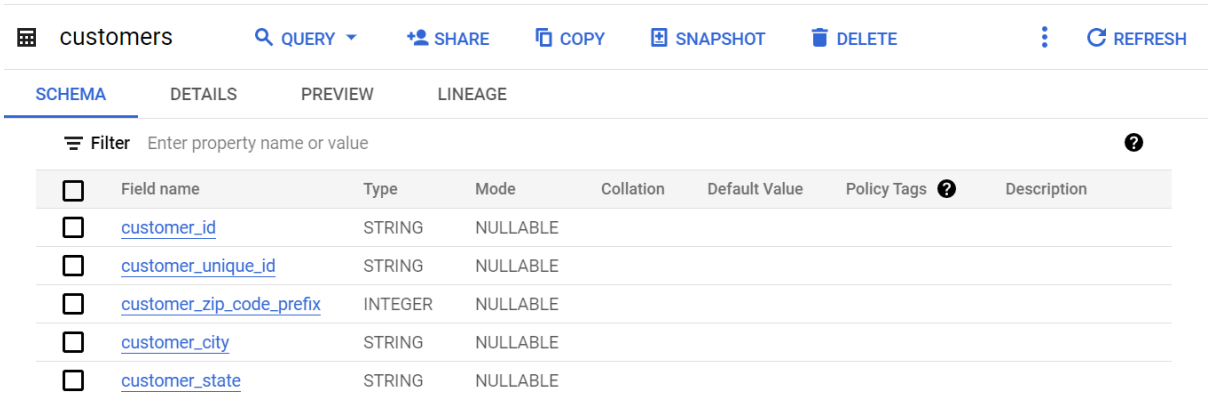
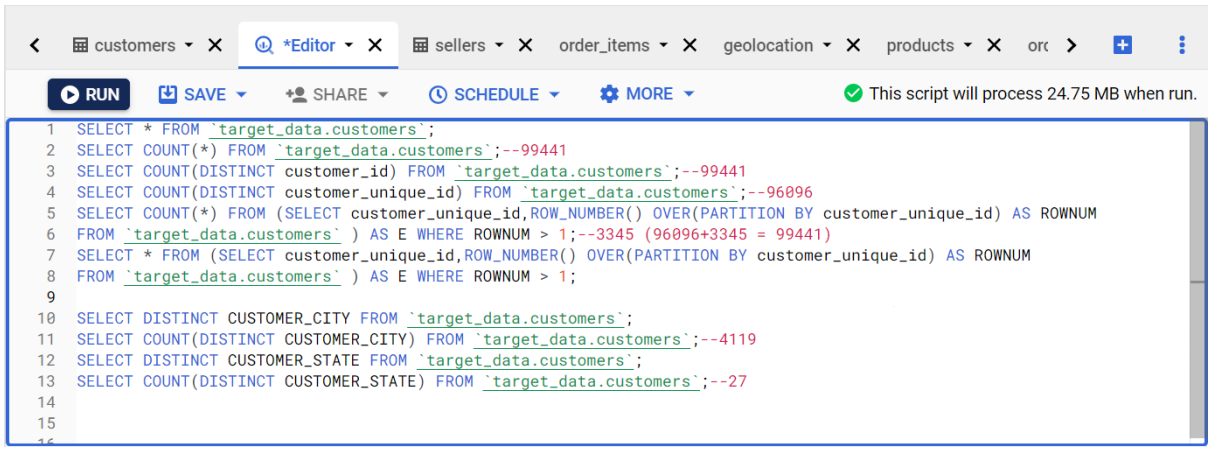
TARGET SQL

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

TABLE NAME : **customers**

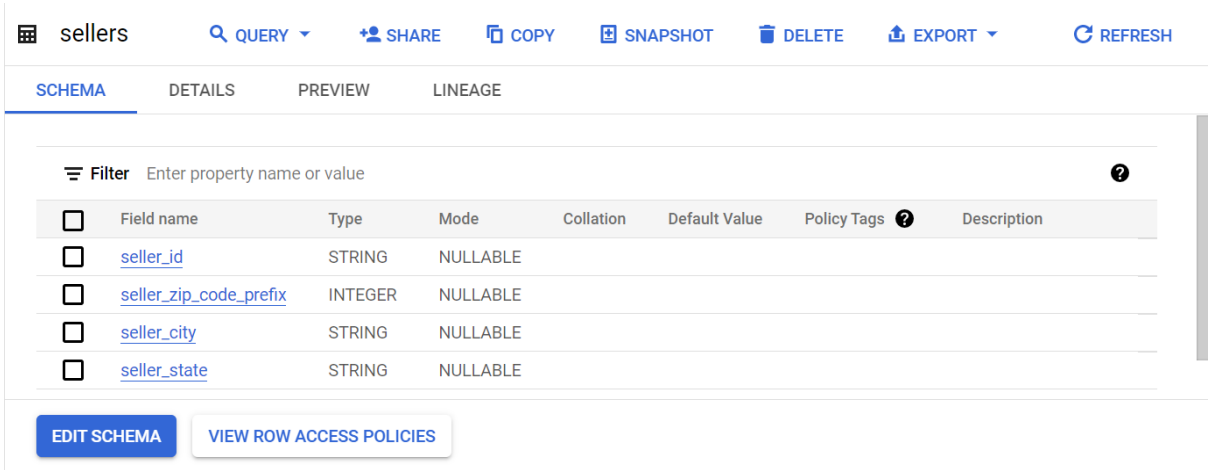


Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
customer_id	STRING	NULLABLE				
customer_unique_id	STRING	NULLABLE				
customer_zip_code_prefix	INTEGER	NULLABLE				
customer_city	STRING	NULLABLE				
customer_state	STRING	NULLABLE				




```sql
SELECT * FROM `target_data.customers`;
SELECT COUNT(*) FROM `target_data.customers`;--99441
SELECT COUNT(DISTINCT customer_id) FROM `target_data.customers`;--99441
SELECT COUNT(DISTINCT customer_unique_id) FROM `target_data.customers`;--96096
SELECT COUNT(*) FROM (SELECT customer_unique_id,ROW_NUMBER() OVER(PARTITION BY customer_unique_id) AS ROWNUM
FROM `target_data.customers` ) AS E WHERE ROWNUM > 1;--3345 (96096+3345 = 99441)
SELECT * FROM (SELECT customer_unique_id,ROW_NUMBER() OVER(PARTITION BY customer_unique_id) AS ROWNUM
FROM `target_data.customers` ) AS E WHERE ROWNUM > 1;

SELECT DISTINCT CUSTOMER_CITY FROM `target_data.customers`;
SELECT COUNT(DISTINCT CUSTOMER_CITY) FROM `target_data.customers`;--4119
SELECT DISTINCT CUSTOMER_STATE FROM `target_data.customers`;
SELECT COUNT(DISTINCT CUSTOMER_STATE) FROM `target_data.customers`;--27
```

TABLE NAME : **sellers**



Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
seller_id	STRING	NULLABLE				
seller_zip_code_prefix	INTEGER	NULLABLE				
seller_city	STRING	NULLABLE				
seller_state	STRING	NULLABLE				

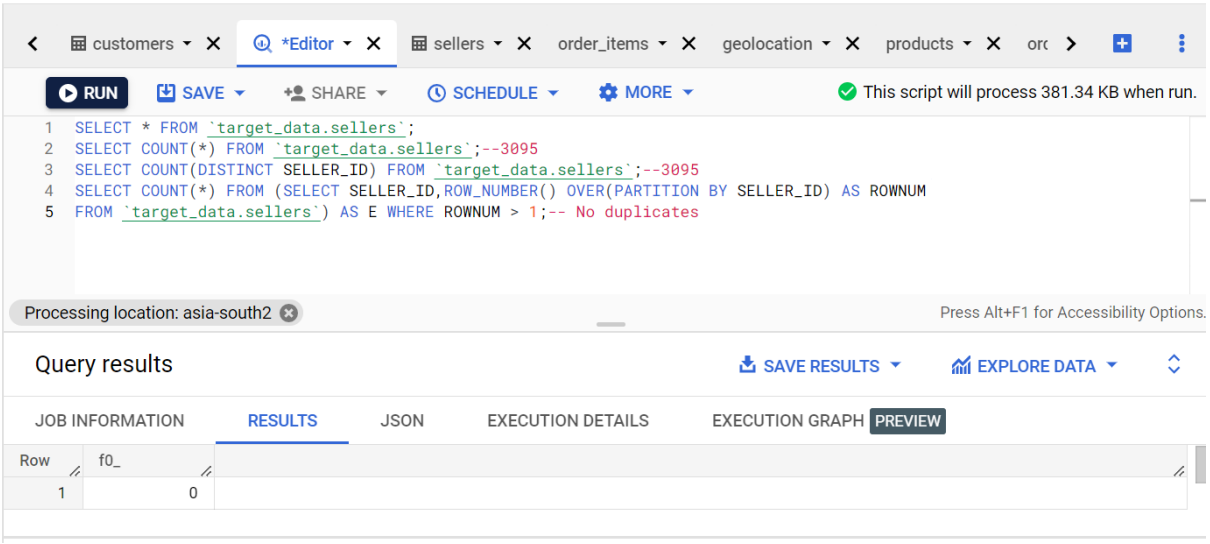
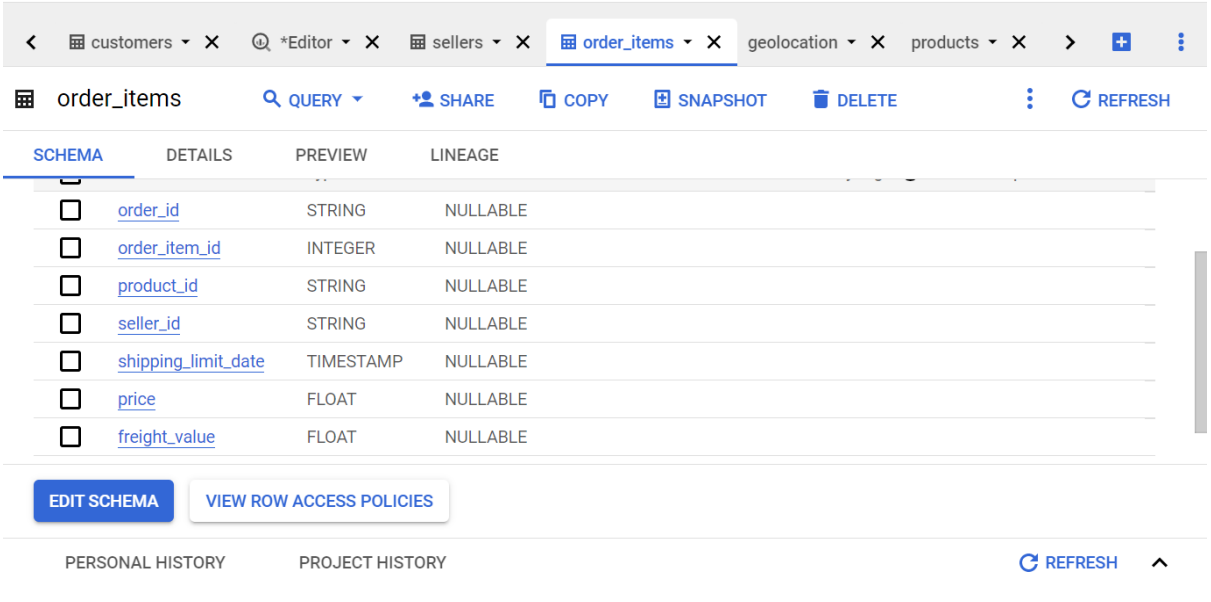


