Target Case Study

Problem Statement:

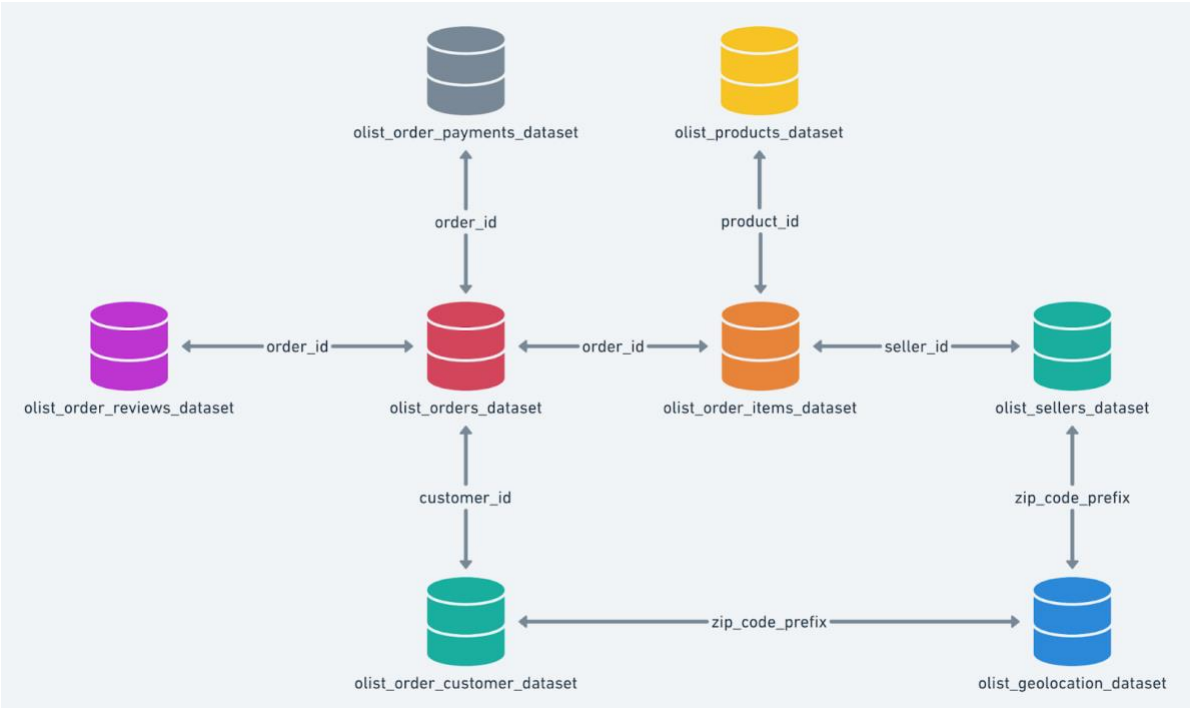
Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

Context:

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

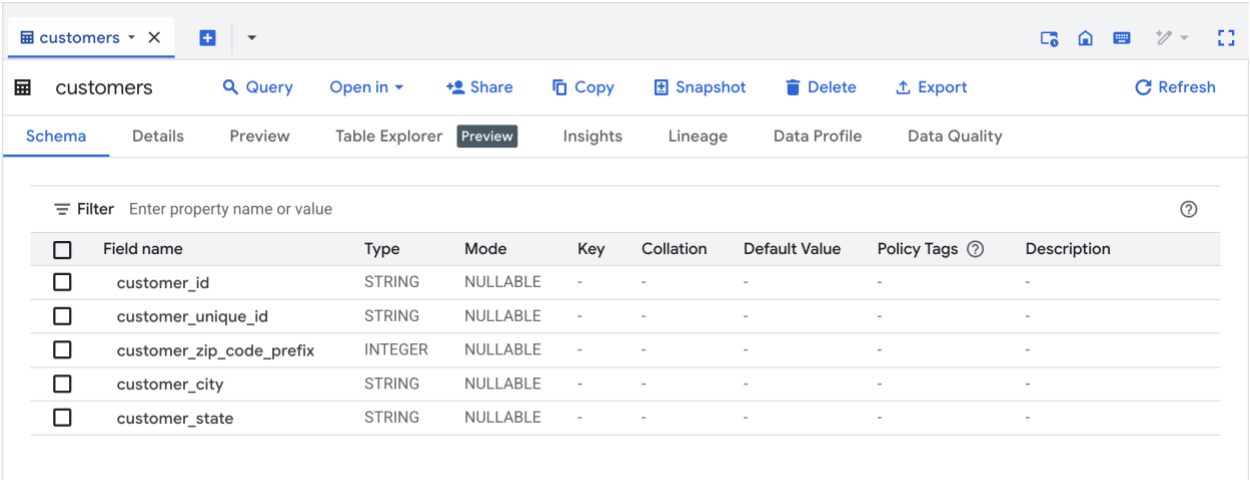
This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.



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

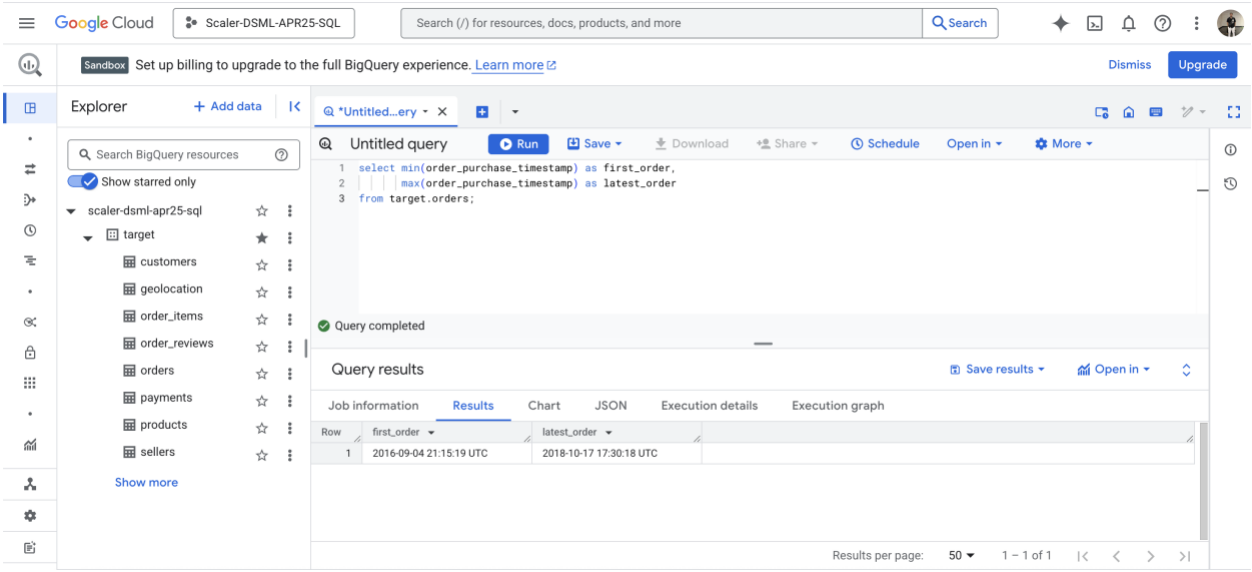
1.a. Data type of all columns in the "customers" table.



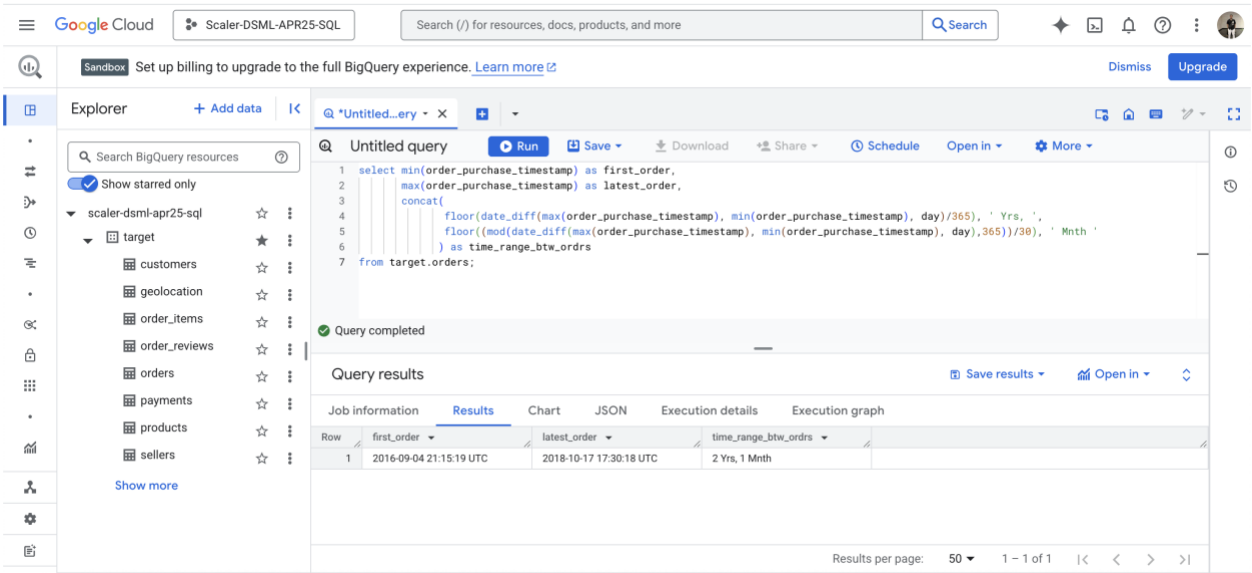
The "customers" table consists of 5 fields. Their data types are:
customer_id -> STRING
customer_unique_id -> STRING
customer_zip_code_prefix -> INTEGER
customer_city -> STRING
customer_state -> STRING

1.b. Get the time range between which the orders were placed.

```
select min(order_purchase_timestamp) as first_order,
       max(order_purchase_timestamp) as latest_order,
from target.orders;
```



