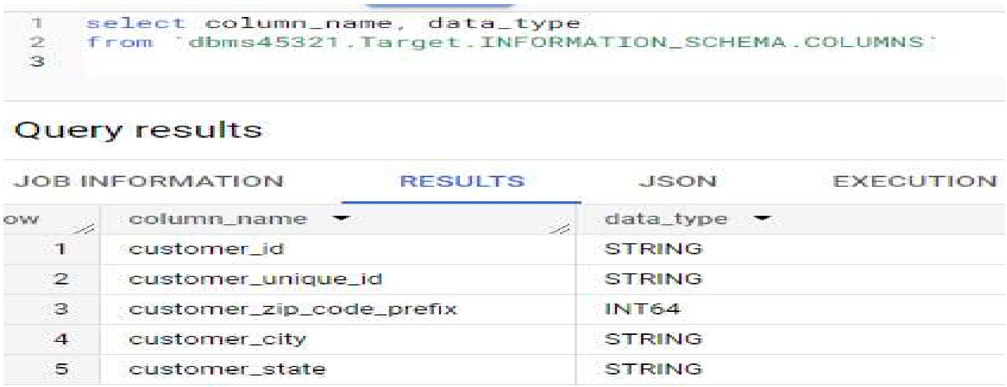
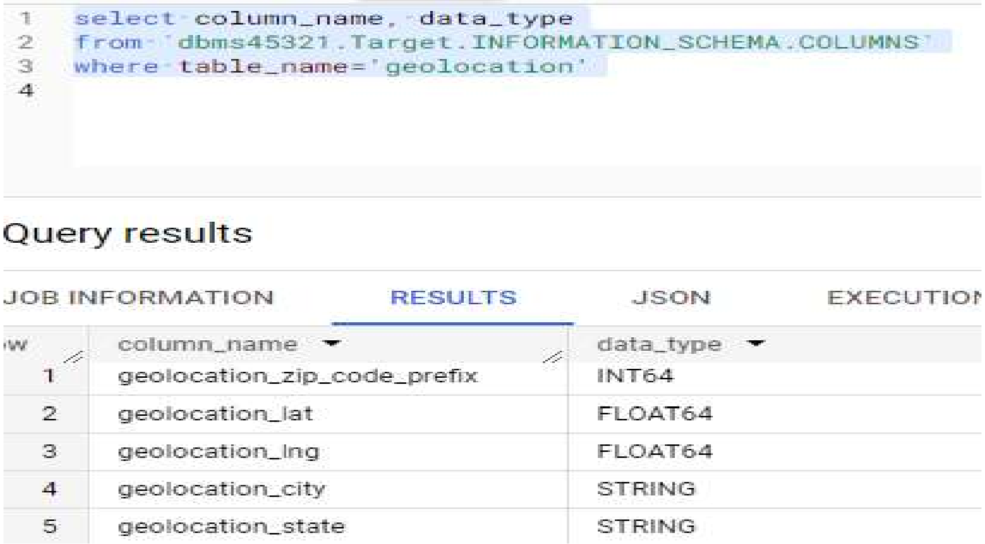
## 1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

### 1. Data type of columns in a table



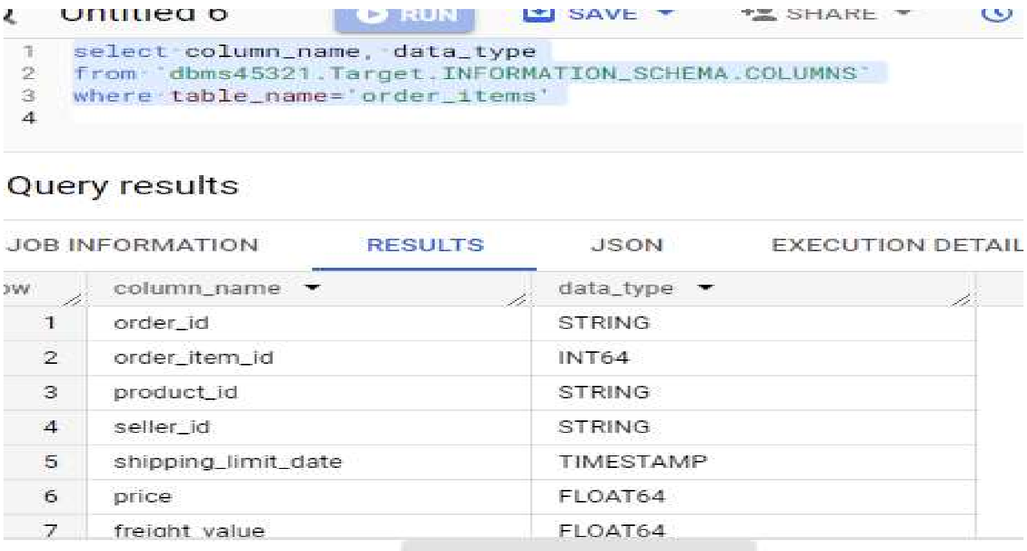
```sql
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`

```

| ow | 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 |



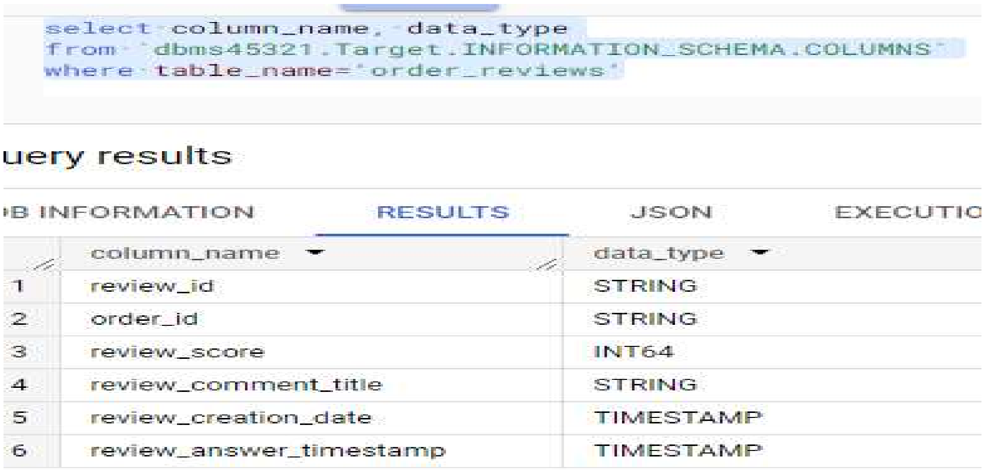
```sql
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name='geolocation'

```

| w | column_name | data_type |
|---|---|---|
| 1 | geolocation_zip_code_prefix | INT64 |
| 2 | geolocation_lat | FLOAT64 |
| 3 | geolocation_lng | FLOAT64 |
| 4 | geolocation_city | STRING |
| 5 | geolocation_state | STRING |



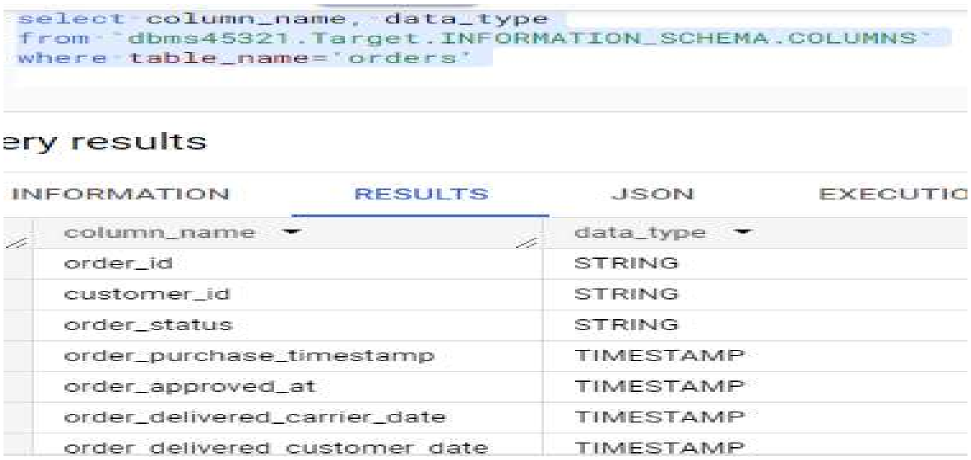
```sql
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name='order_items'

```

| ow | column_name | data_type |
|---|---|---|
| 1 | order_id | STRING |
| 2 | order_item_id | INT64 |
| 3 | product_id | STRING |
| 4 | seller_id | STRING |
| 5 | shipping_limit_date | TIMESTAMP |
| 6 | price | FLOAT64 |
| 7 | freight_value | FLOAT64 |



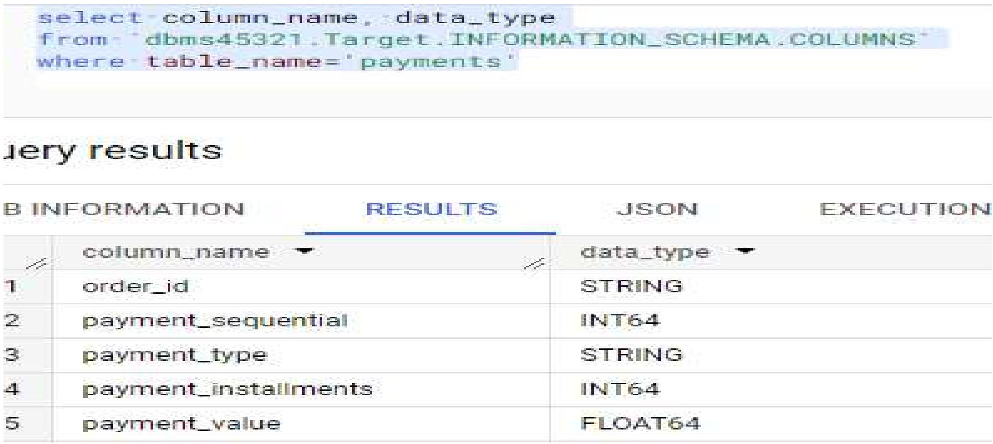
```
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name='order_reviews'
```

|  | column_name | data_type |
|---|---|---|
| 1 | review_id | STRING |
| 2 | order_id | STRING |
| 3 | review_score | INT64 |
| 4 | review_comment_title | STRING |
| 5 | review_creation_date | TIMESTAMP |
| 6 | review_answer_timestamp | TIMESTAMP |