TABLE NAME : **order_items**



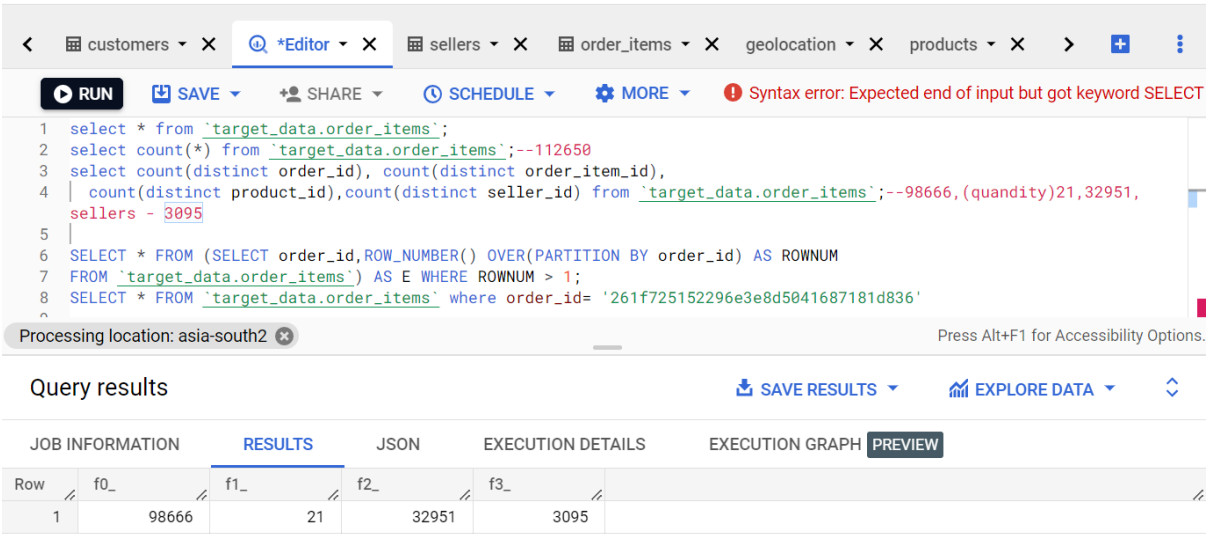


TABLE NAME : **geolocations**

geolocation QUERY SHARE COPY SNAPSHOT DELETE REFRESH

SCHEMA DETAILS PREVIEW LINEAGE

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
geolocation_zip_code_prefix	INTEGER	NULLABLE				
geolocation_lat	FLOAT	NULLABLE				
geolocation_lng	FLOAT	NULLABLE				
geolocation_city	STRING	NULLABLE				
geolocation_state	STRING	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

customers *Editor sellers order_items geolocation products

RUN SAVE SHARE SCHEDULE MORE Query completed.

```
1 select * from `target_data.geolocation`;
2 select count(distinct geolocation_city) from `target_data.geolocation`;--8011
3 select count(distinct geolocation_state) from `target_data.geolocation`;--27
4
5
```

Processing location: asia-south2 Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	f0_
1	8011

TABLE NAME : **payments**

payments QUERY SHARE COPY SNAPSHOT DELETE REFRESH

SCHEMA DETAILS PREVIEW LINEAGE

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
order_id	STRING	NULLABLE				
payment_sequential	INTEGER	NULLABLE				
payment_type	STRING	NULLABLE				
payment_installments	INTEGER	NULLABLE				
payment_value	FLOAT	NULLABLE				

EDIT SCHEMA VIEW ROW ACCESS POLICIES

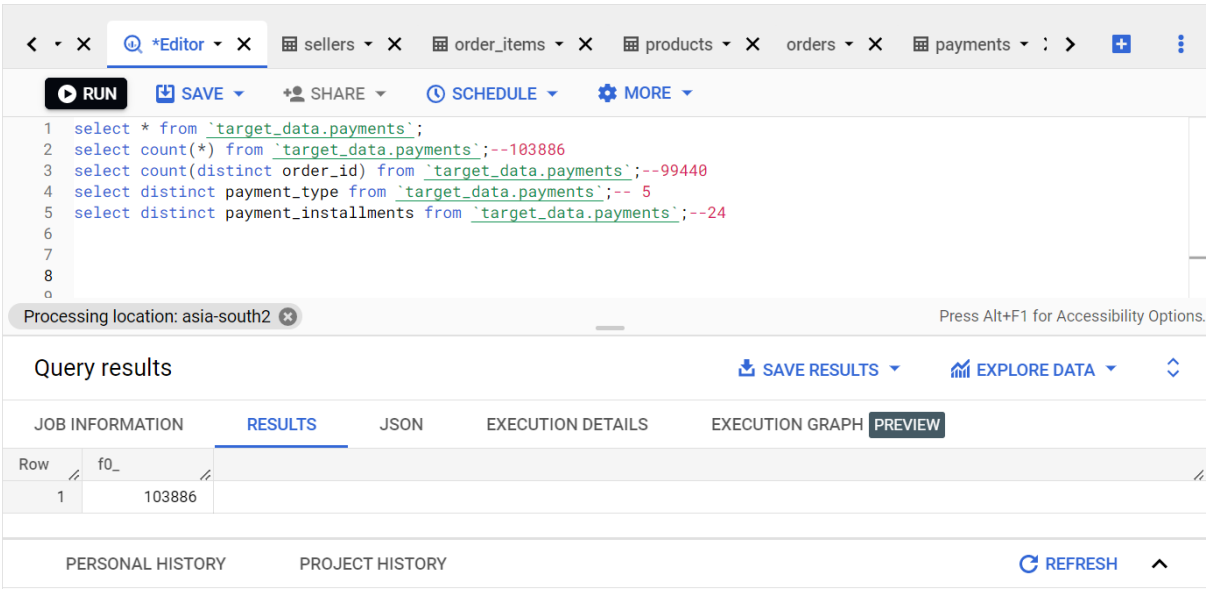
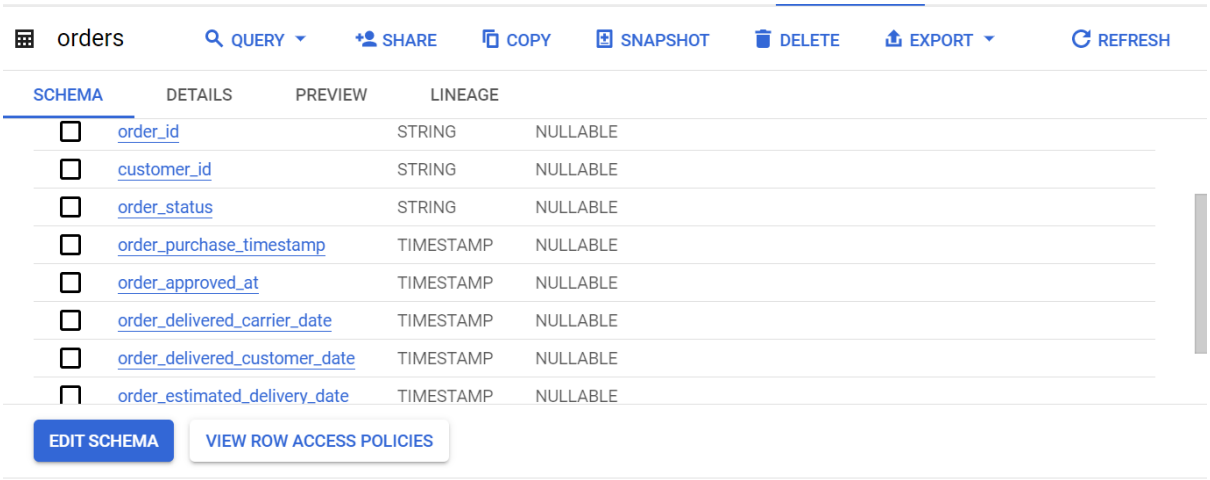


TABLE NAME : **orders (8 columns were there in actual table- order_approved_at)**



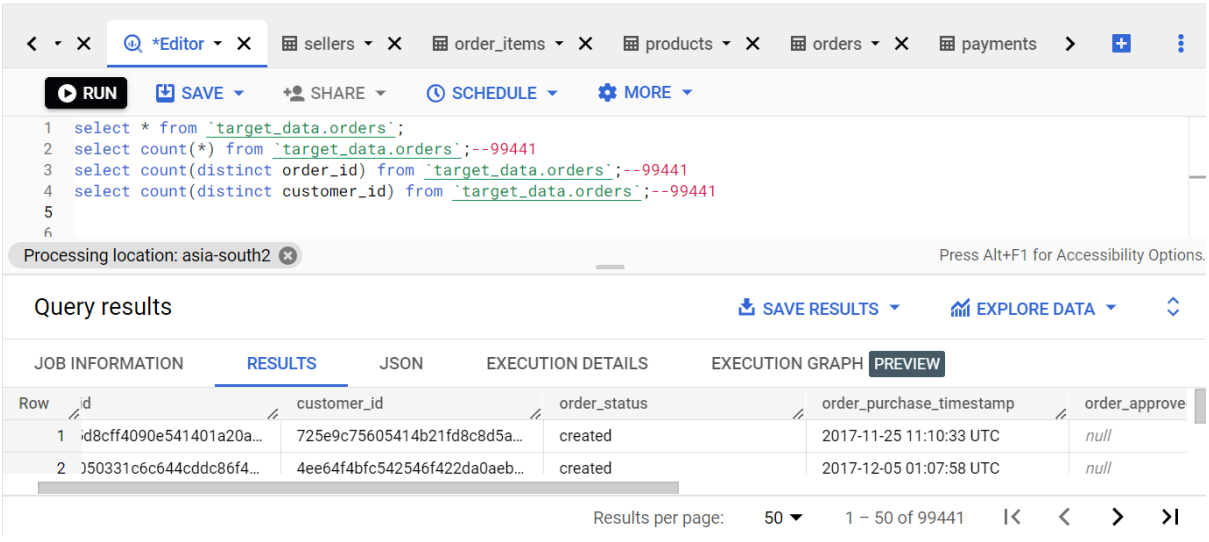
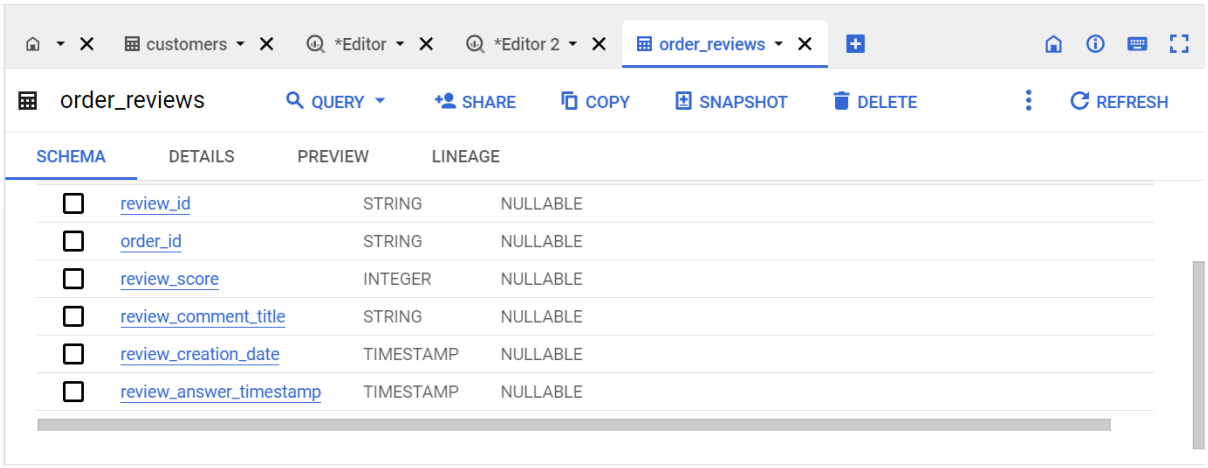


