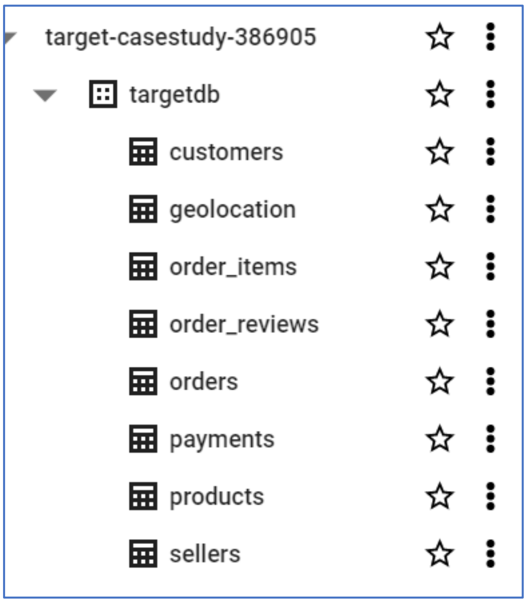
Business Case: Target SQL

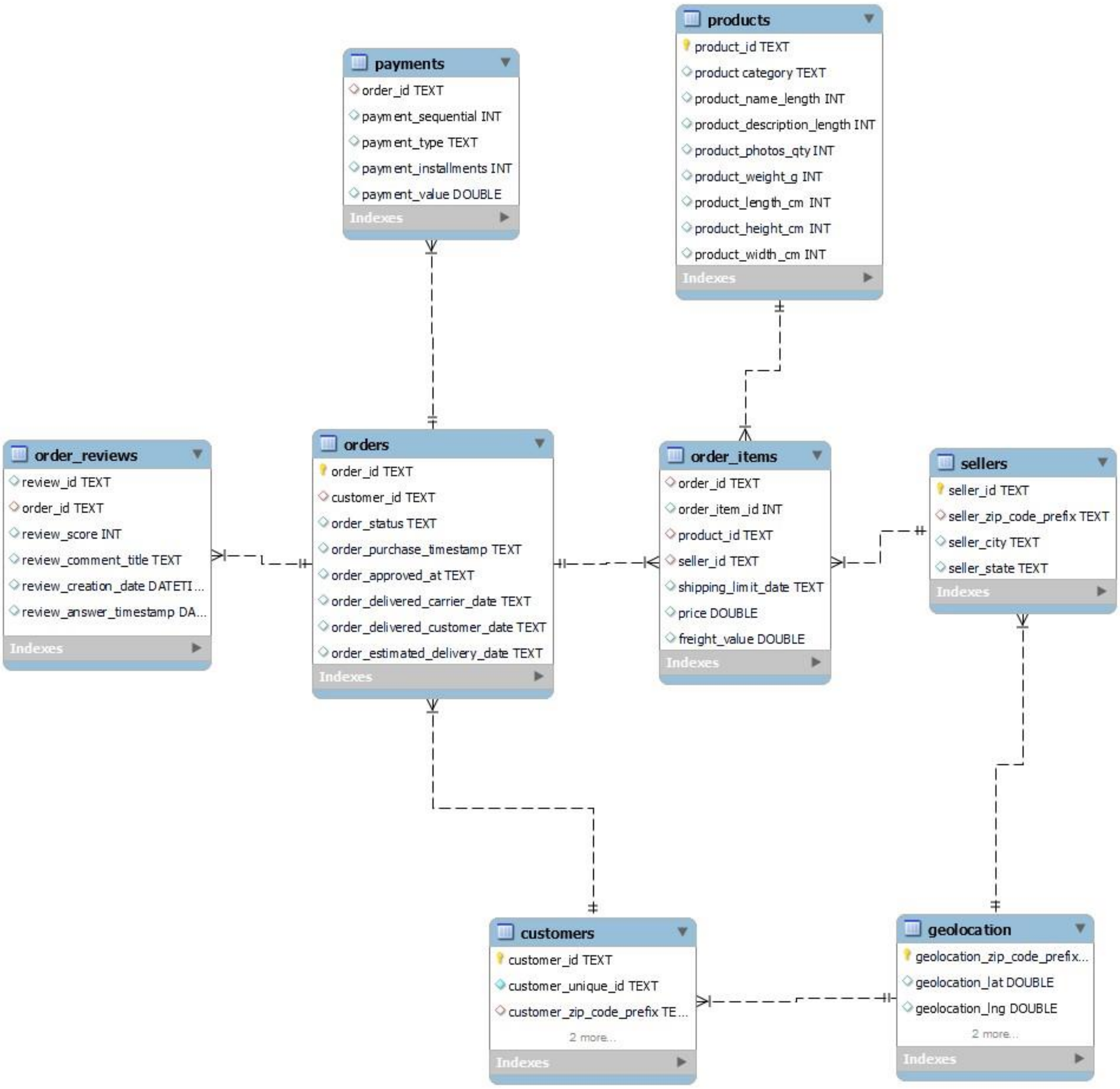
1. Importing the dataset into Big Query



target-casestudy-386905
targetdb
customers
geolocation
order_items
order_reviews
orders
payments
products
sellers

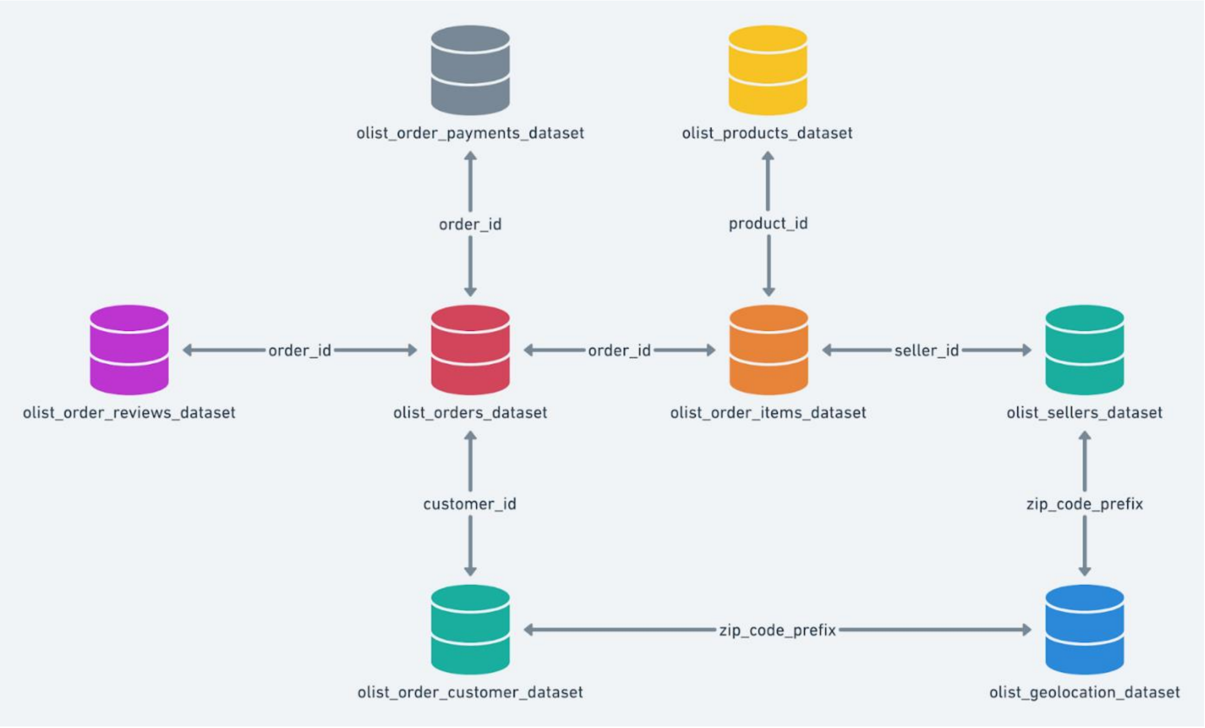
2. Exploratory analysis of the data

- In exploratory analysis of the data , I have analysed the ER diagram in below picture , along with it , studied the schema of each of the 8 tables in BigQuery



ER diagram

PS - geolocation_zip_code_prefix is shown as PK but its not , I have kept it only so that customers and seller table can be connected to geolocation on this key.



- Identified the primary key in each of the table.
- Also edited the schema for each table in Big Query to add the description.
- Below are details and structure of each table.