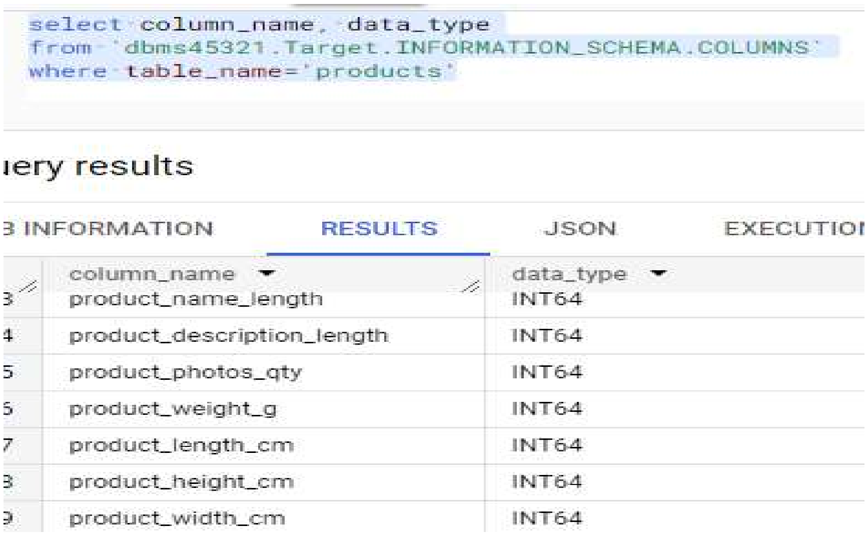
```
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name='orders'
```

| column_name | data_type |
|---|---|
| order_id | STRING |
| customer_id | STRING |
| order_status | STRING |
| order_purchase_timestamp | TIMESTAMP |
| order_approved_at | TIMESTAMP |
| order_delivered_carrier_date | TIMESTAMP |
| order_delivered_customer_date | TIMESTAMP |



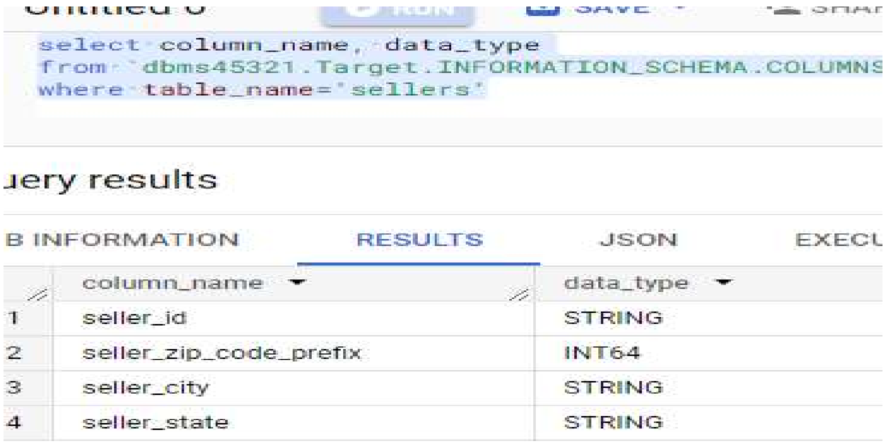
```
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name='payments'
```

|  | column_name | data_type |
|---|---|---|
| 1 | order_id | STRING |
| 2 | payment_sequential | INT64 |
| 3 | payment_type | STRING |
| 4 | payment_installments | INT64 |
| 5 | payment_value | FLOAT64 |



```
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name='products'
```

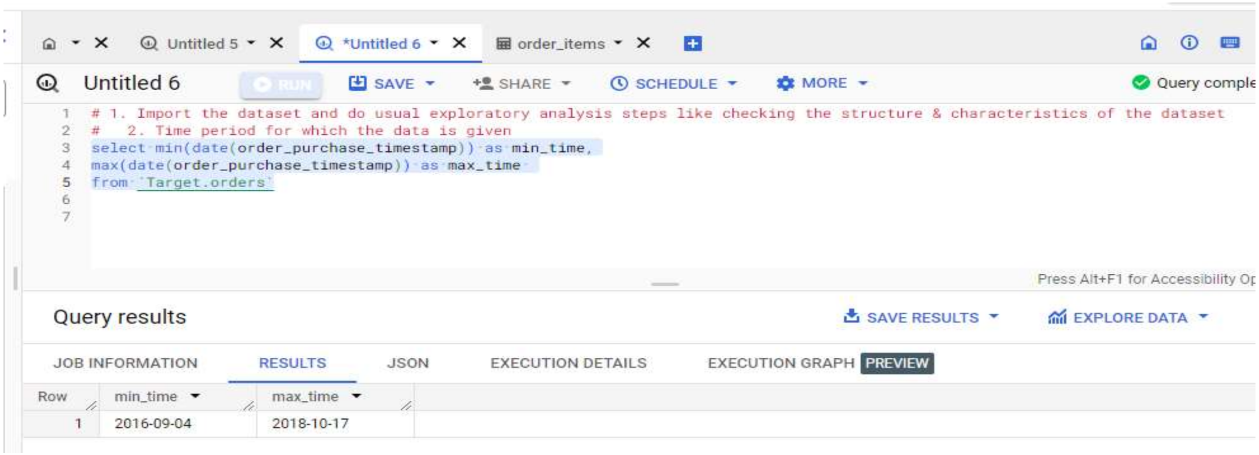
| | column_name | data_type |
|---|---|---|
| 3 | product_name_length | INT64 |
| 4 | product_description_length | INT64 |
| 5 | product_photos_qty | INT64 |
| 6 | product_weight_g | INT64 |
| 7 | product_length_cm | INT64 |
| 8 | product_height_cm | INT64 |
| 9 | product_width_cm | INT64 |



```
select column_name, data_type
from `dbms45321.Target.INFORMATION_SCHEMA.COLUMNS
where table_name='sellers'
```

| | column_name | data_type |
|---|---|---|
| 1 | seller_id | STRING |
| 2 | seller_zip_code_prefix | INT64 |
| 3 | seller_city | STRING |
| 4 | seller_state | STRING |

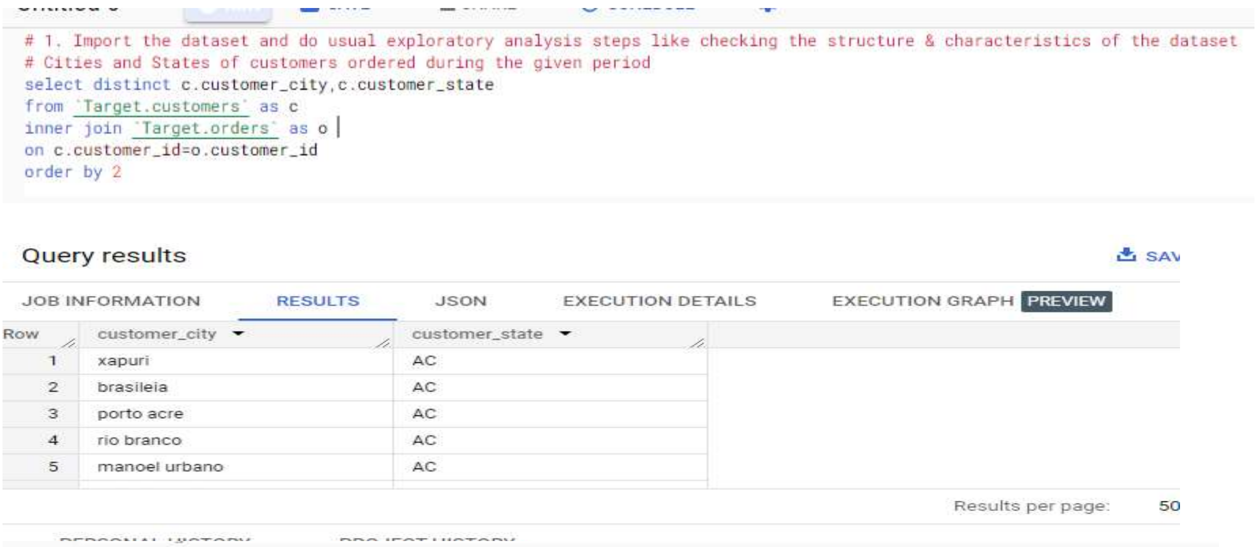
2. Time period for which the data is given



```
# 1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
#   2. Time period for which the data is given
select min(date(order_purchase_timestamp)) as min_time,
max(date(order_purchase_timestamp)) as max_time
from `Target.orders`
```

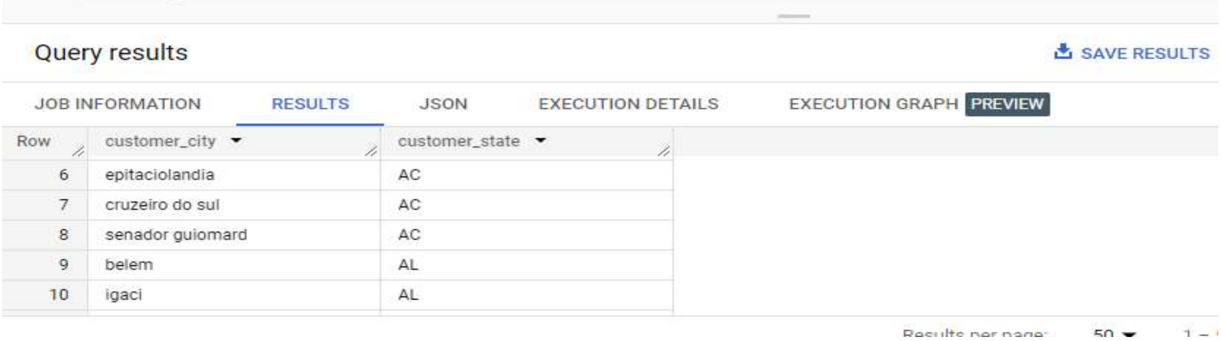
| Row | min_time | max_time |
|---|---|---|
| 1 | 2016-09-04 | 2018-10-17 |

