Akul Vinod

Business Case: Target SQL

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.

Dataset:

https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb?usp=sharing

The data is available in 8 different csv files:

1. customers.csv
2. geolocation.csv
3. order_items.csv
4. payments.csv
5. reviews.csv
6. orders.csv
7. products.csv
8. sellers.csv

akulvinod22@gmail.com

The column description for these csv files is given below.

The customers.csv contain following features:

Features	**Description**
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer's location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The orders.csv contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.
order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery
order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products

akulvinod22@gmail.com

The order_items.csv contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The payments.csv contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

The geolocations.csv contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code
geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The sellers.csv contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller's location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The reviews.csv contain following features:

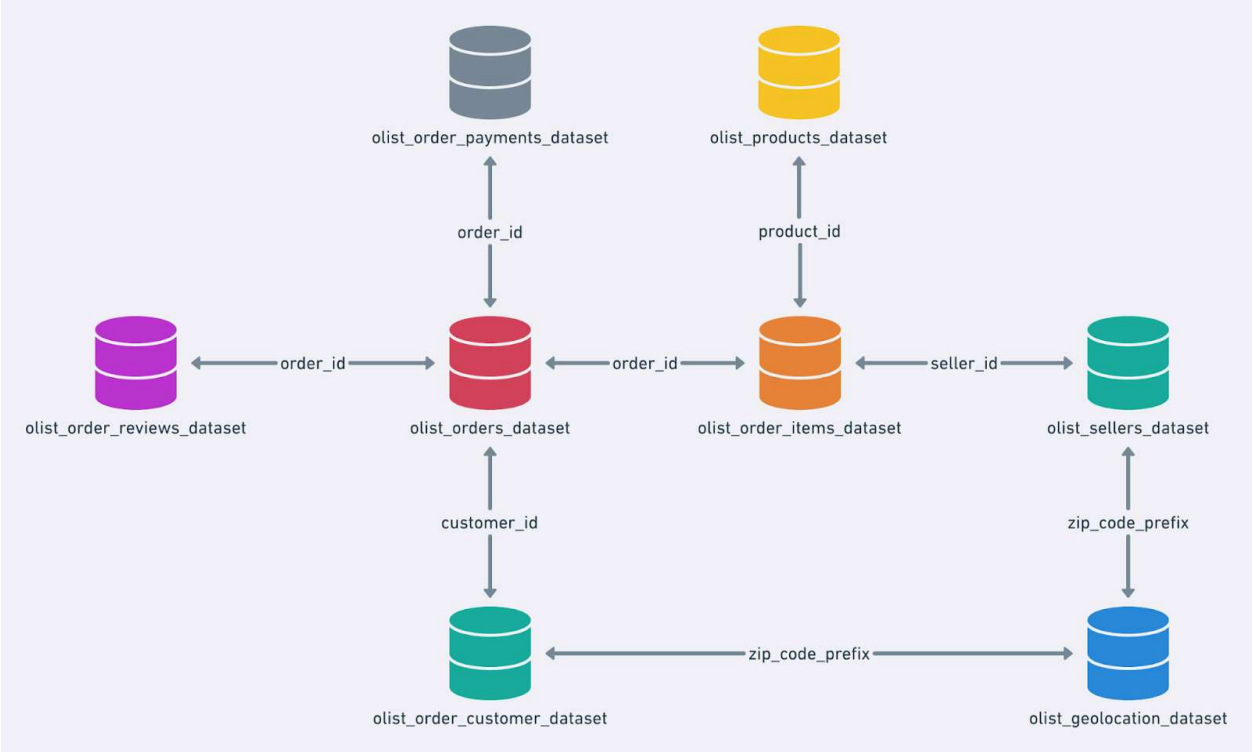
Features	Description
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

akulvinod22@gmail.com

The products.csv contain following features:

Features	Description
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	Length of the string which specifies the name given to the products ordered
product_description_lenght	Length of the description written for each product ordered on the site
product_photos_qty	Number of photos of each product ordered available on the shopping portal
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

akulvinod22@gmail.com




```sql
select * from `target.customers`;
select * from `target.geolocation`;
select * from `target.order_items`;
select * from `target.order_reviews`;
select * from `target.orders`;
select * from `target.payments`;
select * from `target.products`;
select * from `target.sellers`;
```

akulvinod22@gmail.com

Akul Vinod

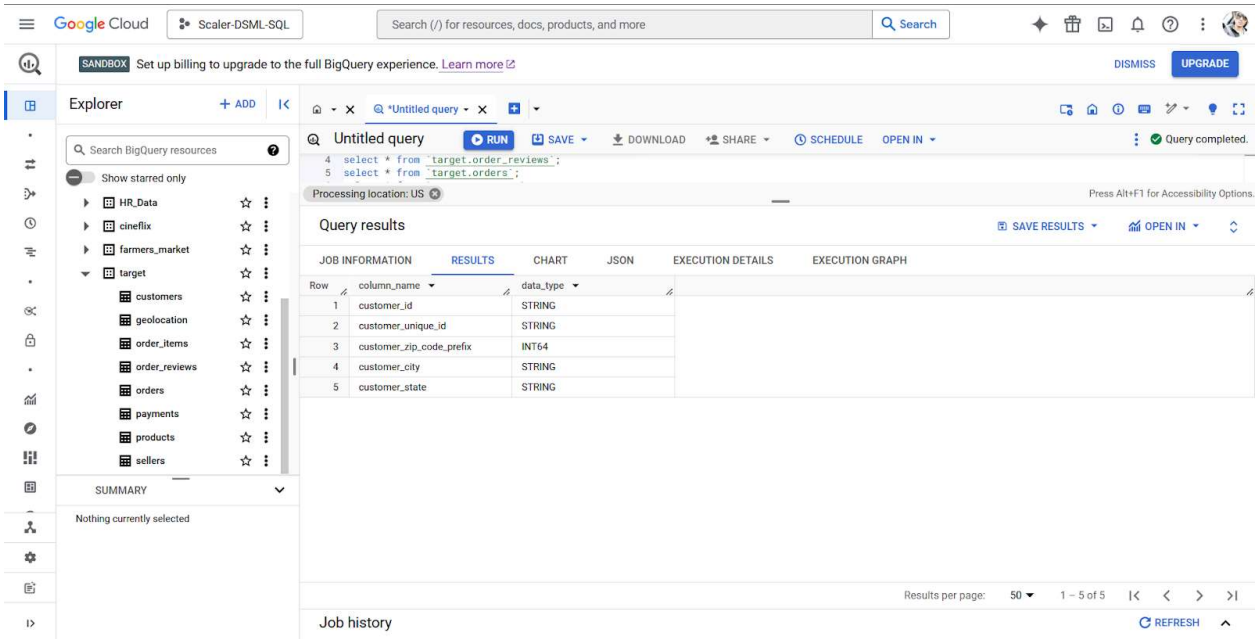
Problem Statement:

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.
What does 'good' look like?