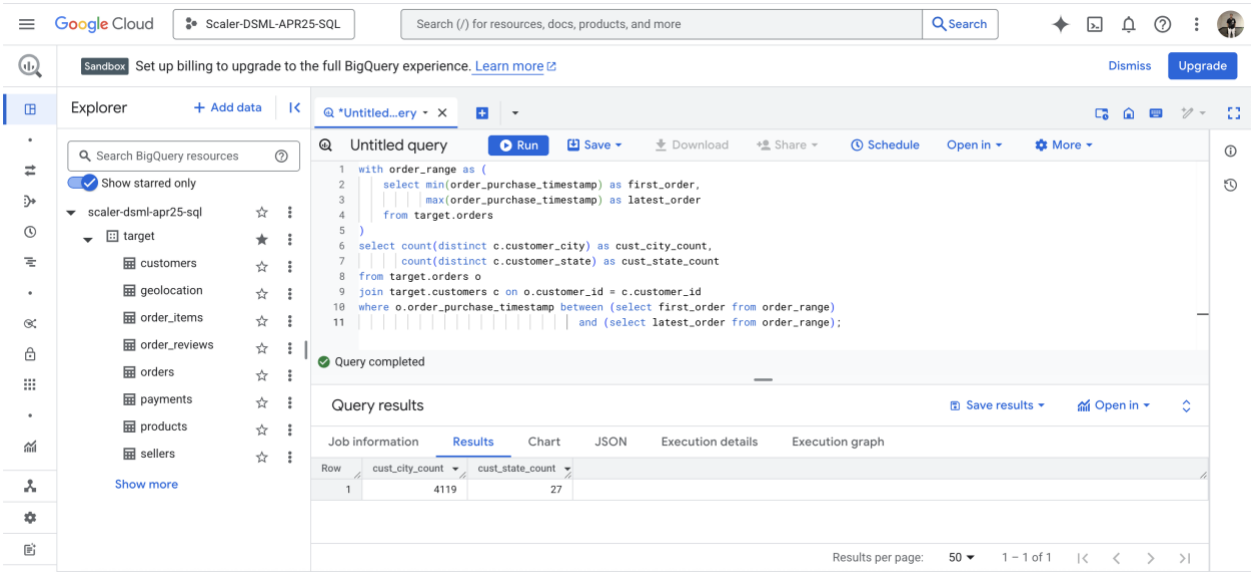

```
Alternative Way:

select min(order_purchase_timestamp) as first_order,
       max(order_purchase_timestamp) as latest_order,
       concat(
              floor(date_diff(max(order_purchase_timestamp),
min(order_purchase_timestamp), day) / 365), ' Yrs, ',
              floor((mod(date_diff(max(order_purchase_timestamp),
min(order_purchase_timestamp), day), 365)) / 30), ' Mnth '
            ) as time_range_btw_ordrs
from target.orders;
```



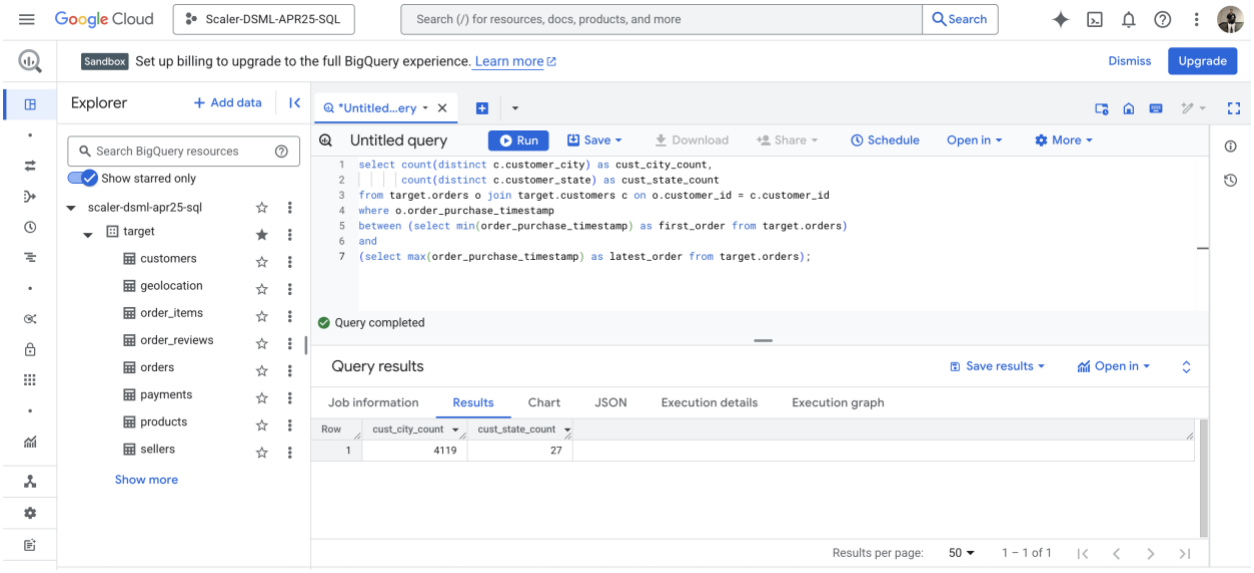
The first order was placed on 2016-09-04 21:15:19 UTC, and the latest order was placed on 2018-10-17 17:30:18 UTC. The difference, 2 years 1 month, represents the time range during which the orders were placed. **Note:** The reason for considering the first and latest order timestamps is that each order, regardless of its status, was initially in a "placed" status before it was canceled or approved or delivered etc.

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

```
with order_range as (
    select min(order_purchase_timestamp) as first_order,
           max(order_purchase_timestamp) as latest_order
    from target.orders
)
select count(distinct c.customer_city) as cust_city_count,
       count(distinct c.customer_state) as cust_state_count
from target.orders o
join target.customers c on o.customer_id = c.customer_id
where o.order_purchase_timestamp between (select first_order from order_range)
                                     and (select latest_order from order_range);
```




```
Alternative Way:

select count(distinct c.customer_city) as cust_city_count,
       count(distinct c.customer_state) as cust_state_count
from target.orders o join target.customers c on o.customer_id = c.customer_id
where o.order_purchase_timestamp
between (select min(order_purchase_timestamp) as first_order from target.orders)
and
(select max(order_purchase_timestamp) as latest_order from target.orders);
```

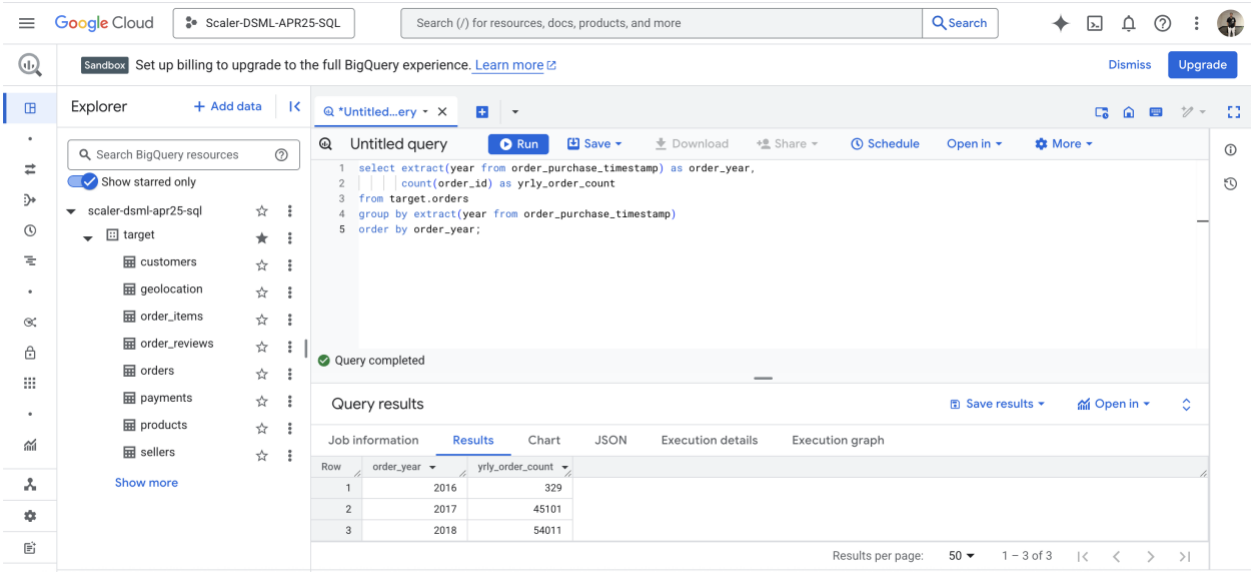


Customers from 4119 cities and 27 states ordered during the period between the first order and the latest order.

2. In-depth Exploration:

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

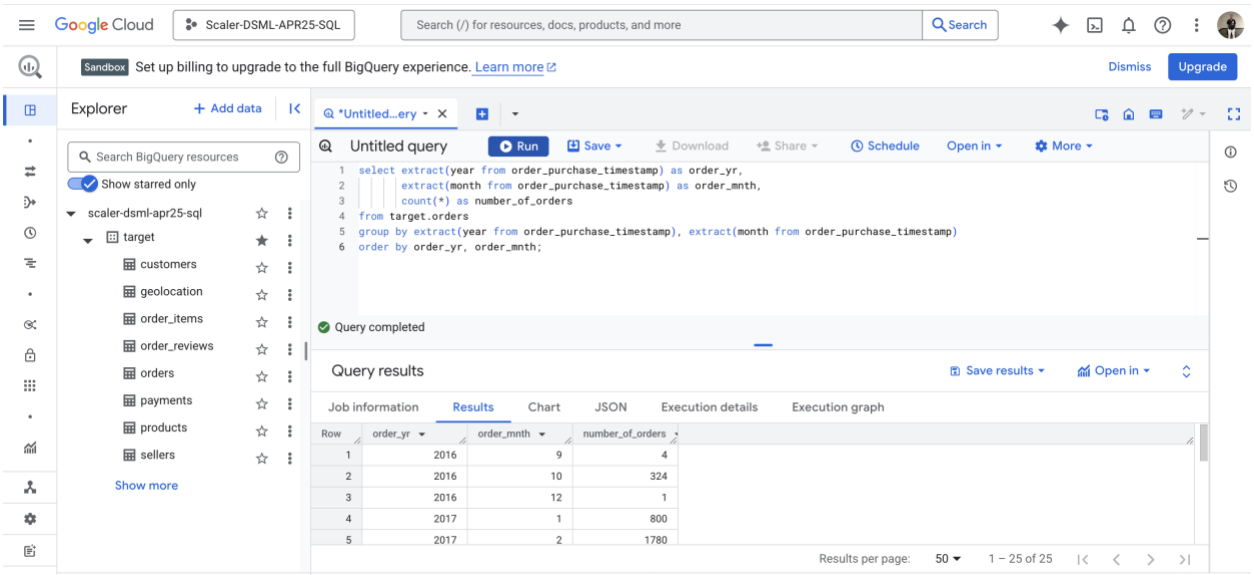
```
select extract(year from order_purchase_timestamp) as order_year,
       count(order_id) as yrly_order_count
from target.orders
group by extract(year from order_purchase_timestamp)
order by order_year;
```

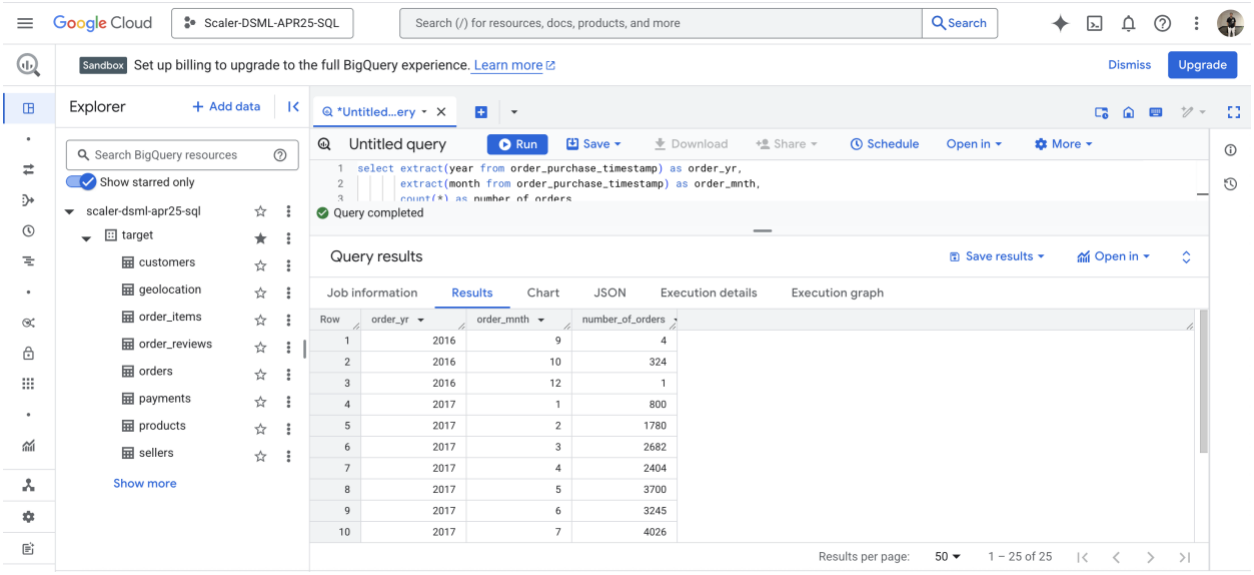


The result indicates that there is a growing trend in the number of orders placed over the past years as the yearly order count increased from 329 in 2016 to 45101 in 2017 and further increased to 54011 in 2018.

