Business Case: Target SQL

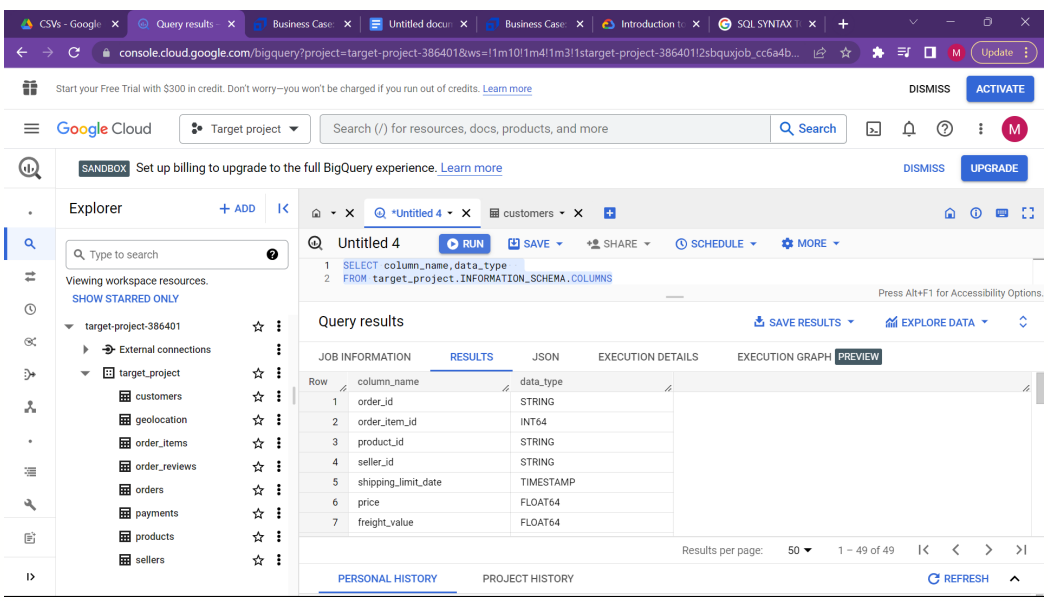
Target is one of the world's most recognized brands and one of America's leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

 # 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 :

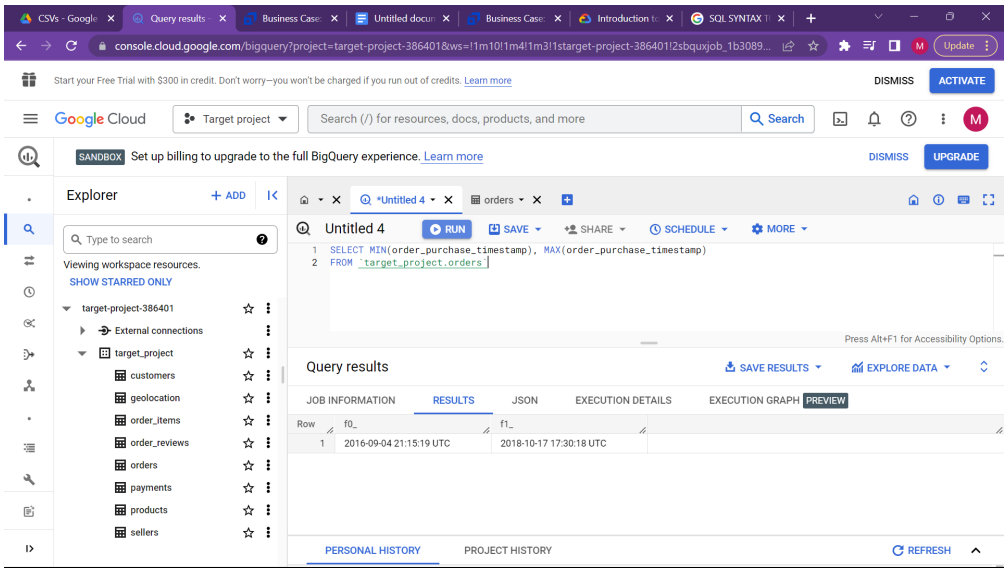
```
SELECT column_name,data_type
FROM target_project.INFORMATION_SCHEMA.COLUMNS
```



INSIGHTS : shows all column names and data types.