TABLE NAME : **reviews** (review_comment_message column is missing in table)



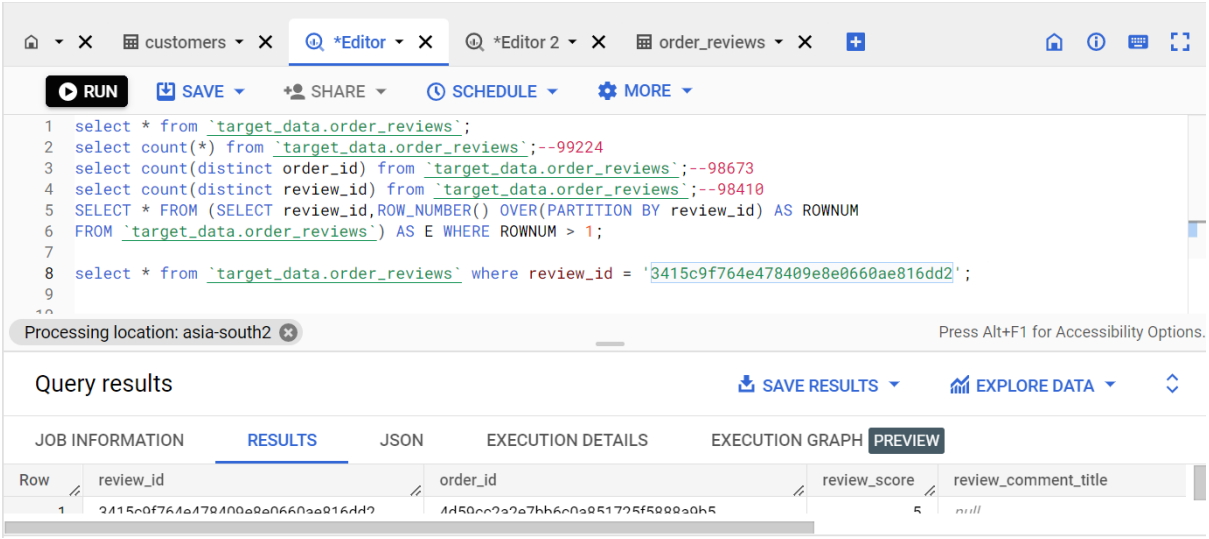
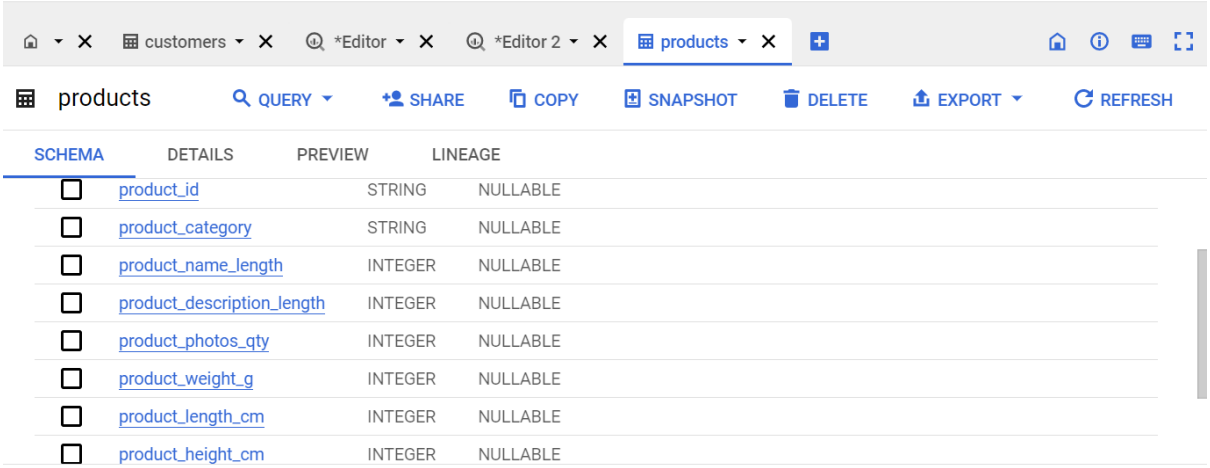
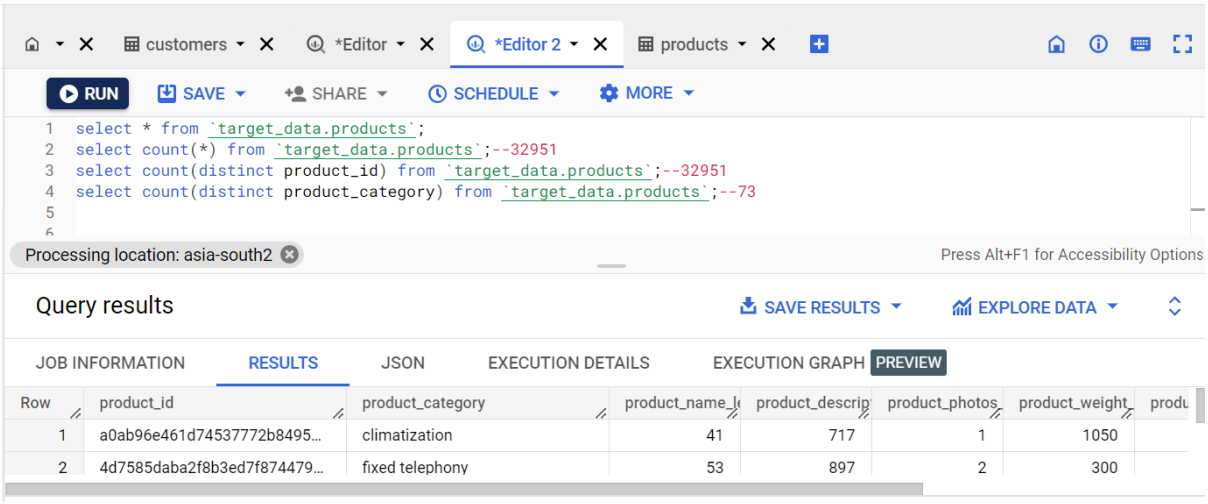
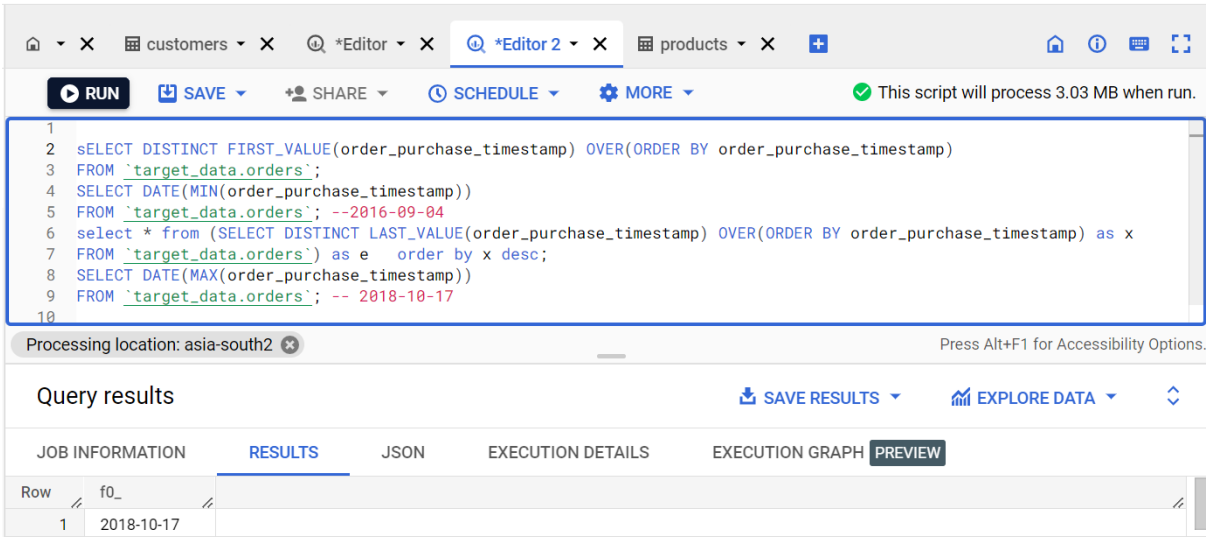