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

```
select extract(year from order_purchase_timestamp) as order_yr,
       extract(month from order_purchase_timestamp) as order_mnth,
       count(*) as number_of_orders
from target.orders
group by extract(year from order_purchase_timestamp), extract(month from
order_purchase_timestamp)
order by order_yr, order_mnth;
```

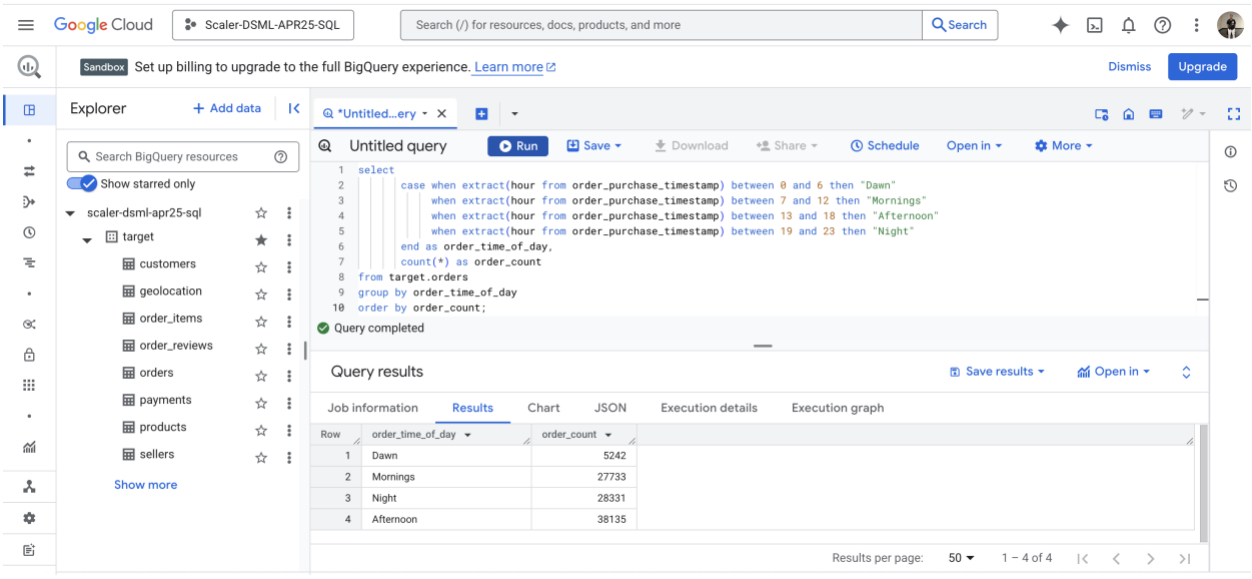




The number of orders in February – March showed a spike due to the onset of the Carnival, the biggest festival in Brazil. Towards the end of the year, the total number of orders also increased due to Christmas and the beginning of a new year. This indicates a trend of monthly seasonality among the customers to purchase products for the onset of the festivals and other relevant seasons.

2.c. 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: Morning, 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 "Mornings"
            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 order_time_of_day,
       count(*) as order_count
from target.orders
group by order_time_of_day
order by order_count;
```

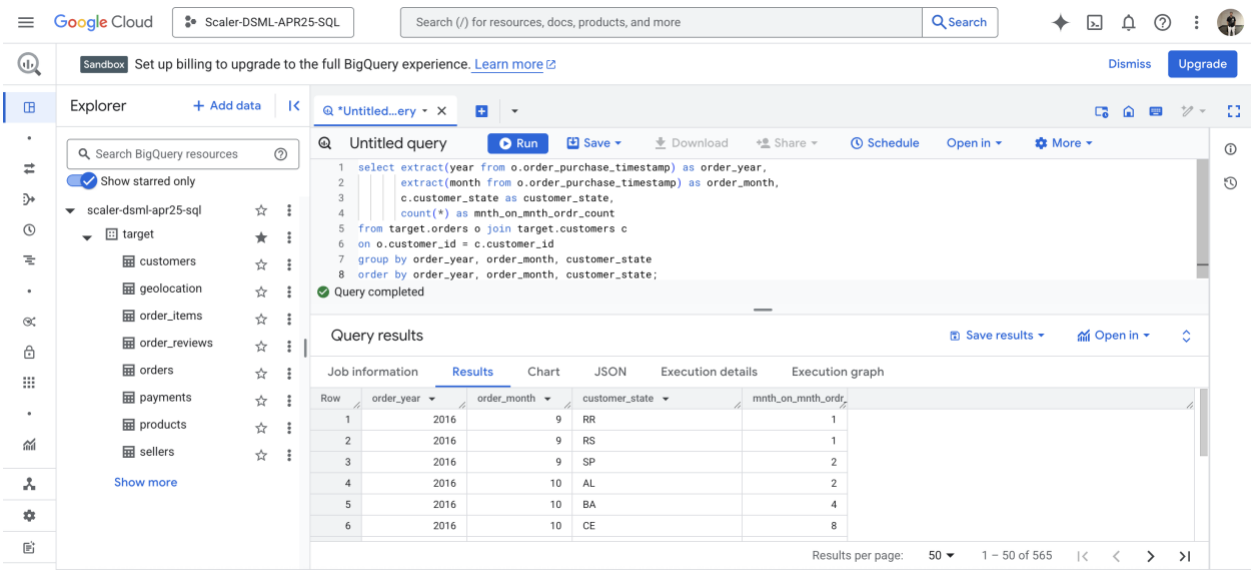


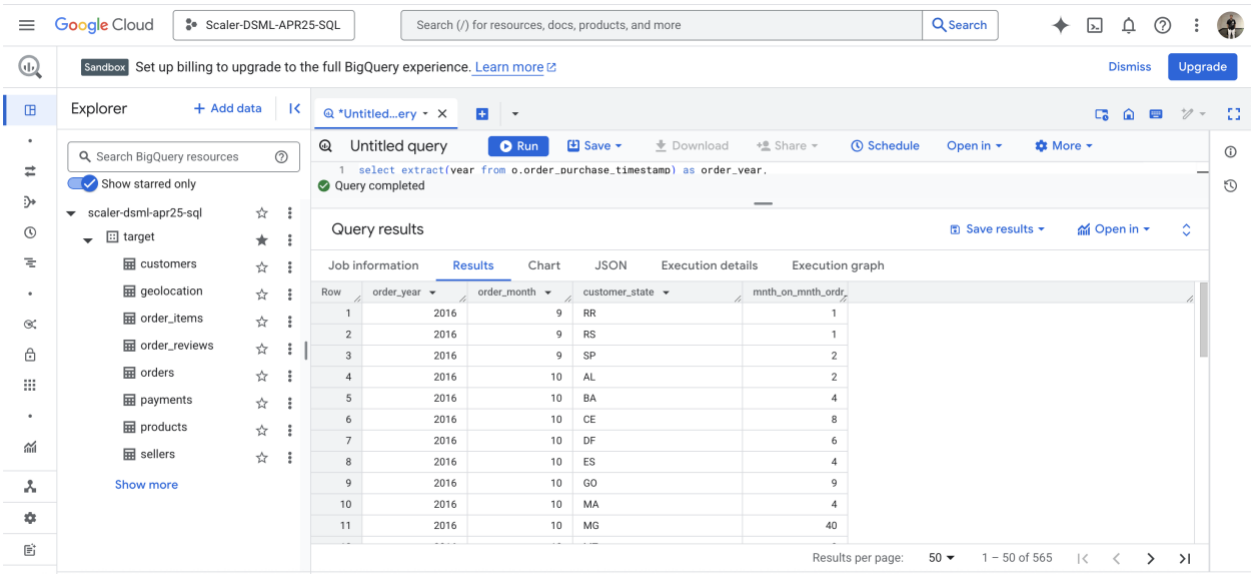
The Brazilian customers mostly place their orders during “Afternoon” time with 38135 orders placed between 13-18 hours. Additionally, the customers have also placed more than 20,000 orders during the

morning as well as the night hours which suggests that majority of the orders in total are placed post dawn.

3. Evolution of E-commerce orders in the Brazil region:
3.a. Get the month-on-month no. of orders placed in each state.

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

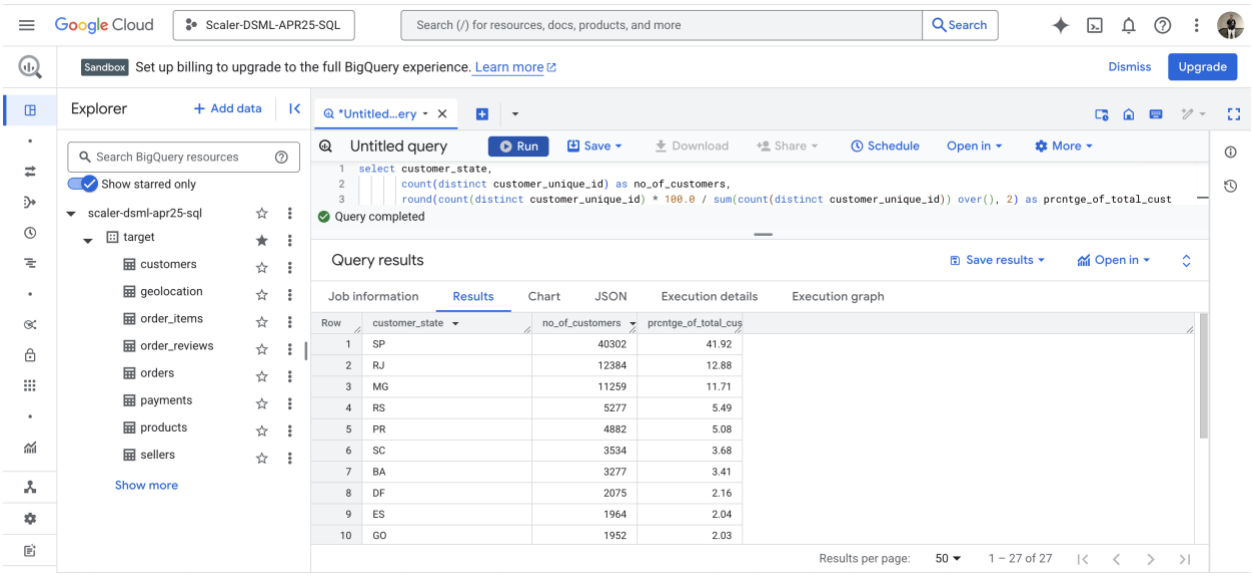


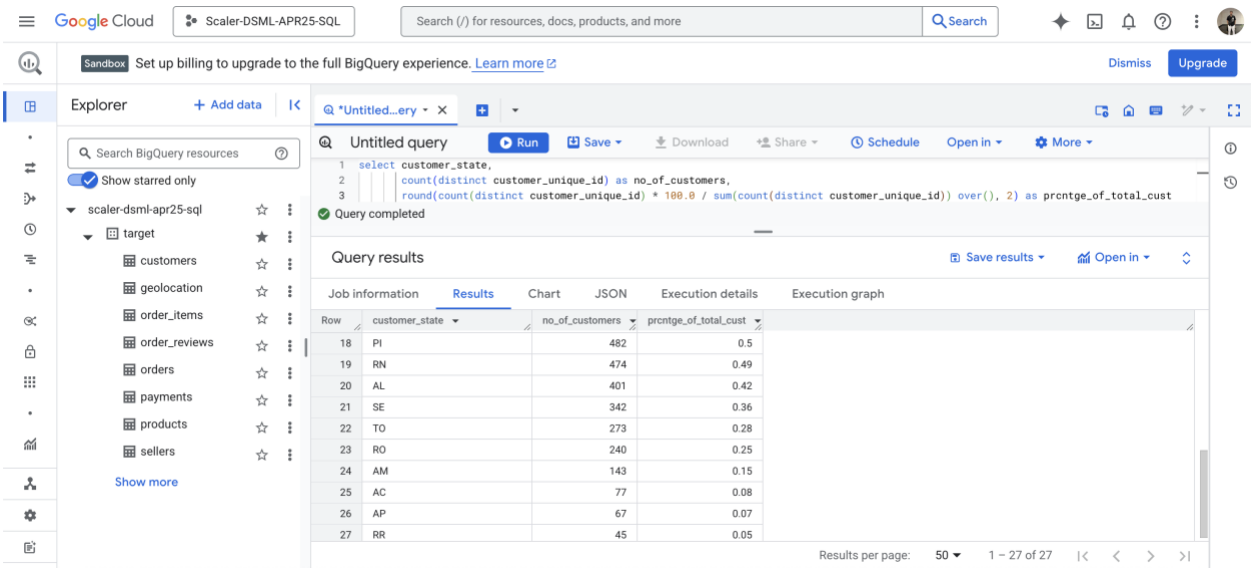


São Paulo (SP) is the dominating state followed by Rio de Janeiro (RJ) and Minas Gerais (MG) in the state-wise distribution of the orders placed which is the expected e-commerce market in Brazil. The dominance of e-commerce in São Paulo can be leveraged in emerging and fast-growing states such as Minas Gerais. Regional growth in under-performing states such as Rio Grande do Sul (RS), Bahia (BA) and others is required through enhanced marketing, fast-delivery and product availability.

3.b. How are the customers distributed across all the states?