2) Time period for which the data is given :

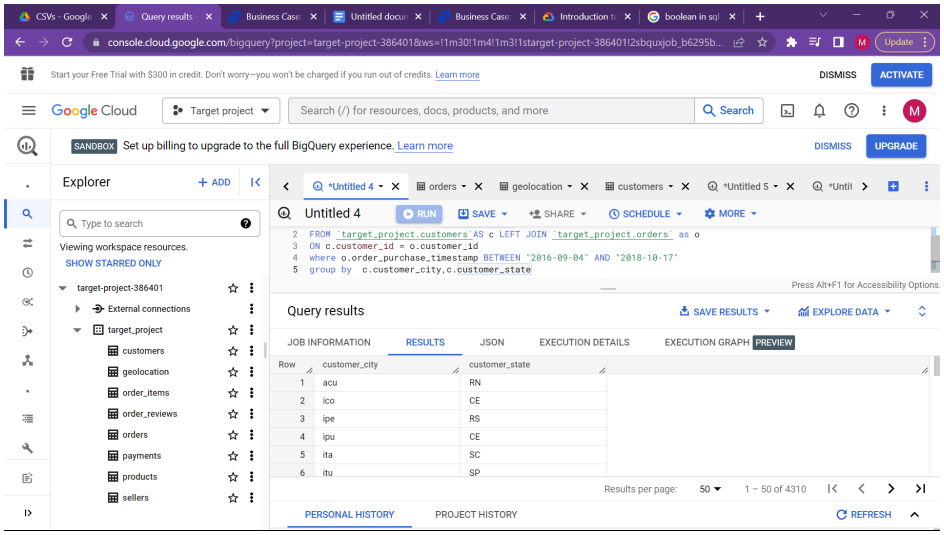
```
SELECT MIN(order_purchase_timestamp), MAX(order_purchase_timestamp)
FROM `target_project.orders`
```



INSIGHTS : Time frame for the given dataset is between 04/09/2016 - 17/10/2018

3) Cities and States of customers ordered during the given period:

```
select c.customer_city, c.customer_state
FROM `target_project.customers`AS c LEFT JOIN `target_project.orders` as o
ON c.customer_id = o.customer_id
where o.order_purchase_timestamp BETWEEN "2016-09-04" AND "2018-10-17"
group by  c.customer_city,c.customer_state
```

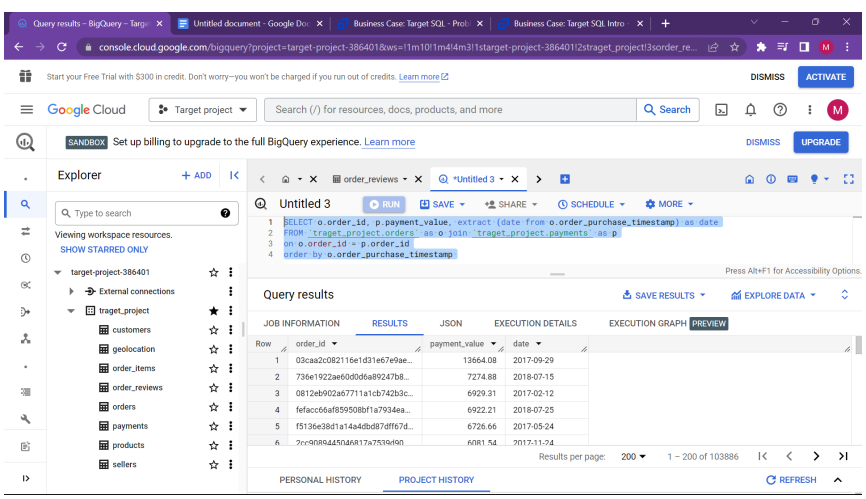


INSIGHTS : Represents city and state names from where customers have ordered.