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

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

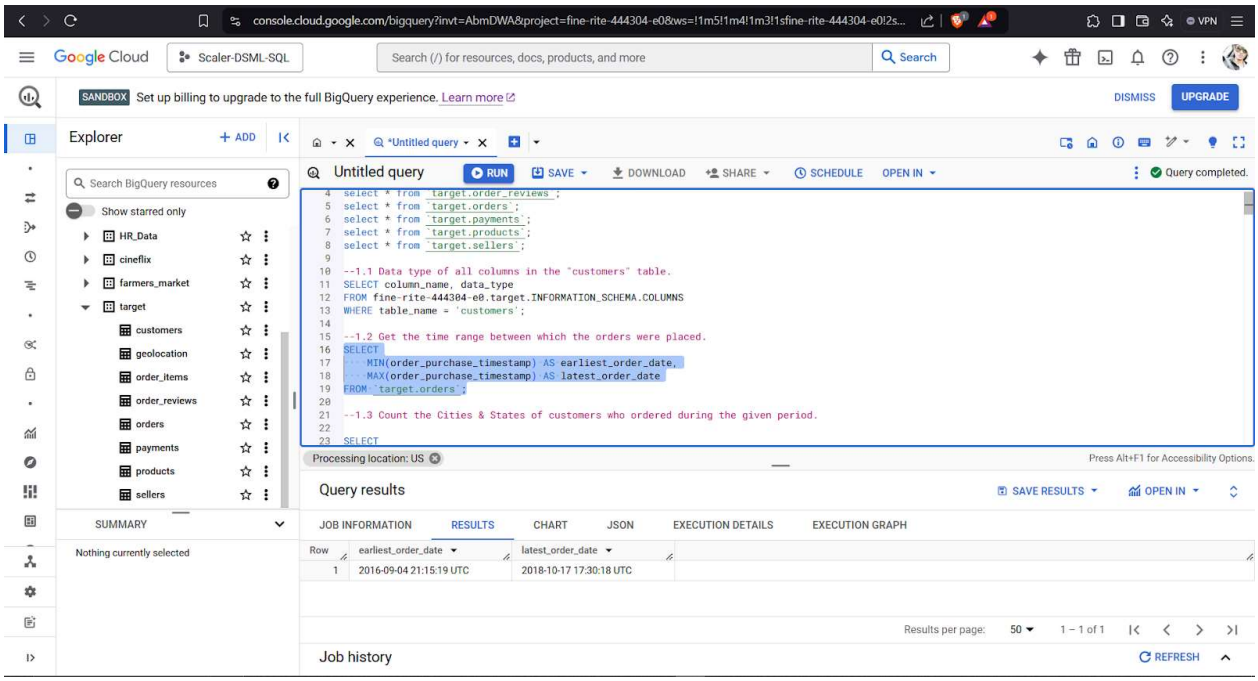
```
SELECT column_name, data_type
FROM fine-rite-444304-e0.target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customers';
```



akulvinod22@gmail.com

1.2 Get the time range between which the orders were placed.

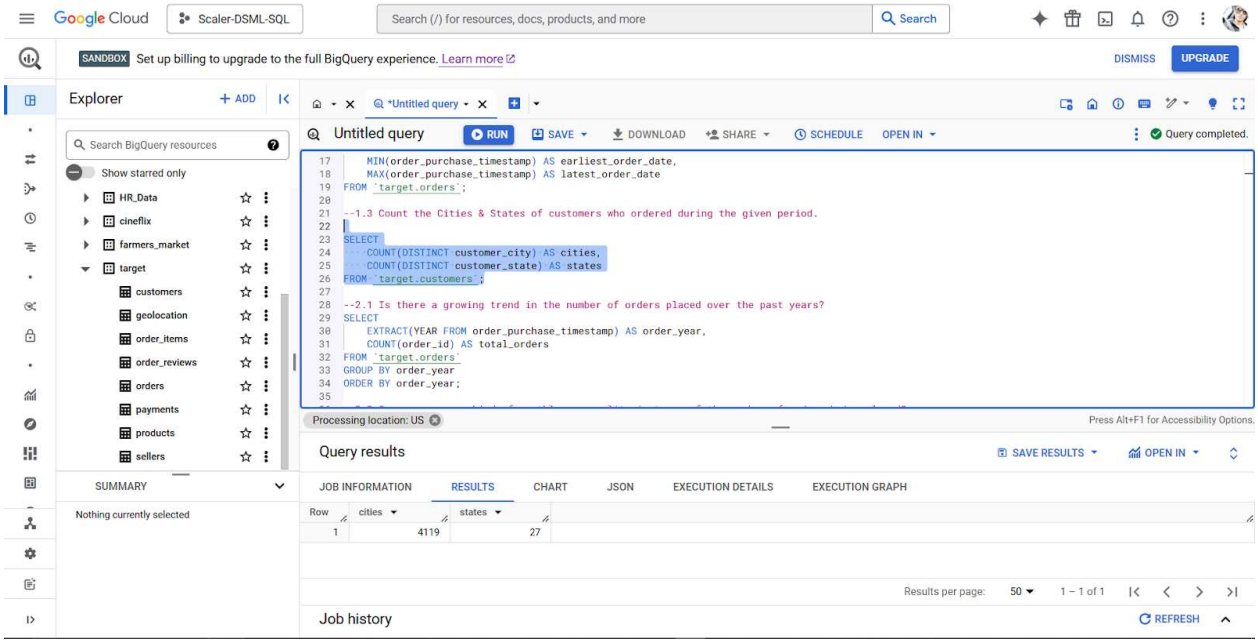
```sql
SELECT
    MIN(order_purchase_timestamp) AS earliest_order_date,
    MAX(order_purchase_timestamp) AS latest_order_date
FROM `target.orders`;
```



akulvinod22@gmail.com

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

```sql
SELECT
    COUNT(DISTINCT customer_city) AS cities,
    COUNT(DISTINCT customer_state) AS states
FROM `target.customers`;
```