TABLE NAME : **products**

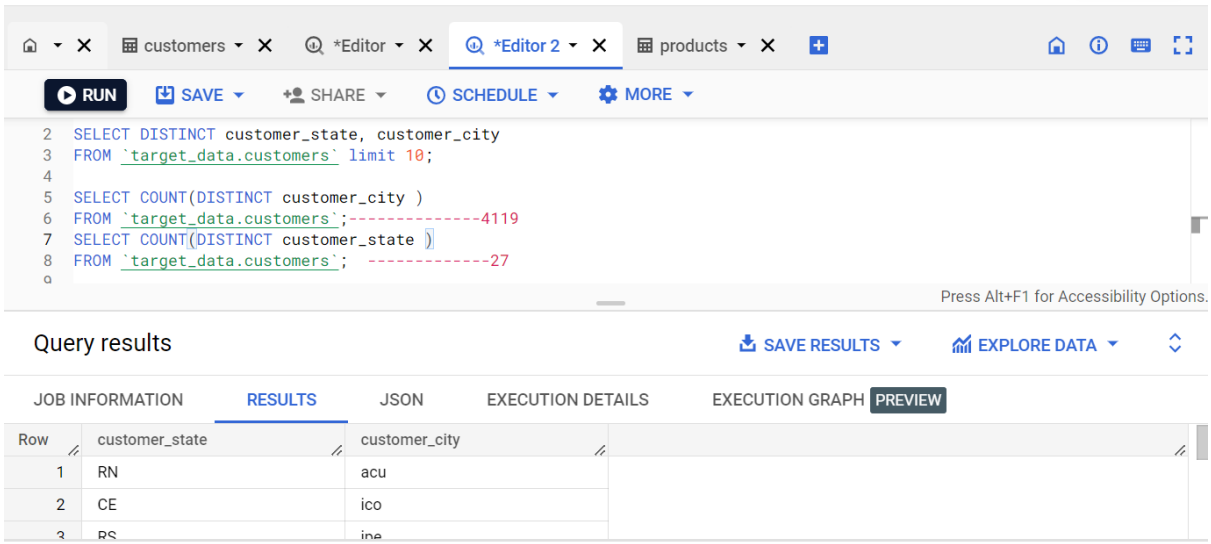




2. Time period for which the data is given (2016-09-04 TO 2018-10-17)

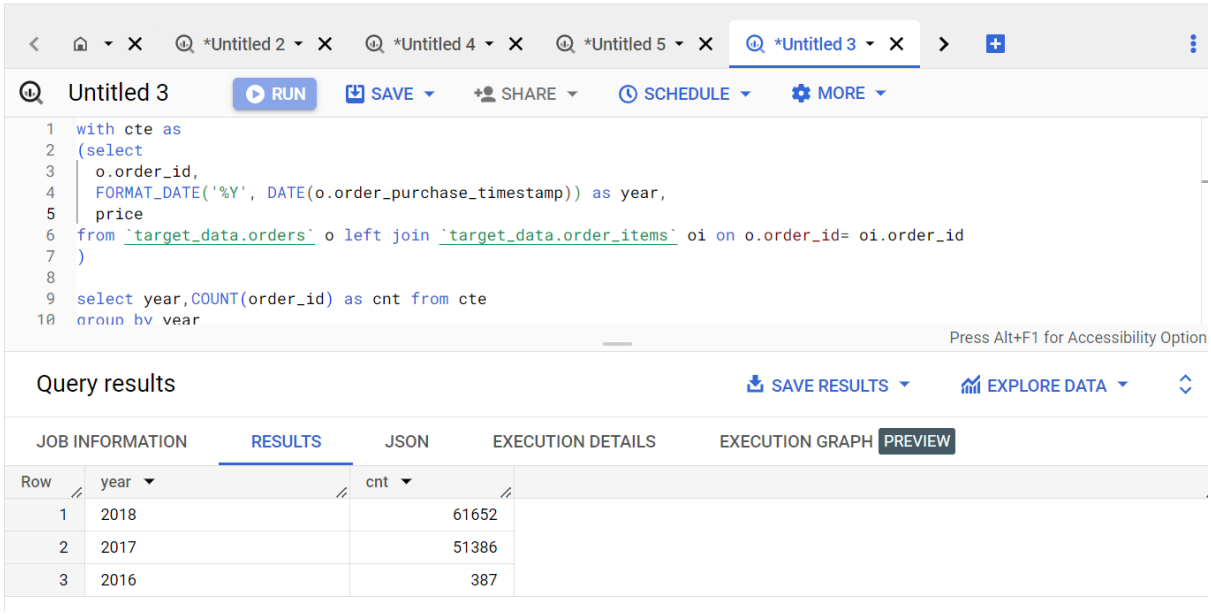


3. Cities and States of customers ordered during the given period



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?
        1. Yearly number of sales was increasing

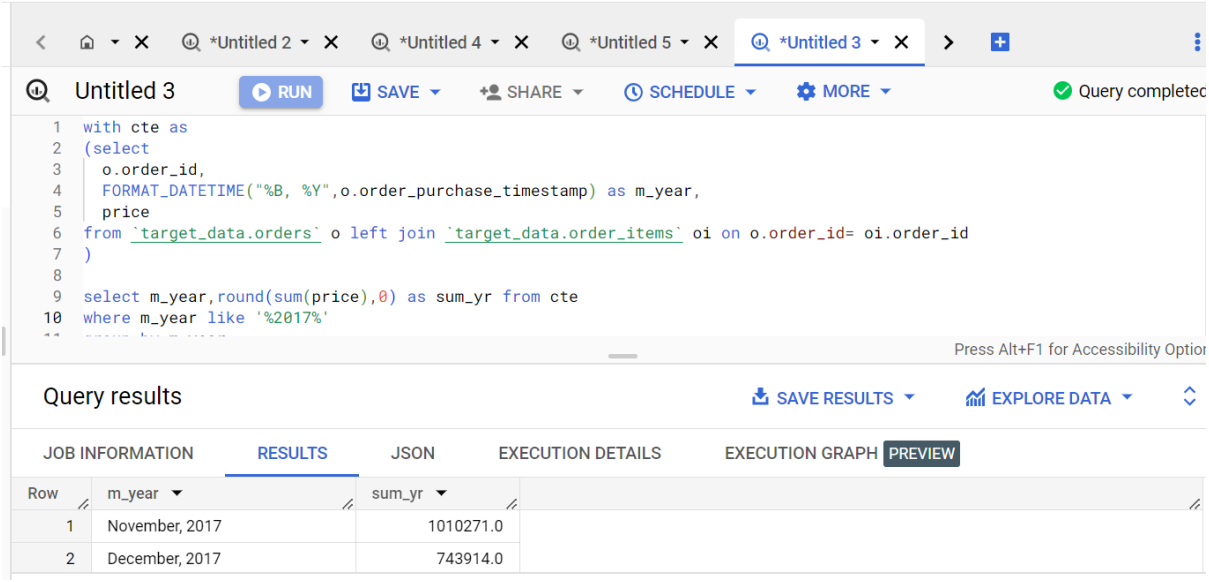
```sql
with cte as
(select
  o.order_id,
  FORMAT_DATE('%Y', DATE(o.order_purchase_timestamp)) as year,
  price
from `target_data.orders` o left join `target_data.order_items` oi on o.order_id= oi.order_id
)

select year,COUNT(order_id) as cnt from cte
group by year
order by cnt desc;
```



Row	year	cnt
1	2018	61652
2	2017	51386
3	2016	387

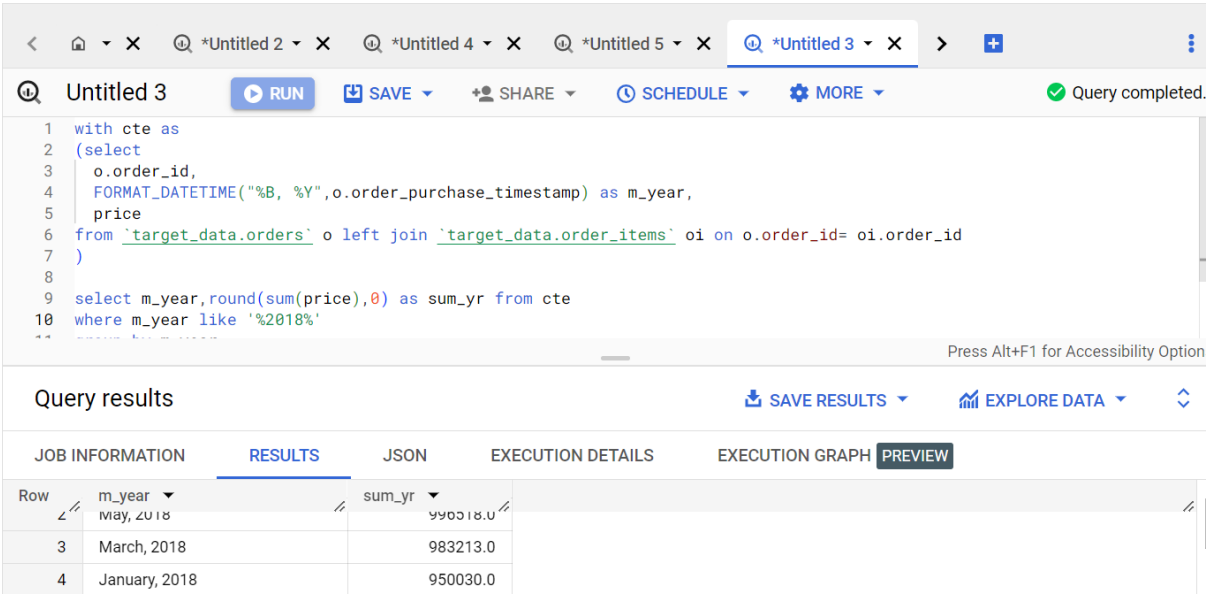
2. In 2017 sales started increasing, at NOV -17 it reached highest number of sales

```
with cte as
(select
  o.order_id,
  FORMAT_DATETIME("%B, %Y",o.order_purchase_timestamp) as m_year,
  price
from `target_data.orders` o left join `target_data.order_items` oi on o.order_id= oi.order_id
)

select m_year,round(sum(price),0) as sum_yr from cte
where m_year like '%2017%'
group by m_year
order by sum_yr desc;
```



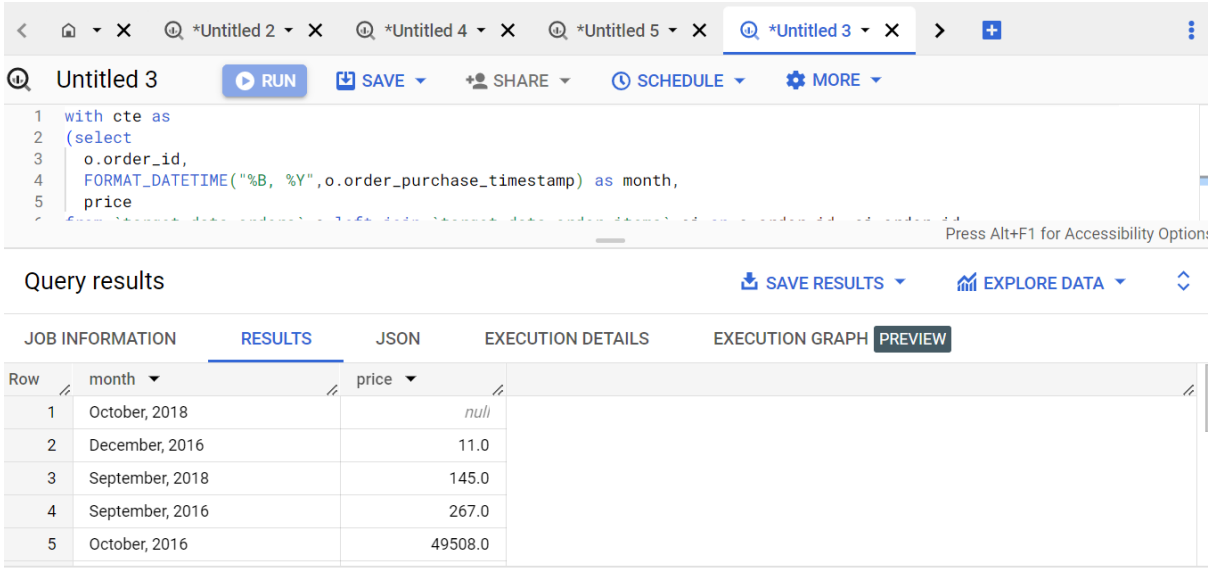
3. In 2018 – Initially the sales is in Peak, JAN-18, MAR-18 the sales is in peak, later sales maintained above 6k

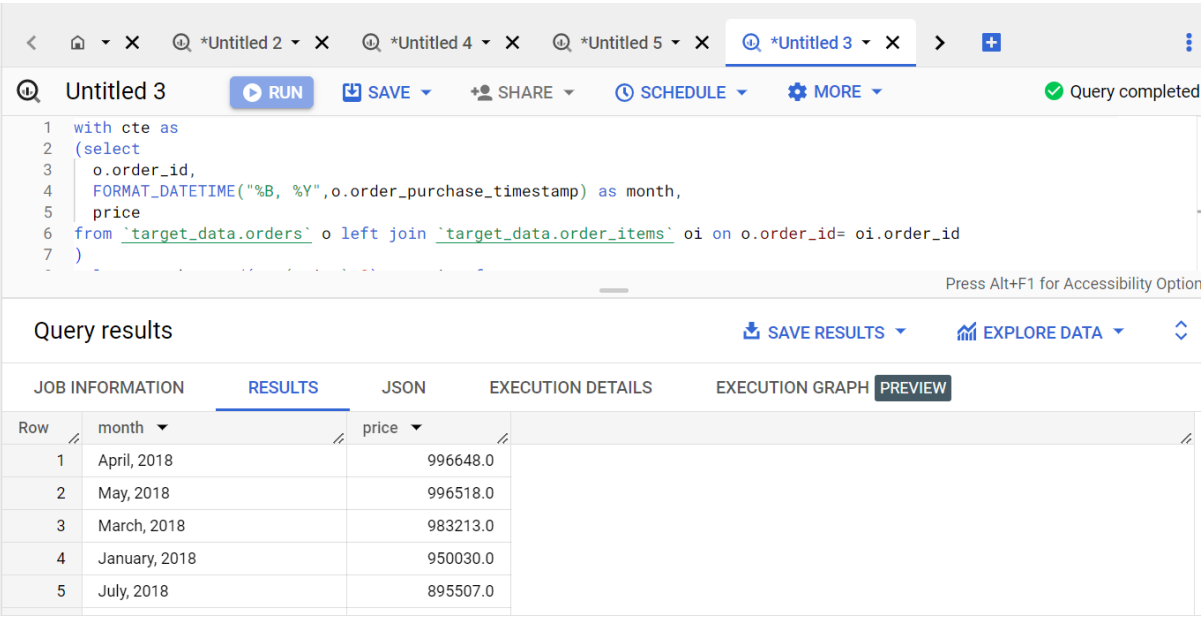
```sql
with cte as
(select
  o.order_id,
  FORMAT_DATETIME("%B, %Y",o.order_purchase_timestamp) as m_year,
  price
from `target_data.orders` o left join `target_data.order_items` oi on o.order_id= oi.order_id
)

select m_year,round(sum(price),0) as sum_yr from cte
where m_year like '%2018%'
group by m_year
order by sum_yr desc;
```

