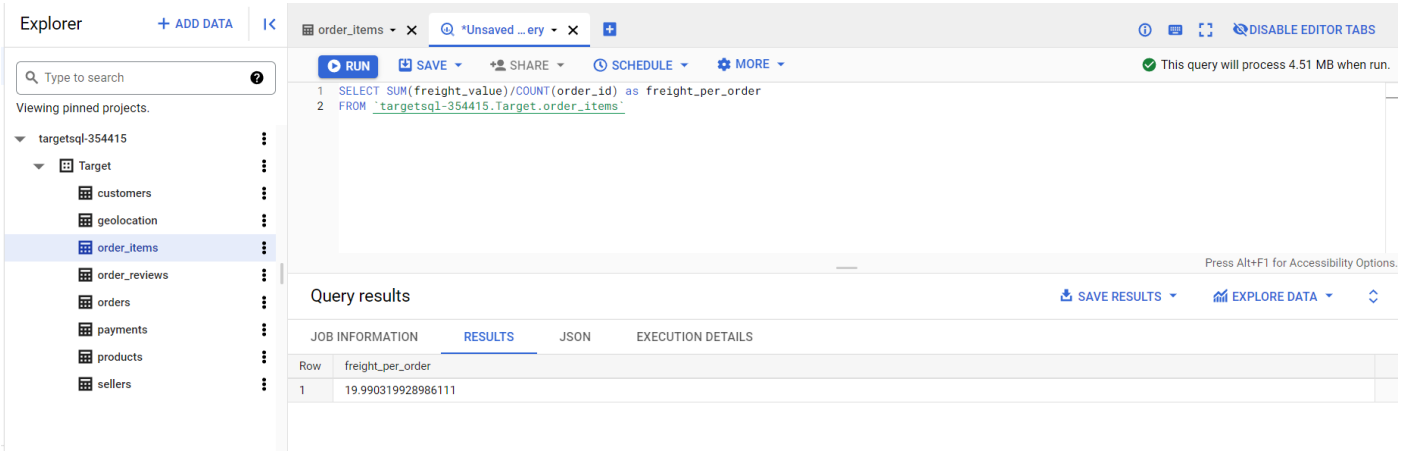


3. Create new columns:

price_per_order = sum(price) / count(order_id)

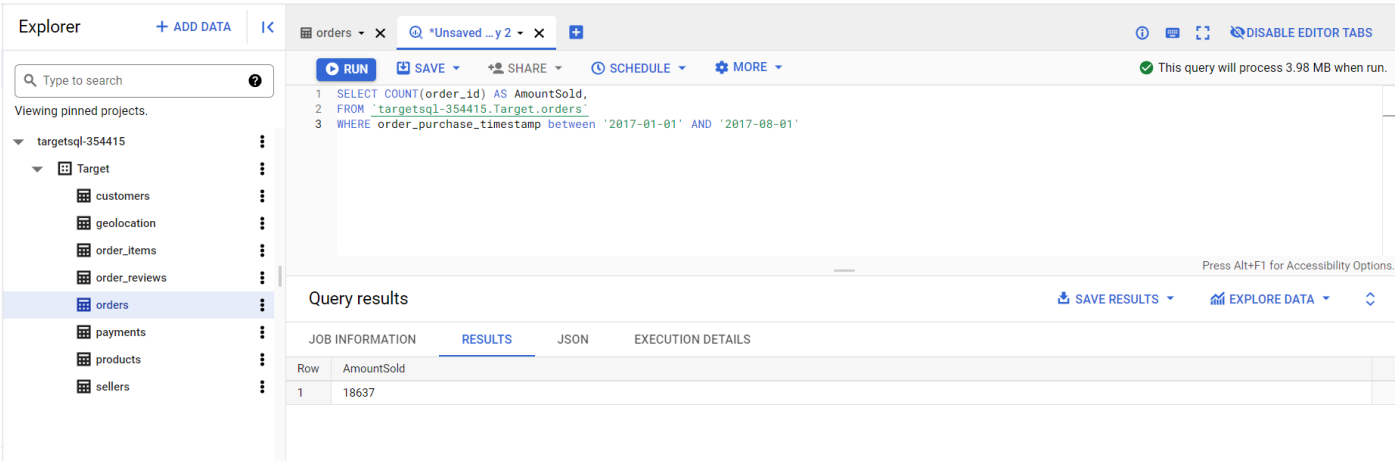


freight_per_order= sum(freight_value) / count(order_id)