 # 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?**

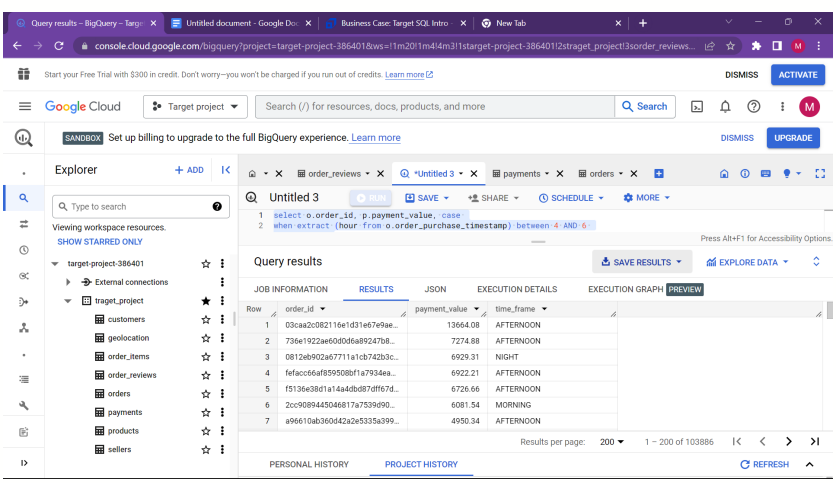
```
SELECT o.order_id, p.payment_value, extract (date from
o.order_purchase_timestamp) as date
FROM `traget_project.orders` as o join `traget_project.payments` as p
on o.order_id = p.order_id
order by date
```



INSIGHTS : Growing trend on ecommerce in Brazil was normal. Between August to November for both 2017 and 2018 the payment value was huge when compared to other months.

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

```
select o.order_id, p.payment_value, case
when extract (hour from o.order_purchase_timestamp) between 4 AND 6
THEN "DAWN"
when extract (hour from o.order_purchase_timestamp) between 6 AND 12
THEN "MORNING"
when extract (hour from o.order_purchase_timestamp) between 12 AND 18
THEN "AFTERNOON"
else "NIGHT"
END AS time_frame
from `traget_project.orders` as o join `traget_project.payments`as p
on o.order_id =p.order_id
order by p.payment_value desc
```

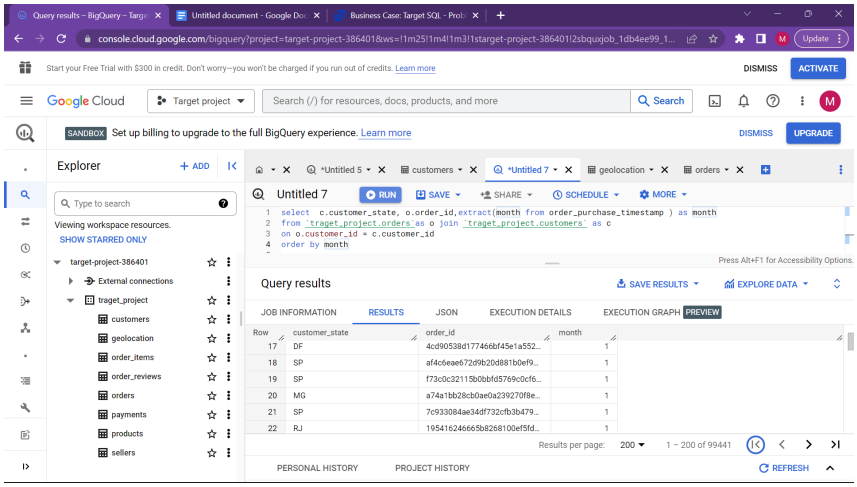


INSIGHTS : From the results, most of the high value payments happened during the “AFTERNOON” time frame.