```sql
select customer_state,
       count(distinct customer_unique_id) as no_of_customers,
       round(count(distinct customer_unique_id) * 100.0 / sum(count(distinct
customer_unique_id)) over(), 2) as prcntge_of_total_cust
from target.customers
group by customer_state
order by no_of_customers desc;
```

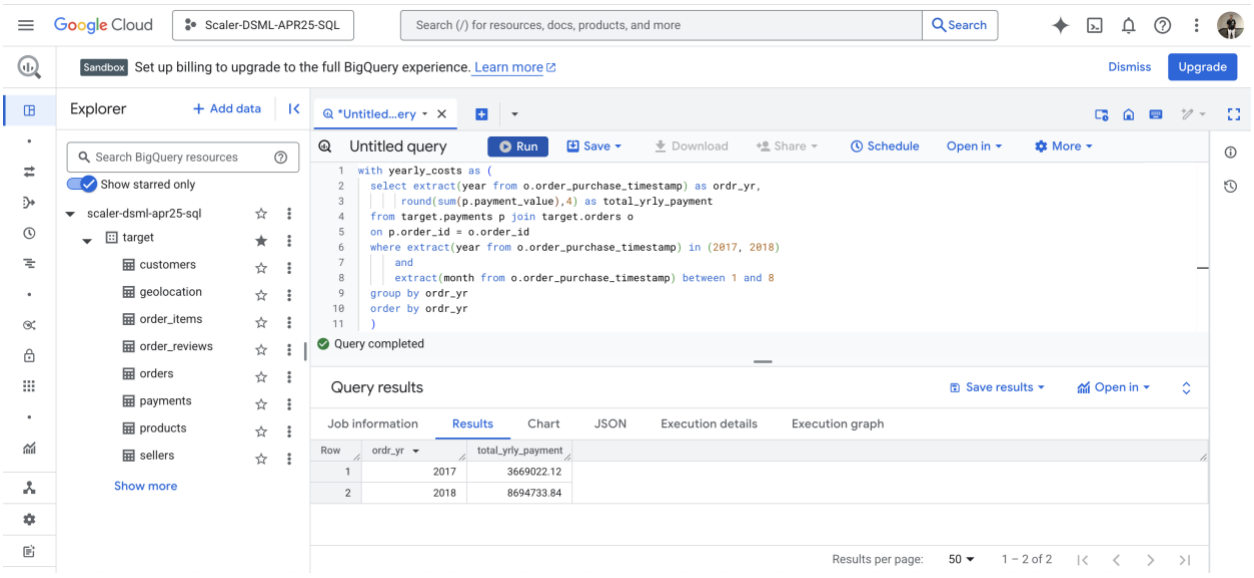


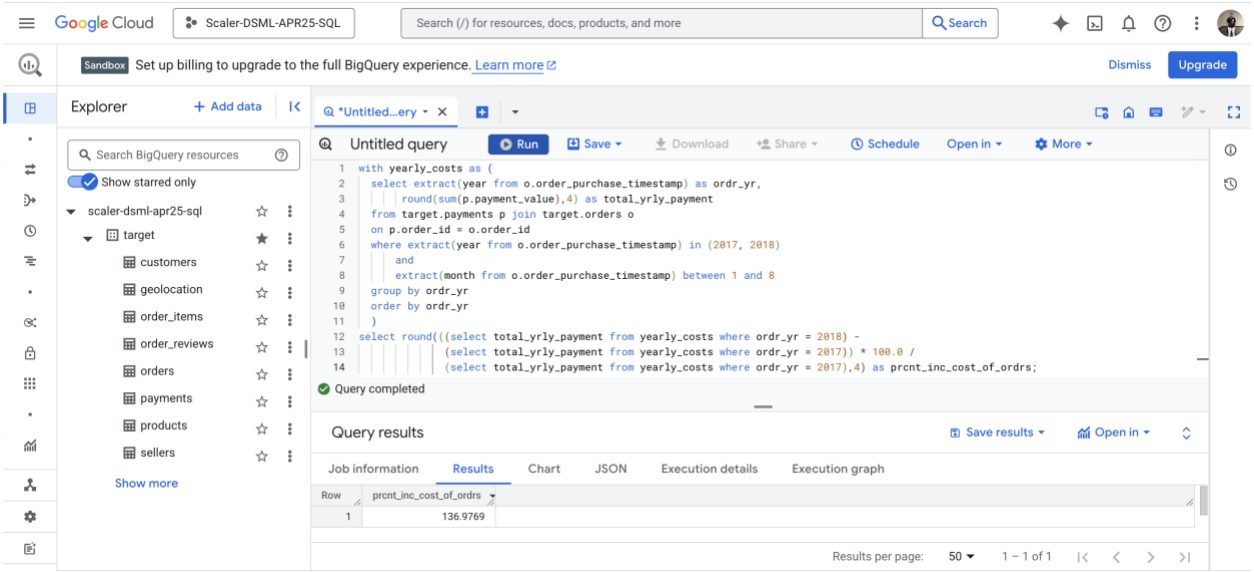


Majority of the customers are based in São Paulo and a significant proportion of them are based in emerging states of Rio de Janeiro and Minas Gerais as indicated by the percentage of total customers. Some of the states that require e-commerce growth and development are Acre (AC), Amapá (AP) and Roraima (RR).

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

```
with yearly_costs as (
  select extract(year from o.order_purchase_timestamp) as ordr_yr,
       sum(p.payment_value) as total_yrly_payment
  from target.payments p join target.orders o
  on p.order_id = o.order_id
  where extract(year from o.order_purchase_timestamp) in (2017, 2018)
      and
      extract(month from o.order_purchase_timestamp) between 1 and 8
  group by ordr_yr
  order by ordr_yr
  )
select round(((select total_yrly_payment from yearly_costs where ordr_yr = 2018) -
              (select total_yrly_payment from yearly_costs where ordr_yr = 2017)) *
100.0 /
              (select total_yrly_payment from yearly_costs where ordr_yr = 2017),4)
as prcnt_inc_cost_of_ordrs;
```

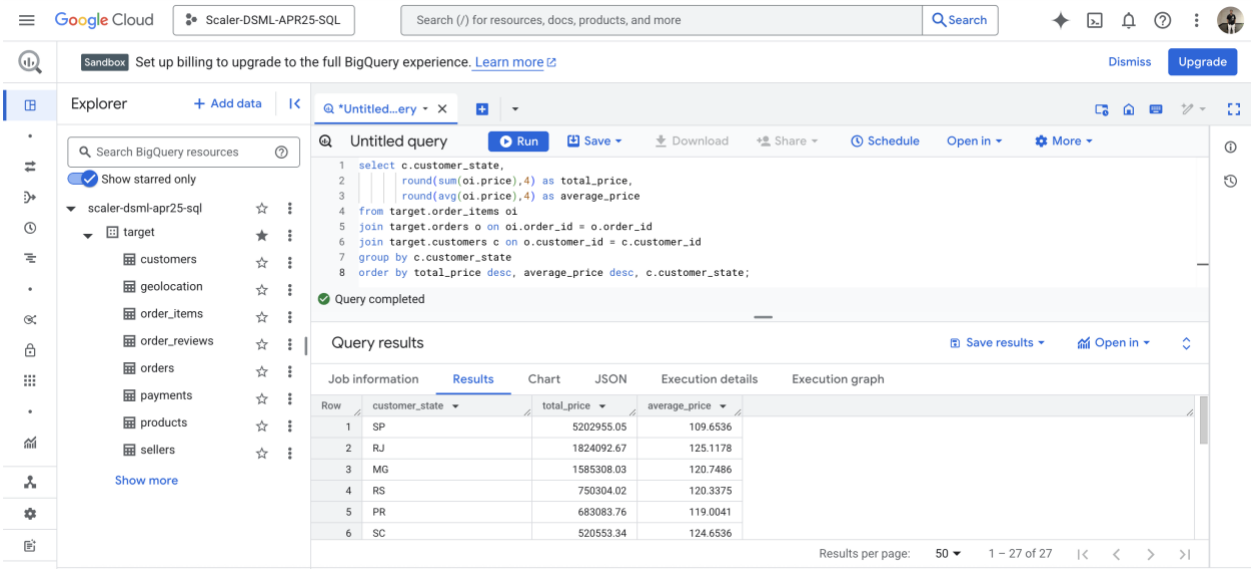


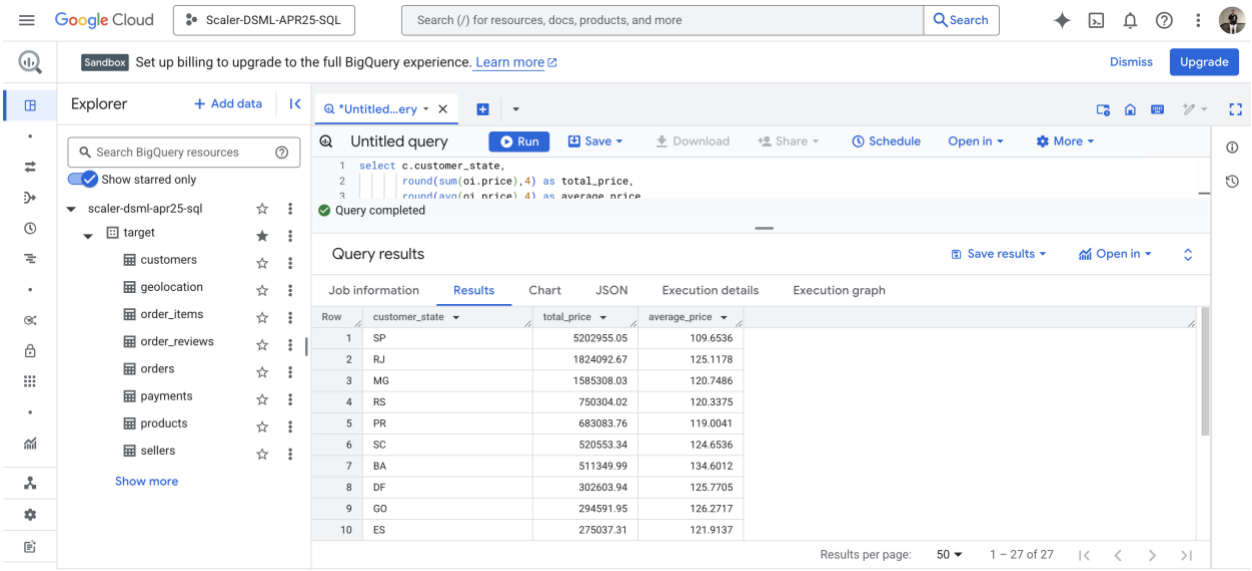


The total cost of orders was $3669022.12 and $8695733.84 in the year 2017 and 2018 respectively (include months between Jan to Aug only). The percentage increase between the defined period was 136.9769% which indicates a significant growth in the e-commerce industry in Brazil.

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