1. Data type of columns in a table

Customers table

fullname	mode	type	description
customer_id	NULLABLE	STRING, **PRIMARY KEY**	Id of the consumer who made the purchase
customer_unique_id	NULLABLE	STRING, **NOT NULL**	Unique Id of the consumer
customer_zip_code_prefix	NULLABLE	INTEGER	Zip Code of the location of the consumer
customer_city	NULLABLE	STRING	Name of the City from where order is made
customer_state	NULLABLE	STRING	State Code from where order is made(Ex- sao paulo-SP)

Sellers table

fullname	mode	type	description
seller_id	NULLABLE	STRING, **PRIMARY KEY**	Unique Id of the seller registered
seller_zip_code_prefix	NULLABLE	INTEGER	Zip Code of the location of the seller
seller_city	NULLABLE	STRING	Name of the City of the seller
seller_state	NULLABLE	STRING	State Code (Ex- sao paulo-SP)

Geolocations table

fullname	mode	type	description
geolocation_zip_code_prefix	NULLABLE	INTEGER	first 5 digits of zip code
geolocation_lat	NULLABLE	FLOAT	latitude
geolocation_lng	NULLABLE	FLOAT	longitude
geolocation_city	NULLABLE	STRING	city name
geolocation_state	NULLABLE	STRING	state

order_items table

fullname	mode	type	description
order_id	NULLABLE	STRING **FOREIGN KEY**	A unique id of order made by the consumers
order_item_id	NULLABLE	INTEGER	A Unique id given to each item ordered in the order
product_id	NULLABLE	STRING, **FOREIGN KEY**	A unique id given to each product available on the site
seller_id	NULLABLE	STRING	Unique Id of the seller registered in Target
shipping_limit_date	NULLABLE	TIMESTAMP	The date before which shipping of the ordered product must be completed
price	NULLABLE	FLOAT	Actual price of the products ordered

freight_value	NULLABLE	FLOAT	Price rate at which a product is delivered from one point to another

Payments table

fullname	mode	type	description
order_id	NULLABLE	STRING	A unique id of order made by the consumers
payment_sequential	NULLABLE	INTEGER	sequences of the payments made in case of EMI
payment_type	NULLABLE	STRING	mode of payment used.(Ex-Credit Card)
payment_installments	NULLABLE	INTEGER	number of installments in case of EMI purchase
payment_value	NULLABLE	FLOAT	Total amount paid for the purchase order

Orders table

This table got an extra column **order_approved_at** (apart from what mentioned in case study)

fullname	mode	type	description
order_id	NULLABLE	STRING, **PRIMARY KEY**	A unique id of order made by the consumers
customer_id	NULLABLE	STRING, **FOREIGN KEY**	Id of the consumer who made the purchase
order_status	NULLABLE	STRING	status of the order made i.e delivered, shipped etc
order_purchase_timestamp	NULLABLE	TIMESTAMP	Timestamp of the purchase

order_approved_at	NULLABLE	TIMESTAMP	
order_delivered_carrier_date	NULLABLE	TIMESTAMP	delivery date at which carrier made the delivery
order_delivered_customer_date	NULLABLE	TIMESTAMP	date at which customer got the product
order_estimated_delivery_date	NULLABLE	TIMESTAMP	estimated delivery date of the products

Reviews Table

This table has got one less column **review_comment_message** (apart from what mentioned in case study)

fullname	mode	type	description
review_id	NULLABLE	STRING	Id of the review given on the product ordered by the order id
order_id	NULLABLE	STRING.	A unique id of order made by the consumers
review_score	NULLABLE	INTEGER	review score given by the customer for each order on the scale of 1–5
review_comment_title	NULLABLE	STRING	Title of the review
review_creation_date	NULLABLE	TIMESTAMP	Timestamp of the review when it is created
review_answer_timestamp	NULLABLE	TIMESTAMP	Timestamp of the review answered

Products Table

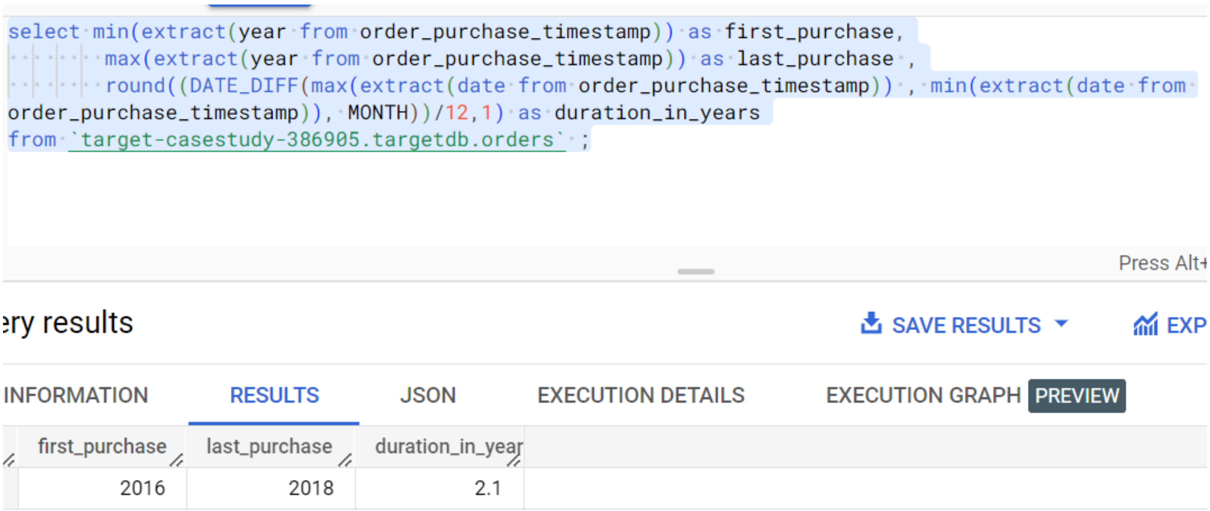
fullname	mode	type	description
product_id	NULLABLE	STRING, **PRIMARY KEY**	A unique identifier for the proposed project
product_category	NULLABLE	STRING	Name of the product category
product_name_length	NULLABLE	INTEGER	length of the string which specifies the name given to the products ordered
product_description_length	NULLABLE	INTEGER	length of the description written for each product ordered on the site

product_photos_qty	NULLABLE	INTEGER	Number of photos of each product ordered available on the shopping portal
product_weight_g	NULLABLE	INTEGER	Weight of the products ordered in grams
product_length_cm	NULLABLE	INTEGER	Length of the products ordered in centimeters
product_height_cm	NULLABLE	INTEGER	Height of the products ordered in centimeters
product_width_cm	NULLABLE	INTEGER	width of the product ordered in centimeters

2. Time period for which the data is given.

```
## Time period for which the data is given.
select min(extract(year from order_purchase_timestamp)) as first_purchase,
        max(extract(year from order_purchase_timestamp)) as last_purchase ,
        round((DATE_DIFF(max(extract(date from order_purchase_timestamp)) ,
min(extract(date from order_purchase_timestamp)), MONTH))/12,1) as
duration_in_years
from `target-casestudy-386905.targetdb.orders` ;
```

Duration is **2.1** years **(25 months)**, this can be seen from the below query

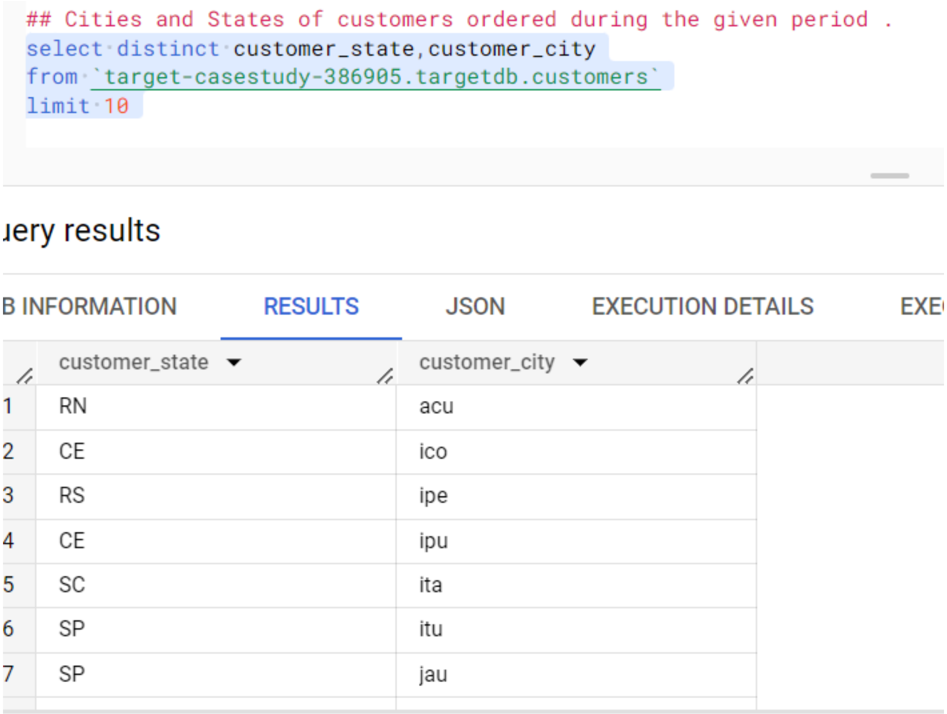


```
select min(extract(year from order_purchase_timestamp)) as first_purchase,
      max(extract(year from order_purchase_timestamp)) as last_purchase ,
      round((DATE_DIFF(max(extract(date from order_purchase_timestamp)) , min(extract(date from
order_purchase_timestamp)), MONTH))/12,1) as duration_in_years
from `target-casestudy-386905.targetdb.orders` ;
```