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

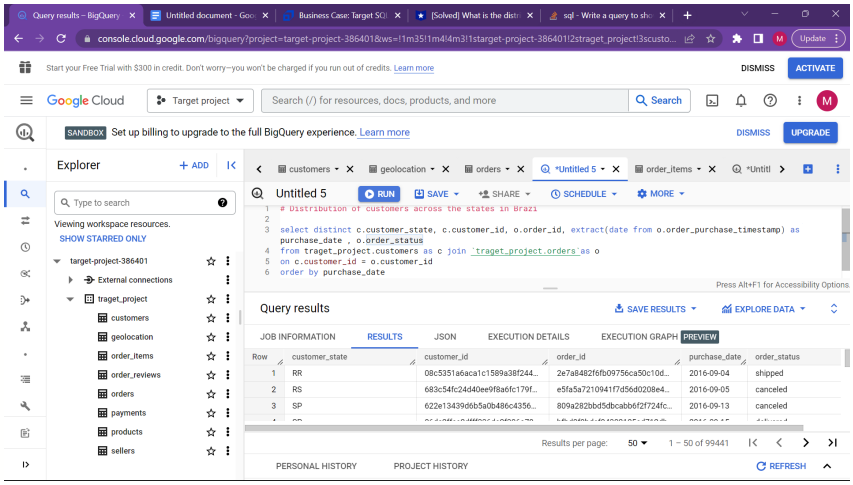
1) **Get month on month orders by states :**

```
select  c.customer_state, o.order_id,extract(month from order_purchase_timestamp ) as
month
from `traget_project.orders`as o join `traget_project.customers` as c
on o.customer_id = c.customer_id
order by month
```



2) Distribution of customers across the states in Brazil:

```
select distinct c.customer_state, c.customer_id, o.order_id, extract(date from
o.order_purchase_timestamp) as purchase_date , o.order_status
from traget_project.customers as c join `traget_project.orders`as o
on c.customer_id = o.customer_id
order by purchase_date
```



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

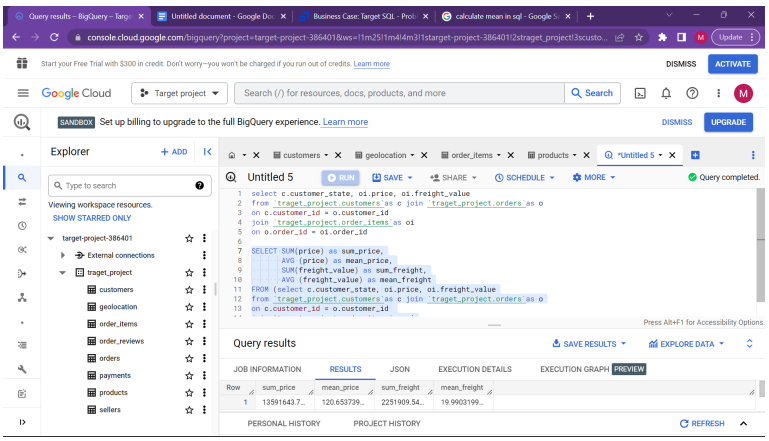
1) **Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table**

```
select (payment_value - previous)/ previous * 100 as percent_change
from (select extract (date from o.order_purchase_timestamp) as date_year, o.order_id,
p.payment_value, lag (p.payment_value) over (partition by o.order_id order by
order_purchase_timestamp ) as previous
from `traget_project.orders`as o join `traget_project.payments`as p
on o.order_id = p.order_id
where o.order_purchase_timestamp between  "2016-12-31" and "2017-09-01"
order by date_year)
```

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

```
SELECT SUM(price) as sum_price,
AVG (price) as mean_price,
SUM(freight_value) as sum_freight,
AVG (freight_value) as mean_freight
```

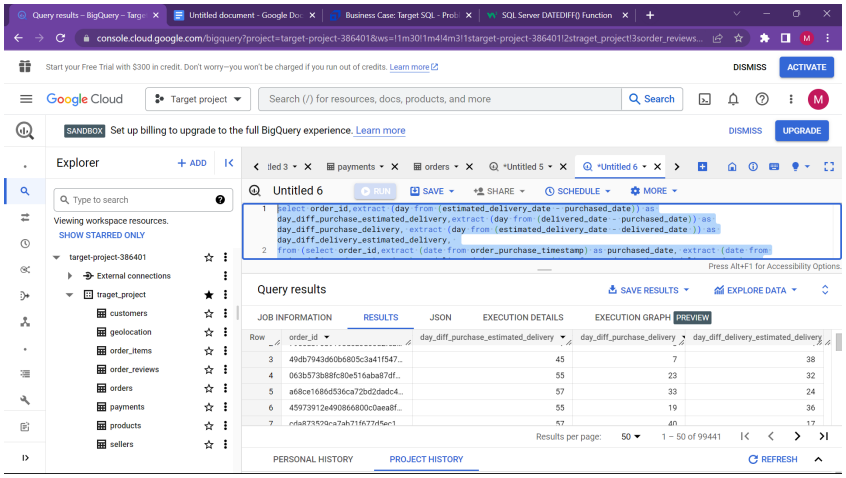
```
FROM (select c.customer_state, oi.price, oi.freight_value
from `traget_project.customers`as c join `traget_project.orders`as o
on c.customer_id = o.customer_id
join `traget_project.order_items`as oi
on o.order_id = oi.order_id)
```