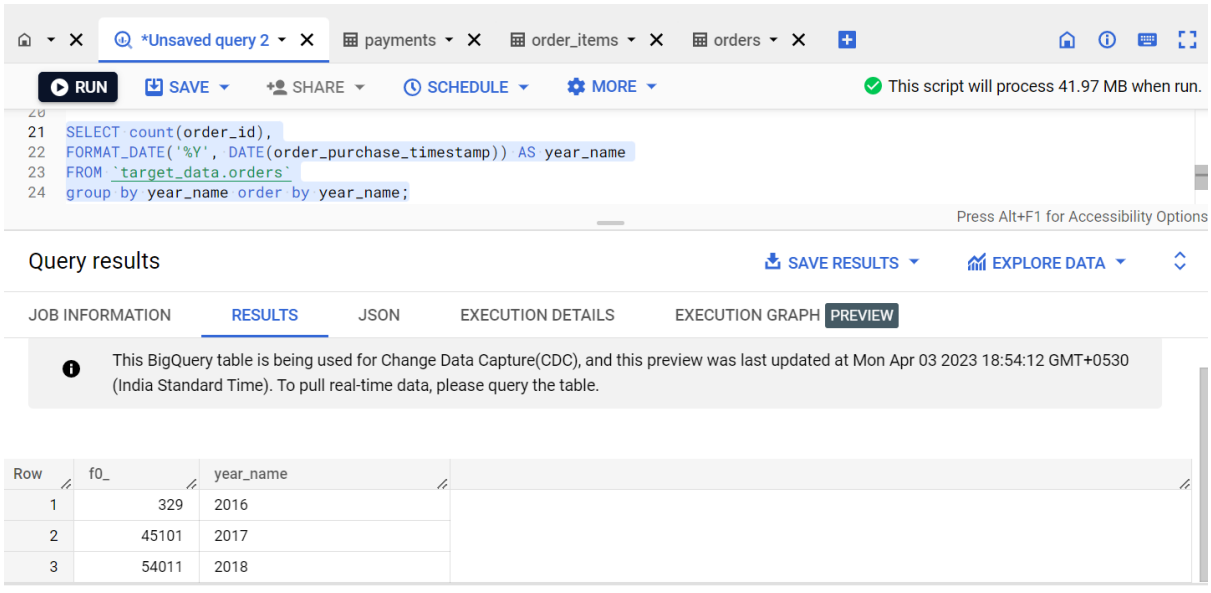


4. Lowest sales were in before DEC-16, SEP-18, OCT-18

```sql
with cte as
(select
  o.order_id,
  FORMAT_DATETIME("%B, %Y",o.order_purchase_timestamp) as month,
  price
from `target_data.orders` o left join `target_data.order_items` oi on o.order_id= oi.order_id
)
select month,round(sum(price),0) as price from cte
--where month in ('September, 2018','October, 2018','December, 2016')
group by month
order by price;
```


*Untitled 2
*Untitled 4
*Untitled 5
*Untitled 3
Untitled 3
RUN
SAVE
SHARE
SCHEDULE
MORE
1 with cte as
2 (select
3 o.order_id,
4 FORMAT_DATETIME("%B, %Y",o.order_purchase_timestamp) as month,
5 price
Press Alt+F1 for Accessibility Options
Query results
SAVE RESULTS
EXPLORE DATA
JOB INFORMATION
RESULTS
JSON
EXECUTION DETAILS
EXECUTION GRAPH
PREVIEW
Row month price
1 October, 2018 null
2 December, 2016 11.0
3 September, 2018 145.0
4 September, 2016 267.0
5 October, 2016 49508.0


*Untitled 2
*Untitled 4
*Untitled 5
*Untitled 3
Untitled 3
RUN
SAVE
SHARE
SCHEDULE
MORE
Query completed
1 with cte as
2 (select
3 o.order_id,
4 FORMAT_DATETIME("%B, %Y",o.order_purchase_timestamp) as month,
5 price
6 from `target_data.orders` o left join `target_data.order_items` oi on o.order_id= oi.order_id
7)
Press Alt+F1 for Accessibility Option
Query results
SAVE RESULTS
EXPLORE DATA
JOB INFORMATION
RESULTS
JSON
EXECUTION DETAILS
EXECUTION GRAPH
PREVIEW
Row month price
1 April, 2018 996648.0
2 May, 2018 996518.0
3 March, 2018 983213.0
4 January, 2018 950030.0
5 July, 2018 895507.0


*Unsaved query 2
payments
order_items
orders
RUN
SAVE
SHARE
SCHEDULE
MORE
This script will process 41.97 MB when run.
21 SELECT count(order_id),
22 FORMAT_DATE('%Y', DATE(order_purchase_timestamp)) AS year_name
23 FROM `target_data.orders`
24 group by year_name order by year_name;
Press Alt+F1 for Accessibility Options
Query results
SAVE RESULTS
EXPLORE DATA
JOB INFORMATION
RESULTS
JSON
EXECUTION DETAILS
EXECUTION GRAPH
PREVIEW
This BigQuery table is being used for Change Data Capture(CDC), and this preview was last updated at Mon Apr 03 2023 18:54:12 GMT+0530 (India Standard Time). To pull real-time data, please query the table.
Row f0_ year_name
1 329 2016
2 45101 2017
3 54011 2018

```sql
select * from `target_data.payments`;


SELECT order_purchase_timestamp, FORMAT_DATE('%B', DATE(order_purchase_timestamp)) AS month_name,
FORMAT_DATE('%Y', DATE(order_purchase_timestamp)) AS year_name
FROM `target_data.orders`;

SELECT order_purchase_timestamp,concat(FORMAT_DATE('%B', DATE(order_purchase_timestamp)),' ',FORMAT_DATE('%Y', DATE
(order_purchase_timestamp))) as month_year
FROM `target_data.orders`;

select count(order_id),month_year from
(
SELECT order_id,concat(FORMAT_DATE('%B', DATE(order_purchase_timestamp)),' ',FORMAT_DATE('%Y', DATE
(order_purchase_timestamp))) as month_year
FROM `target_data.orders`
) as a
group by  month_year
order by substr(month_year,-2,4);

```

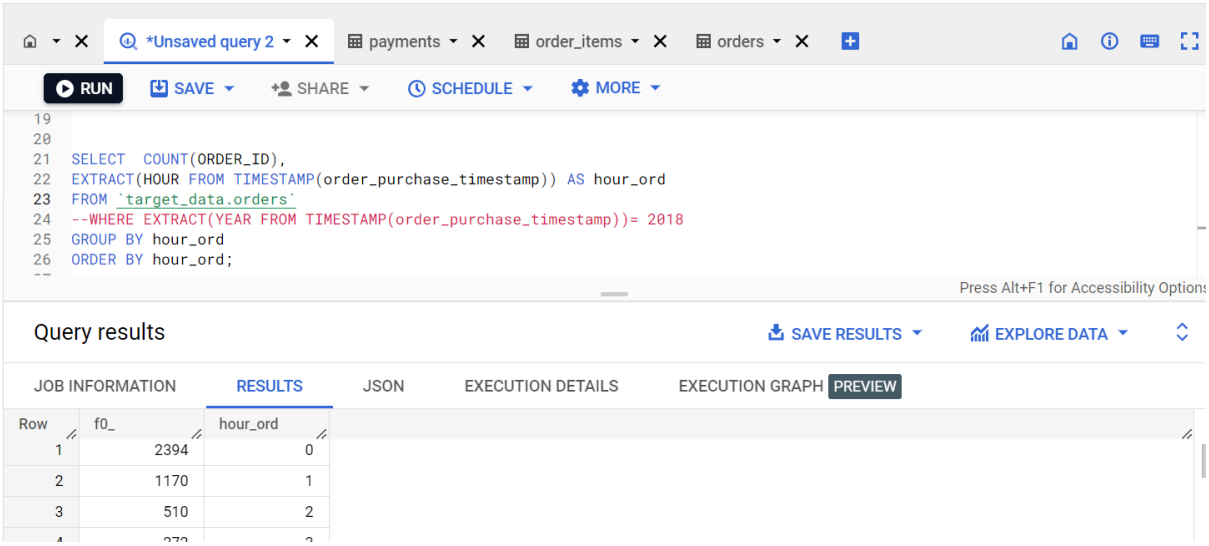
Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

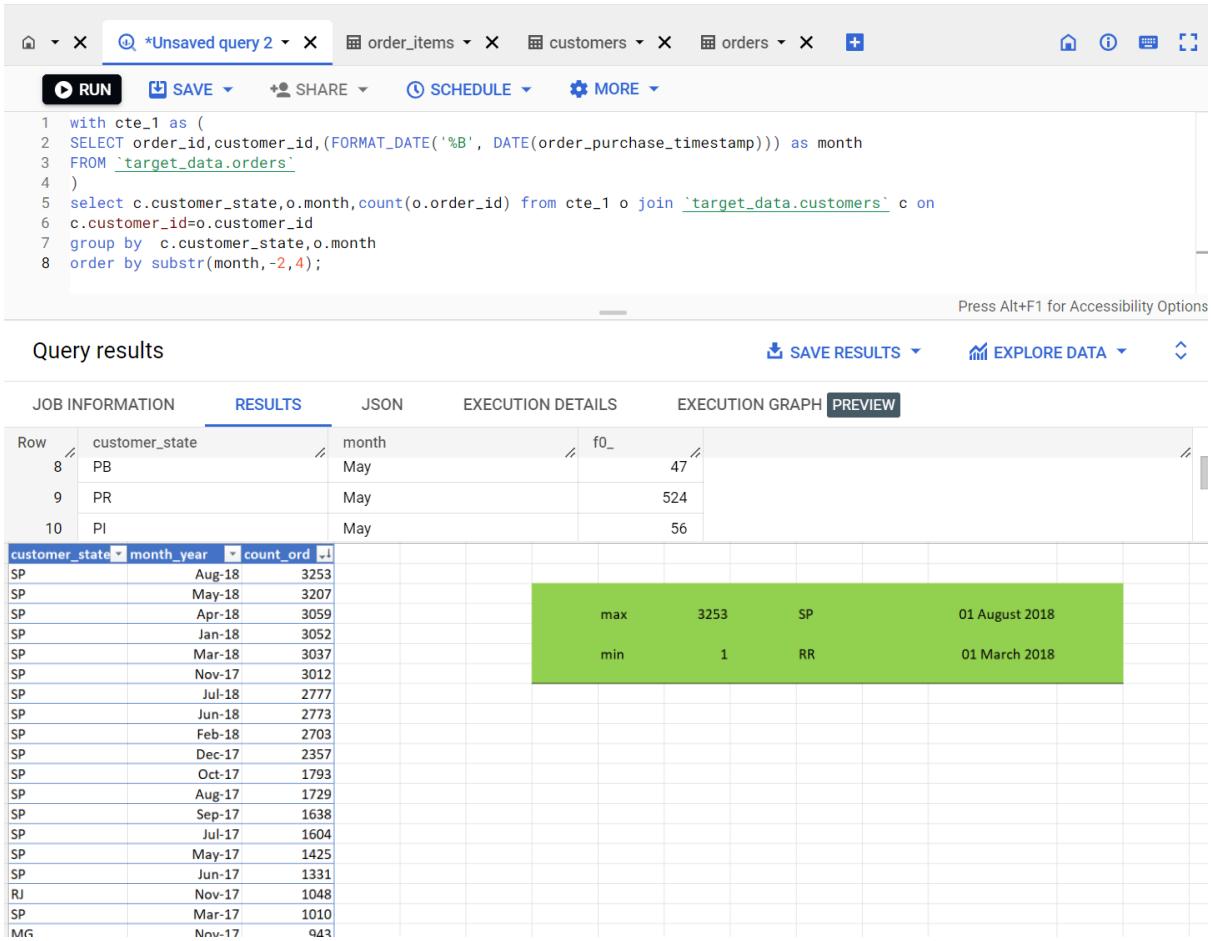
Row	f0_	month_year
1	324	October 2016
2	4	September 2016
3	1	December 2016
4	7544	November 2017
5	5673	December 2017
6	2404	April 2017