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



```
# 1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
# Cities and States of customers ordered during the given period
select distinct c.customer_city,c.customer_state
from `Target.customers` as c
inner join `Target.orders` as o
on c.customer_id=o.customer_id
order by 2
```

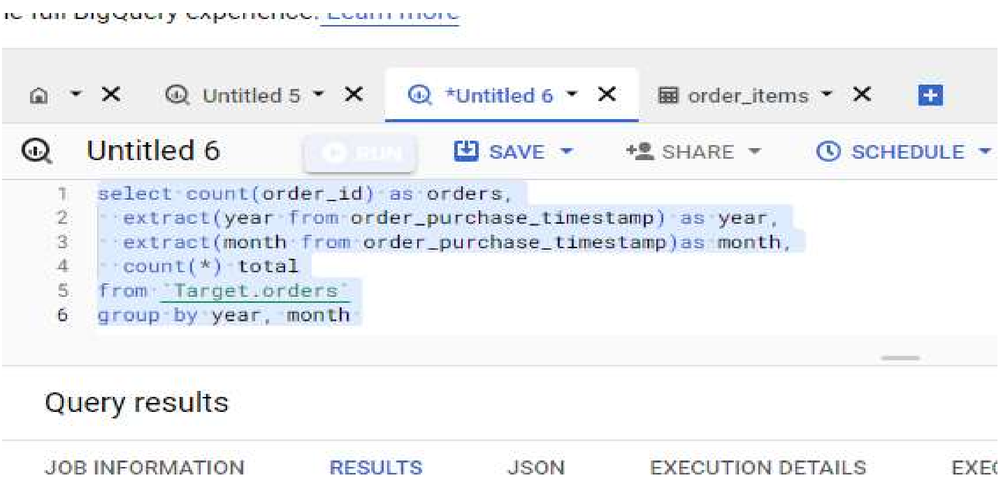
| Row | customer_city | customer_state |
|---|---|---|
| 1 | xapuri | AC |
| 2 | brasileia | AC |
| 3 | porto acre | AC |
| 4 | rio branco | AC |
| 5 | manoel urbano | AC |



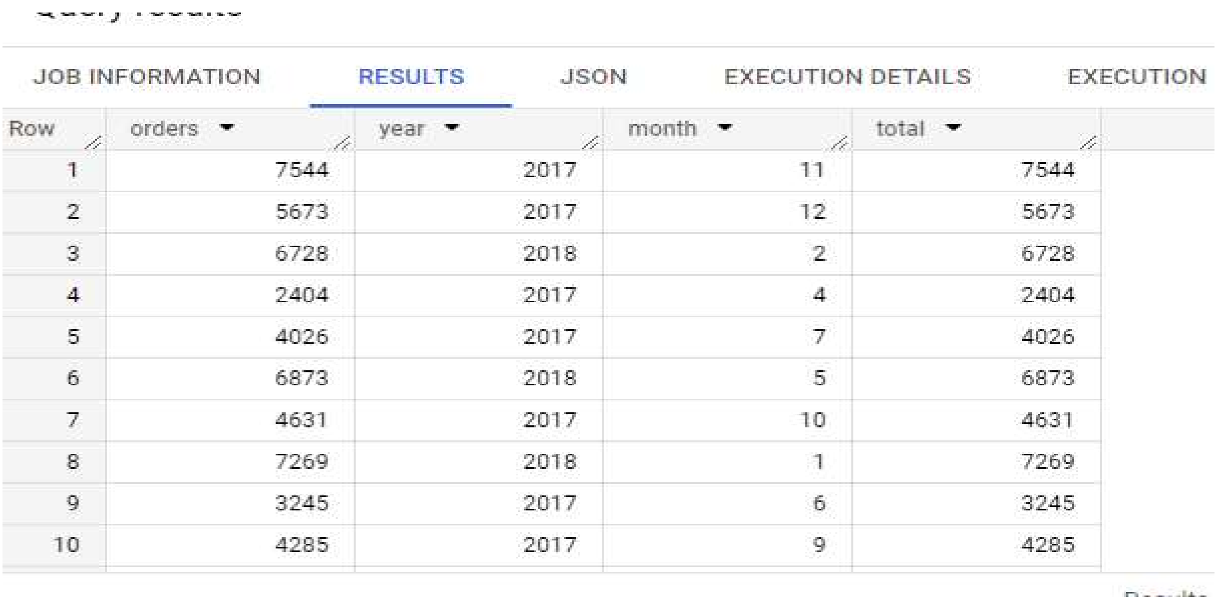
| Row | customer_city | customer_state |
|---|---|---|
| 6 | epitaciolandia | AC |
| 7 | cruzeiro do sul | AC |
| 8 | senador guiomard | AC |
| 9 | belem | AL |
| 10 | igaci | AL |

## 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?



```
select count(order_id) as orders,
  extract(year from order_purchase_timestamp) as year,
  extract(month from order_purchase_timestamp)as month,
  count(*) total
from `Target.orders`
group by year, month
```

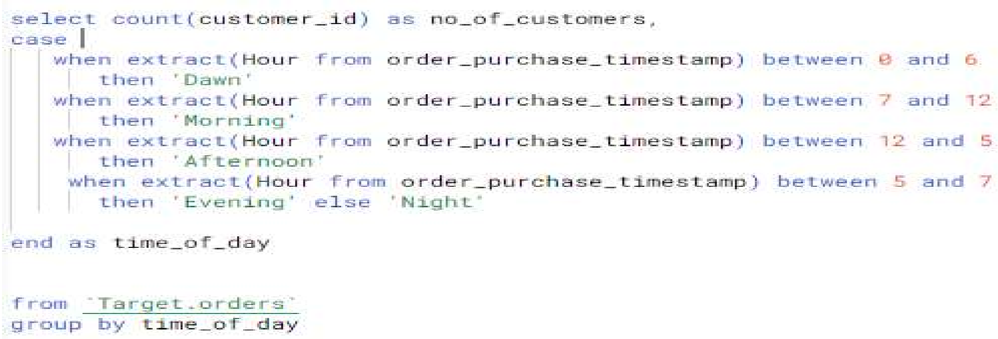


| Row | orders | year | month | total |
|---|---|---|---|---|
| 1 | 7544 | 2017 | 11 | 7544 |
| 2 | 5673 | 2017 | 12 | 5673 |
| 3 | 6728 | 2018 | 2 | 6728 |
| 4 | 2404 | 2017 | 4 | 2404 |
| 5 | 4026 | 2017 | 7 | 4026 |
| 6 | 6873 | 2018 | 5 | 6873 |
| 7 | 4631 | 2017 | 10 | 4631 |
| 8 | 7269 | 2018 | 1 | 7269 |
| 9 | 3245 | 2017 | 6 | 3245 |
| 10 | 4285 | 2017 | 9 | 4285 |

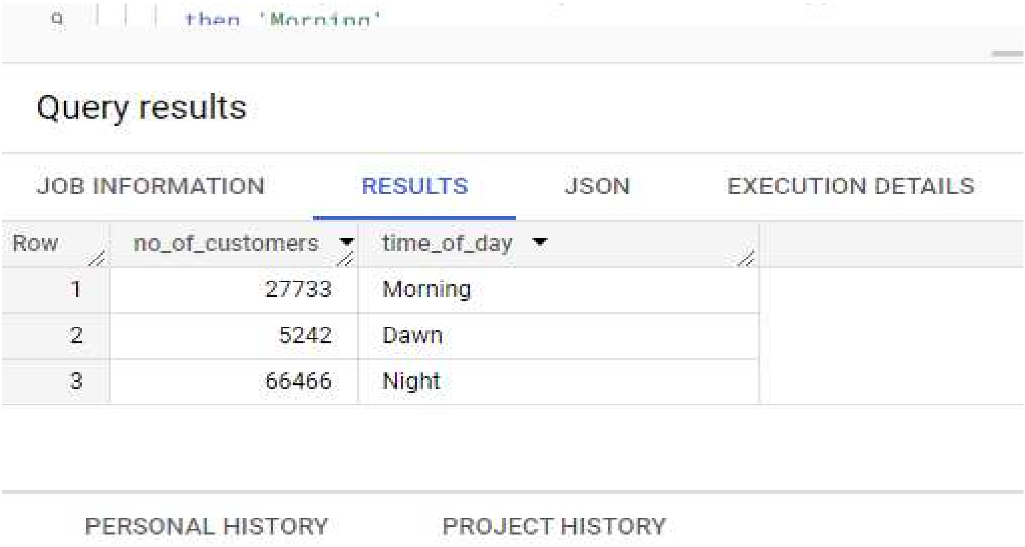
**Growing Trend: -**

- There is a Growing trend in orders, trend sustains during 2018. There is a light fall we can observe during October 2017 following a great hike in November month and again a fall in end of December 2017 and January 2018.
- We can observe there's 815% growth increased in terms of orders and 707% growth increment in terms of revenue in January from 2017 to 2018.
- growth rate for July and august in 2017 to 2018 is relatively very low!
- 2017-february, 2017-march,2017-november were the highest growing sale month compared to the previous month.
- PCs and Musical Instruments category have relatively less number of products, but contributes in a high revenue. PCs, house pastels oven and cafe, agro industry and commerce, musical instruments, Kitchen portable and food coach are having highest average product price categories.