```
select c.customer_state,
       round(sum(oi.price),4) as total_price,
       round(avg(oi.price),4) as average_price
from target.order_items oi
join target.orders o on oi.order_id = o.order_id
join target.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by total_price desc, average_price desc, c.customer_state;
```

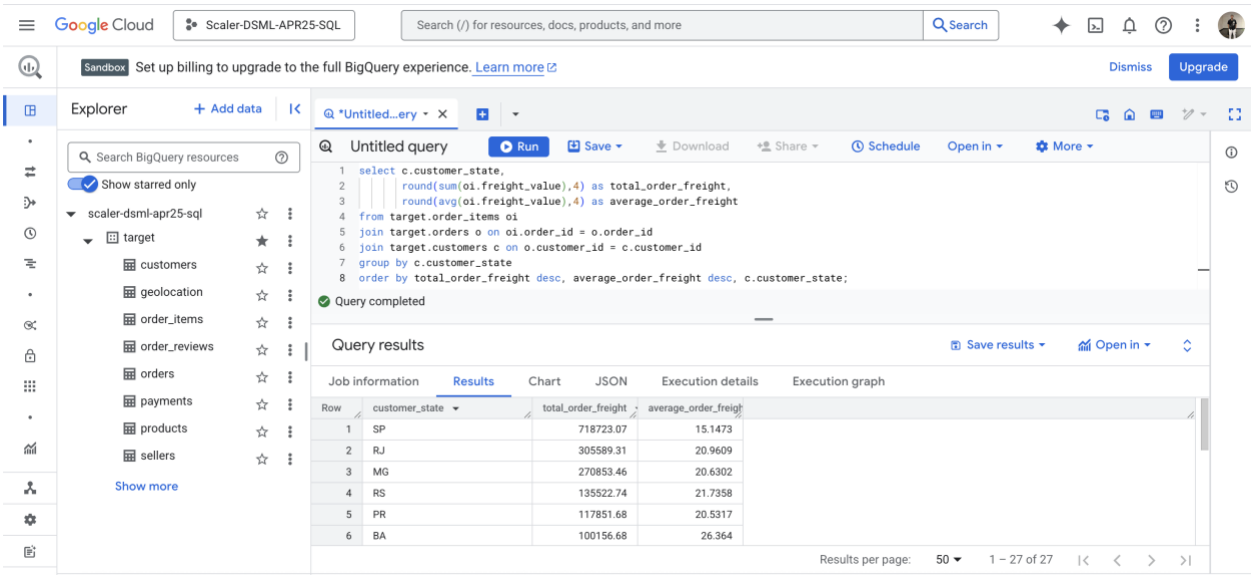


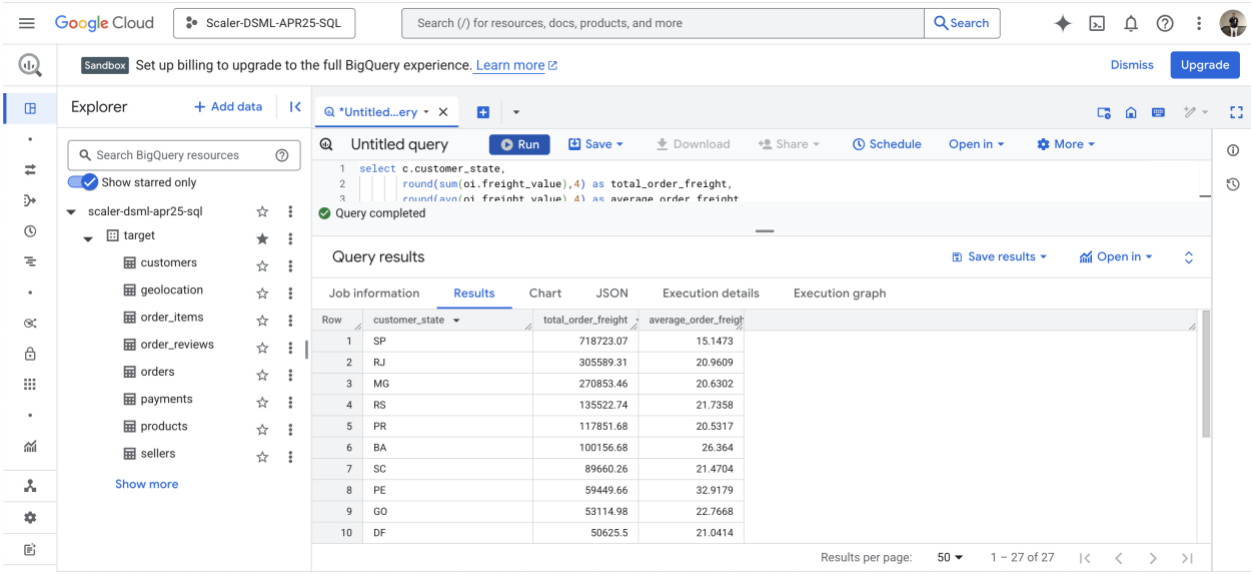


The total value and average value of order price for each state was calculated as shown in the image above. São Paulo (SP), Rio de Janeiro (RJ), Minas Gerais (MG), Rio Grande do Sul (RS) and Paraná (PR) are the top 5 states (Ordered by total_price in descending order followed by average_price in descending order and then by customer_state alphabetically).

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