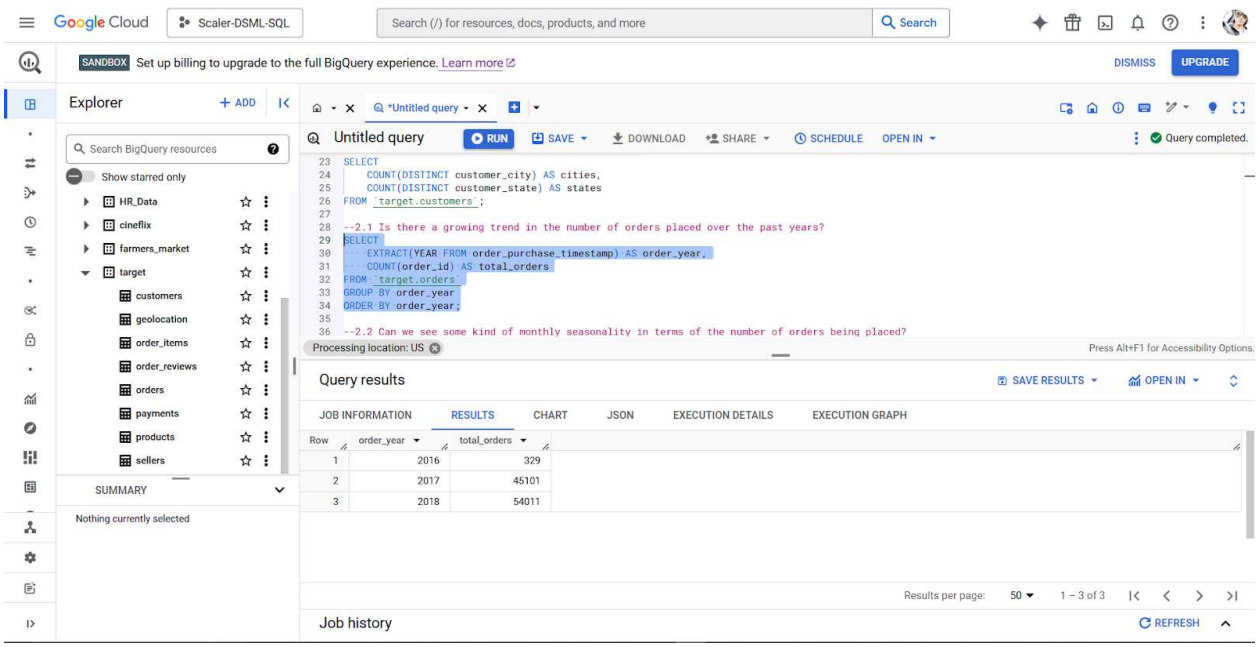


akulvinod22@gmail.com

2.In-depth Exploration:

2.1 Is there a growing trend in the number of orders placed over the past years?

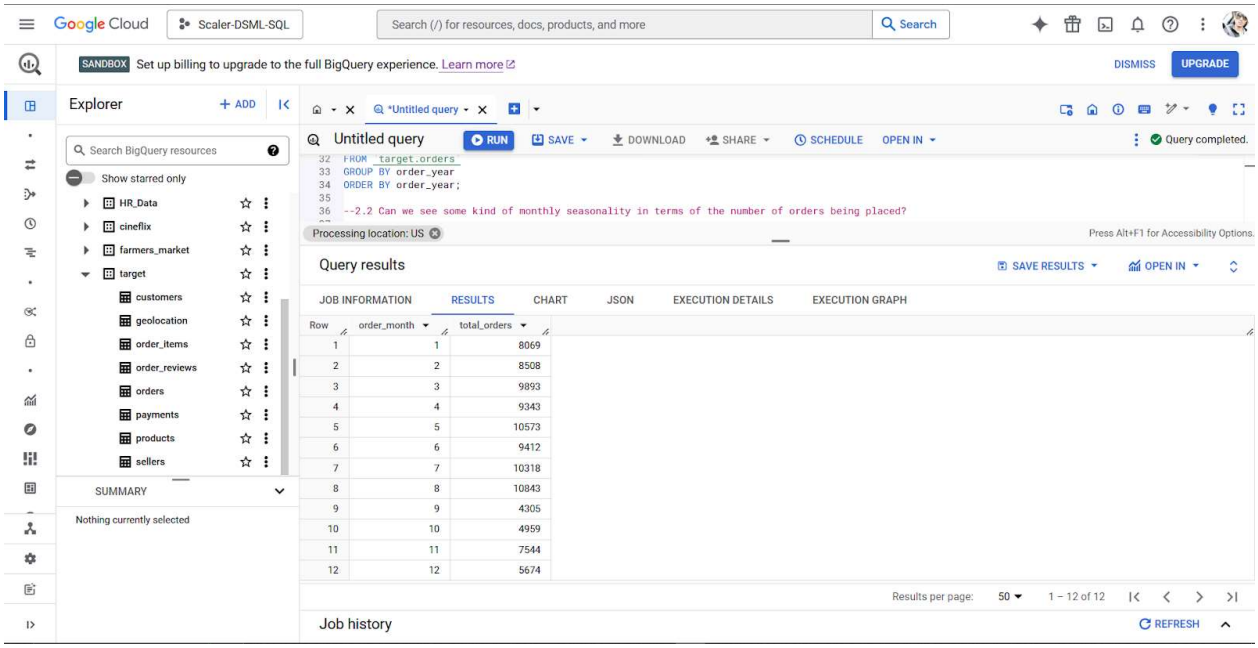
```
SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
    COUNT(order_id) AS total_orders
FROM `target.orders`
GROUP BY order_year
ORDER BY order_year;
```



akulvinod22@gmail.com

2.2 Can we see some kind of monthly seasonality in terms of the number of orders being placed?

```
SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
    COUNT(order_id) AS total_orders
FROM `target.orders`
GROUP BY order_month
ORDER BY order_month;
```