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



```
select count(customer_id) as no_of_customers,
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 12 and 5
      then 'Afternoon'
     when extract(Hour from order_purchase_timestamp) between 5 and 7
      then 'Evening' else 'Night'

end as time_of_day


from `Target.orders`
group by time_of_day
```



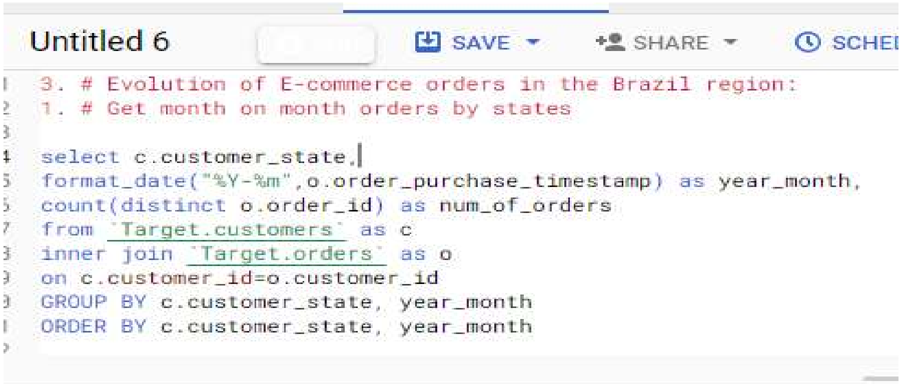
```
      then 'Morning'
```

Query results

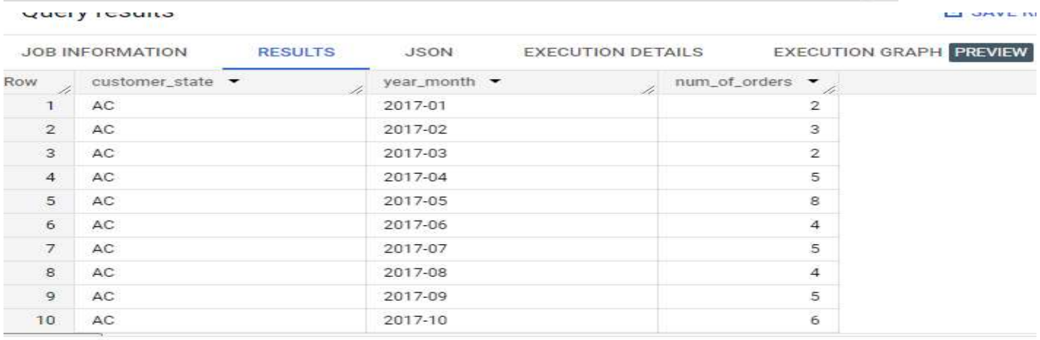
| Row | no_of_customers | time_of_day |
|---|---|---|
| 1 | 27733 | Morning |
| 2 | 5242 | Dawn |
| 3 | 66466 | Night |

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

**1.** Get month on month orders by states:



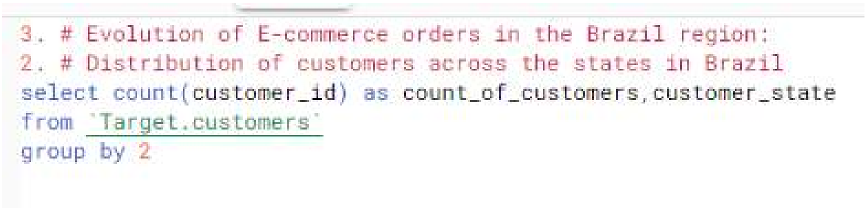
```
3. # Evolution of E-commerce orders in the Brazil region:
1. # Get month on month orders by states

select c.customer_state,
format_date("%Y-%m",o.order_purchase_timestamp) as year_month,
count(distinct o.order_id) as num_of_orders
from `Target.customers` as c
inner join `Target.orders` as o
on c.customer_id=o.customer_id
GROUP BY c.customer_state, year_month
ORDER BY c.customer_state, year_month
```



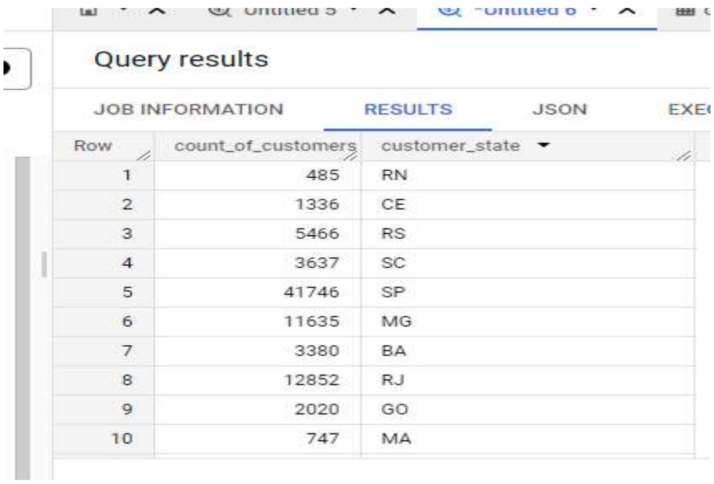
Query results

| Row | customer_state | year_month | num_of_orders |
|---|---|---|---|
| 1 | AC | 2017-01 | 2 |
| 2 | AC | 2017-02 | 3 |
| 3 | AC | 2017-03 | 2 |
| 4 | AC | 2017-04 | 5 |
| 5 | AC | 2017-05 | 8 |
| 6 | AC | 2017-06 | 4 |
| 7 | AC | 2017-07 | 5 |
| 8 | AC | 2017-08 | 4 |
| 9 | AC | 2017-09 | 5 |
| 10 | AC | 2017-10 | 6 |

2. Distribution of customers across the states in Brazil :



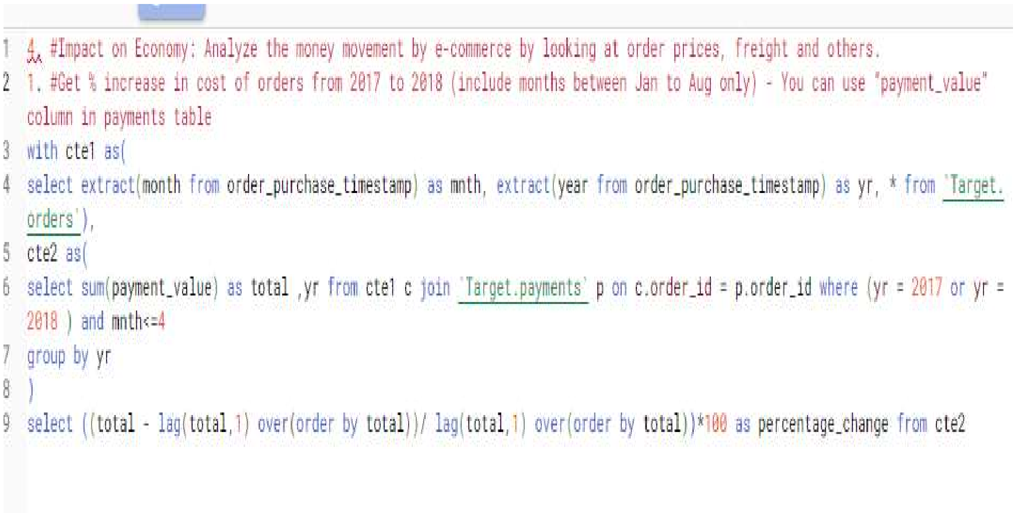
```
3. # Evolution of E-commerce orders in the Brazil region:
2. # Distribution of customers across the states in Brazil
select count(customer_id) as count_of_customers,customer_state
from `Target.customers`
group by 2
```



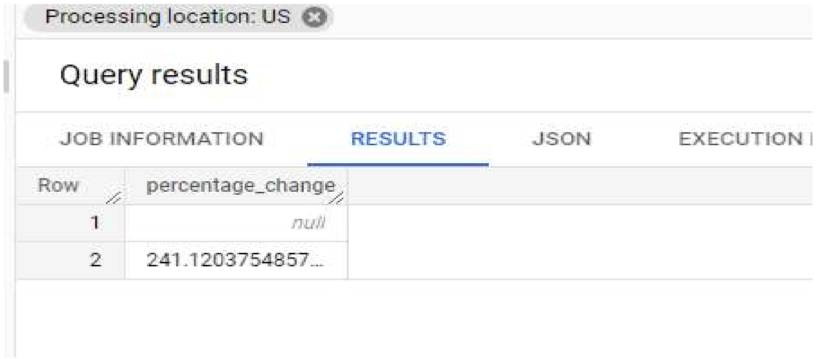
| Row | count_of_customers | customer_state |
|---|---|---|
| 1 | 485 | RN |
| 2 | 1336 | CE |
| 3 | 5466 | RS |
| 4 | 3637 | SC |
| 5 | 41746 | SP |
| 6 | 11635 | MG |
| 7 | 3380 | BA |
| 8 | 12852 | RJ |
| 9 | 2020 | GO |
| 10 | 747 | MA |