first_purchase	last_purchase	duration_in_year
2016	2018	2.1

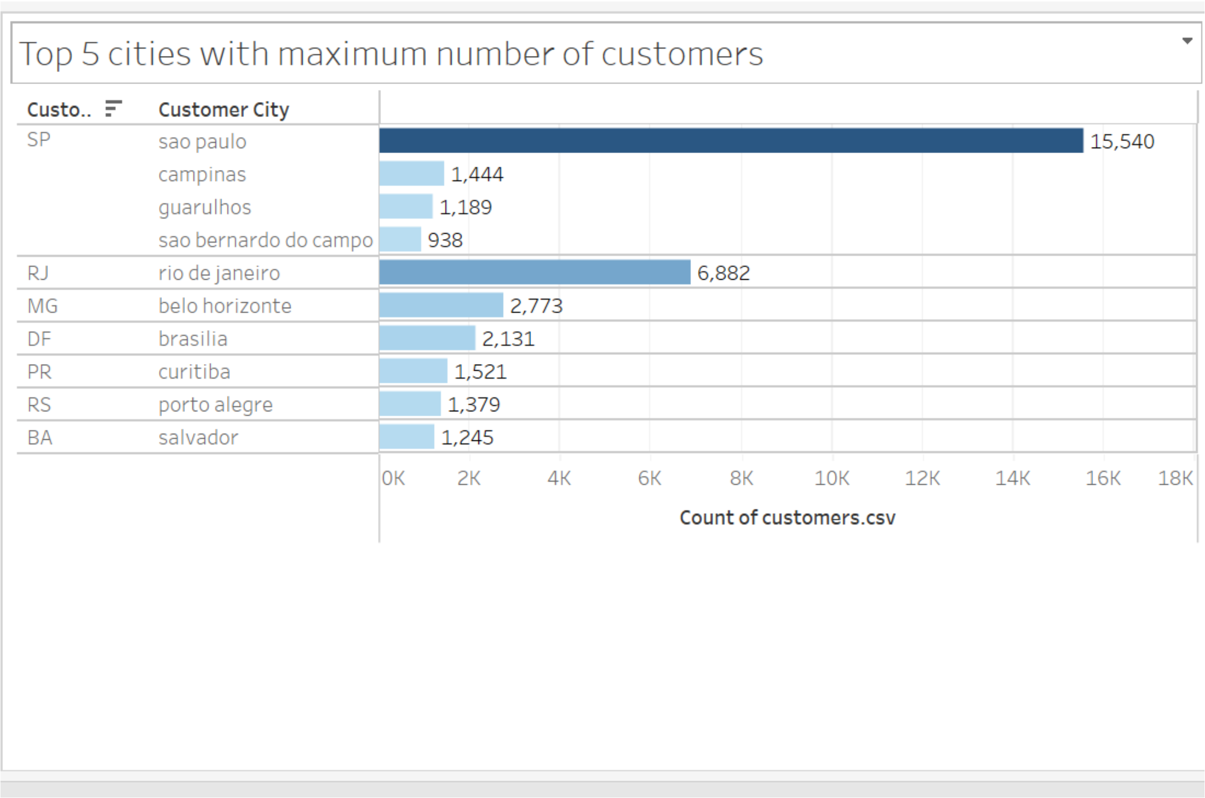
3. Cities and States of customers ordered during the given period .

```
## Cities and States of customers ordered during the given period .
select distinct customer_state,customer_city
from `target-casestudy-386905.targetdb.customers`
```



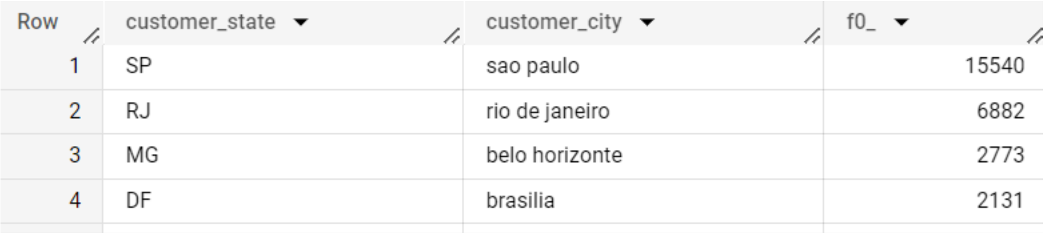
```
## Cities and States of customers ordered during the given period .
select distinct customer_state,customer_city
from `target-casestudy-386905.targetdb.customers`
limit 10
```

	customer_state	customer_city
1	RN	acu
2	CE	ico
3	RS	ipe
4	CE	ipu
5	SC	ita
6	SP	itu
7	SP	jau



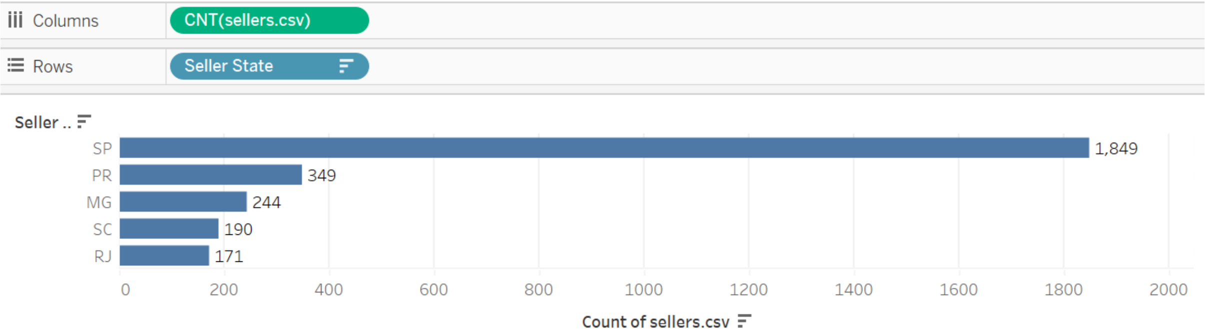
Highest number of customers are from Sao Paulo

```
select customer_state,customer_city,count(customer_id)
from `target-casestudy-386905.targetdb.customers`
group by 1,2
order by 3 desc;
```



Row	customer_state	customer_city	f0_
1	SP	sao paulo	15540
2	RJ	rio de janeiro	6882
3	MG	belo horizonte	2773
4	DF	brasilia	2131

Top 5 States with the greatest number of sellers



Actionable Insights:

- Since number of customers (1521) in PR state are lesser even though we have enough # of sellers for that state (349),
- Similarly, # of seller (171) in RJ state should be increased to suffice the demand of 6882 customers.

Recommendations:

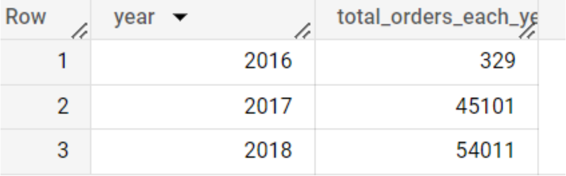
- Customers in PR state should be approached with additional offers.

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?

   Since the number of orders are increasing each year, we can see a growing trend each year from 2016 to 2018

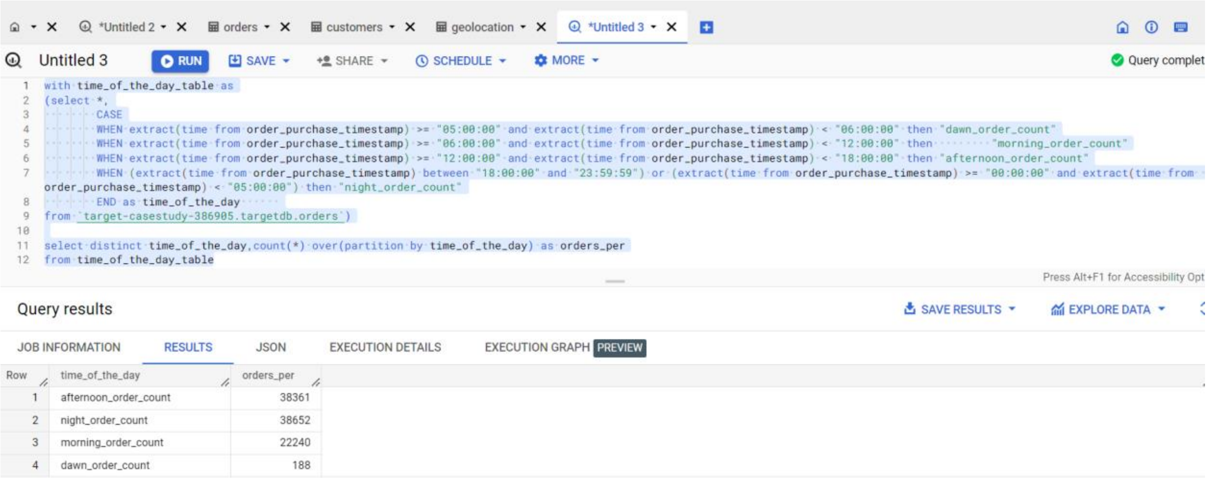
```
## Is there a growing trend on e-commerce in Brazil? How can we describe a complete
scenario? Can we see some seasonality with peaks at specific months?

select distinct extract(year from order_purchase_timestamp) as year,
        count(order_id) over(partition by extract(year from
order_purchase_timestamp)) as total_orders_each_year from `target-casestudy-
386905.targetdb.orders`
order by 1;
```