```
select c.customer_state,
       round(sum(oi.freight_value),4) as total_order_freight,
       round(avg(oi.freight_value),4) as average_order_freight
from target.order_items oi
join target.orders o on oi.order_id = o.order_id
join target.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by total_order_freight desc, average_order_freight desc, c.customer_state;
```





The total value and average value of order freight for each state was calculated as shown in the image above. São Paulo (SP), Rio de Janeiro (RJ), Minas Gerais (MG), Rio Grande do Sul (RS) and Paraná (PR) are the top 5 states (Ordered by total_order_freight in descending order followed by average_order_freight in descending order and then by customer_state alphabetically).

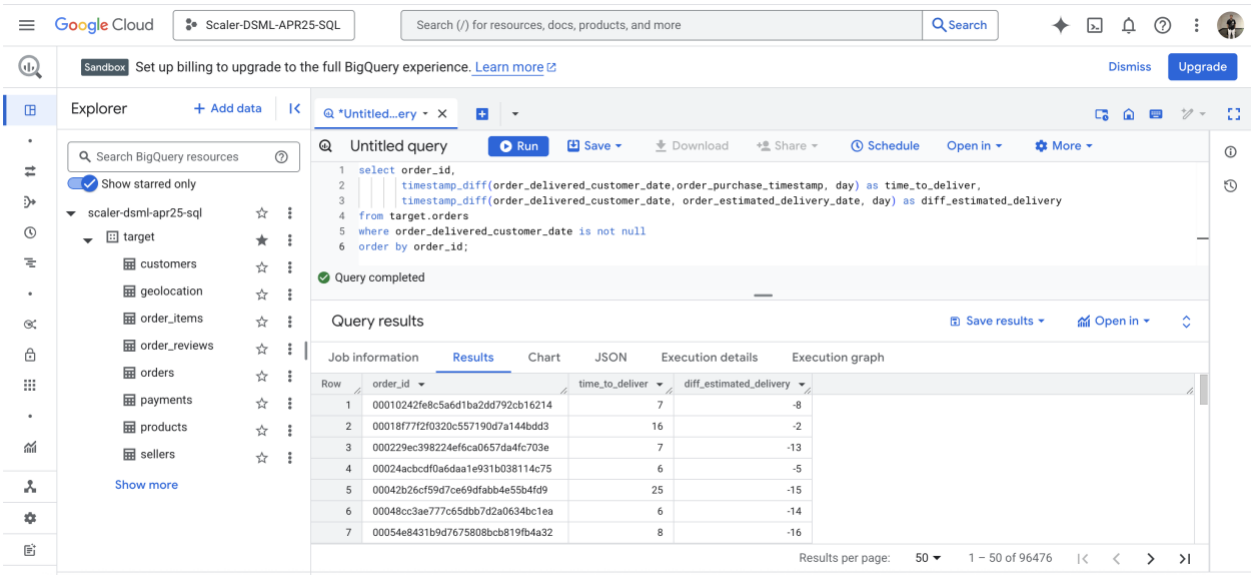
5. **Analysis based on sales, freight and delivery time.**

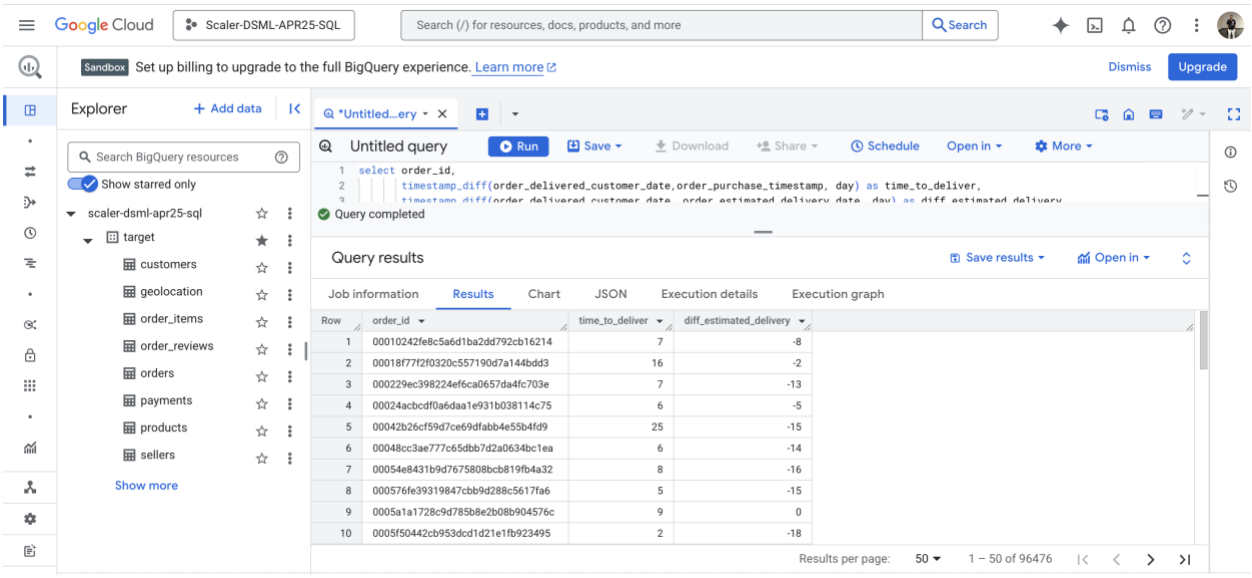
5.a. **Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query. You can calculate the**

delivery time and the difference between the estimated & actual delivery date using the given formula:

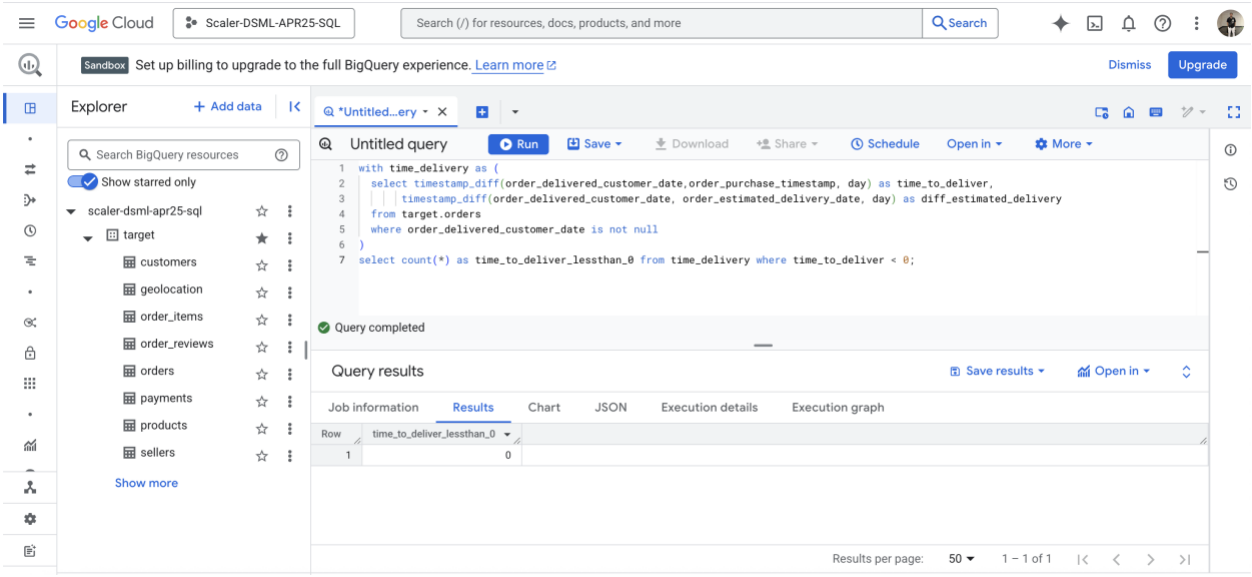
i. time_to_deliver = **order_delivered_customer_date - order_purchase_timestamp**
ii. diff_estimated_delivery = **order_delivered_customer_date - order_estimated_delivery_date**

```
select order_id,
       timestamp_diff(order_delivered_customer_date,order_purchase_timestamp, day)
as time_to_deliver,
       timestamp_diff(order_delivered_customer_date, order_estimated_delivery_date,
day) as diff_estimated_delivery
from target.orders
where order_delivered_customer_date is not null
order by order_id;
```



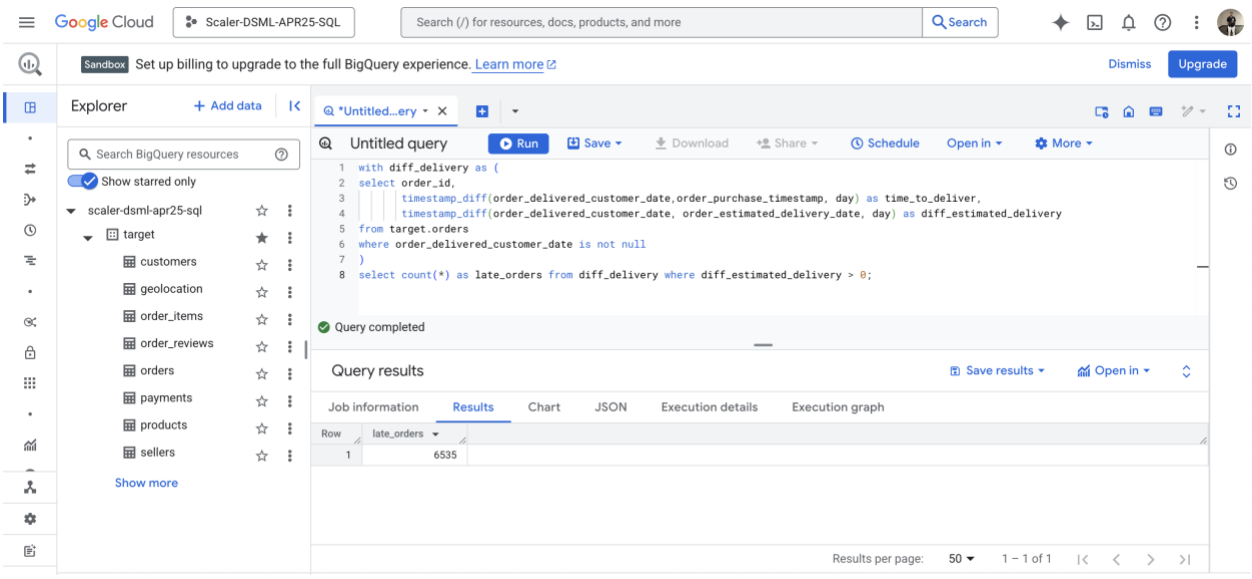



```sql
with time_delivery as (
  select timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,
day) as time_to_deliver,
      timestamp_diff(order_delivered_customer_date,
order_estimated_delivery_date, day) as diff_estimated_delivery
  from target.orders
  where order_delivered_customer_date is not null
)
select count(*) as time_to_deliver_lessthan_0 from time_delivery where
time_to_deliver < 0;
```



Time to deliver is the total time taken from when the customer placed the order to when the customer received the product, showing how long customers wait for their orders. This difference has to be zero or positive since the date when the customer receives the order is always greater than or equal to the purchase date. If this value is negative, it means that the delivery occurred before purchase, indicating a data error. There are no such values where time to deliver is less than 0 which indicates that there is no error.

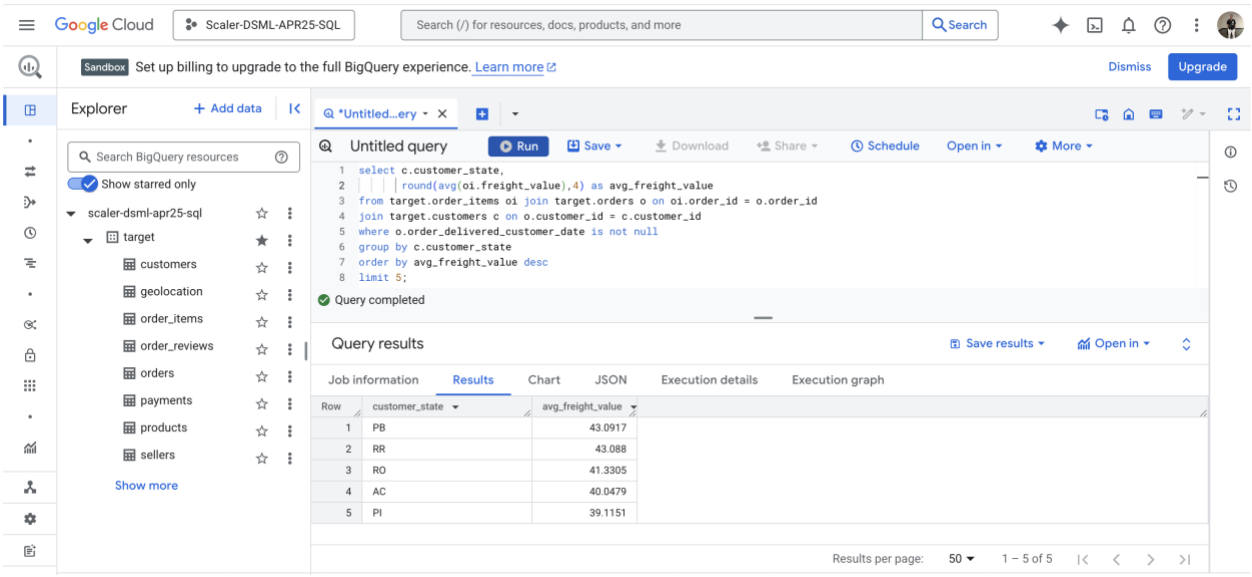
```
with diff_delivery as (
select order_id,
       timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,
day) as time_to_deliver,
       timestamp_diff(order_delivered_customer_date,
order_estimated_delivery_date, day) as diff_estimated_delivery
from target.orders
where order_delivered_customer_date is not null
)
select count(*) as late_orders from diff_delivery where diff_estimated_delivery >
0;
```



On the other hand, diff_estimated_delivery measures the difference between the actual delivery date and the estimated delivery date. If this difference is negative, it means that the order was delivered early, zero means delivered on time and positive indicates that the order was delivered late. Out of total orders (96,476 orders), 6,535 orders have a positive diff_estimated_delivery value. Target should ensure that year-on-year, this value remains zero or below zero.

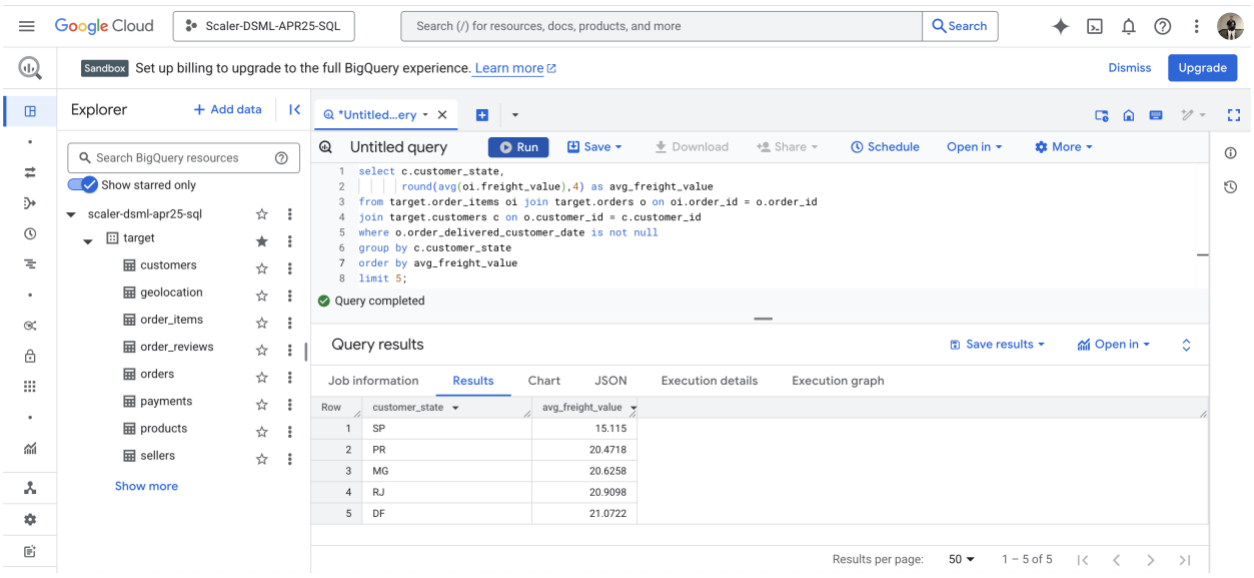
5.b. Find out the top 5 states with the highest & lowest average freight value.