 # Analysis on sales, freight and delivery time

1) Calculate days between purchasing, delivering and estimated delivery

```
select order_id,extract (day from (estimated_delivery_date - purchased_date))
as day_diff_purchase_estimated_delivery,extract (day from (delivered_date -
purchased_date)) as day_diff_purchase_delivery, extract (day from
(estimated_delivery_date - delivered_date )) as
day_diff_delivery_estimated_delivery,
from (select order_id,extract (date from order_purchase_timestamp) as
purchased_date, extract (date from order_delivered_carrier_date) as
delivered_date,  extract (date from order_estimated_delivery_date) as
estimated_delivery_date
from `traget_project.orders`)
```

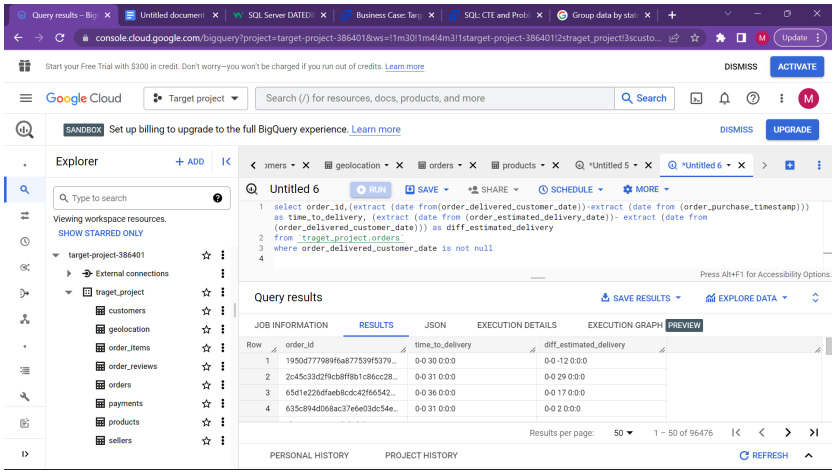


Insights : This results shows the day's difference between purchased date, delivered date and estimated delivery date.

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 order_id,(extract (date from(order_delivered_customer_date))-extract
(date from (order_purchase_timestamp))) as time_to_delivery, (extract (date from
(order_estimated_delivery_date))- extract (date from (order_delivered_customer_date)))
as diff_estimated_delivery
from `traget_project.orders`
where order_delivered_customer_date is not null
```



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