Row	year	total_orders_each_ye
1	2016	329
2	2017	45101
3	2018	54011

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

Count of orders each time of the day

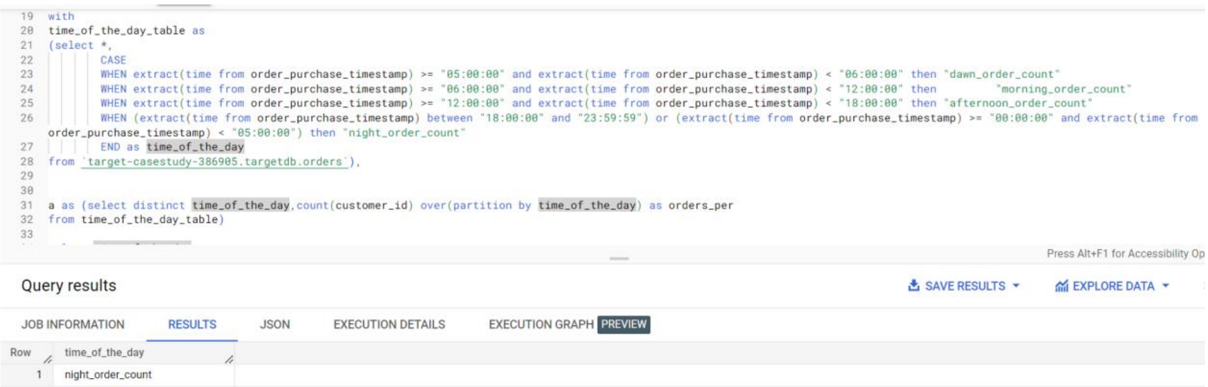


```
with time_of_the_day_table as
(select *,
      CASE
      WHEN extract(time from order_purchase_timestamp) >= "05:00:00" and extract(time from order_purchase_timestamp) < "06:00:00" then "dawn_order_count"
      WHEN extract(time from order_purchase_timestamp) >= "06:00:00" and extract(time from order_purchase_timestamp) < "12:00:00" then         "morning_order_count"
      WHEN extract(time from order_purchase_timestamp) >= "12:00:00" and extract(time from order_purchase_timestamp) < "18:00:00" then "afternoon_order_count"
      WHEN (extract(time from order_purchase_timestamp) between "18:00:00" and "23:59:59") or (extract(time from order_purchase_timestamp) >= "00:00:00" and extract(time from 
order_purchase_timestamp) < "05:00:00") then "night_order_count"
      END as time_of_the_day
from `target-casestudy-386905.targetdb.orders`)

select distinct time_of_the_day,count(*) over(partition by time_of_the_day) as orders_per
from time_of_the_day_table
```

Row	time_of_the_day	orders_per
1	afternoon_order_count	38361
2	night_order_count	38652
3	morning_order_count	22240
4	dawn_order_count	188

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

with
time_of_the_day_table as
(select *,
        CASE
        WHEN extract(time from order_purchase_timestamp) >= "05:00:00" and
extract(time from order_purchase_timestamp) < "06:00:00" then "dawn_order_count"
        WHEN extract(time from order_purchase_timestamp) >= "06:00:00" and
extract(time from order_purchase_timestamp) < "12:00:00"
then             "morning_order_count"
        WHEN extract(time from order_purchase_timestamp) >= "12:00:00" and
extract(time from order_purchase_timestamp) < "18:00:00" then
"afternoon_order_count"
```

```
        WHEN (extract(time from order_purchase_timestamp) between "18:00:00" and
"23:59:59") or (extract(time from order_purchase_timestamp) >= "00:00:00" and
extract(time from   order_purchase_timestamp) < "05:00:00") then
"night_order_count"
        END as time_of_the_day
from `target-casestudy-386905.targetdb.orders`),


a as (select distinct time_of_the_day,count(customer_id) over(partition by
time_of_the_day) as orders_per
from time_of_the_day_table)

select time_of_the_day
from a
where orders_per = (select max(orders_per) from a);
```



```
with
time_of_the_day_table as
(select *,
        CASE
        WHEN extract(time from order_purchase_timestamp) >= "05:00:00" and extract(time from order_purchase_timestamp) < "06:00:00" then "dawn_order_count"
        WHEN extract(time from order_purchase_timestamp) >= "06:00:00" and extract(time from order_purchase_timestamp) < "12:00:00" then         "morning_order_count"
        WHEN extract(time from order_purchase_timestamp) >= "12:00:00" and extract(time from order_purchase_timestamp) < "18:00:00" then "afternoon_order_count"
        WHEN (extract(time from order_purchase_timestamp) between "18:00:00" and "23:59:59") or (extract(time from order_purchase_timestamp) >= "00:00:00" and extract(time from
order_purchase_timestamp) < "05:00:00") then "night_order_count"
        END as time_of_the_day
from `target-casestudy-386905.targetdb.orders`),


a as (select distinct time_of_the_day,count(customer_id) over(partition by time_of_the_day) as orders_per
from time_of_the_day_table)

```

Row	time_of_the_day
1	night_order_count

Most of the orders per placed during Night.

Actionable Insights:

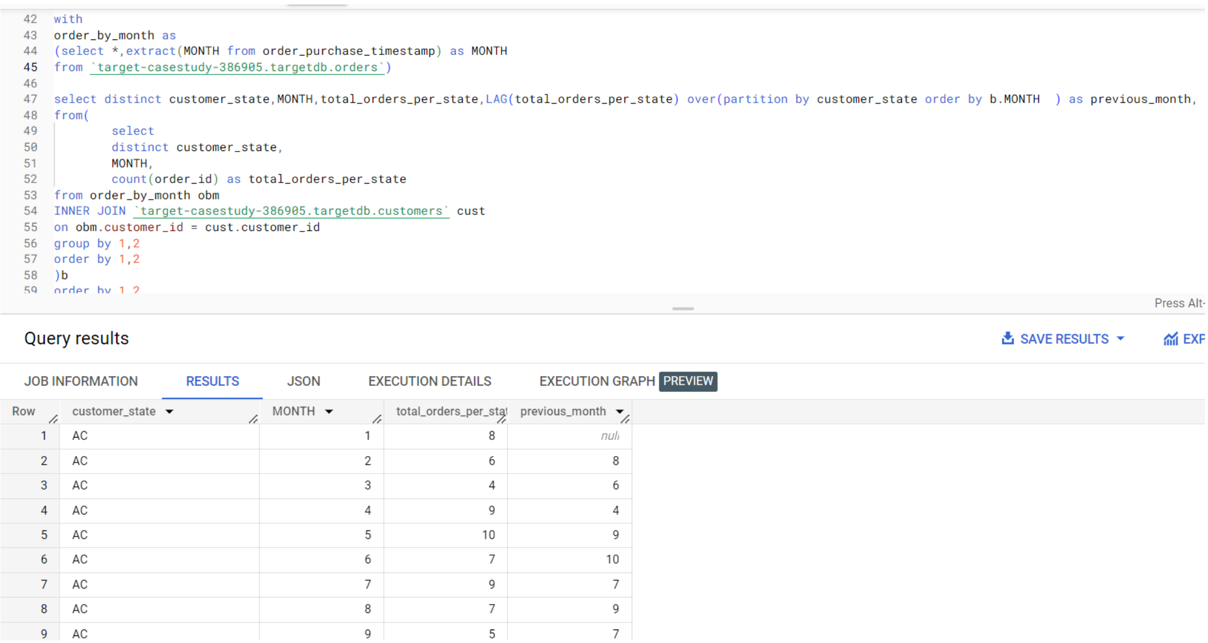
- Since number of orders in dawn are very less, personalised notifications could be sent to customers with additional discounts during dawn hour (5 to 6 AM)

Recommendations:

- There should be a mechanism to identify early risers as targeted customer .

Evolution of E-commerce orders in the Brazil region:

1. Get month on month orders by states



```
with
order_by_month as
(select *,extract(MONTH from order_purchase_timestamp) as MONTH
from `target-casestudy-386905.targetdb.orders`)

select distinct customer_state,MONTH,total_orders_per_state,LAG(total_orders_per_state) over(partition by customer_state order by b.MONTH  ) as previous_month,
from(
        select
        distinct customer_state,
        MONTH,
        count(order_id) as total_orders_per_state
from order_by_month obm
INNER JOIN `target-casestudy-386905.targetdb.customers` cust
on obm.customer_id = cust.customer_id
group by 1,2
order by 1,2
)b
order by 1,2
```

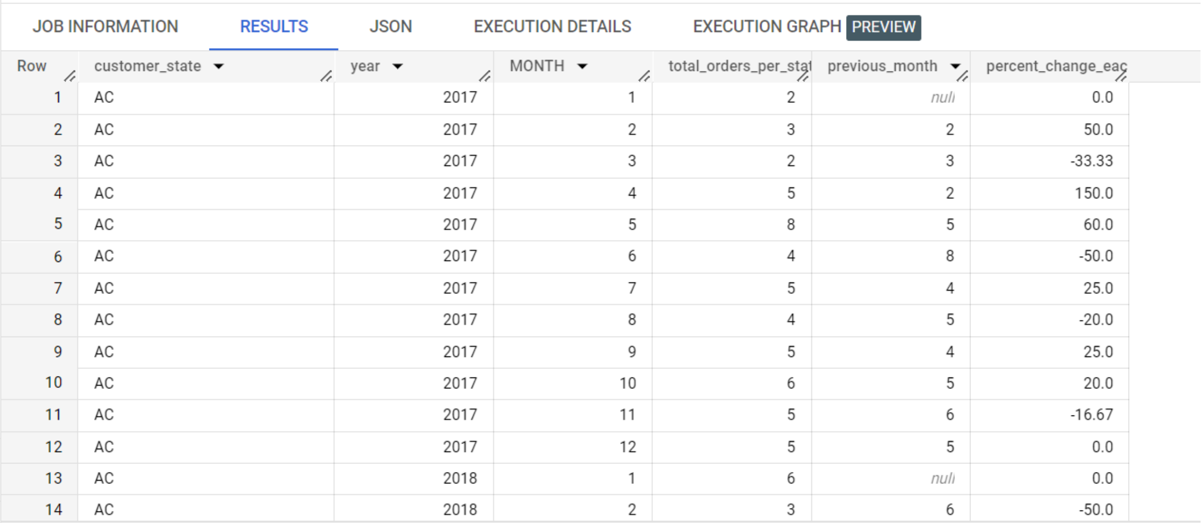
Press Alt-

Query results

Row	customer_state	MONTH	total_orders_per_stat	previous_month
1	AC	1	8	null
2	AC	2	6	8
3	AC	3	4	6
4	AC	4	9	4
5	AC	5	10	9
6	AC	6	7	10
7	AC	7	9	7
8	AC	8	7	9
9	AC	9	5	7

Second approach, the results attached from the below query