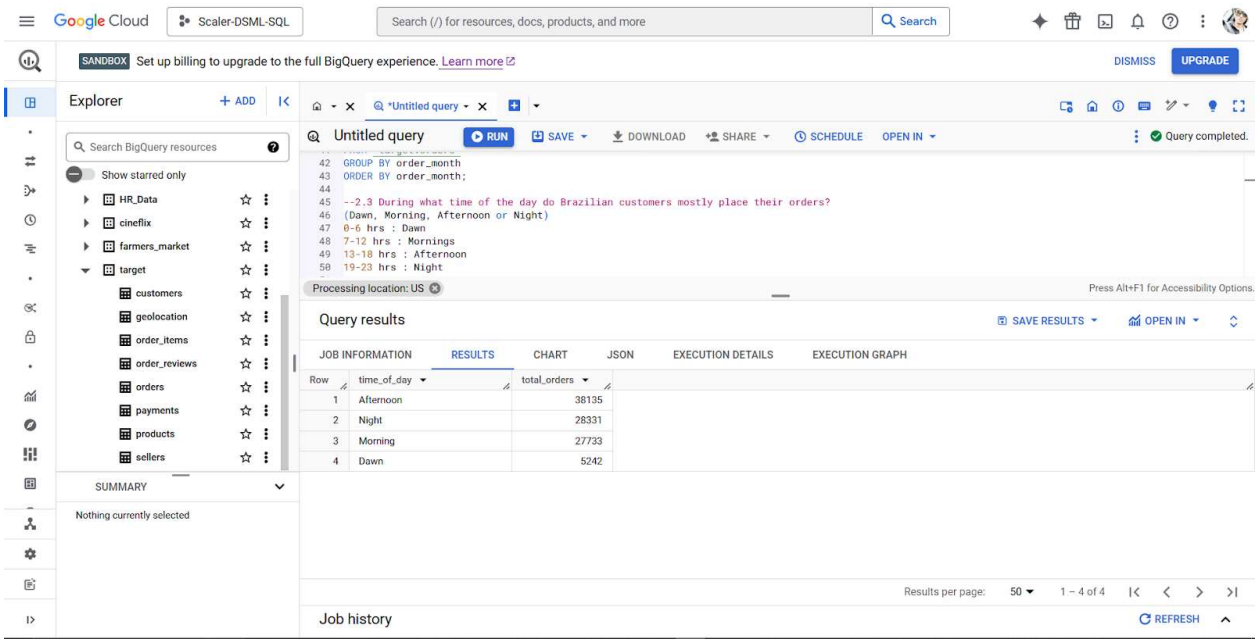


2.3 During what time of the day do Brazilian customers mostly place their orders?

(Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

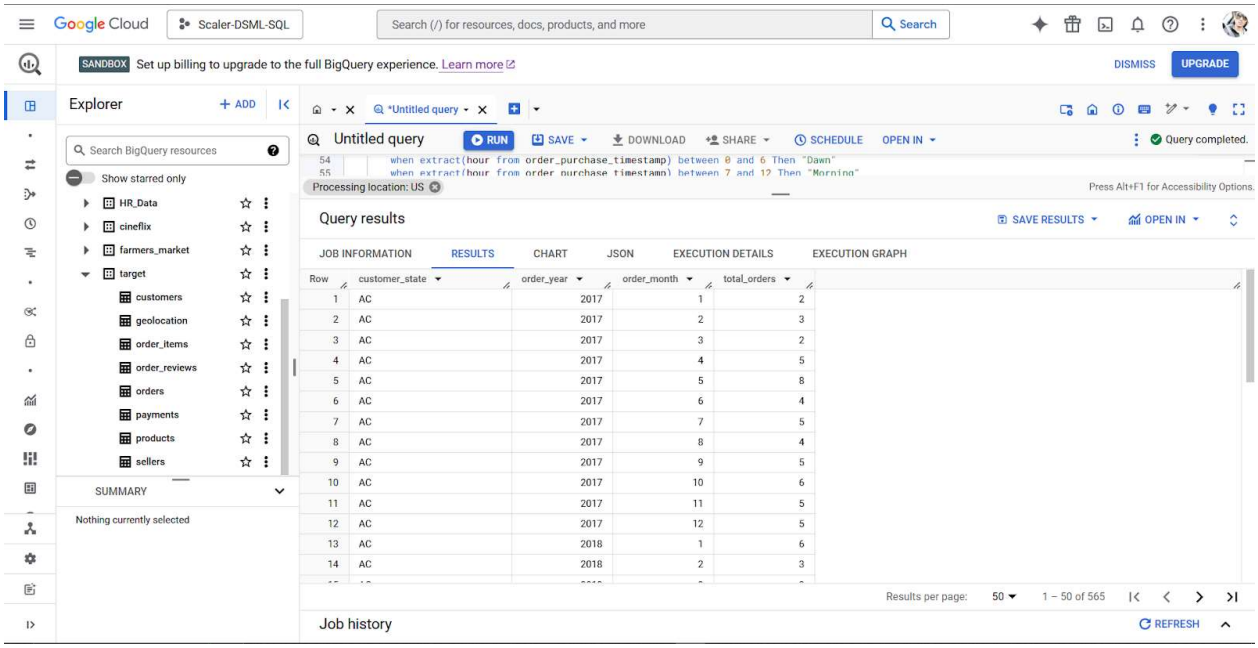
```
SELECT
    case
        when extract(hour from order_purchase_timestamp) between 0 and 6 Then "Dawn"
        when extract(hour from order_purchase_timestamp) between 7 and 12 Then
"Morning"
        when extract(hour from order_purchase_timestamp) between 13 and 18 Then
"Afternoon"
        when extract(hour from order_purchase_timestamp) between 19 and 23 Then
"Night"
    end as time_of_day,
COUNT(order_id) AS total_orders
FROM `target.orders`
GROUP BY time_of_day
ORDER BY total_orders desc;
```



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

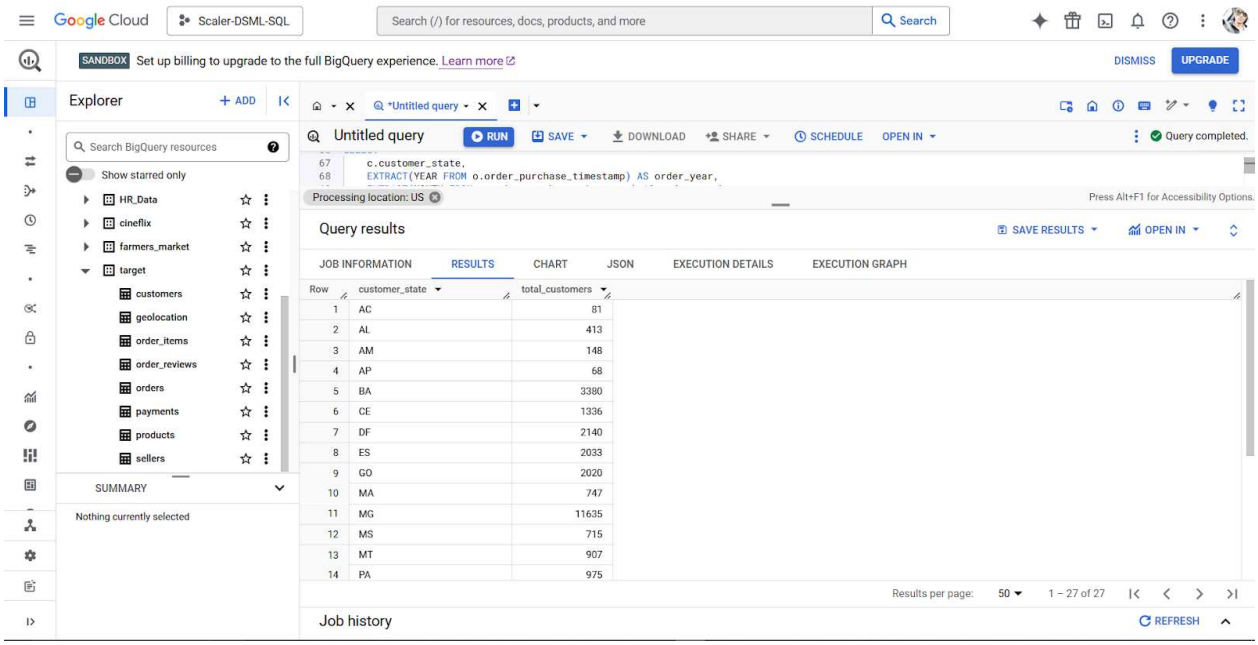
3.1 Get the month-on-month number of orders placed in each state

```
SELECT
    c.customer_state,
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
    COUNT(o.order_id) AS total_orders
FROM `target.orders` o
JOIN `target.customers` c
ON o.customer_id = c.customer_id
GROUP BY order_year, order_month, c.customer_state
ORDER BY c.customer_state,order_year, order_month, total_orders;
```