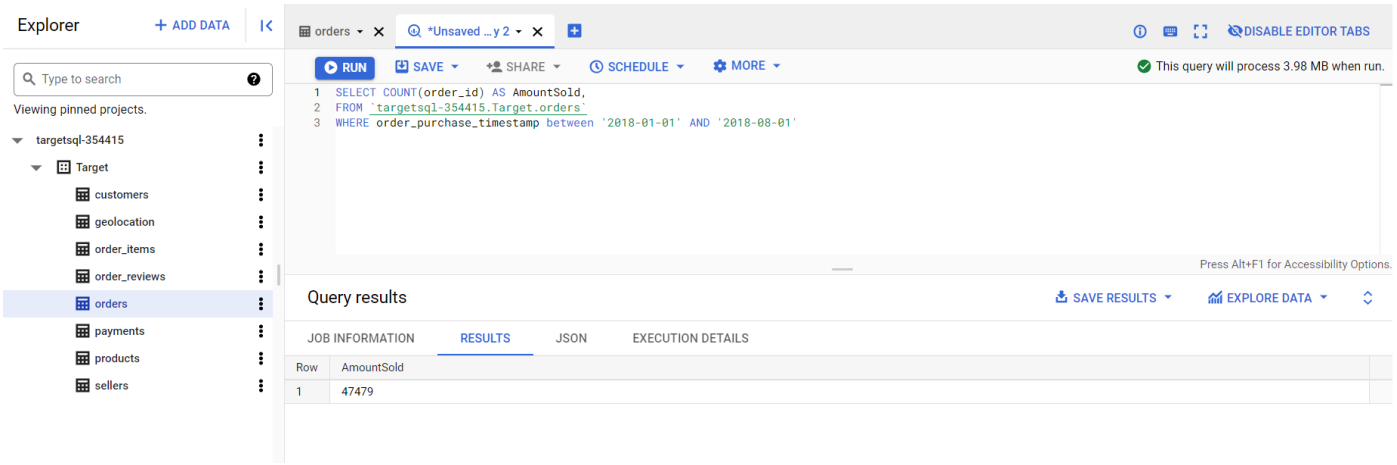


Step 2: Answer the following questions:

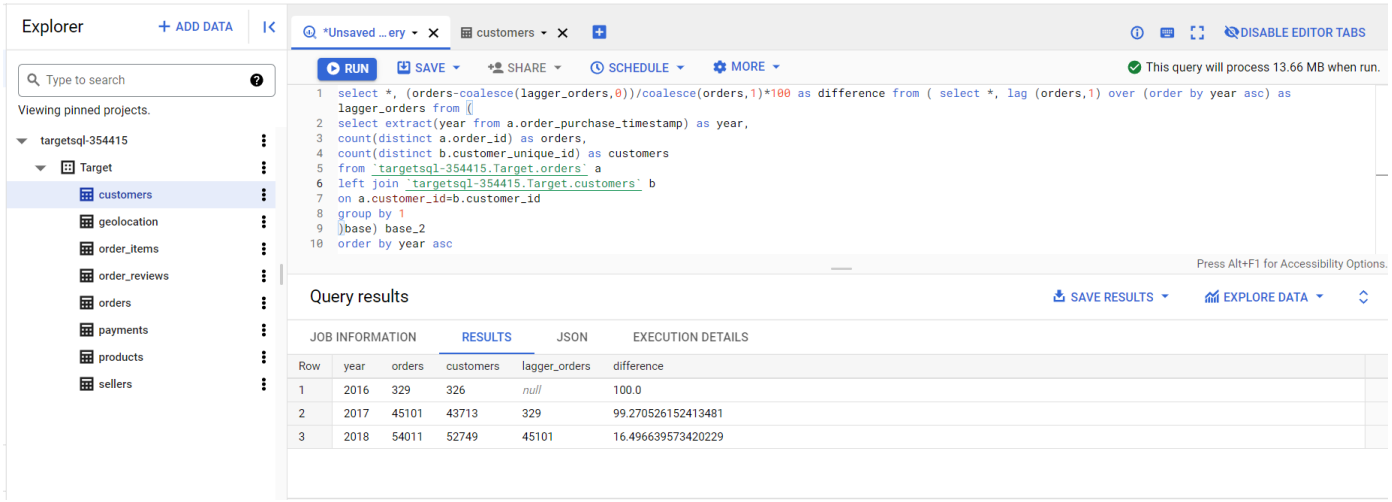
1. Total amount sold in 2017 between Jan to August