```
## Get month on month orders by states

## Get month on month orders by states

with
order_by_month as
(
        select *,
        extract(year from order_purchase_timestamp) as year,
        extract(MONTH from order_purchase_timestamp) as MONTH
from `target-casestudy-386905.targetdb.orders`
)

select distinct customer_state,year, MONTH,total_orders_per_state,previous_month,
        CASE
        WHEN previous_month is NULL then 0
        ELSE round(((total_orders_per_state -
previous_month)/previous_month)*100,2)
        END as percent_change_each_month
from(
        select distinct
customer_state,year,MONTH,total_orders_per_state,LAG(total_orders_per_state)
over(partition by customer_state,year order by b.MONTH  asc) as previous_month,
from(
        select
        distinct customer_state,
        year,
        MONTH,
        count(order_id) as total_orders_per_state
from order_by_month obm
INNER JOIN `target-casestudy-386905.targetdb.customers` cust
on obm.customer_id = cust.customer_id
group by 1,2,3
order by 1,2,3
)b
order by 1,2,3
)c
order by 1,2,3;
```



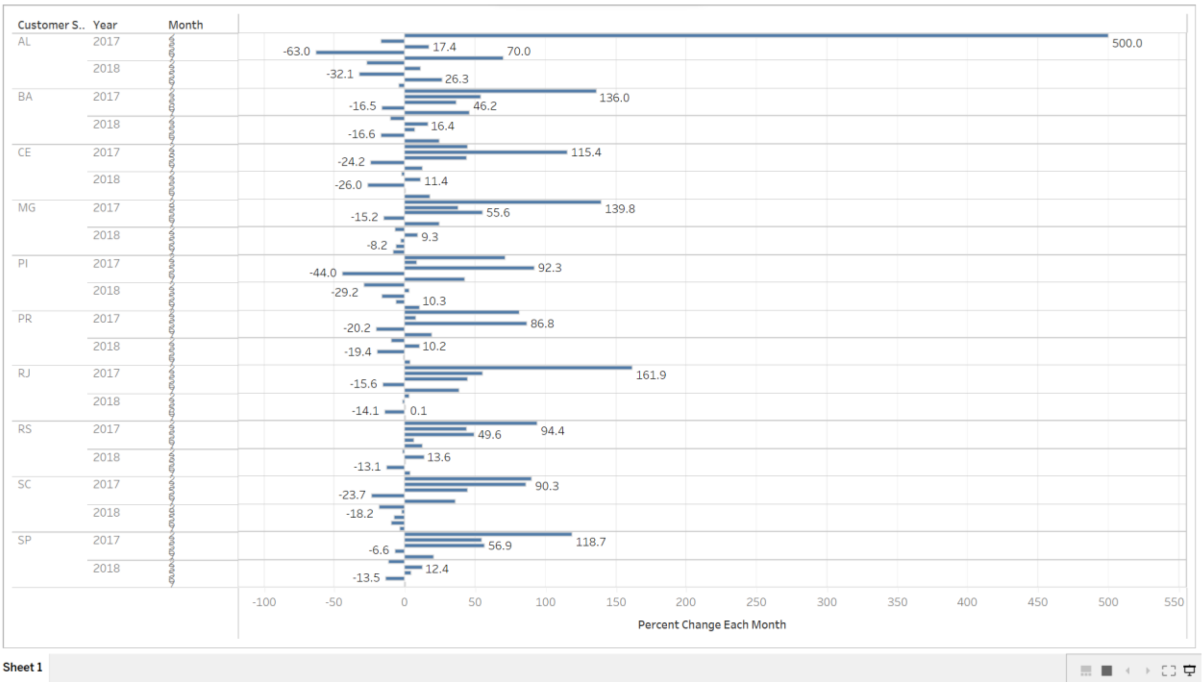
Row	customer_state	year	MONTH	total_orders_per_stat	previous_month	percent_change_eac
1	AC	2017	1	2	null	0.0
2	AC	2017	2	3	2	50.0
3	AC	2017	3	2	3	-33.33
4	AC	2017	4	5	2	150.0
5	AC	2017	5	8	5	60.0
6	AC	2017	6	4	8	-50.0
7	AC	2017	7	5	4	25.0
8	AC	2017	8	4	5	-20.0
9	AC	2017	9	5	4	25.0
10	AC	2017	10	6	5	20.0
11	AC	2017	11	5	6	-16.67
12	AC	2017	12	5	5	0.0
13	AC	2018	1	6	null	0.0
14	AC	2018	2	3	6	-50.0

https://docs.google.com/spreadsheets/d/1lrPikScqS3dDYfQUD_ug9I05uI5uQoguvlSbwk4Ky2E/edit?usp=sharing

Complete data attached above

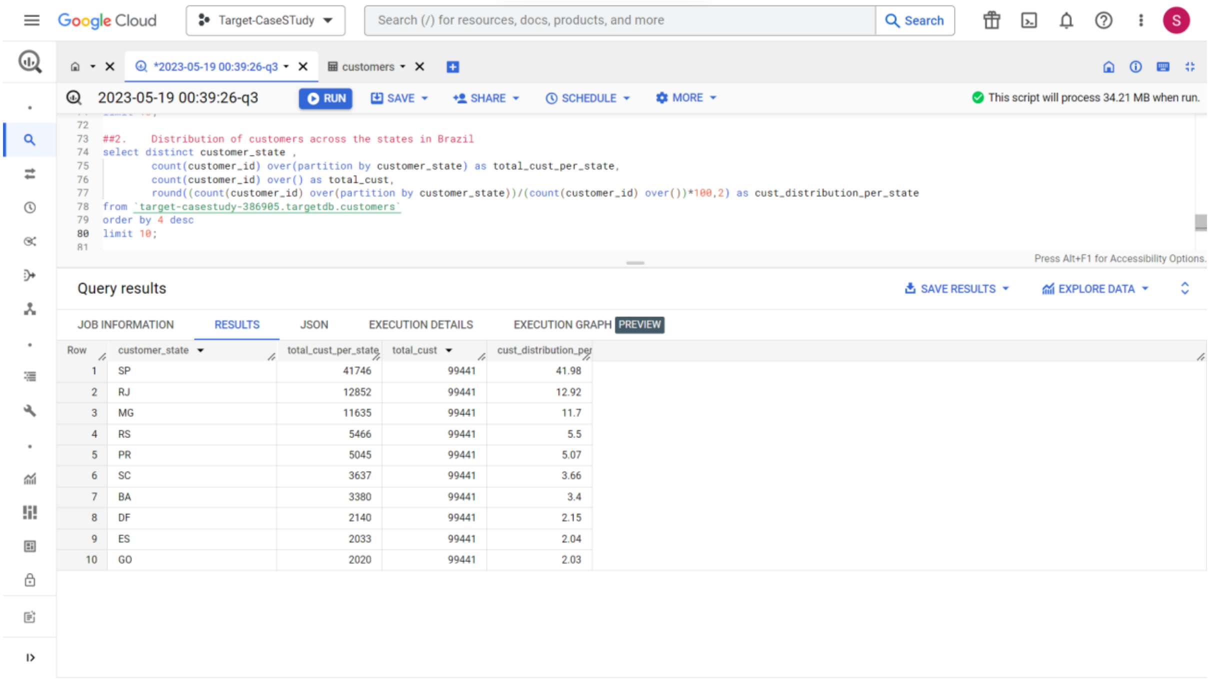
Overall orders are placed more during Feb for most of the state

Top 5 state with highest percentage change



2. Distribution of customers across the states in Brazil

```
##2.	Distribution of customers across the states in Brazil
select distinct customer_state ,
        count(customer_id) over(partition by customer_state) as
total_cust_per_state,
        count(customer_id) over() as total_cust,
        round((count(customer_id) over(partition by
customer_state))/(count(customer_id) over())*100,2) as cust_distribution_per_state
from `target-casestudy-386905.targetdb.customers`
order by 4 desc
limit 10;
```