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
    1. Most orders placed in 9 to 22 (Morning and Afternoon)
    2. Maximum orders received 4 o clock (Afternoon)
    3. Minimum numbers of orders received between 1 and 6 o clock (Morning)

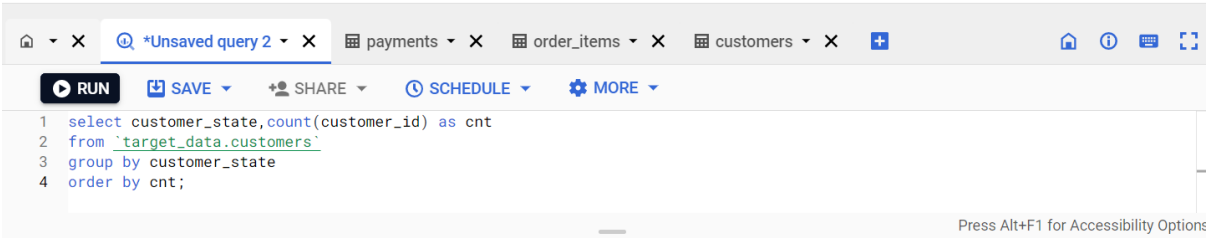


order_Count	hour
2394	0
1170	1
510	2
272	3
206	4
188	5
502	6
1231	7
2967	8
4785	9
6177	10
6578	11
5995	12
6518	13
6569	14
6454	15
6675	16
6150	17
5769	18
5982	19
6193	20
6217	21
5816	22
4123	23

3. Evolution of E-commerce orders in the Brazil region:
    1. Get month on month orders by states
        1. Maximum orders placed in the state of SP -São Paulo in the month of AUG-18 and also many orders were placed in the month of MAR- 2018 to MAY -2018 in the state of São Paulo
        2. Minimum ordered state are RR - Roraima in SEP,OCT,DEC months



    2. Distribution of customers across the states in Brazil
        1. In Brazil maximum customers are in state SP -São Paulo
        2. Minimum customers are in state of RR- Roraima



Query results

SAVE RESULTS | EXPLORE DATA

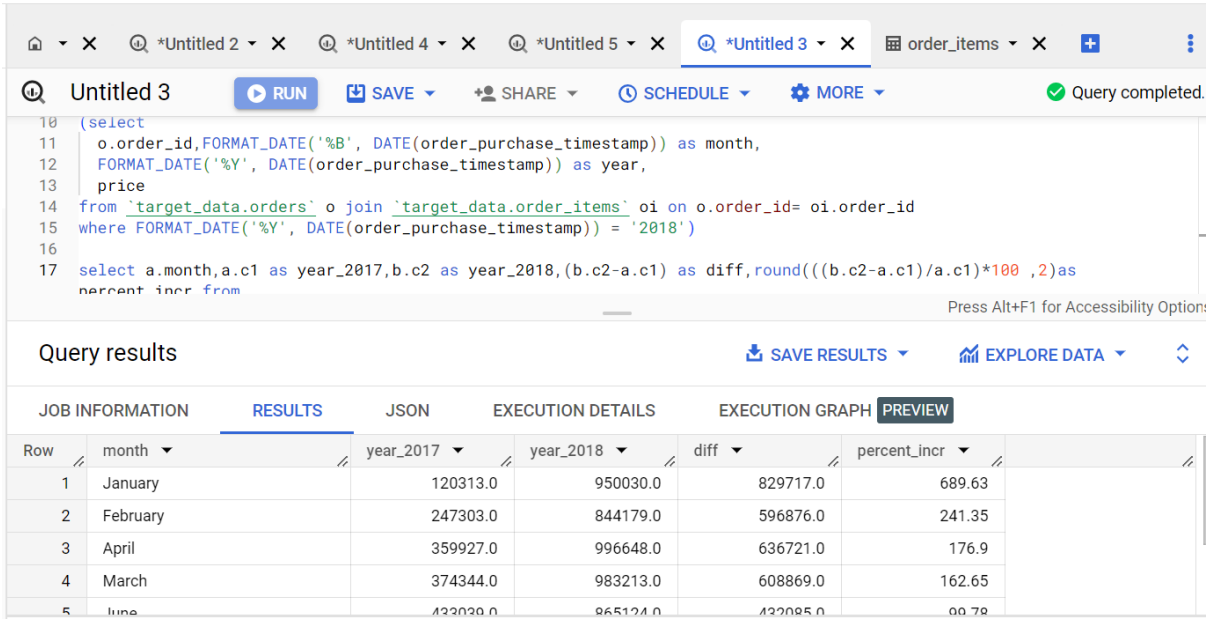
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	cnt
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253

customer_state	cnt				
RR	46				
AP	68				
AC	81		Minimum	46	RR
AM	148				
RO	253		Maximum	41746	SP
TO	280				
SE	350				
AL	413				
RN	485				
PI	495				
PB	536				
MS	715				
MA	747				
MT	907				
PA	975				
CE	1336				
PE	1652				
GO	2020				
ES	2033				
DF	2140				
BA	3380				
SC	3637				
PR	5045				
RS	5466				
MG	11635				
RJ	12852				
SP	41746				

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

```sql
with cte as
(select
  o.order_id,FORMAT_DATE('%B', DATE(order_purchase_timestamp)) as month,
  FORMAT_DATE('%Y', DATE(order_purchase_timestamp)) as year,
  price
from `target_data.orders` o join `target_data.order_items` oi on o.order_id= oi.order_id
where FORMAT_DATE('%Y', DATE(order_purchase_timestamp)) = '2017'),

cte2 as
(select
  o.order_id,FORMAT_DATE('%B', DATE(order_purchase_timestamp)) as month,
  FORMAT_DATE('%Y', DATE(order_purchase_timestamp)) as year,
  price
from `target_data.orders` o join `target_data.order_items` oi on o.order_id= oi.order_id
where FORMAT_DATE('%Y', DATE(order_purchase_timestamp)) = '2018')

select a.month,a.c1,b.c2,(b.c2-a.c1) as diff,round(((b.c2-a.c1)/a.c1)*100 ,2)as percent_incr from
(select cte.month, round(sum(cte.price),0) as c1 from cte group by cte.month ) as a
join
(select cte2.month, round(sum(cte2.price),0) as c2 from cte2 group by cte2.month) as b
on a.month = b.month
and a.month not in ('September','October','November','December')
order by percent_incr desc;
```



Row	month	year_2017	year_2018	diff	percent_incr
1	January	120313.0	950030.0	829717.0	689.63
2	February	247303.0	844179.0	596876.0	241.35
3	April	359927.0	996648.0	636721.0	176.9
4	March	374344.0	983213.0	608869.0	162.65
5	June	433039.0	865124.0	432085.0	99.78

2. Mean & Sum of price and freight value by customer state
    1. Maximum Mean price value of customer state is PB – Paraíba
    2. Minimum mean price of customer state is SP- São Paulo
    3. Maximum mean freight value of customer state is – RR – Roraima
    4. Minimum mean freight value of customer state is – SP- São Paulo
    5. Maximum total price of customer state is - is SP- São Paulo
    6. Minimum total price of customer state is - RR – Roraima

```
6
7  select c.customer_state,sum(oi.price),sum(oi.freight_value),avg(oi.price),avg(oi.freight_value)
8  from `target_data.customers` c join `target_data.orders` o on c.customer_id=o.customer_id
9  join `target_data.order_items` oi on o.order_id=oi.order_id
10 group by c.customer_state
11 order by avg(oi.price)desc;
```

Press Alt+F1 for Accessibility Option

Query results

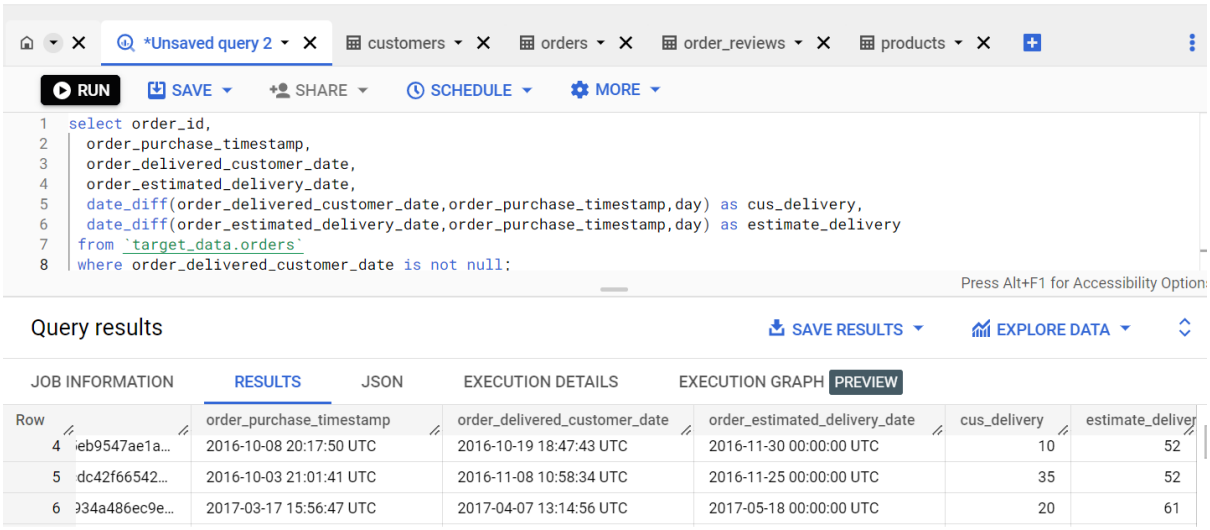
SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