3.2 How are the customers distributed across all the states?

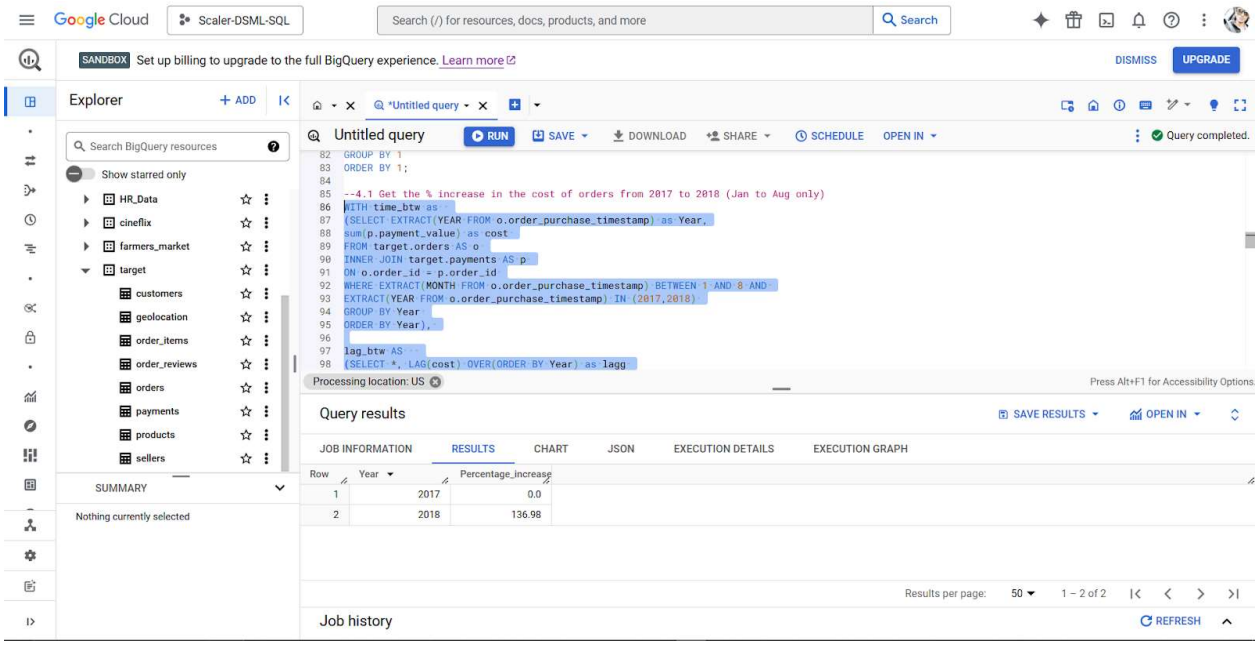
```sql
SELECT
    customer_state,
    COUNT(DISTINCT customer_id) AS total_customers
FROM `target.customers`
GROUP BY 1
ORDER BY 1;
```



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

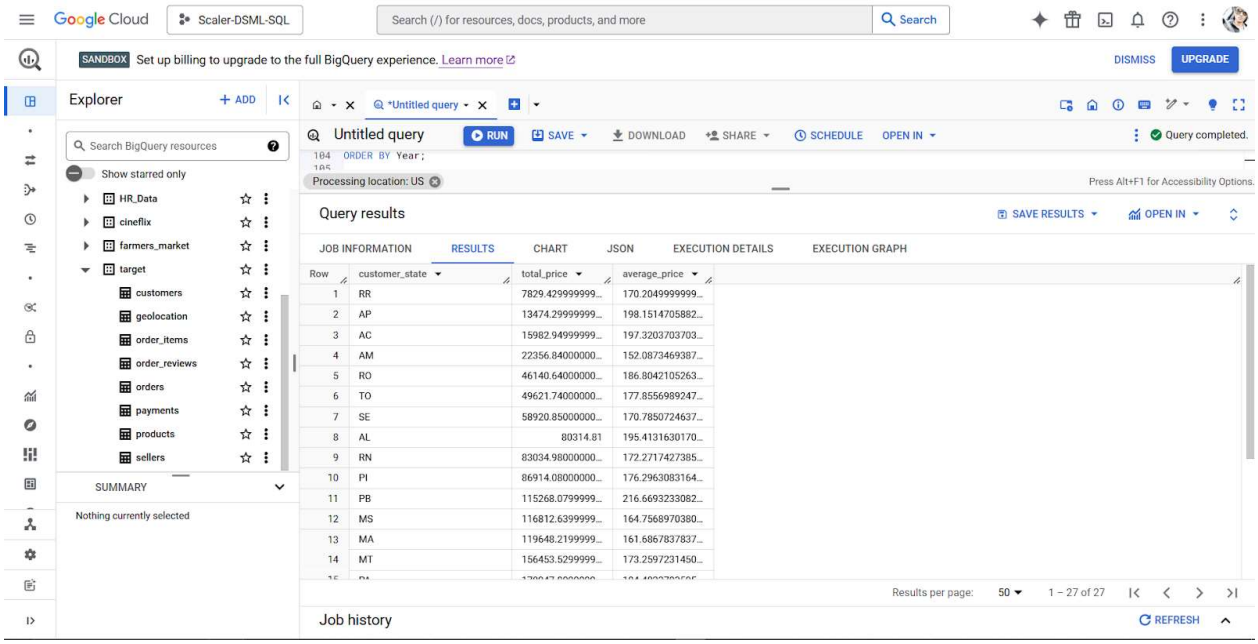
4.1 Get the % increase in the cost of orders from 2017 to 2018 (Jan to Aug only)

```
WITH time_btw as
(SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) as Year,
sum(p.payment_value) as cost
FROM target.orders AS o
INNER JOIN target.payments AS p
ON o.order_id = p.order_id
WHERE EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8 AND
EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017,2018)
GROUP BY Year
ORDER BY Year),
lag_btw AS
(SELECT *, LAG(cost) OVER(ORDER BY Year) as lagg
FROM time_btw)

SELECT Year,
ROUND(ifnull(((cost-lagg)/lagg)*100,0),2) as Percentage_increase
FROM lag_btw
ORDER BY Year;
```