## 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

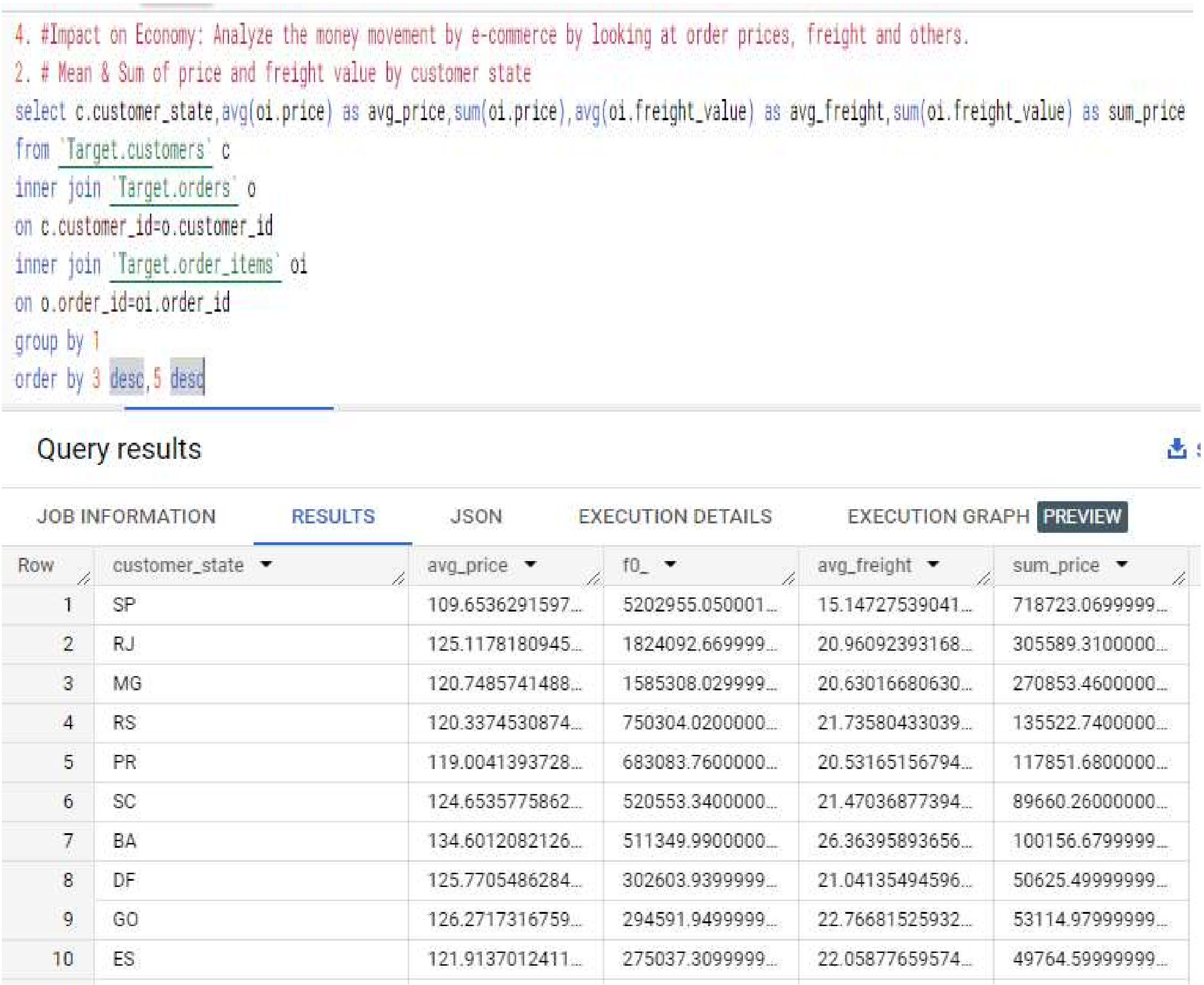


```
4. #Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
1. #Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value"
column in payments table
with cte1 as(
select extract(month from order_purchase_timestamp) as mnth, extract(year from order_purchase_timestamp) as yr, * from `Target.
orders`),
cte2 as(
select sum(payment_value) as total ,yr from cte1 c join `Target.payments` p on c.order_id = p.order_id where (yr = 2017 or yr =
2018 ) and mnth<=4
group by yr
)
select ((total - lag(total,1) over(order by total))/ lag(total,1) over(order by total))*100 as percentage_change from cte2
```



| Row | percentage_change |
|---|---|
| 1 | null |
| 2 | 241.1203754857... |

2. Mean & Sum of price and freight value by customer state

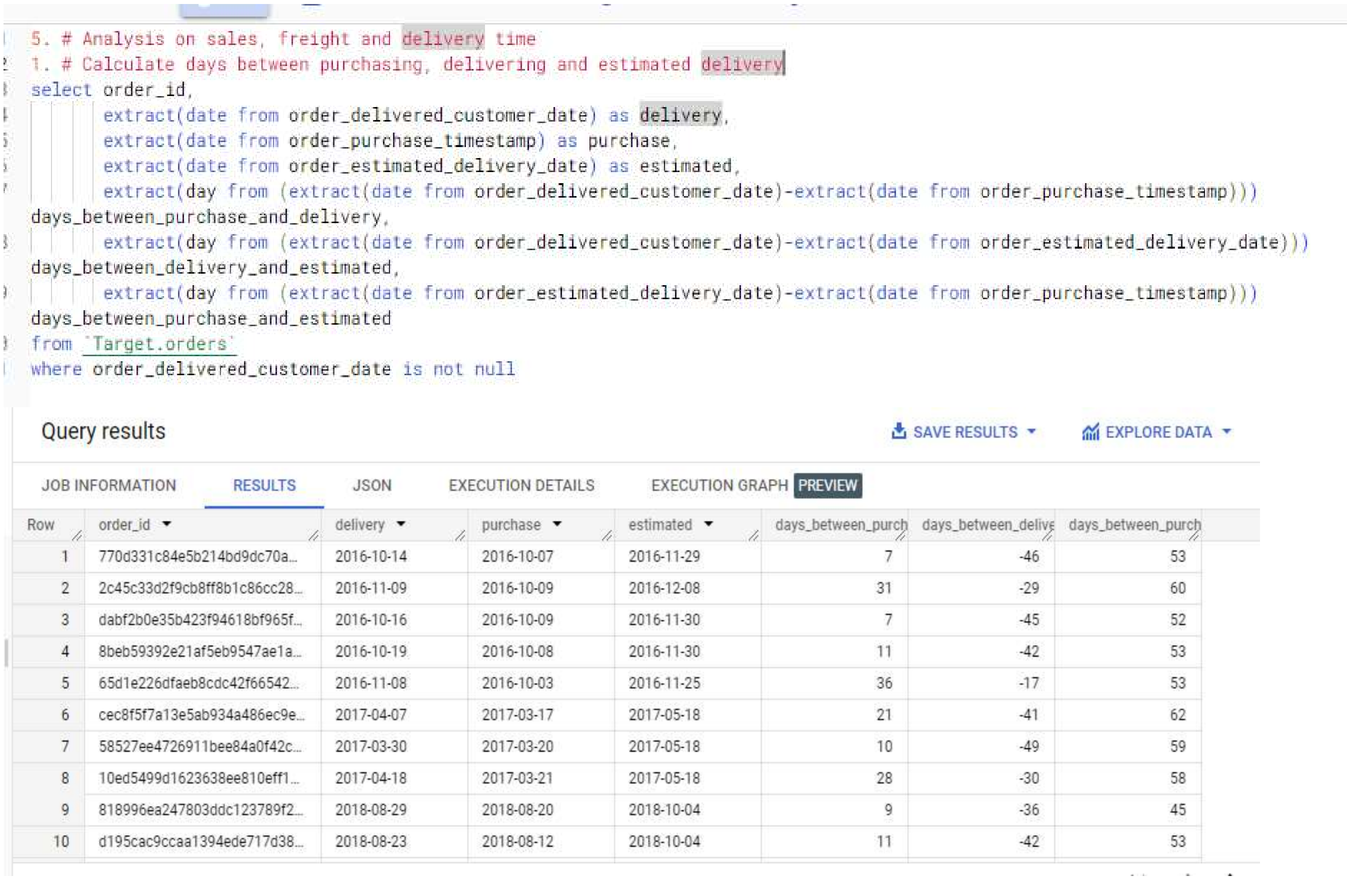


```
4. #Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
2. # Mean & Sum of price and freight value by customer state
select c.customer_state,avg(oi.price) as avg_price,sum(oi.price),avg(oi.freight_value) as avg_freight,sum(oi.freight_value) as sum_price
from `Target.customers` c
inner join `Target.orders` o
on c.customer_id=o.customer_id
inner join `Target.order_items` oi
on o.order_id=oi.order_id
group by 1
order by 3 desc,5 desc
```

| Row | customer_state | avg_price | f0_ | avg_freight | sum_price |
|---|---|---|---|---|---|
| 1 | SP | 109.6536291597... | 5202955.050001... | 15.14727539041... | 718723.0699999... |
| 2 | RJ | 125.1178180945... | 1824092.669999... | 20.96092393168... | 305589.3100000... |
| 3 | MG | 120.7485741488... | 1585308.029999... | 20.63016680630... | 270853.4600000... |
| 4 | RS | 120.3374530874... | 750304.0200000... | 21.73580433039... | 135522.7400000... |
| 5 | PR | 119.0041393728... | 683083.7600000... | 20.53165156794... | 117851.6800000... |
| 6 | SC | 124.6535775862... | 520553.3400000... | 21.47036877394... | 89660.26000000... |
| 7 | BA | 134.6012082126... | 511349.9900000... | 26.36395893656... | 100156.6799999... |
| 8 | DF | 125.7705486284... | 302603.9399999... | 21.04135494596... | 50625.49999999... |
| 9 | GO | 126.2717316759... | 294591.9499999... | 22.76681525932... | 53114.97999999... |
| 10 | ES | 121.9137012411... | 275037.3099999... | 22.05877659574... | 49764.59999999... |