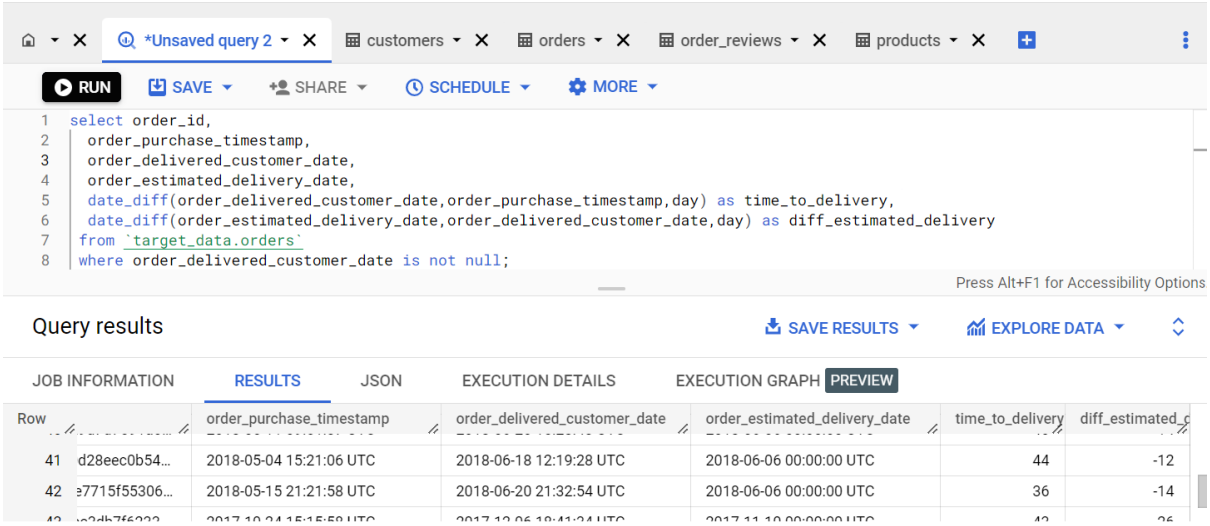
Row	customer_state	f0_	f1_	f2_	f3_
1	PB	115268.080...	25719.7299...	191.475215...	42.7238039...
2	AL	80314.8100...	15914.5899...	180.889211...	35.8436711...
3	AC	15982.9499...	3686.74999...	173.727717...	40.0733695...
4	RO	46140.6400...	11417.38	165.973525...	41.0697122...

5. Analysis on sales, freight and delivery time

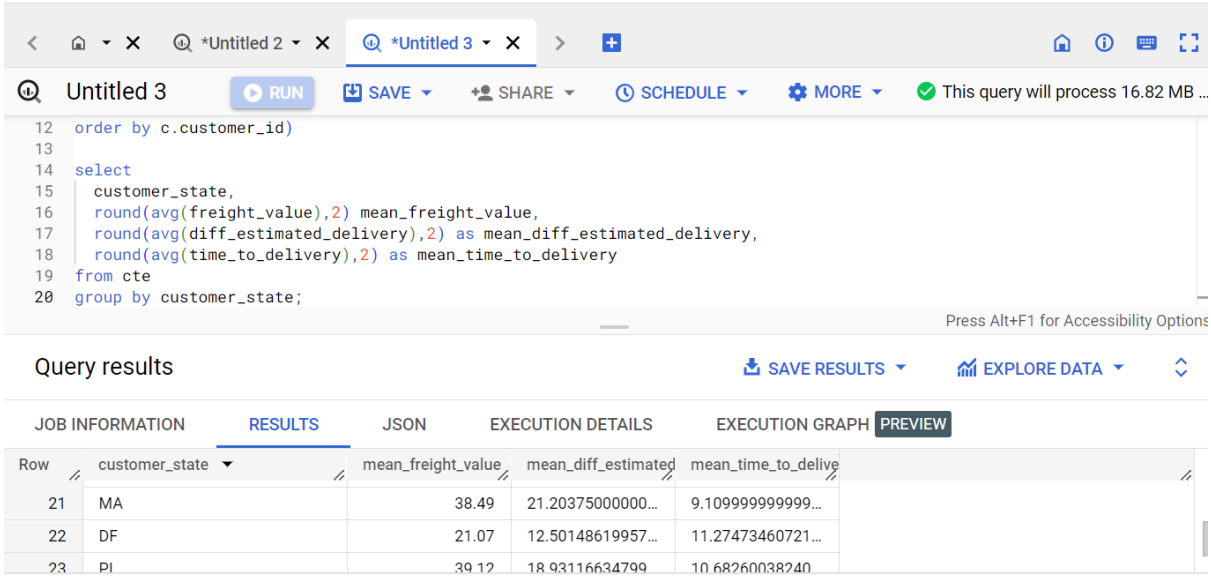
    1. Calculate days between purchasing, delivering and estimated delivery



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

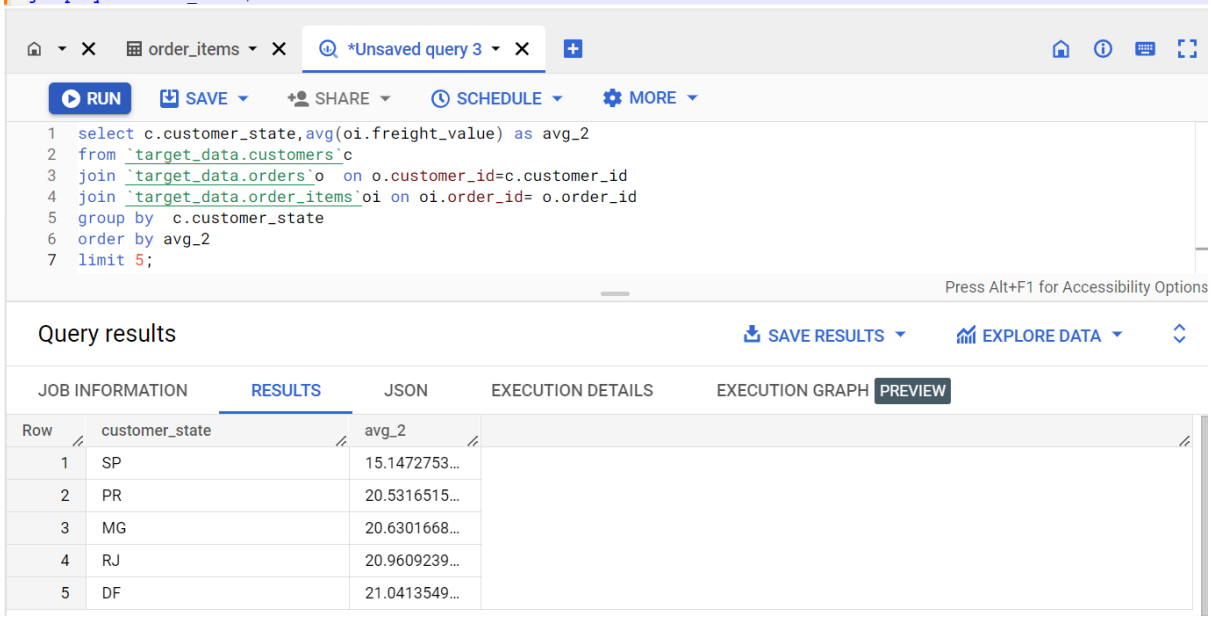


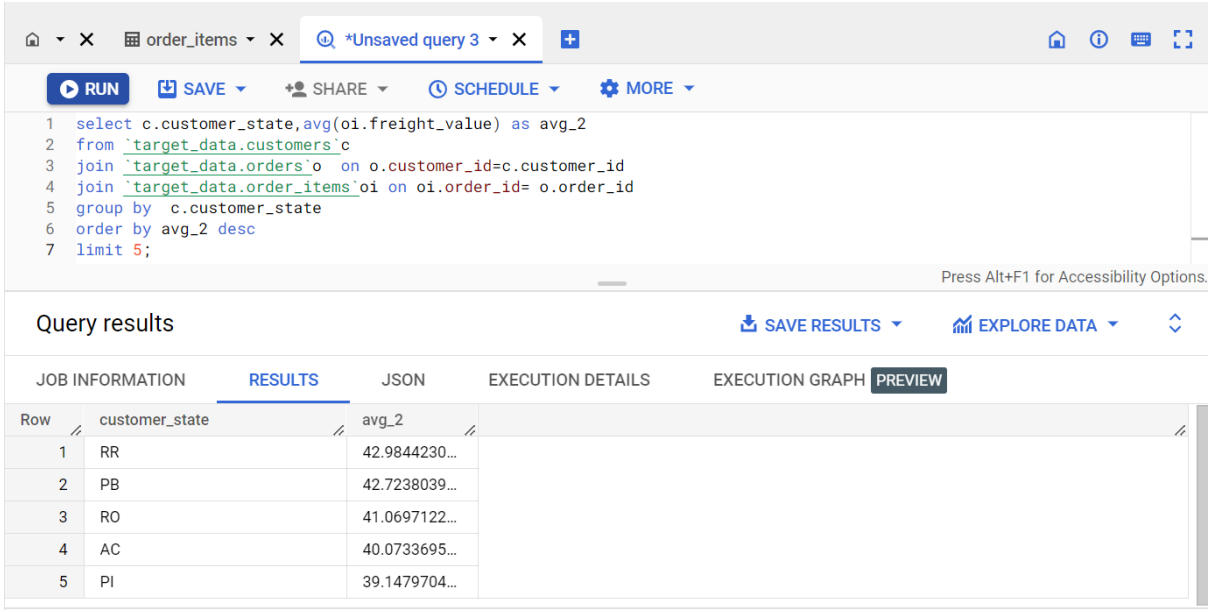
    3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery



4. Sort the data to get the following:
    1. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

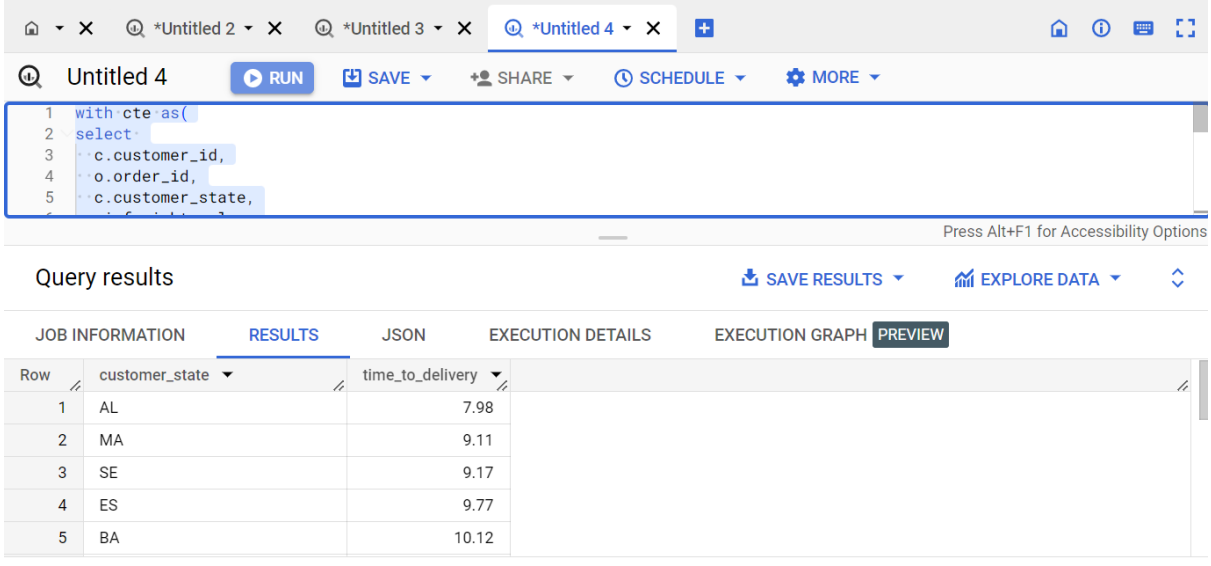
```sql
with cte as(
select
  c.customer_id,
  o.order_id,
  c.customer_state,
  oi.freight_value,
  date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as diff_estimated_delivery,
  date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as time_to_delivery
from `target-382407.target_data.customers` c join `target-382407.target_data.orders` o on c.customer_id= o.customer_id
join `target-382407.target_data.order_items` oi on o.order_id= oi.order_id
where o.order_delivered_customer_date is not null
order by c.customer_id)

select
  customer_state,
  round(avg(freight_value),2) mean_freight_value,
  round(avg(diff_estimated_delivery),2) as mean_diff_estimated_delivery,
  round(avg(time_to_delivery),2) as mean_time_to_delivery
from cte
group by customer_state;
```