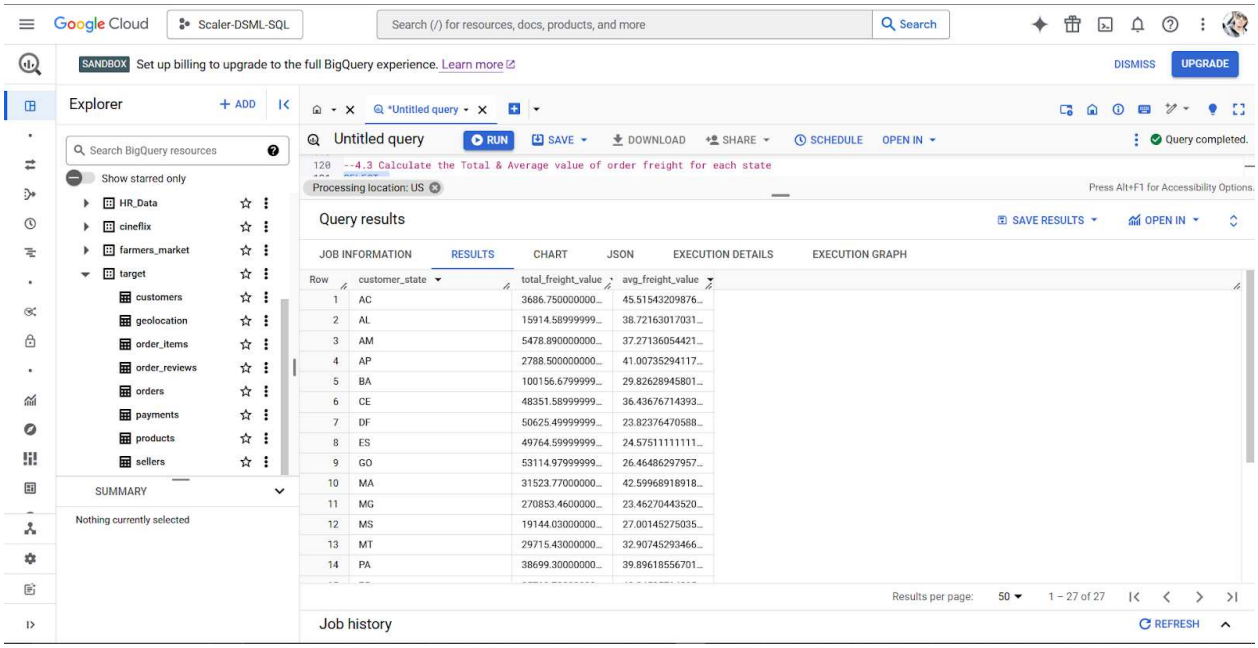
4.2 Calculate the Total & Average value of order price for each state

```sql
select c.customer_state ,
sum(oi.price) as total_price,
SUM(oi.price)/count(distinct o.order_id) as average_price
from `target.order_items` oi
inner join `target.orders` o
on oi.order_id = o.order_id
inner join `target.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by total_price
```



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

```sql
SELECT

    c.customer_state,

    SUM(oi.freight_value) AS total_freight_value,

    SUM(oi.freight_value)/count(distinct o.order_id) AS avg_freight_value

FROM `target.order_items` oi

INNER JOIN `target.orders` o

ON oi.order_id = o.order_id

INNER JOIN `target.customers` c

ON o.customer_id = c.customer_id

GROUP BY 1

ORDER BY 1
```



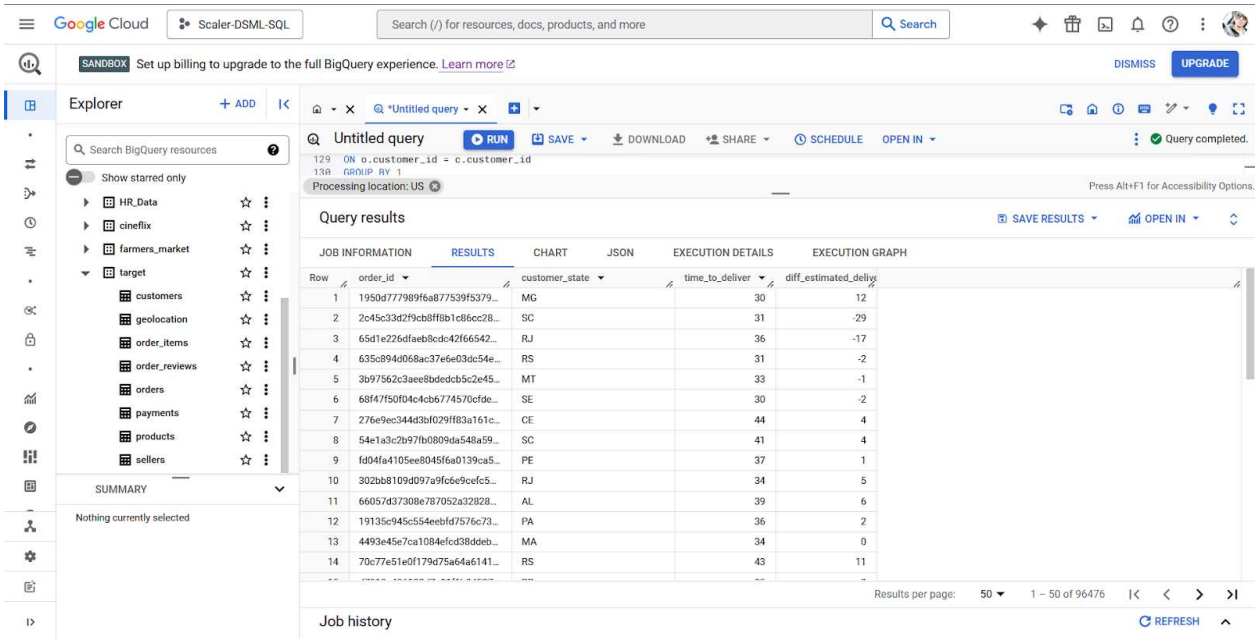
5 Analysis based on sales, freight and delivery time.