```sql
select c.customer_state,
       round(avg(oi.freight_value),4) as avg_freight_value
from target.order_items oi join target.orders o on oi.order_id = o.order_id
join target.customers c on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state
order by avg_freight_value desc
limit 5;
```



The top 5 states with the highest average freight value are Paraíba (PB), Roraima (RR), Rondônia (RO), Acre (AC) and Piauí (PI) with Paraíba (PB) having highest the average freight value of 43.0917 and Piauí (PI) with the 5th highest average freight value of 39.1151.

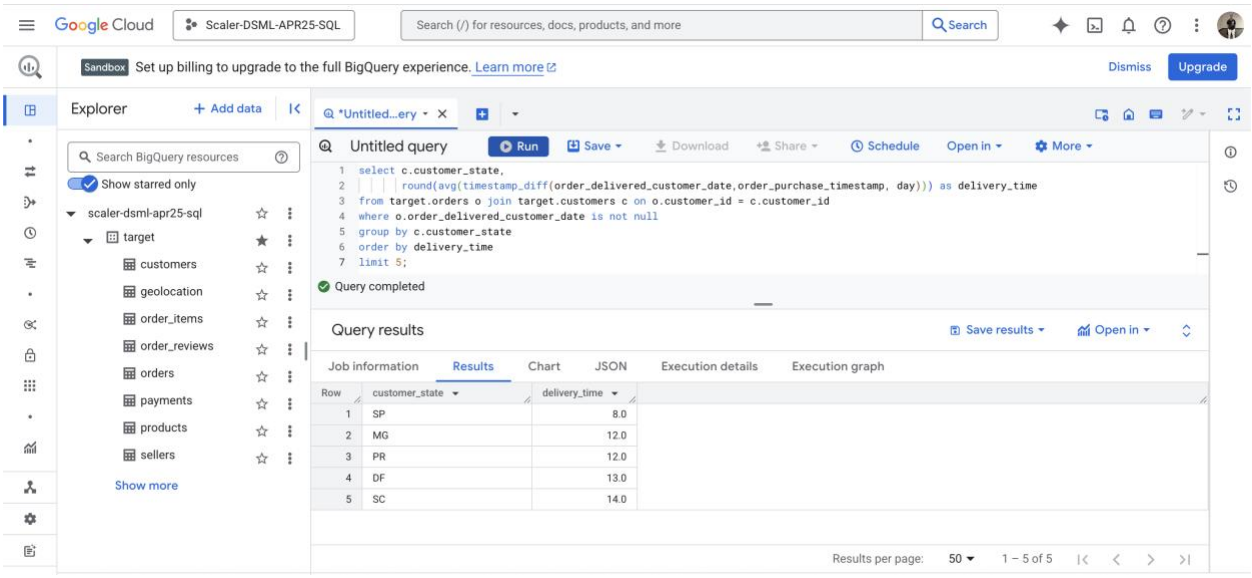
```sql
select c.customer_state,
       round(avg(oi.freight_value),4) as avg_freight_value
from target.order_items oi join target.orders o on oi.order_id = o.order_id
join target.customers c on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state
order by avg_freight_value
limit 5;
```



The top 5 states with the lowest average freight value are São Paulo (SP), Paraná (PR), Minas Gerais (MG), Rio de Janeiro (RJ) and Distrito Federal (DF) with São Paulo (SP) having the lowest average freight value of 15.115 and Distrito Federal (DF) with the 5th lowest average freight value of 21.0722.

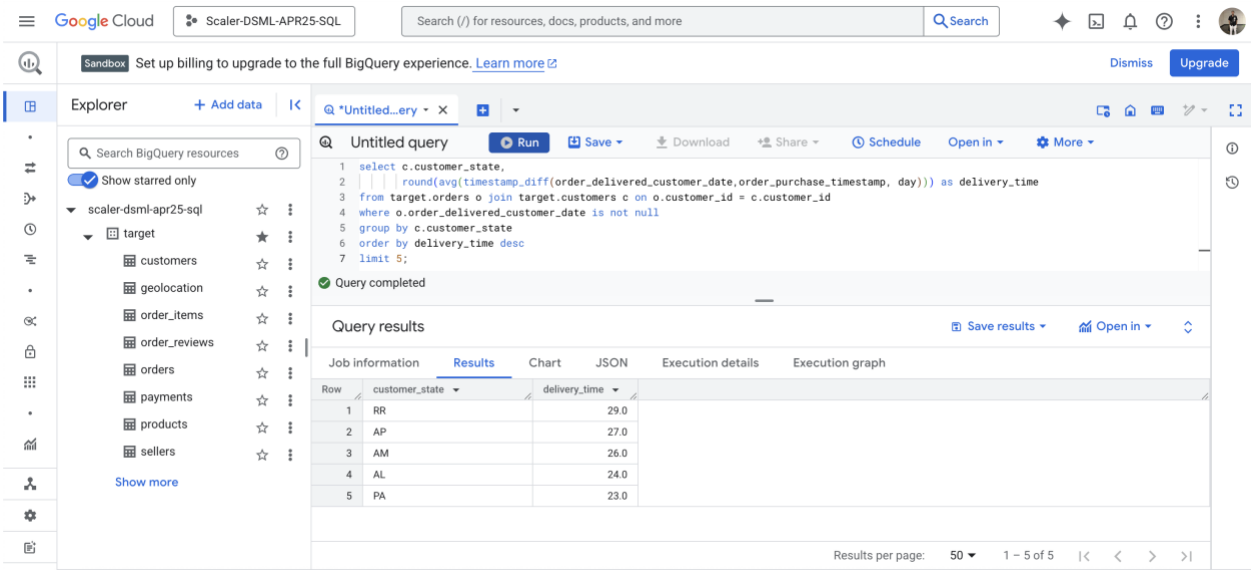
5.c. Find out the top 5 states with the highest & lowest average delivery time.

```
select c.customer_state,

round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,
day))) as delivery_time
from target.orders o join target.customers c on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state
order by delivery_time
limit 5;
```



The top 5 states with the lowest average delivery time are the southern/southeastern states of São Paulo (SP), Minas Gerais (MG), Paraná (PR), Distrito Federal (DF) and Santa Catarina (SC). That benefit from better infrastructure. Rio de Janeiro (RJ) didn't make it to the top 5 (6th place) given its proximity to São Paulo (SP).

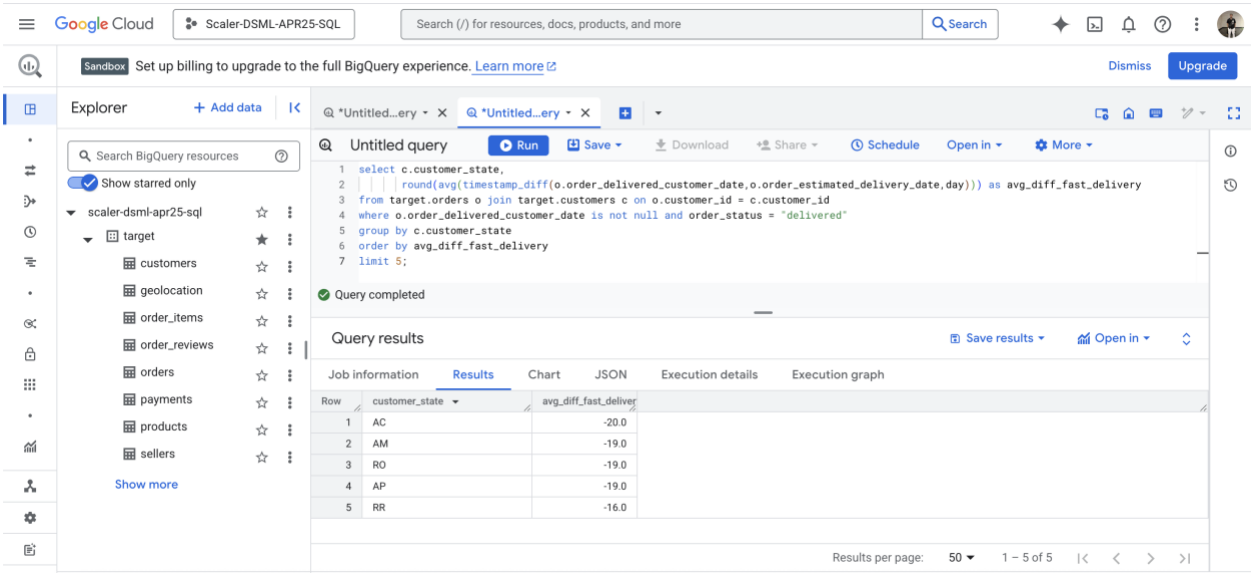
```
select c.customer_state,