## 5. Analysis on sales, freight and delivery time :

1. Calculate days between purchasing, delivering and estimated delivery



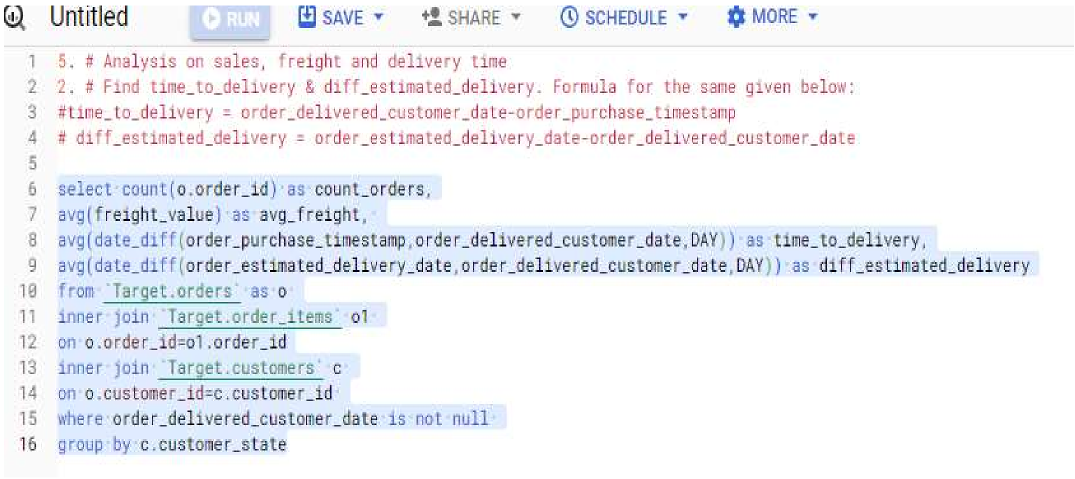
```
5. # Analysis on sales, freight and delivery time
1. # Calculate days between purchasing, delivering and estimated delivery
select order_id,
        extract(date from order_delivered_customer_date) as delivery,
        extract(date from order_purchase_timestamp) as purchase,
        extract(date from order_estimated_delivery_date) as estimated,
        extract(day from (extract(date from order_delivered_customer_date)-extract(date from order_purchase_timestamp)))
days_between_purchase_and_delivery,
        extract(day from (extract(date from order_delivered_customer_date)-extract(date from order_estimated_delivery_date)))
days_between_delivery_and_estimated,
        extract(day from (extract(date from order_estimated_delivery_date)-extract(date from order_purchase_timestamp)))
days_between_purchase_and_estimated
from `Target.orders`
where order_delivered_customer_date is not null
```

Query results

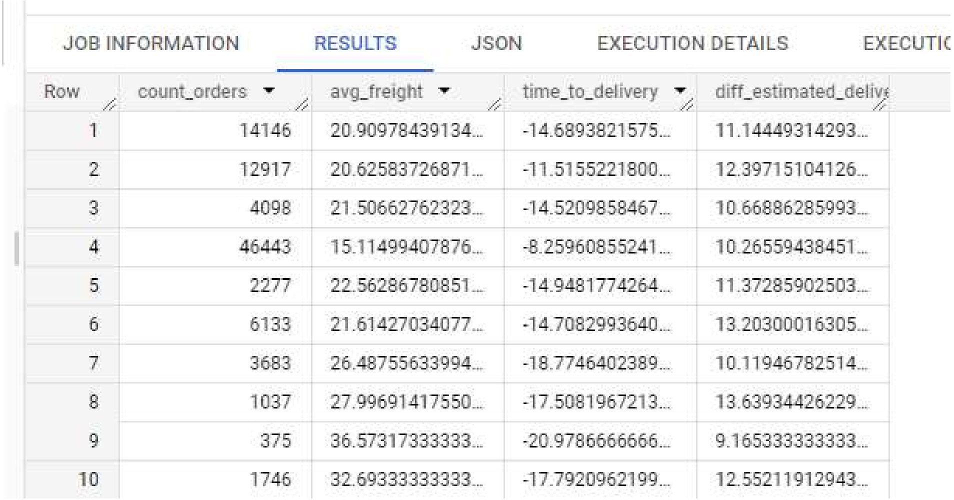
| Row | order_id | delivery | purchase | estimated | days_between_purch | days_between_delive | days_between_purch |
|---|---|---|---|---|---|---|---|
| 1 | 770d331c84e5b214bd9dc70a... | 2016-10-14 | 2016-10-07 | 2016-11-29 | 7 | -46 | 53 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 2016-11-09 | 2016-10-09 | 2016-12-08 | 31 | -29 | 60 |
| 3 | dabf2b0e35b423f94618bf965f... | 2016-10-16 | 2016-10-09 | 2016-11-30 | 7 | -45 | 52 |
| 4 | 8beb59392e21af5eb9547ae1a... | 2016-10-19 | 2016-10-08 | 2016-11-30 | 11 | -42 | 53 |
| 5 | 65d1e226dfaeb8cdc42f66542... | 2016-11-08 | 2016-10-03 | 2016-11-25 | 36 | -17 | 53 |
| 6 | cec8f5f7a13e5ab934a486ec9e... | 2017-04-07 | 2017-03-17 | 2017-05-18 | 21 | -41 | 62 |
| 7 | 58527ee4726911bee84a0f42c... | 2017-03-30 | 2017-03-20 | 2017-05-18 | 10 | -49 | 59 |
| 8 | 10ed5499d1623638ee810eff1... | 2017-04-18 | 2017-03-21 | 2017-05-18 | 28 | -30 | 58 |
| 9 | 818996ea247803ddc123789f2... | 2018-08-29 | 2018-08-20 | 2018-10-04 | 9 | -36 | 45 |
| 10 | d195cac9ccaa1394ede717d38... | 2018-08-23 | 2018-08-12 | 2018-10-04 | 11 | -42 | 53 |

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

- time_to_delivery = order_delivered_customer_date-order_purchase_timestamp
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

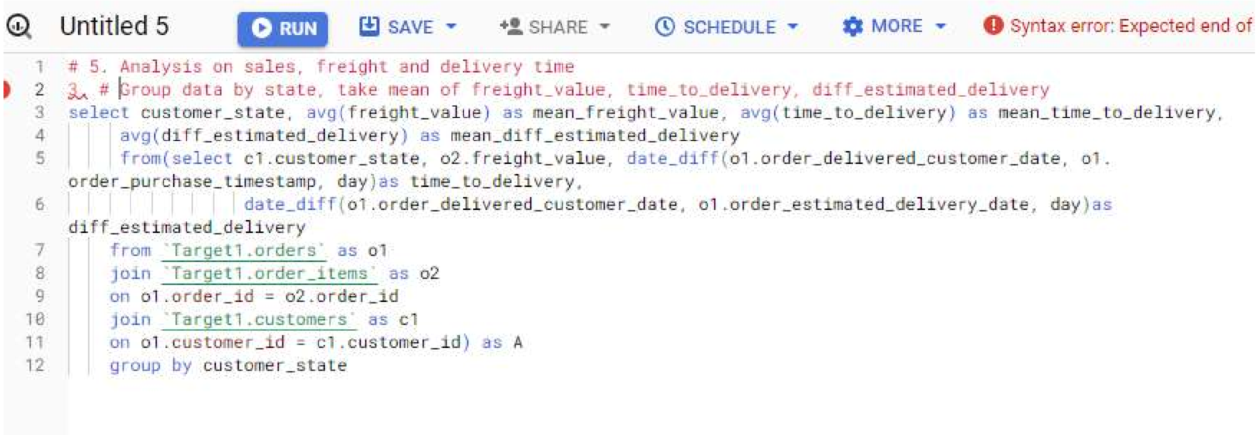


```
5. # Analysis on sales, freight and delivery time
2. # Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
#time_to_delivery = order_delivered_customer_date-order_purchase_timestamp
# diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

select count(o.order_id) as count_orders,
avg(freight_value) as avg_freight,
avg(date_diff(order_purchase_timestamp,order_delivered_customer_date,DAY)) as time_to_delivery,
avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)) as diff_estimated_delivery
from `Target.orders` as o
inner join `Target.order_items` o1
on o.order_id=o1.order_id
inner join `Target.customers` c
on o.customer_id=c.customer_id
where order_delivered_customer_date is not null
group by c.customer_state
```

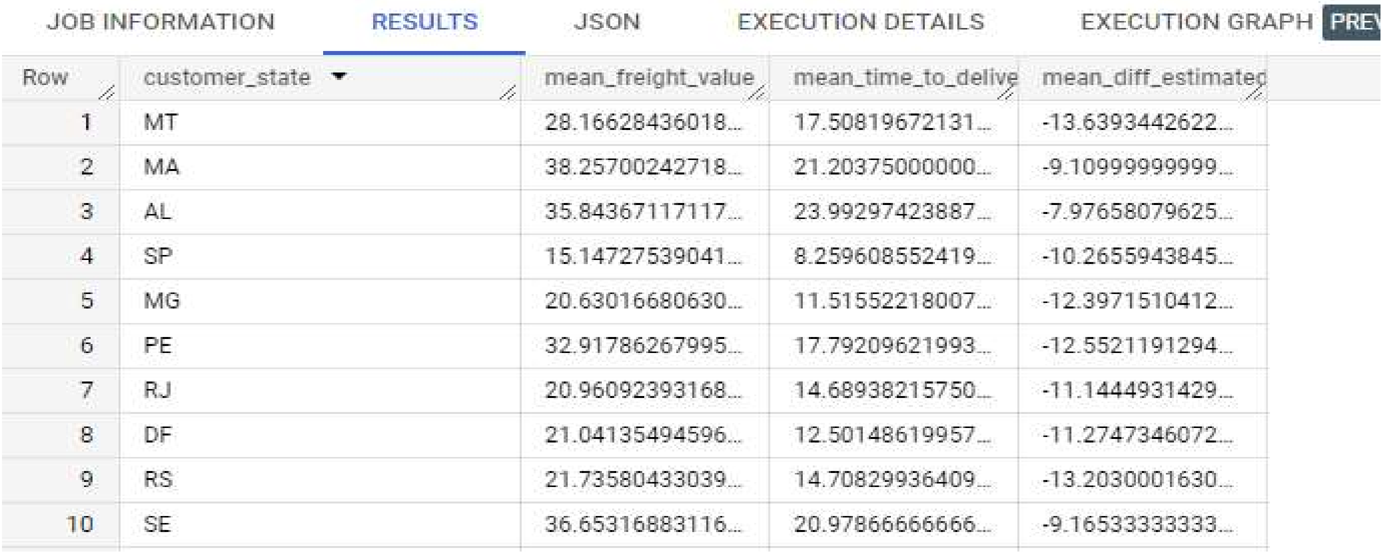


| Row | count_orders | avg_freight | time_to_delivery | diff_estimated_delive |
|---|---|---|---|---|
| 1 | 14146 | 20.90978439134... | -14.6893821575... | 11.14449314293... |
| 2 | 12917 | 20.62583726871... | -11.5155221800... | 12.39715104126... |
| 3 | 4098 | 21.50662762323... | -14.5209858467... | 10.66886285993... |
| 4 | 46443 | 15.11499407876... | -8.25960855241... | 10.26559438451... |
| 5 | 2277 | 22.56286780851... | -14.9481774264... | 11.37285902503... |
| 6 | 6133 | 21.61427034077... | -14.7082993640... | 13.20300016305... |
| 7 | 3683 | 26.48755633994... | -18.7746402389... | 10.11946782514... |
| 8 | 1037 | 27.99691417550... | -17.5081967213... | 13.63934426229... |
| 9 | 375 | 36.57317333333... | -20.9786666666... | 9.165333333333... |
| 10 | 1746 | 32.69333333333... | -17.7920962199... | 12.55211912943... |