5.1 Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

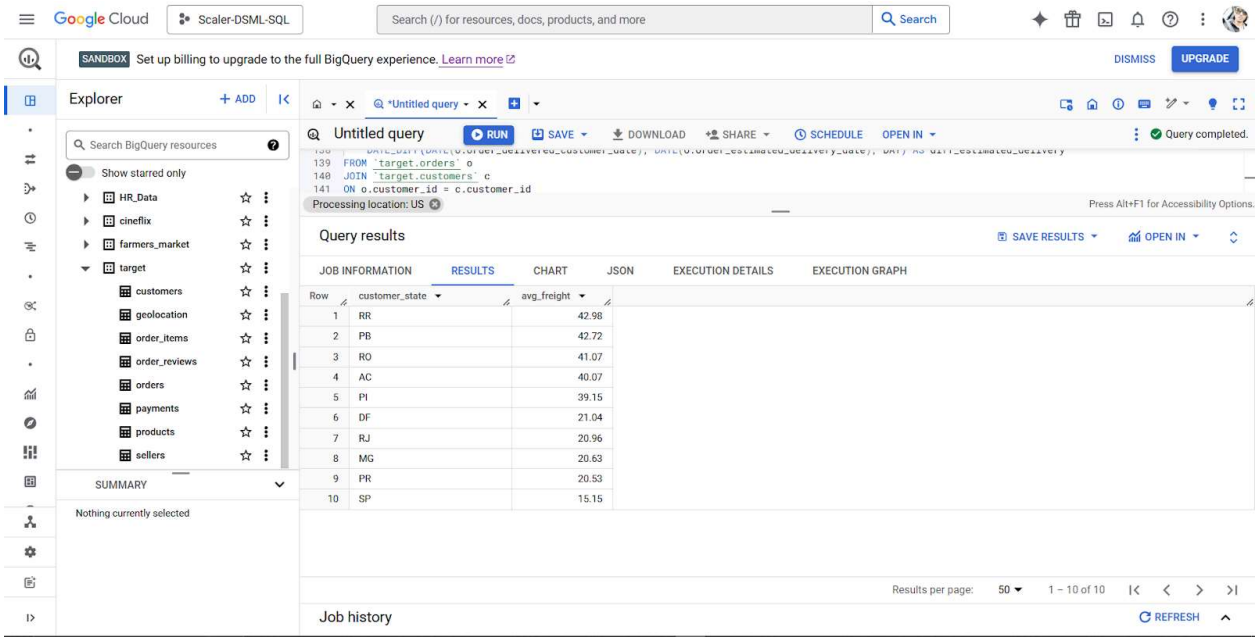
You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
* time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
* diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date

```
SELECT
    o.order_id,
    c.customer_state,
    DATE_DIFF(DATE(o.order_delivered_customer_date), DATE(o.order_purchase_timestamp),
DAY) AS time_to_deliver,
    DATE_DIFF(DATE(o.order_delivered_customer_date),
DATE(o.order_estimated_delivery_date), DAY) AS diff_estimated_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;
```



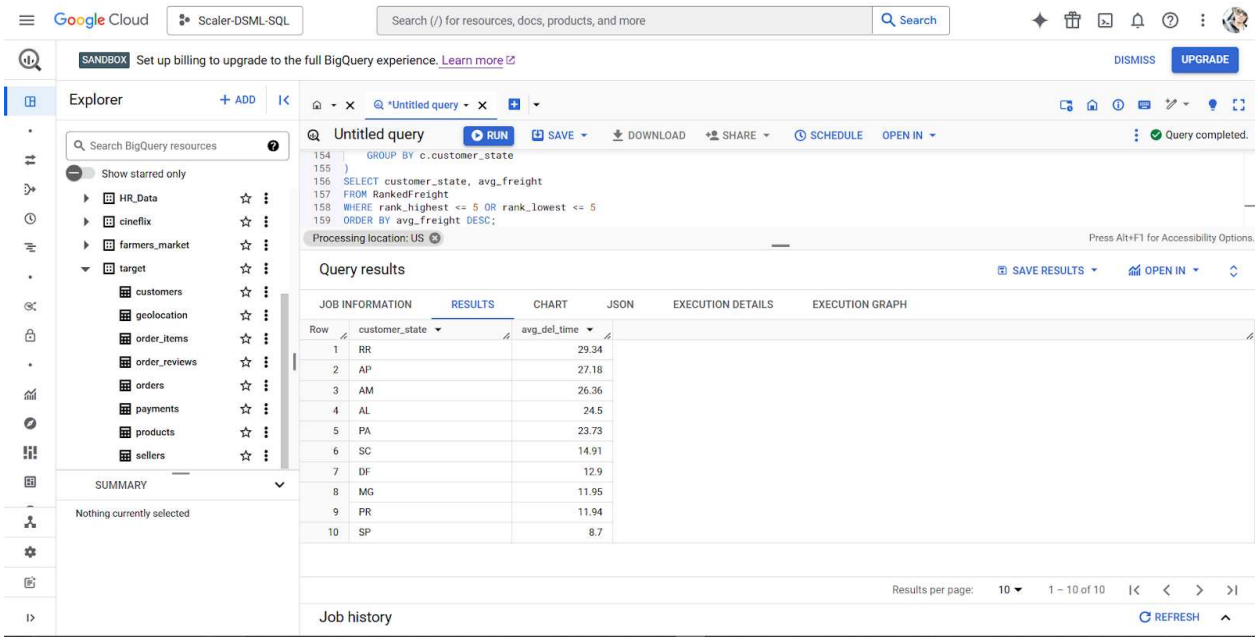
5.2 Find out the top 5 states with the highest & lowest average freight value.

```sql
WITH RankedFreight AS (
    SELECT
        c.customer_state,
        ROUND(AVG(oi.freight_value), 2) AS avg_freight,
        DENSE_RANK() OVER(ORDER BY AVG(oi.freight_value) DESC) AS rank_highest,
        DENSE_RANK() OVER(ORDER BY AVG(oi.freight_value)) AS rank_lowest
    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
)
SELECT customer_state, avg_freight
FROM RankedFreight
WHERE rank_highest <= 5 OR rank_lowest <= 5
ORDER BY avg_freight DESC;
```