```
##2.   Distribution of customers across the states in Brazil
select distinct customer_state ,
        count(customer_id) over(partition by customer_state) as total_cust_per_state,
        count(customer_id) over() as total_cust,
        round((count(customer_id) over(partition by customer_state))/(count(customer_id) over())*100,2) as cust_distribution_per_state
from `target-casestudy-386905.targetdb.customers`
order by 4 desc
limit 10;
```

Row	customer_state	total_cust_per_state	total_cust	cust_distribution_per
1	SP	41746	99441	41.98
2	RJ	12852	99441	12.92
3	MG	11635	99441	11.7
4	RS	5466	99441	5.5
5	PR	5045	99441	5.07
6	SC	3637	99441	3.66
7	BA	3380	99441	3.4
8	DF	2140	99441	2.15
9	ES	2033	99441	2.04
10	GO	2020	99441	2.03

Most of the customers are from SP and very few from GO.

Actionable Insights:

- Customers in SC, BA, DF,ES,GO etc should be targeted

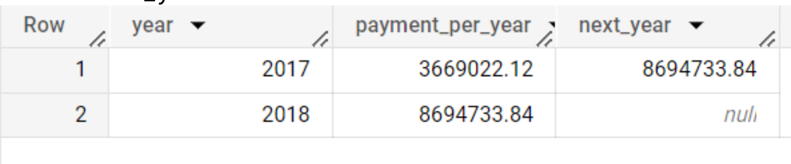
Recommendations:

- Marketing campaign in various cities across the state where the customer distribution is less than 5%

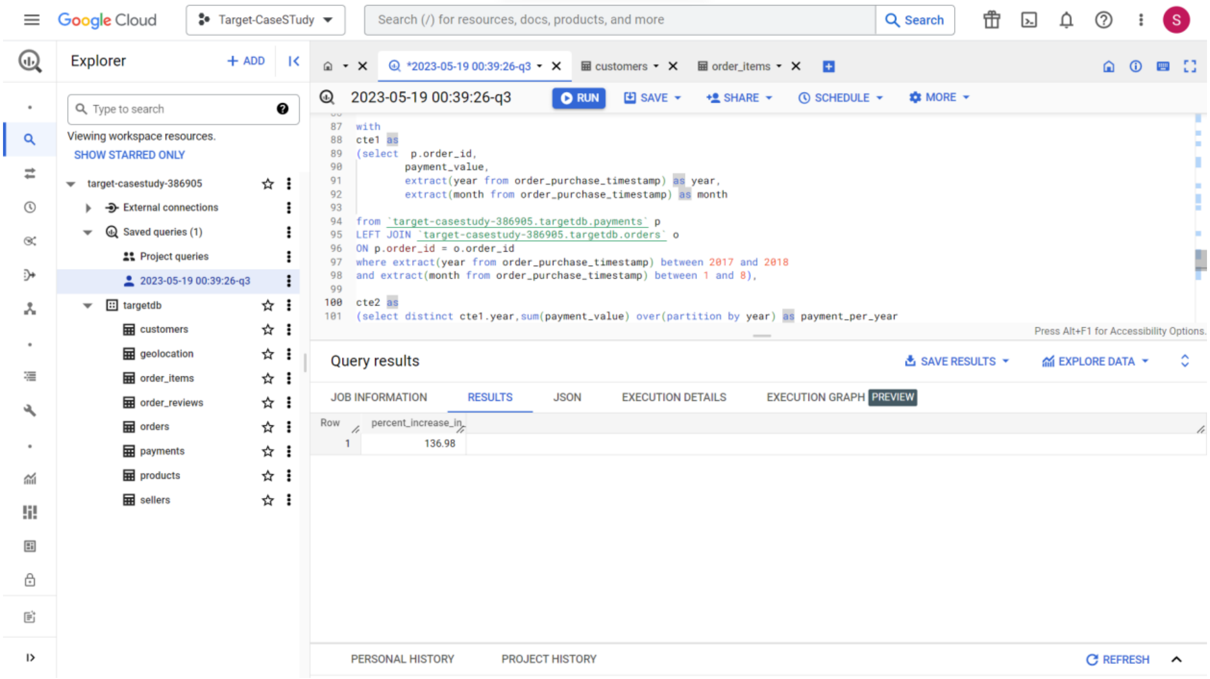
Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.

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  p.order_id,
        payment_value,
        extract(year from order_purchase_timestamp) as year,
        extract(month from order_purchase_timestamp) as month

from `target-casestudy-386905.targetdb.payments` p
LEFT JOIN `target-casestudy-386905.targetdb.orders` o
ON p.order_id = o.order_id
where extract(year from order_purchase_timestamp) between 2017 and 2018
and extract(month from order_purchase_timestamp) between 1 and 8),

cte2 as
(select distinct cte1.year,sum(payment_value) over(partition by year) as
payment_per_year
from cte1)

select round(((next_year - payment_per_year)/payment_per_year)*100,2) as
percent_increase_in_cost
from(
        select year ,payment_per_year,LEAD(payment_per_year) over(order by
payment_per_year) as next_year
from cte2
)a
where next_year is not null
```



Row	year	payment_per_year	next_year
1	2017	3669022.12	8694733.84
2	2018	8694733.84	null

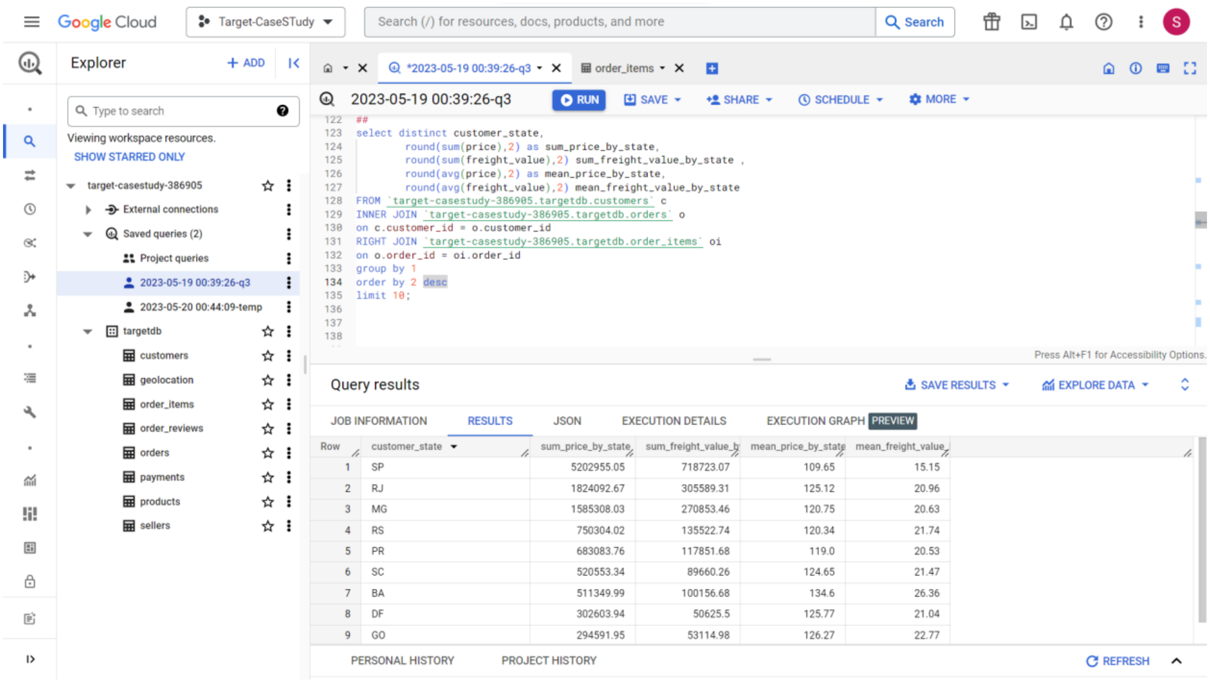


```
with
cte1 as
(select  p.order_id,
        payment_value,
        extract(year from order_purchase_timestamp) as year,
        extract(month from order_purchase_timestamp) as month

from `target-casestudy-386905.targetdb.payments` p
LEFT JOIN `target-casestudy-386905.targetdb.orders` o
ON p.order_id = o.order_id
where extract(year from order_purchase_timestamp) between 2017 and 2018
and extract(month from order_purchase_timestamp) between 1 and 8),

cte2 as
(select distinct cte1.year,sum(payment_value) over(partition by year) as payment_per_year
```

Row	percent_increase_in
1	136.98

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

```sql
select distinct customer_state,
        round(sum(price),2) as sum_price_by_state,
        round(sum(freight_value),2) sum_freight_value_by_state ,
        round(avg(price),2) as mean_price_by_state,
        round(avg(freight_value),2) mean_freight_value_by_state
FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
RIGHT JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 2 desc
limit 10;
```



```sql
##
select distinct customer_state,
        round(sum(price),2) as sum_price_by_state,
        round(sum(freight_value),2) sum_freight_value_by_state ,
        round(avg(price),2) as mean_price_by_state,
        round(avg(freight_value),2) mean_freight_value_by_state
FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
RIGHT JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 2 desc
limit 10;
```

Query results

Row	customer_state	sum_price_by_state	sum_freight_value_b	mean_price_by_state	mean_freight_value_
1	SP	5202955.05	718723.07	109.65	15.15
2	RJ	1824092.67	305589.31	125.12	20.96
3	MG	1585308.03	270853.46	120.75	20.63
4	RS	750304.02	135522.74	120.34	21.74
5	PR	683083.76	117851.68	119.0	20.53
6	SC	520553.34	89660.26	124.65	21.47
7	BA	511349.99	100156.68	134.6	26.36
8	DF	302603.94	50625.5	125.77	21.04
9	GO	294591.95	53114.98	126.27	22.77

State SP is a major contributor to Brazil’s economy.

Actionable Insights:

- Measures should be taken to reduce average freight value more than 15.