```
select avg(count(freight_value) as Avg_freight_value, avg(time_to_delivery) as
Avg_ttd, avg (diff_estimated_delivery) as avg_ded
from(select oi.freight_value,(count(o.order_delivered_customer_date)-count
(o.order_purchase_timestamp)) as time_to_delivery,
(count(o.order_estimated_delivery_date)- count
(o.order_delivered_customer_date)) as diff_estimated_delivery
from `traget_project.order_items` as oi join `traget_project.orders`as o
on oi.order_id = o.order_id
join `traget_project.customers`as c
on o.customer_id = c.customer_id)
group by time_to_delivery,  c.customer_state, oi.freight_value
```

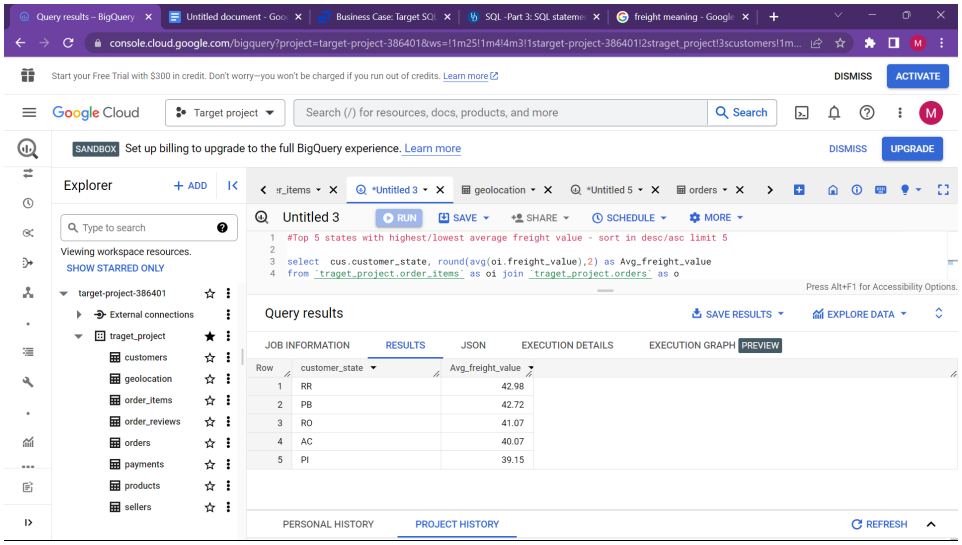
4) Sort the data to get the following:

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

<u>Highest Freight value :</u>

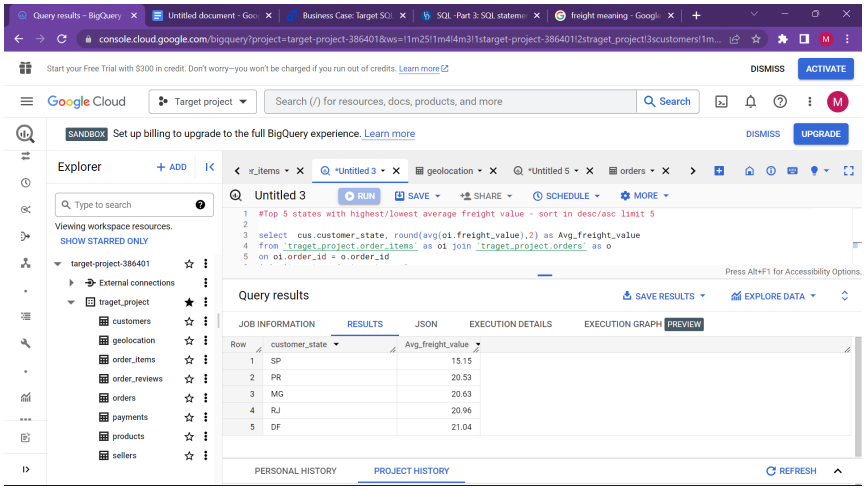
```
select  cus.customer_state, round(avg(oi.freight_value),2) as Avg_freight_value
from `traget_project.order_items` as oi join `traget_project.orders` as o
on oi.order_id = o.order_id
```

```
join `traget_project.customers` as cus
on o.customer_id = cus.customer_id
group by cus.customer_state
order by avg(oi.freight_value) desc
limit 5
```



Lowest Freight value :

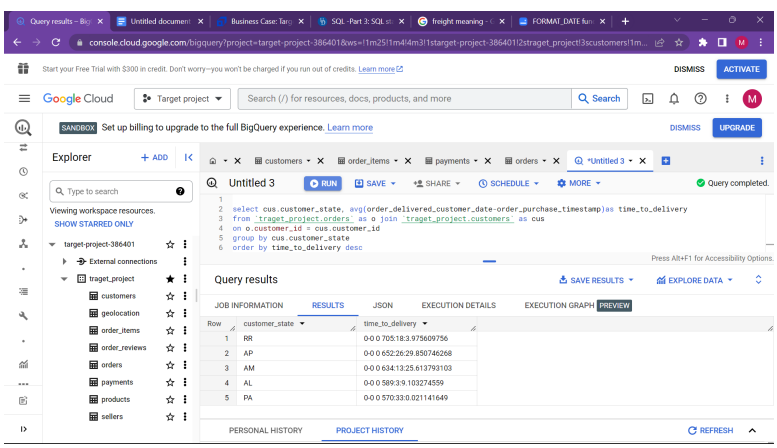
```
select  cus.customer_state, round(avg(oi.freight_value),2) as Avg_freight_value
from `traget_project.order_items` as oi join `traget_project.orders` as o
on oi.order_id = o.order_id
join `traget_project.customers` as cus
on o.customer_id = cus.customer_id
group by cus.customer_state
order by avg(oi.freight_value) asc
limit 5
```



- Top 5 states with highest/lowest average time to delivery

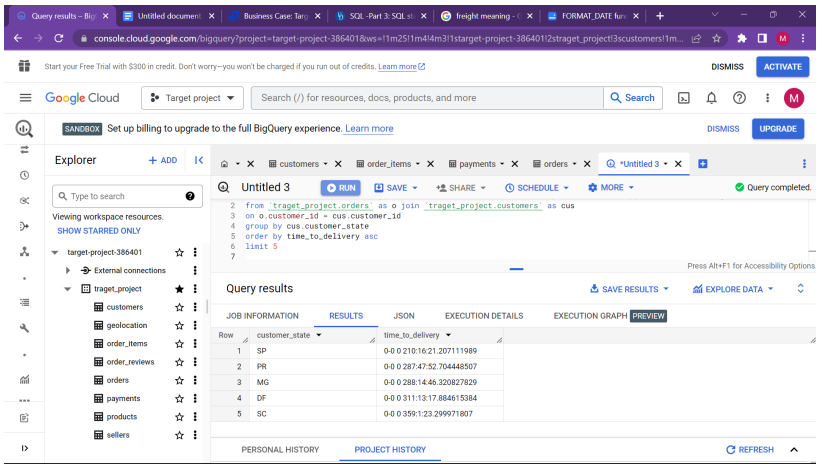
Highest average time to delivery:

```
select cus.customer_state,
avg(order_delivered_customer_date-order_purchase_timestamp)as time_to_delivery
from `traget_project.orders` as o join `traget_project.customers` as cus
on o.customer_id = cus.customer_id
group by cus.customer_state
order by time_to_delivery desc
limit 5
```



Lowest average time to delivery:

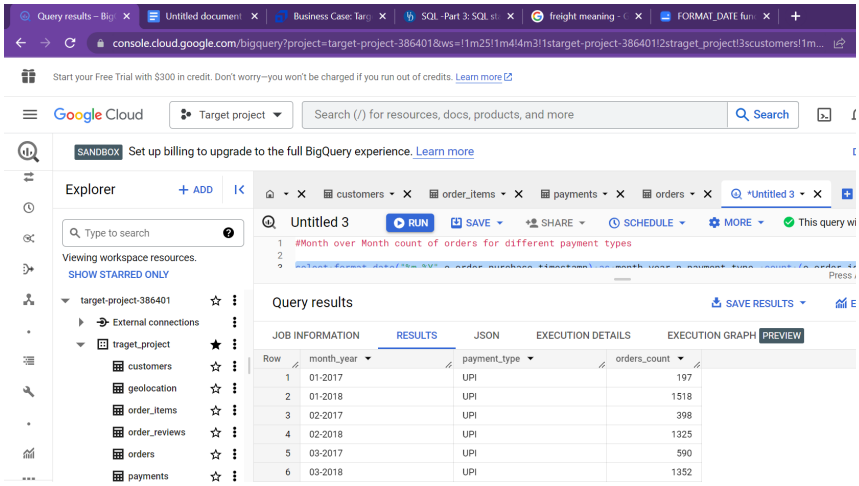
```
select cus.customer_state,
avg(order_delivered_customer_date-order_purchase_timestamp)as time_to_delivery
from `traget_project.orders` as o join `traget_project.customers` as cus
on o.customer_id = cus.customer_id
group by cus.customer_state
order by time_to_delivery asc
limit 5
```



 # Payment type analysis:

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

```
select format_date("%m-%Y",o.order_purchase_timestamp) as month_year,p.payment_type,
count (o.order_id) as orders_count,
from `traget_project.orders` as o join `traget_project.payments` as p
on o.order_id = p.order_id
group by p.payment_type, month_year
order by p.payment_type, month_year,orders_count
```



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

```
select format_date("%m-%Y",o.order_purchase_timestamp) as
month_year,p.payment_installments, count (o.order_id) as orders_count,
from `traget_project.orders` as o join `traget_project.payments` as p
on o.order_id = p.order_id
group by p.payment_installments, month_year
order by p.payment_installments, month_year,orders_count
```