round(avg(timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,
day))) as delivery_time
from target.orders o join target.customers c on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state
order by delivery_time desc
limit 5;
```



On the other hand, the top 5 states with the highest average delivery time are the northeastern/northern states of Roraima (RR), Amapá (AP), Amazonas (AM), Alagoas (AL) and Pará (PA) reflecting logistical challenges in those regions of Brazil.

5.d. 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.

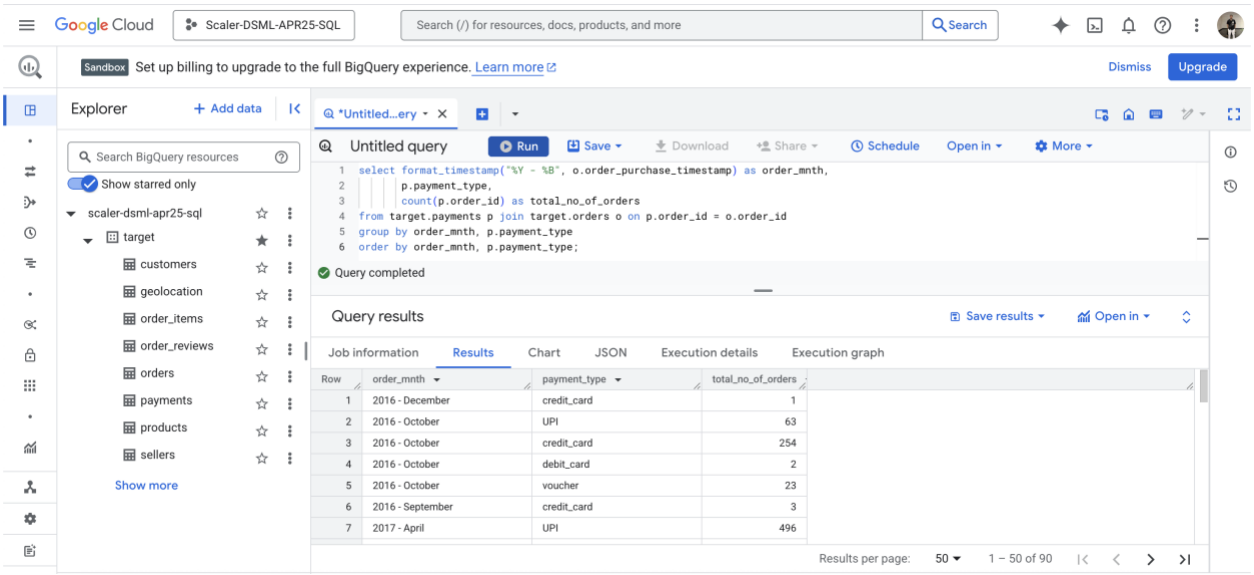
```
select c.customer_state,

round(avg(timestamp_diff(o.order_delivered_customer_date,o.order_estimated_delivery
_date,day))) as avg_diff_fast_delivery
from target.orders o join target.customers c on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null and order_status = "delivered"
group by c.customer_state
order by avg_diff_fast_delivery
limit 5;
```

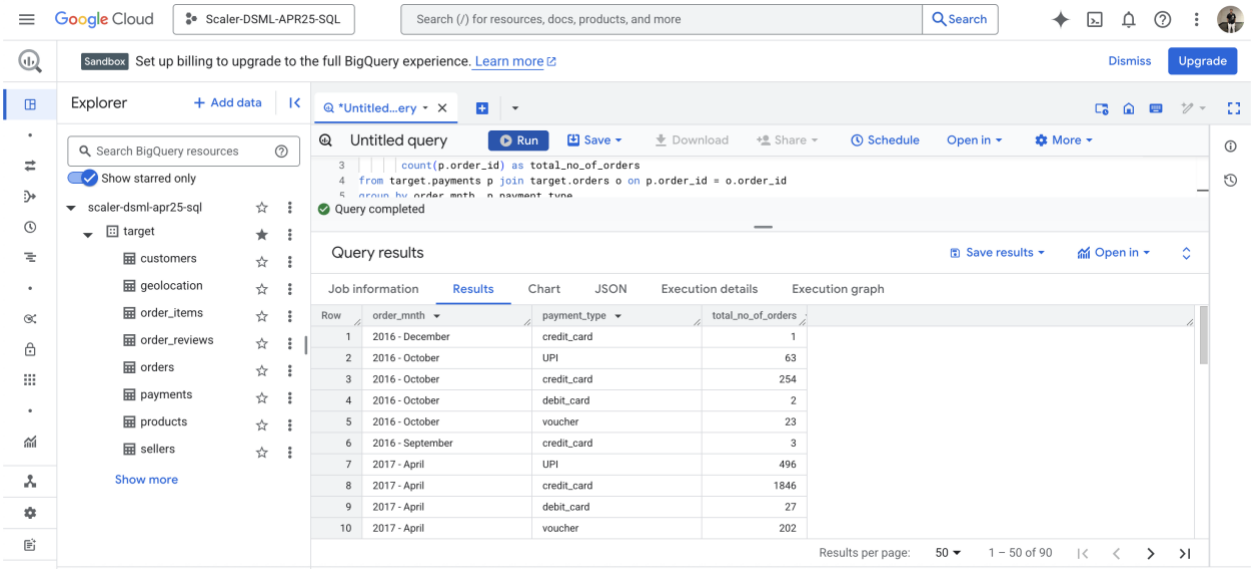


Acre (AC), Amazonas (AM), Rondônia (RO), Amapá (AP) and Roraima (RR) are the top 5 states where the order delivery is really fast as compared to the estimated date of delivery. A higher negative value of the difference between order_delivered_customer_date and order_estimated_delivery_date indicates much faster delivery. Customers in the state of Acre (AC) have reported fastest delivery of their orders with an approx. average of 20 days.

6. Analysis based on the payments:
6.a. Find the month-on-month no. of orders placed using different payment types.

```
select format_timestamp("%Y - %B", o.order_purchase_timestamp) as order_mnth,
       p.payment_type,
       count(p.order_id) as total_no_of_orders
from target.payments p join target.orders o on p.order_id = o.order_id
group by order_mnth, p.payment_type
order by order_mnth, p.payment_type;
```

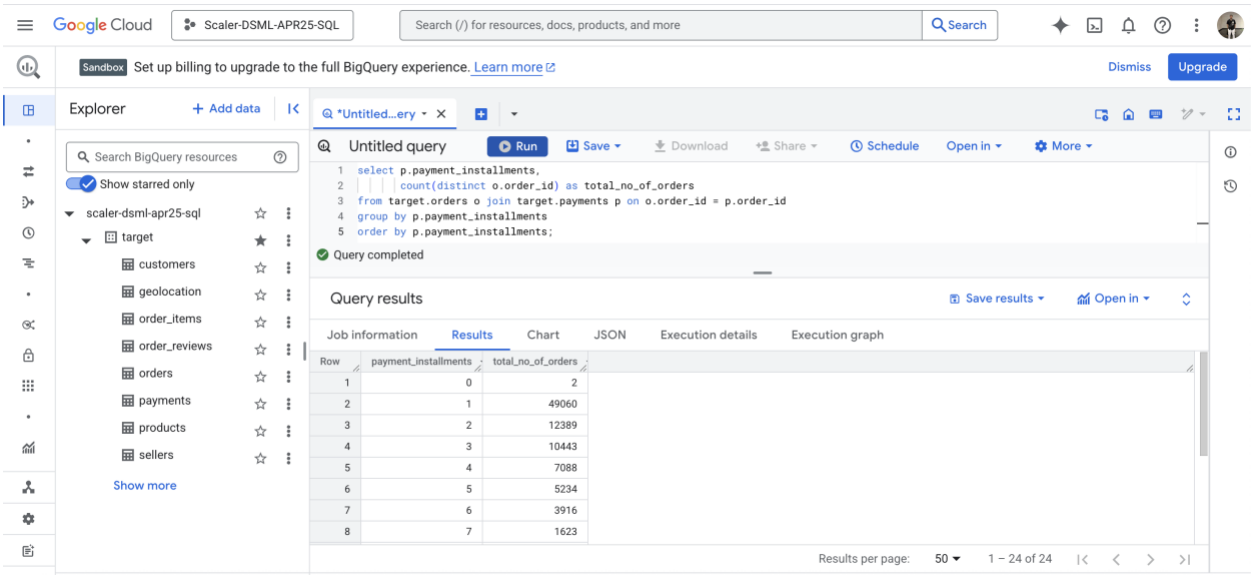


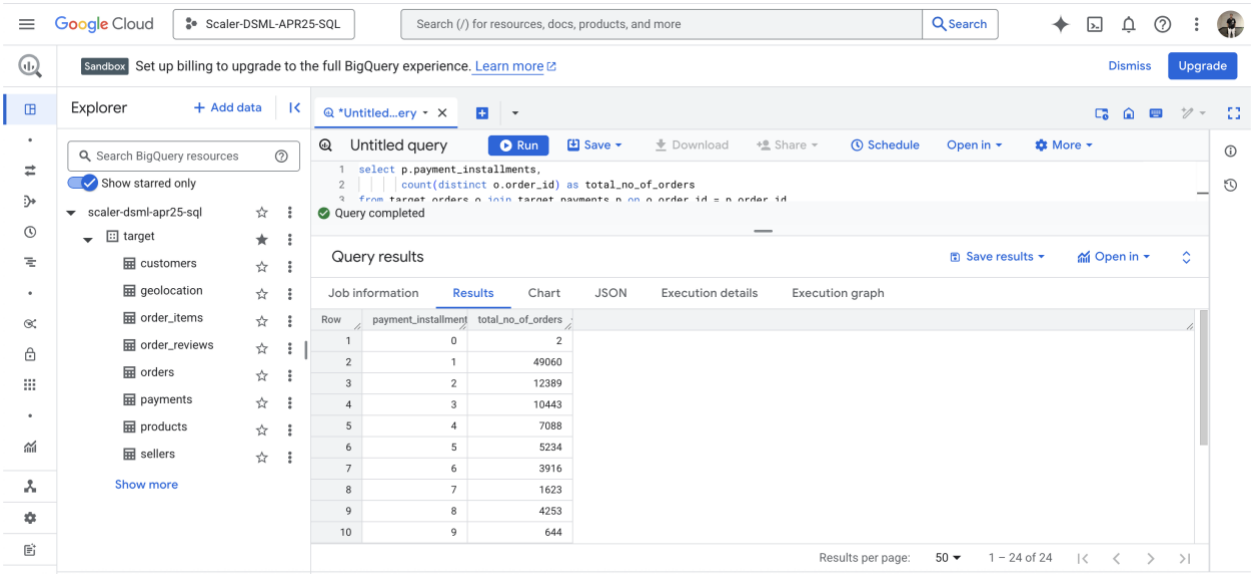


The result indicates that customers can comfortably place orders using a credit card, debit card, UPI or vouchers as the payment_type and majority of the orders have been placed with a credit card.

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

```sql
select p.payment_installments,
       count(distinct o.order_id) as total_no_of_orders
from target.orders o join target.payments p on o.order_id = p.order_id
group by p.payment_installments
order by p.payment_installments;
```





Customers that choose 0 or 1 installment indicates a strong preference for full payment or short-term installments. Many customers opt for flexible installments plan of 3-12 months suggesting that the full amount will be paid within a year of purchase.

However, long term installments plan (12+ months) indicate that the customers rely heavily on credit depending on the product. This means delayed revenue realization which affects cash flow. Hence, Target can opt for optimized payment plans with incentives for faster cash flow and offer flexibility to the customers.