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

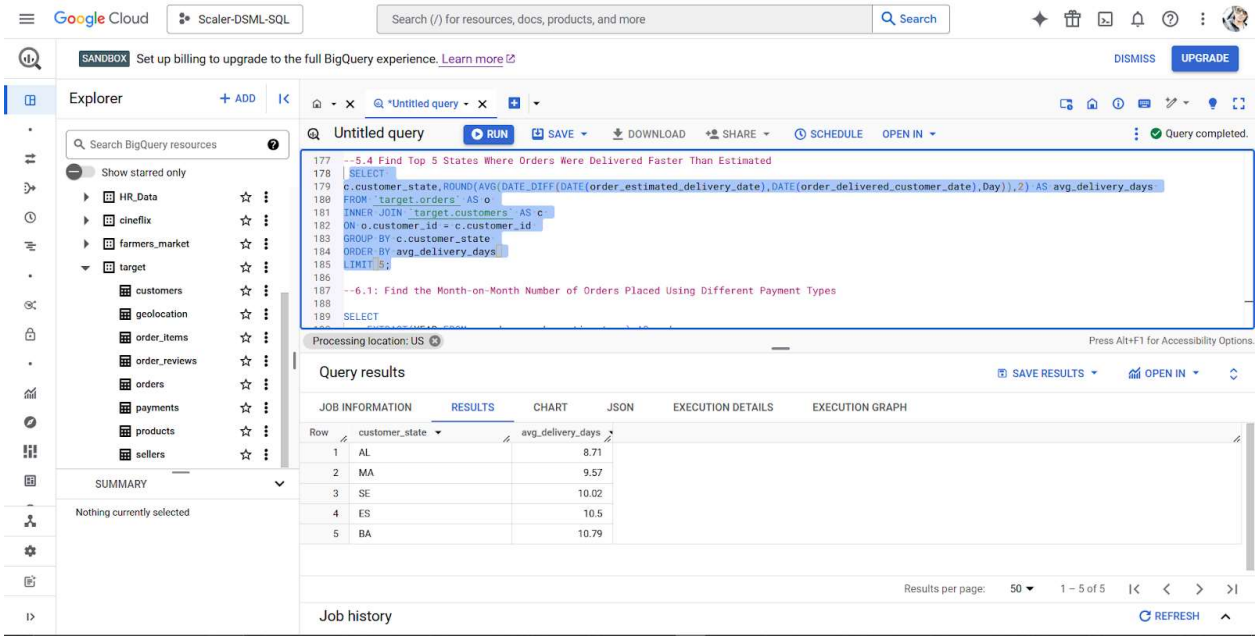
```sql
WITH RankedStates AS (
    SELECT
        c.customer_state,
        ROUND(AVG(DATE_DIFF(DATE(order_delivered_customer_date),
DATE(order_purchase_timestamp), DAY)), 2) AS avg_del_time,
        DENSE_RANK() OVER(ORDER BY AVG(DATE_DIFF(DATE(order_delivered_customer_date),
DATE(order_purchase_timestamp), DAY)) DESC) AS rank_highest,
        DENSE_RANK() OVER(ORDER BY AVG(DATE_DIFF(DATE(order_delivered_customer_date),
DATE(order_purchase_timestamp), DAY))) AS rank_lowest
    FROM target.orders o
    JOIN target.customers c ON o.customer_id = c.customer_id
    GROUP BY c.customer_state
)
SELECT customer_state, avg_del_time
FROM RankedStates
WHERE rank_highest <= 5 OR rank_lowest <= 5
ORDER BY avg_del_time DESC;
```



5.4 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(DATE_DIFF(DATE(order_estimated_delivery_date),DATE(order_de
livered_customer_date),Day)),2) AS avg_delivery_days
FROM `target.orders` AS o
INNER JOIN `target.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY avg_delivery_days
LIMIT 5;
```

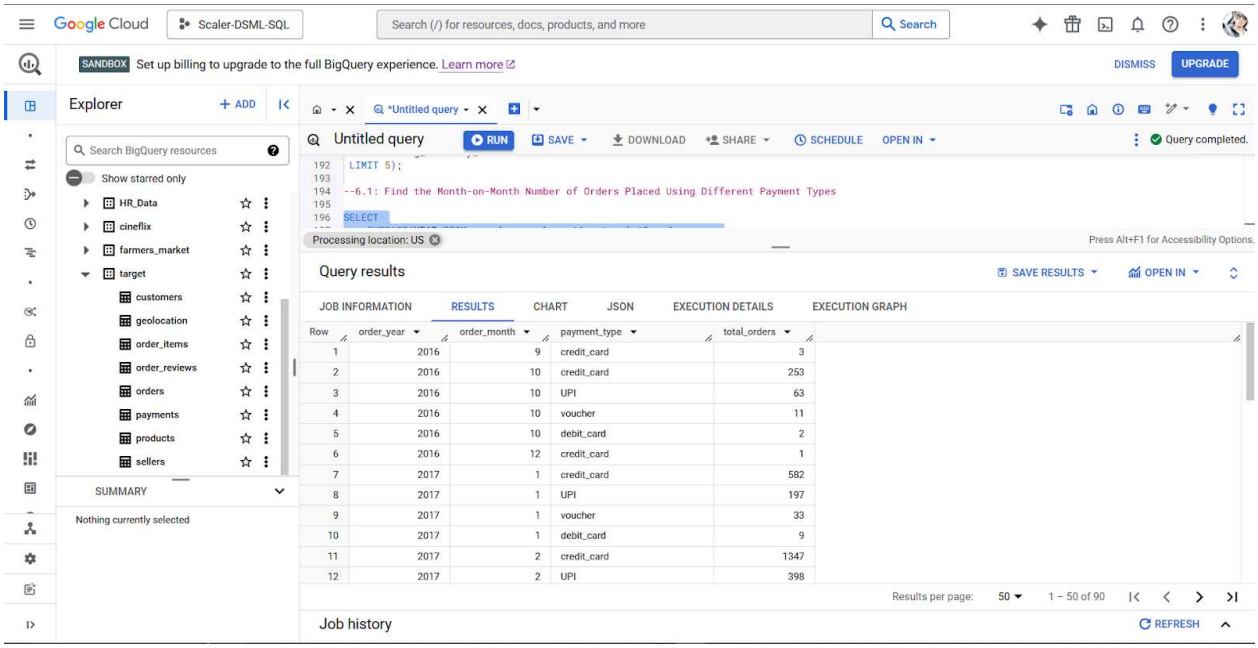


6 Analysis based on the payments:

6.1Find the month on month no. of orders placed using different payment types.

```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
    p.payment_type,
```

```
    COUNT(DISTINCT o.order_id) AS total_orders
FROM `target.orders` o
JOIN `target.payments` p ON o.order_id = p.order_id
GROUP BY order_year, order_month, p.payment_type
ORDER BY order_year, order_month, total_orders DESC;
```



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

```
SELECT
  payment_installments AS installments,
COUNT(order_id) AS num_orders,
FROM `target.payments`
WHERE payment_installments >= 1
GROUP BY payment_installments
ORDER BY num_orders DESC
```