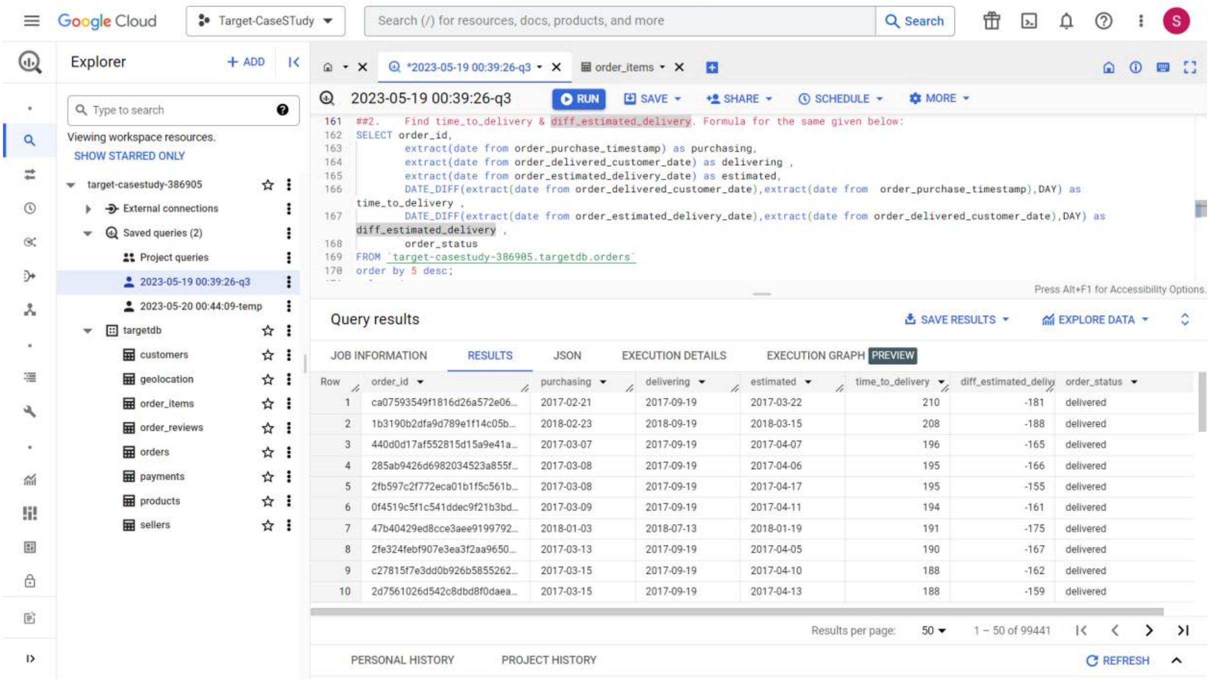
Recommendations:

- Creation of new Hubs/warehouse in proximity of states where freight values are on higher side.
- Onboarding of new seller in these areas

Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
SELECT order_id,
        extract(date from order_purchase_timestamp) as purchasing,
        extract(date from order_delivered_customer_date) as delivering ,
        extract(date from order_estimated_delivery_date) as estimated,
        DATE_DIFF(extract(date from order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY) as time_to_delivery ,
        DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date
from order_delivered_customer_date),DAY) as diff_estimated_delivery ,
        order_status
FROM `target-casestudy-386905.targetdb.orders`
order by 5 desc;
```

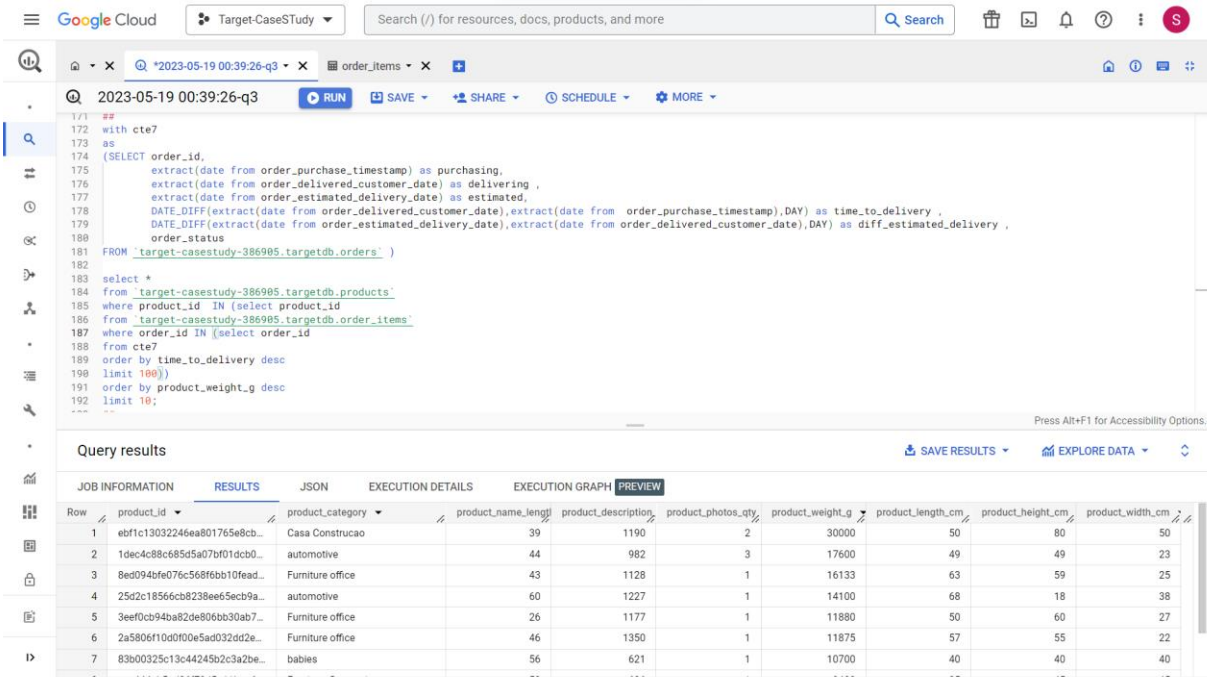


```
##2.   Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
SELECT order_id,
        extract(date from order_purchase_timestamp) as purchasing,
        extract(date from order_delivered_customer_date) as delivering ,
        extract(date from order_estimated_delivery_date) as estimated,
        DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY) as
time_to_delivery ,
        DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY) as
diff_estimated_delivery ,
        order_status
FROM `target-casestudy-386905.targetdb.orders`
order by 5 desc;
```

Row	order_id	purchasing	delivering	estimated	time_to_delivery	diff_estimated_deliv	order_status
1	ca07593549f1816d26a572e06...	2017-02-21	2017-09-19	2017-03-22	210	-181	delivered
2	1b3190b2dfa9d789e1f14c05b...	2018-02-23	2018-09-19	2018-03-15	208	-188	delivered
3	440d0d17af552815d15a9e41a...	2017-03-07	2017-09-19	2017-04-07	196	-165	delivered
4	285ab9426d6982034523a855f...	2017-03-08	2017-09-19	2017-04-06	195	-166	delivered
5	2fb597c2f772eca01b1f5c561b...	2017-03-08	2017-09-19	2017-04-17	195	-155	delivered
6	0f4519c5f1c541ddec9f21b3bd...	2017-03-09	2017-09-19	2017-04-11	194	-161	delivered
7	47b40429ed8cce3aee9199792...	2018-01-03	2018-07-13	2018-01-19	191	-175	delivered
8	2fe324febf907e3ea3f2aa9650...	2017-03-13	2017-09-19	2017-04-05	190	-167	delivered
9	c27815f7e3dd0b926b5855262...	2017-03-15	2017-09-19	2017-04-10	188	-162	delivered
10	2d7561026d542c8dbd8f0daea...	2017-03-15	2017-09-19	2017-04-13	188	-159	delivered

- Time_to_delivery for few orders is more than 6 months
- Upon analysing top 100 orders , these are found out to be belonging to product_category , Automotive, Furniture , Construction etc

```
with cte7
as
(SELECT order_id,
        extract(date from order_purchase_timestamp) as purchasing,
        extract(date from order_delivered_customer_date) as delivering ,
        extract(date from order_estimated_delivery_date) as estimated,
        DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY) as time_to_delivery ,
        DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY) as diff_estimated_delivery ,
        order_status
FROM `target-casestudy-386905.targetdb.orders` )

select *
from `target-casestudy-386905.targetdb.products`
where product_id  IN (select product_id
from `target-casestudy-386905.targetdb.order_items`
where order_id IN (select order_id
from cte7
order by time_to_delivery desc
limit 100))
order by product_weight_g desc
limit 10;
```



```
##
with cte7
as
(SELECT order_id,
        extract(date from order_purchase_timestamp) as purchasing,
        extract(date from order_delivered_customer_date) as delivering ,
        extract(date from order_estimated_delivery_date) as estimated,
        DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY) as time_to_delivery ,
        DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY) as diff_estimated_delivery ,
        order_status
FROM `target-casestudy-386905.targetdb.orders` )