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

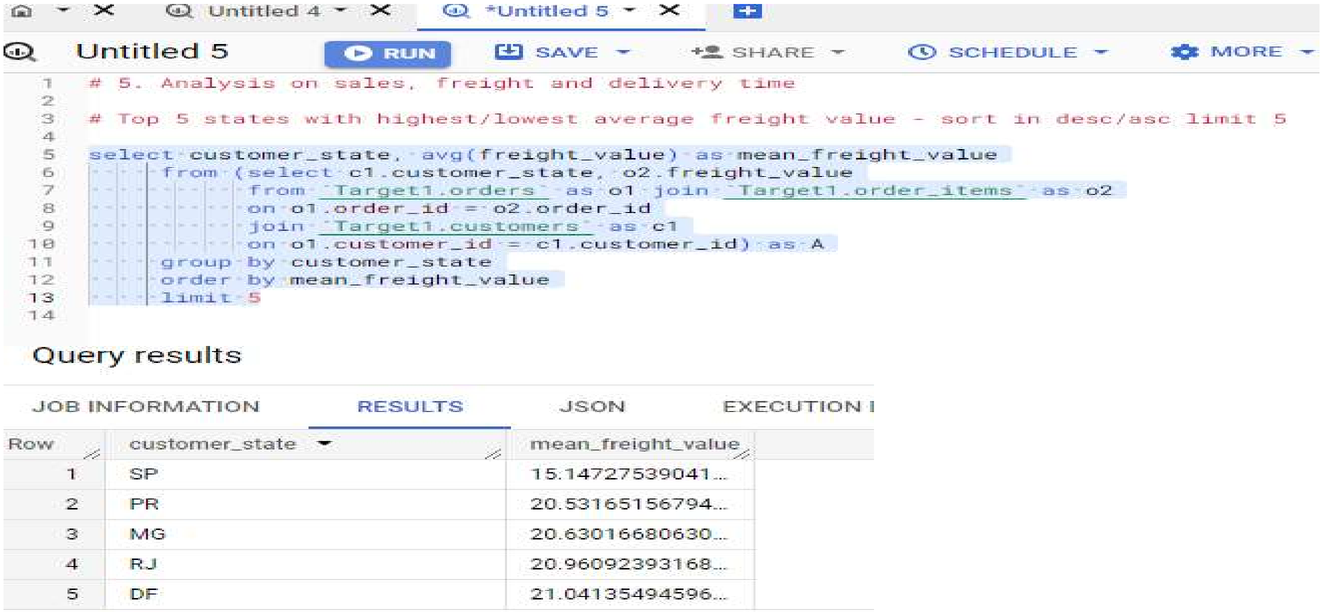


```
# 5. Analysis on sales, freight and delivery time
3. # Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
select customer_state, avg(freight_value) as mean_freight_value, avg(time_to_delivery) as mean_time_to_delivery,
    avg(diff_estimated_delivery) as mean_diff_estimated_delivery
    from(select c1.customer_state, o2.freight_value, date_diff(o1.order_delivered_customer_date, o1.
order_purchase_timestamp, day)as time_to_delivery,
            date_diff(o1.order_delivered_customer_date, o1.order_estimated_delivery_date, day)as
diff_estimated_delivery
   from `Target1.orders` as o1
   join `Target1.order_items` as o2
   on o1.order_id = o2.order_id
   join `Target1.customers` as c1
   on o1.customer_id = c1.customer_id) as A
   group by customer_state
```



| Row | customer_state | mean_freight_value | mean_time_to_delive | mean_diff_estimated |
|---|---|---|---|---|
| 1 | MT | 28.16628436018... | 17.50819672131... | -13.6393442622... |
| 2 | MA | 38.25700242718... | 21.20375000000... | -9.10999999999... |
| 3 | AL | 35.84367117117... | 23.99297423887... | -7.97658079625... |
| 4 | SP | 15.14727539041... | 8.259608552419... | -10.2655943845... |
| 5 | MG | 20.63016680630... | 11.51552218007... | -12.3971510412... |
| 6 | PE | 32.91786267995... | 17.79209621993... | -12.5521191294... |
| 7 | RJ | 20.96092393168... | 14.68938215750... | -11.1444931429... |
| 8 | DF | 21.04135494596... | 12.50148619957... | -11.2747346072... |
| 9 | RS | 21.73580433039... | 14.70829936409... | -13.2030001630... |
| 10 | SE | 36.65316883116... | 20.97866666666... | -9.16533333333... |

5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

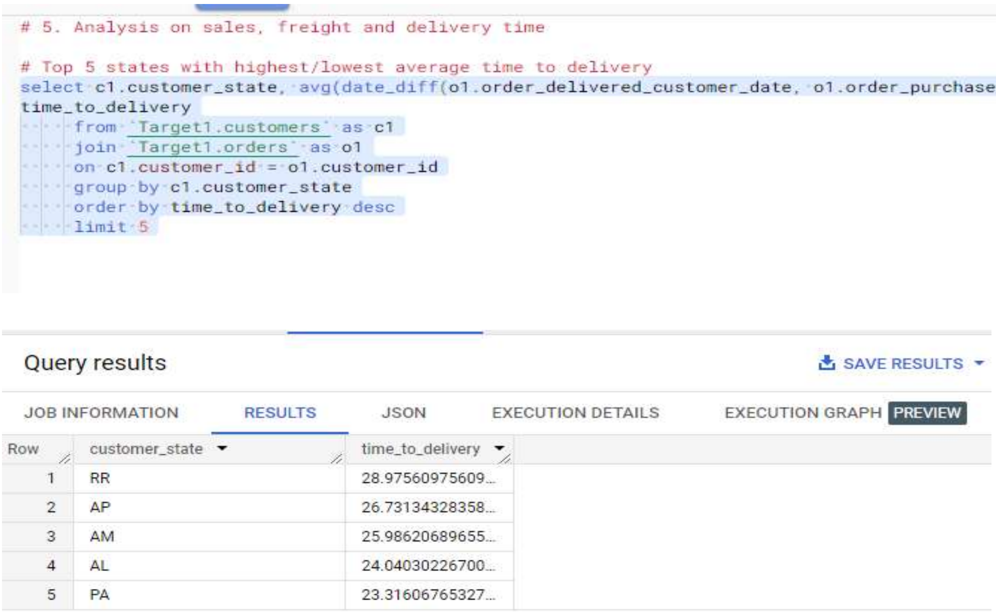


```
# 5. Analysis on sales, freight and delivery time

# Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

select customer_state, avg(freight_value) as mean_freight_value
    from (select c1.customer_state, o2.freight_value
            from `Target1.orders` as o1 join `Target1.order_items` as o2
            on o1.order_id = o2.order_id
            join `Target1.customers` as c1
            on o1.customer_id = c1.customer_id) as A
    group by customer_state
    order by mean_freight_value
    limit 5
```

Query results

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

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

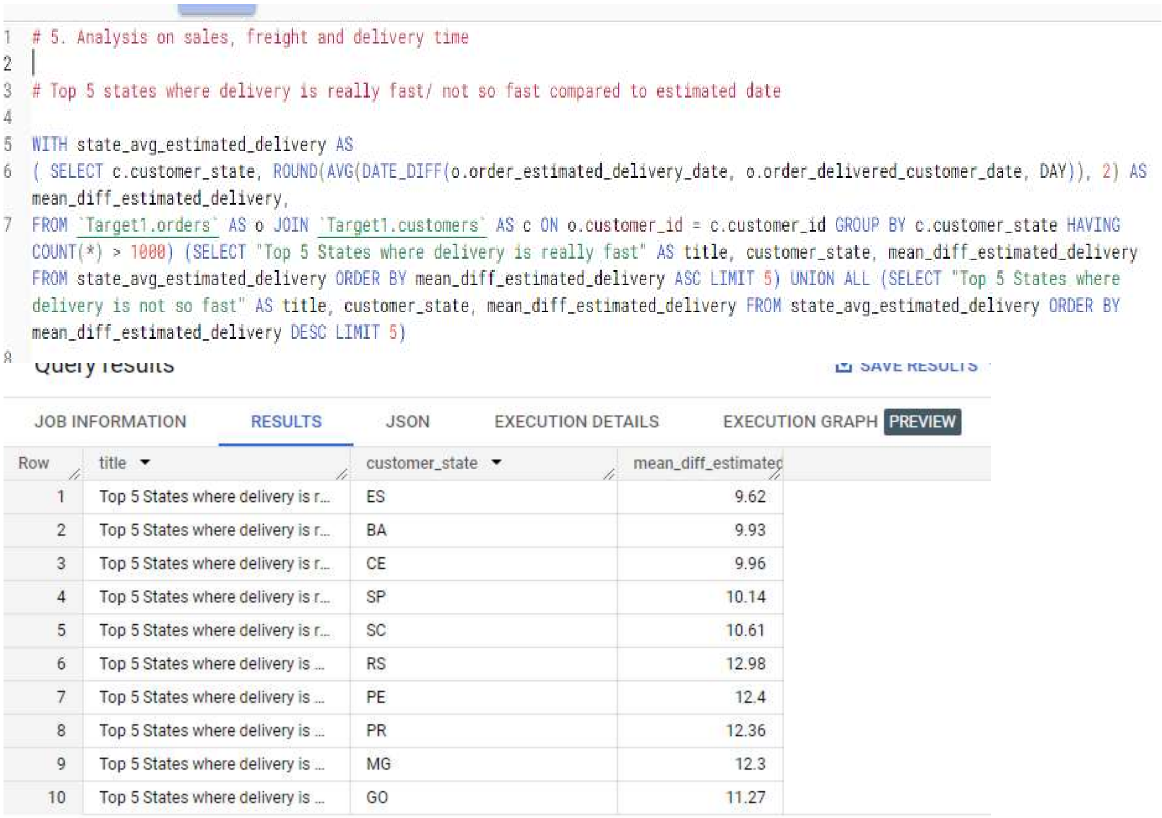


```
# 5. Analysis on sales, freight and delivery time

# Top 5 states with highest/lowest average time to delivery
select c1.customer_state, avg(date_diff(o1.order_delivered_customer_date, o1.order_purchase_timestamp, day))as
time_to_delivery
    from `Target1.customers` as c1
    join `Target1.orders` as o1
    on c1.customer_id = o1.customer_id
    group by c1.customer_state
    order by time_to_delivery desc
    limit 5
```