2. Total amount sold in 2018 between Jan to august



3. % increase from 2017 to 2018

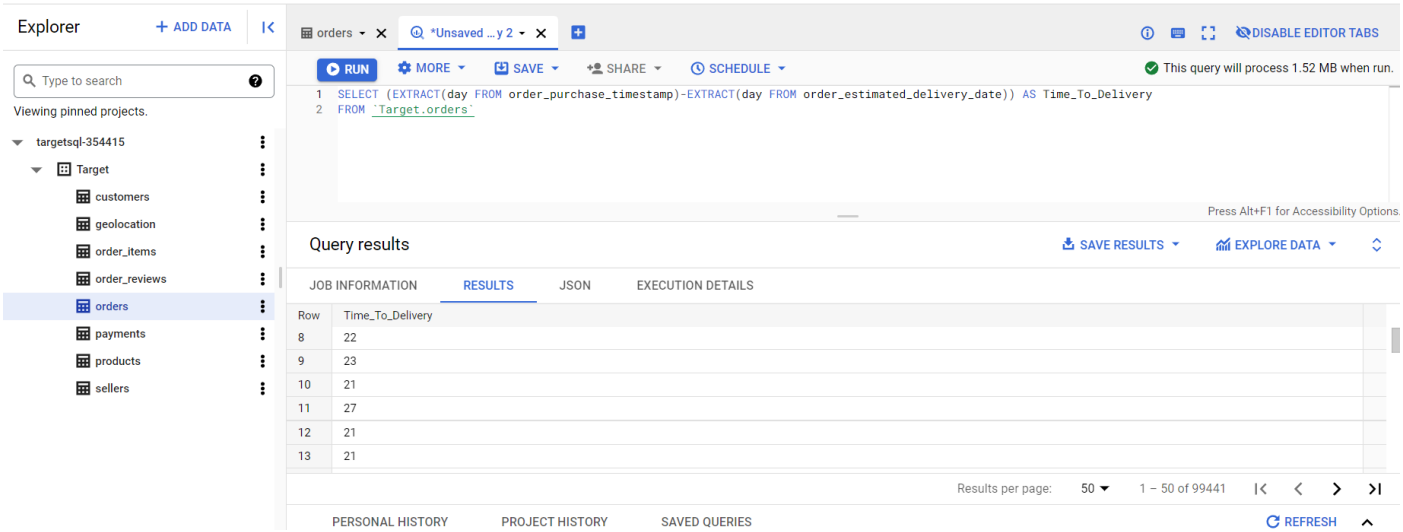


Analysis on sales, freight and delivery time

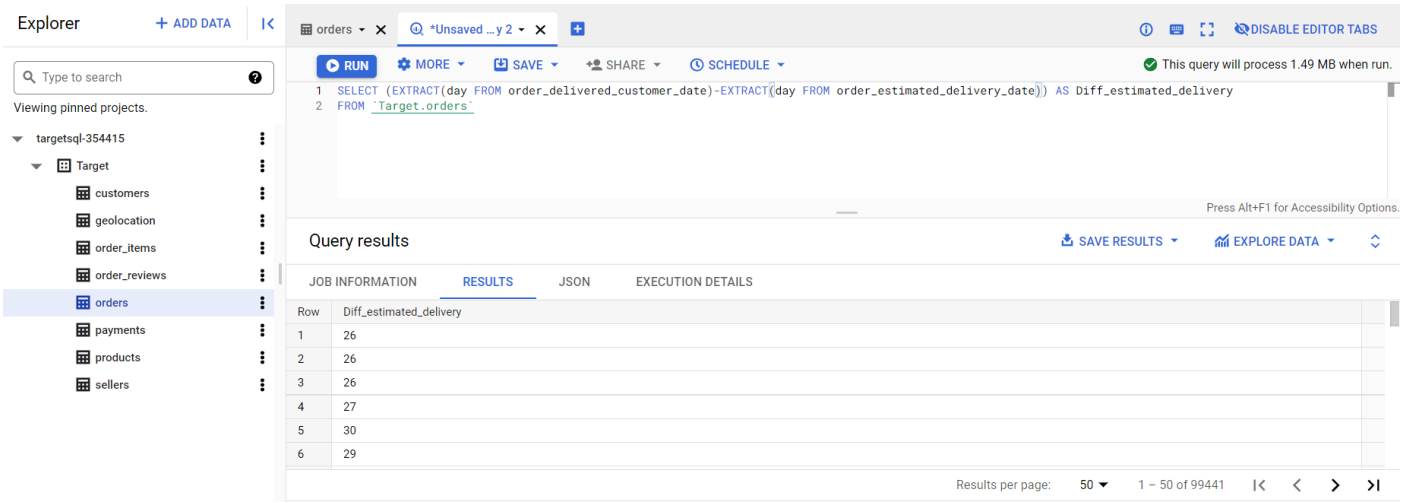
A. Calculating days between purchasing, delivering and estimated delivery

B. Create columns:

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date



diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date



6. Payment type analysis: Join “payments” dataset with the existing data on order_id

a. Count of orders for different payment types