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

```sql
with cte as(
select
  c.customer_id,
  o.order_id,
  c.customer_state,
  oi.freight_value,
  date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as diff_estimated_delivery,
  date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as time_to_delivery
from `target-382407.target_data.customers` c join `target-382407.target_data.orders` o on c.customer_id= o.customer_id
join `target-382407.target_data.order_items` oi on o.order_id= oi.order_id
where o.order_delivered_customer_date is not null
order by c.customer_id)

select
  customer_state,
  round(avg(time_to_delivery),2) as time_to_delivery
from
  cte
group by customer_state
order by time_to_delivery;
```



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

```sql
with cte as(
select
  c.customer_id,
  o.order_id,
  c.customer_state,
  oi.freight_value,
  date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as diff_estimated_delivery,
  date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as time_to_delivery
from `target-382407.target_data.customers` c join `target-382407.target_data.orders` o on c.customer_id= o.customer_id
join `target-382407.target_data.order_items` oi on o.order_id= oi.order_id
where o.order_delivered_customer_date is not null
order by c.customer_id)

select * from
(select
  customer_state,
  round(avg(time_to_delivery),2) as time_to_delivery,
  round(avg(diff_estimated_delivery),2) as estimated_delivery
from
  cte
group by customer_state
order by time_to_delivery)as a
where time_to_delivery > estimated_delivery;
```

Untitled 4 RUN SAVE SHARE SCHEDULE MORE

```sql
  round(avg(time_to_delivery),2) as time_to_delivery,
  round(avg(diff_estimated_delivery),2) as estimated_delivery
from
  cte
group by customer_state
order by time_to_delivery)as a
where time_to_delivery > estimated_delivery;
```

Press Alt+F1 for Accessibility Options.

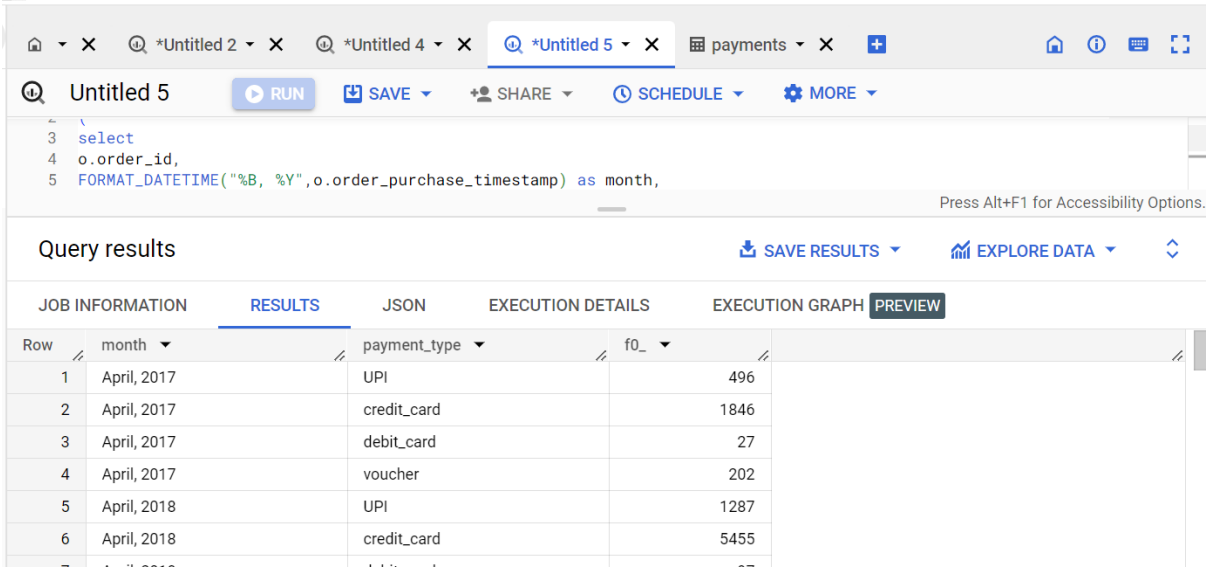
Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	time_to_delivery	estimated_delivery
1	SP	10.27	8.26
2	MG	12.4	11.52
3	PR	12.53	11.48

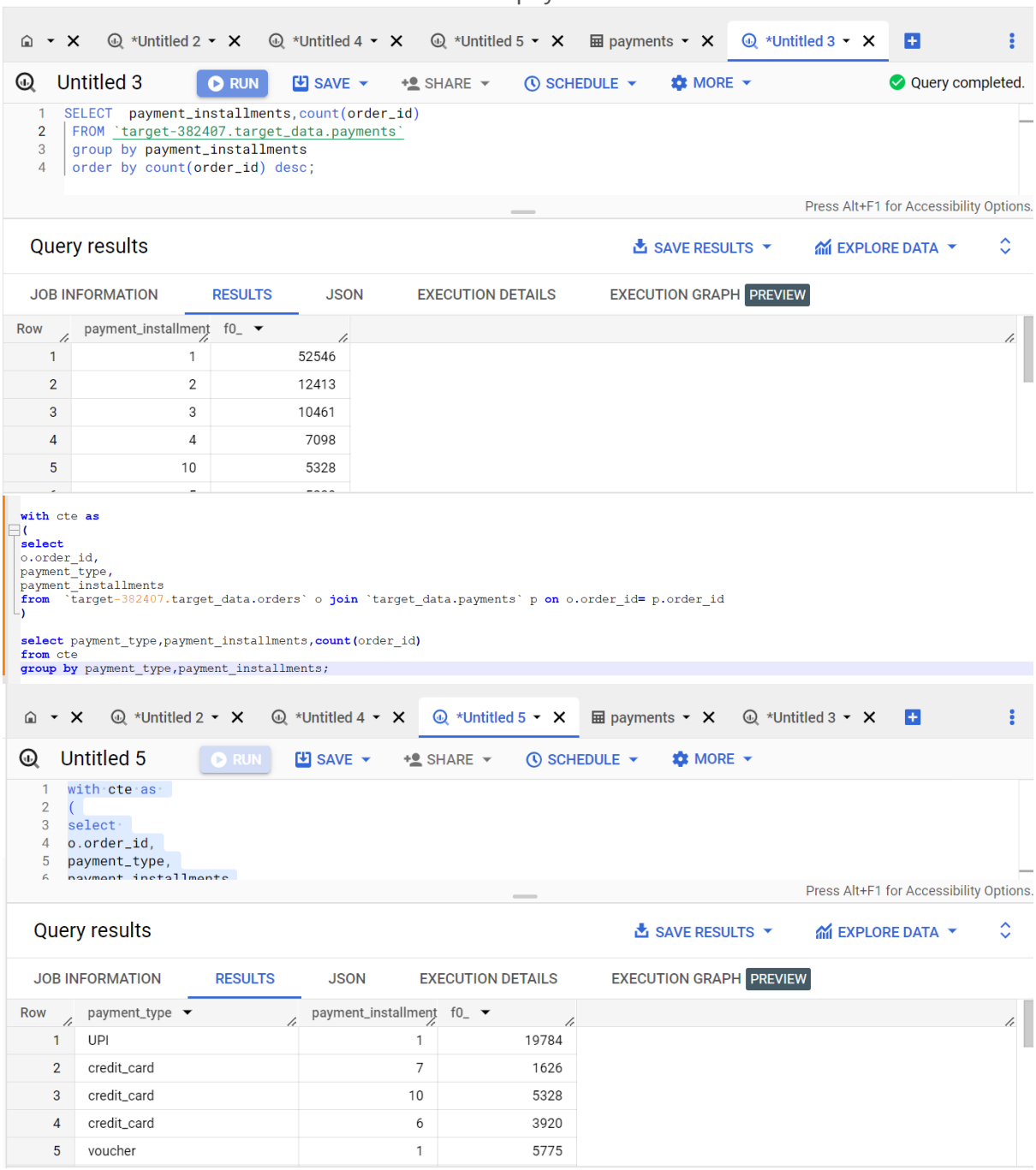
6. Payment type analysis:
    1. Month over Month count of orders for different payment types

```sql
with cte as
(
select
o.order_id,
FORMAT_DATETIME("%B, %Y",o.order_purchase_timestamp) as month,
payment_type
from  `target-382407.target_data.orders` o join `target_data.payments` p on o.order_id= p.order_id
)

select month,payment_type,count(order_id)
from cte
group by month,payment_type
order by month,payment_type;
```



Row	month	payment_type	f0_
1	April, 2017	UPI	496
2	April, 2017	credit_card	1846
3	April, 2017	debit_card	27
4	April, 2017	voucher	202
5	April, 2018	UPI	1287
6	April, 2018	credit_card	5455

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



```
SELECT  payment_installments,count(order_id)
FROM `target-382407.target_data.payments`
group by payment_installments
order by count(order_id) desc;
```

Query results

Row	payment_installment	f0_
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328

```
with cte as
(
select
o.order_id,
payment_type,
payment_installments
from  `target-382407.target_data.orders` o join `target_data.payments` p on o.order_id= p.order_id
)

select payment_type,payment_installments,count(order_id)
from cte
group by payment_type,payment_installments;
```



```
with cte as
(
select
o.order_id,
payment_type,
```

Query results

Row	payment_type	payment_installment	f0_
1	UPI	1	19784
2	credit_card	7	1626
3	credit_card	10	5328
4	credit_card	6	3920
5	voucher	1	5775

7. Actionable Insights
    1. People of buying in the single instalment is more around 52546 so much of wealthiest people living in SP state, we can provide more stores so that they No of customers got increased
    2. From the state RR of a smaller number of people buying items there will give more offers so that will attract more customers it leads to improve the sales over there
    3. May and August having most sales, will minimize the freight value on that month let the customers will purchase more

8. Recommendations
    1. mean fright value varies between 15 to 45
    2. Finding the better place or any good storage places nearby each state it indirectly reduces the fight value
    3. It will more affordable customer can purchase more items more often
    4. September and October month having minimum amount of sales happen, Focus more advertisement on that month and Give more offers to customers
    5. Conduct Big billion days kinds of sales and offers can improve the customers strength
    6. AP and RR state was having more number of days to take delivered, To improve the time to deliver and minimize the amount of days to deliver in these state, Implementing the local goods storage better improve the time to deliver