Query results

| Row | customer_state | time_to_delivery |
|---|---|---|
| 1 | RR | 28.97560975609... |
| 2 | AP | 26.73134328358... |
| 3 | AM | 25.98620689655... |
| 4 | AL | 24.04030226700... |
| 5 | PA | 23.31606765327... |

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



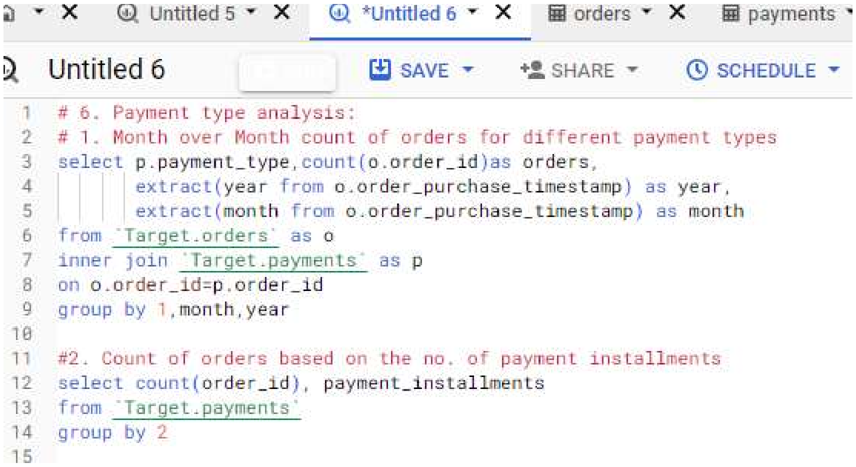
```
# 5. Analysis on sales, freight and delivery time

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

WITH state_avg_estimated_delivery AS
( SELECT c.customer_state, ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)), 2) AS
mean_diff_estimated_delivery,
FROM `Target1.orders` AS o JOIN `Target1.customers` AS c ON o.customer_id = c.customer_id GROUP BY c.customer_state HAVING
COUNT(*) > 1000) (SELECT "Top 5 States where delivery is really fast" AS title, customer_state, mean_diff_estimated_delivery
FROM state_avg_estimated_delivery ORDER BY mean_diff_estimated_delivery ASC LIMIT 5) UNION ALL (SELECT "Top 5 States where
delivery is not so fast" AS title, customer_state, mean_diff_estimated_delivery FROM state_avg_estimated_delivery ORDER BY
mean_diff_estimated_delivery DESC LIMIT 5)
```

| Row | title | customer_state | mean_diff_estimated |
|---|---|---|---|
| 1 | Top 5 States where delivery is r... | ES | 9.62 |
| 2 | Top 5 States where delivery is r... | BA | 9.93 |
| 3 | Top 5 States where delivery is r... | CE | 9.96 |
| 4 | Top 5 States where delivery is r... | SP | 10.14 |
| 5 | Top 5 States where delivery is r... | SC | 10.61 |
| 6 | Top 5 States where delivery is ... | RS | 12.98 |
| 7 | Top 5 States where delivery is ... | PE | 12.4 |
| 8 | Top 5 States where delivery is ... | PR | 12.36 |
| 9 | Top 5 States where delivery is ... | MG | 12.3 |
| 10 | Top 5 States where delivery is ... | GO | 11.27 |

## 6. Payment type analysis:

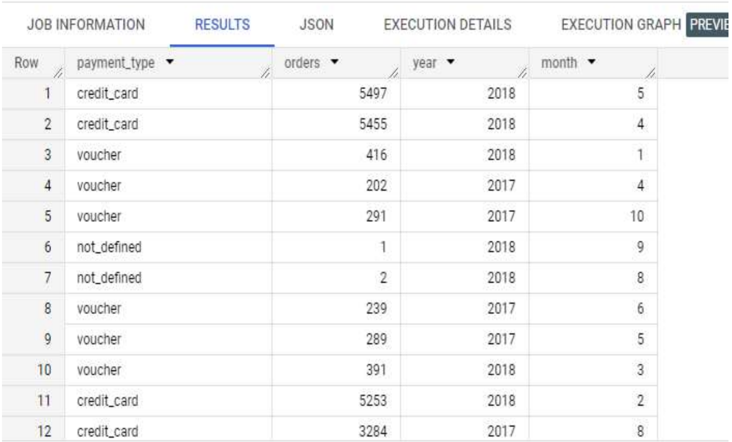
1. Month over Month count of orders for different payment types

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



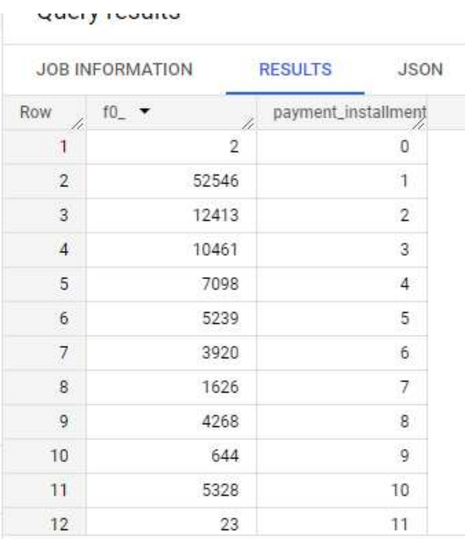
```
# 6. Payment type analysis:
# 1. Month over Month count of orders for different payment types
select p.payment_type,count(o.order_id)as orders,
          extract(year from o.order_purchase_timestamp) as year,
          extract(month from o.order_purchase_timestamp) as month
from `Target.orders` as o
inner join `Target.payments` as p
on o.order_id=p.order_id
group by 1,month,year

#2. Count of orders based on the no. of payment installments
select count(order_id), payment_installments
from `Target.payments`
group by 2
```

1. Result:



| Row | payment_type | orders | year | month |
|---|---|---|---|---|
| 1 | credit_card | 5497 | 2018 | 5 |
| 2 | credit_card | 5455 | 2018 | 4 |
| 3 | voucher | 416 | 2018 | 1 |
| 4 | voucher | 202 | 2017 | 4 |
| 5 | voucher | 291 | 2017 | 10 |
| 6 | not_defined | 1 | 2018 | 9 |
| 7 | not_defined | 2 | 2018 | 8 |
| 8 | voucher | 239 | 2017 | 6 |
| 9 | voucher | 289 | 2017 | 5 |
| 10 | voucher | 391 | 2018 | 3 |
| 11 | credit_card | 5253 | 2018 | 2 |
| 12 | credit_card | 3284 | 2017 | 8 |

2. Result



| Row | f0_ | payment_installment |
|---|---|---|
| 1 | 2 | 0 |
| 2 | 52546 | 1 |
| 3 | 12413 | 2 |
| 4 | 10461 | 3 |
| 5 | 7098 | 4 |
| 6 | 5239 | 5 |
| 7 | 3920 | 6 |
| 8 | 1626 | 7 |
| 9 | 4268 | 8 |
| 10 | 644 | 9 |
| 11 | 5328 | 10 |
| 12 | 23 | 11 |

## 7. Actionable Insights:

- Customers are from different 4119 cities and 27 states from Brazil.
- 68% customers are from southeast Brazil, 14% are from south Brazil and rest are other regions of Brazil.
- We have 96096 number of Unique Customers ids.
- Of a total of 99441 orders, 1107 are shipped, 625 were canceled, 96478 are delivered.
- Compared to 2017, revenue has increased in 2018 by 21%.

- Average number of orders are higher during November month, September and October month average orders are comparatively low, in May and July and august have higher average orders compare to other months.

- The average time taken for a carrier to start the delivery is 2 and a half days.

- States Sao Paulo, Parana, Minas Gerais, Distrito Federal, Santa Catarina and Rio de Janeiro are some of the states having faster delivery time relatively.

## 8. Recommendations :

- Increasing network in north Brazil, having small towns can help increase the customer base.

- • As north Brazil has the world’s largest river and most extensive rain forest, must be a good travel destination, introducing necessary survival/ camping/adventure products can help increase revenue and order from northern region .

- top selling items are between 10-100 dollars, introducing new different more products from top selling categories can increase revenue more.

- In order to reduce the delivery time, if we look at the average time for a carrier to start the delivery itself takes at least 2 and a half days. and order approval time is 0.26 days. These two should be optimized at as low as possible, that can result in delivering faster.
- It was observed that an increasing trend in revenue and orders over time, yet during October and January sales are decreasing probably after Festival Sales.
- Introducing a possible discount on not so running products can help sell more products during those low-going months.
- From the distribution and statistical analysis, we can observe the average time to complete the delivery is 12 days. which should be reduced to at least half, as due to high competition in e-commerce market, it is vital to do so.
- It was observed that an increasing trend in revenue and orders over time , yet during October and January sales are decreasing probably after Festival Sales.