select *
from `target-casestudy-386905.targetdb.products`
where product_id  IN (select product_id
from `target-casestudy-386905.targetdb.order_items`
where order_id IN (select order_id
from cte7
order by time_to_delivery desc
limit 100))
order by product_weight_g desc
limit 10;
```

Query results

Row	product_id	product_category	product_name_lengt	product_description	product_photos_qty	product_weight_g	product_length_cm	product_height_cm	product_width_cm
1	ebf1c13032246ea801765e8cb...	Casa Construcao	39	1190	2	30000	50	80	50
2	1dec4c88c685d5a07bf01dcb0...	automotive	44	982	3	17600	49	49	23
3	8ed094bfe076c568f6bb10fead...	Furniture office	43	1128	1	16133	63	59	25
4	25d2c18566cb8238ee65ecb9a...	automotive	60	1227	1	14100	68	18	38
5	3eef0cb94ba82de806bb30ab7...	Furniture office	26	1177	1	11880	50	60	27
6	2a5806f10d0f00e5ad032dd2e...	Furniture office	46	1350	1	11875	57	55	22
7	83b00325c13c44245b2c3a2be...	babies	56	621	1	10700	40	40	40

Actionable Insights:

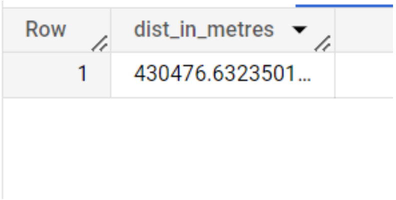
- estimated delivery for the products under heavy good like automotive etc should be a higher date.

Recommendations:

Increasing the sellers in proximity of customer's zip code

For e.g. for order_id `'ca07593549f1816d26a572e06dc1eab6'` **with highest time_to_delivery distance between seller and customer zip code is 431 Kms**

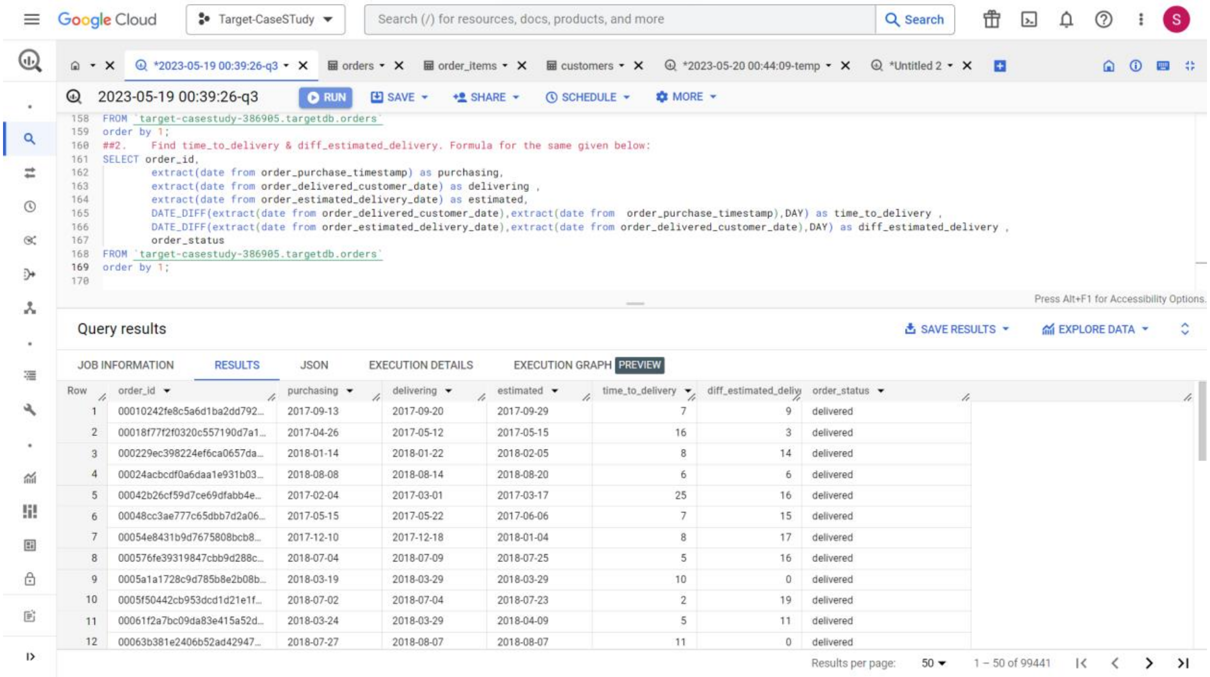
```
select *
from `target-casestudy-386905.targetdb.products`
where product_id = (select *
from `target-casestudy-386905.targetdb.order_items`
where order_id = 'ca07593549f1816d26a572e06dc1eab6');
select *
from `target-casestudy-386905.targetdb.orders`
where order_id = 'ca07593549f1816d26a572e06dc1eab6';
select *
from `target-casestudy-386905.targetdb.customers`
where customer_id = '75683a92331068e2d281b11a7866ba44';
select *
from `target-casestudy-386905.targetdb.sellers`
where seller_id = '2a1348e9addc1af5aaa619b1a3679d6b';
select *
from `target-casestudy-386905.targetdb.geolocation`
where geolocation_zip_code_prefix = 30494;
select *
from `target-casestudy-386905.targetdb.geolocation`
where geolocation_zip_code_prefix = 29890

##distance betwen two long and lat

with points as (
  -- Longitudes and latitudes are in degrees
  select -40.359526081486742 as from_long, -18.125972944247216 as from_lat, -
43.968436347881045 as to_long, -19.956616907484161 as to_lat
)

select
  st_distance(
    st_geogpoint(from_long, from_lat),
    st_geogpoint(to_long,   to_lat)
  ) as dist_in_metres
from points;
```



Row	dist_in_metres
1	430476.6323501...

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

```
##2.    Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
SELECT order_id,
        extract(date from order_purchase_timestamp) as purchasing,
        extract(date from order_delivered_customer_date) as delivering ,
        extract(date from order_estimated_delivery_date) as estimated,
        DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY) as time_to_delivery ,
        DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY) as diff_estimated_delivery ,
        order_status
FROM `target-casestudy-386905.targetdb.orders`
order by 1;
```

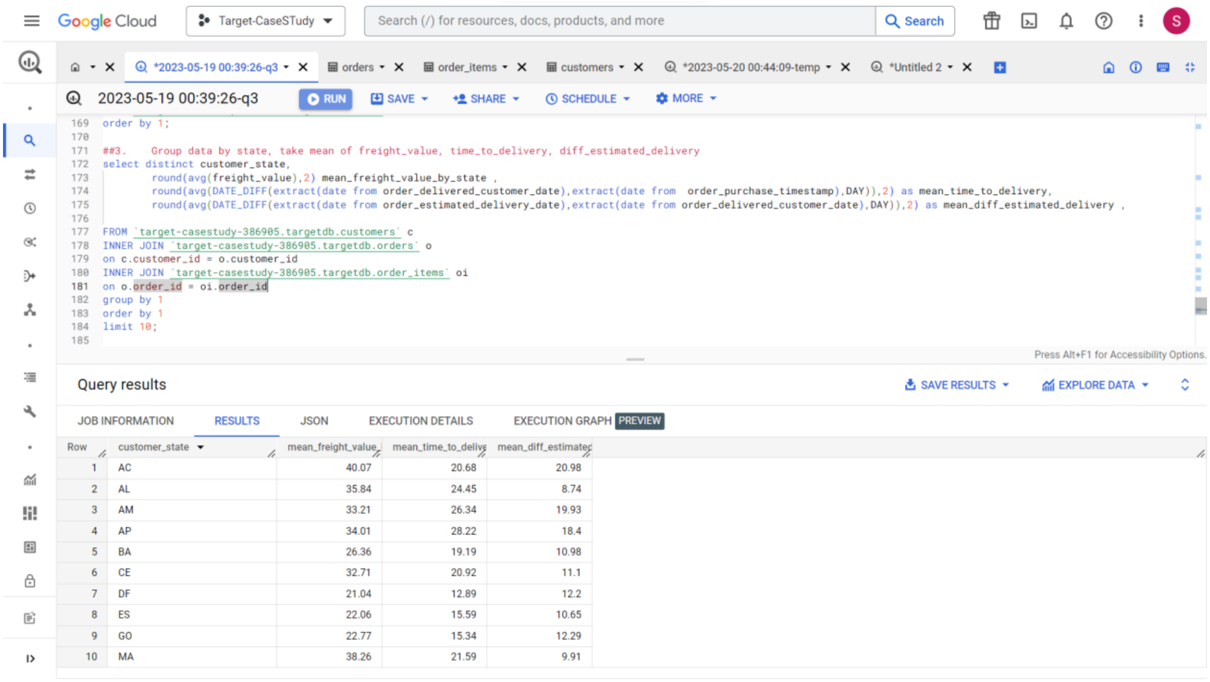


```
FROM `target-casestudy-386905.targetdb.orders`
order by 1;
##2.    Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
SELECT order_id,
        extract(date from order_purchase_timestamp) as purchasing,
        extract(date from order_delivered_customer_date) as delivering ,
        extract(date from order_estimated_delivery_date) as estimated,
        DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY) as time_to_delivery ,
        DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY) as diff_estimated_delivery ,
        order_status
FROM `target-casestudy-386905.targetdb.orders`
order by 1;
```

Row	order_id	purchasing	delivering	estimated	time_to_delivery	diff_estimated_deliv	order_status
1	00010242fe8c5a6d1ba2dd792...	2017-09-13	2017-09-20	2017-09-29	7	9	delivered
2	00018f77f2f0320c557190d7a1...	2017-04-26	2017-05-12	2017-05-15	16	3	delivered
3	000229ec398224ef6ca0657da...	2018-01-14	2018-01-22	2018-02-05	8	14	delivered
4	00024acbcdf0a6daa1e931b03...	2018-08-08	2018-08-14	2018-08-20	6	6	delivered
5	00042b26cf59d7ce69dfabb4e...	2017-02-04	2017-03-01	2017-03-17	25	16	delivered
6	00048cc3ae777c65dbb7d2a06...	2017-05-15	2017-05-22	2017-06-06	7	15	delivered
7	00054e8431b9d7675808bcb8...	2017-12-10	2017-12-18	2018-01-04	8	17	delivered
8	000576fe39319847cbb9d288c...	2018-07-04	2018-07-09	2018-07-25	5	16	delivered
9	0005a1a1728c9d785b8e2b08b...	2018-03-19	2018-03-29	2018-03-29	10	0	delivered
10	0005f50442cb953dcd1d21e1f...	2018-07-02	2018-07-04	2018-07-23	2	19	delivered
11	00061f2a7bc09da83e415a52d...	2018-03-24	2018-03-29	2018-04-09	5	11	delivered
12	00063b381e2406b52ad42947...	2018-07-27	2018-08-07	2018-08-07	11	0	delivered

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

```
##3.    Group data by state, take mean of freight_value, time_to_delivery,
diff_estimated_delivery
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 1
limit 10;
```



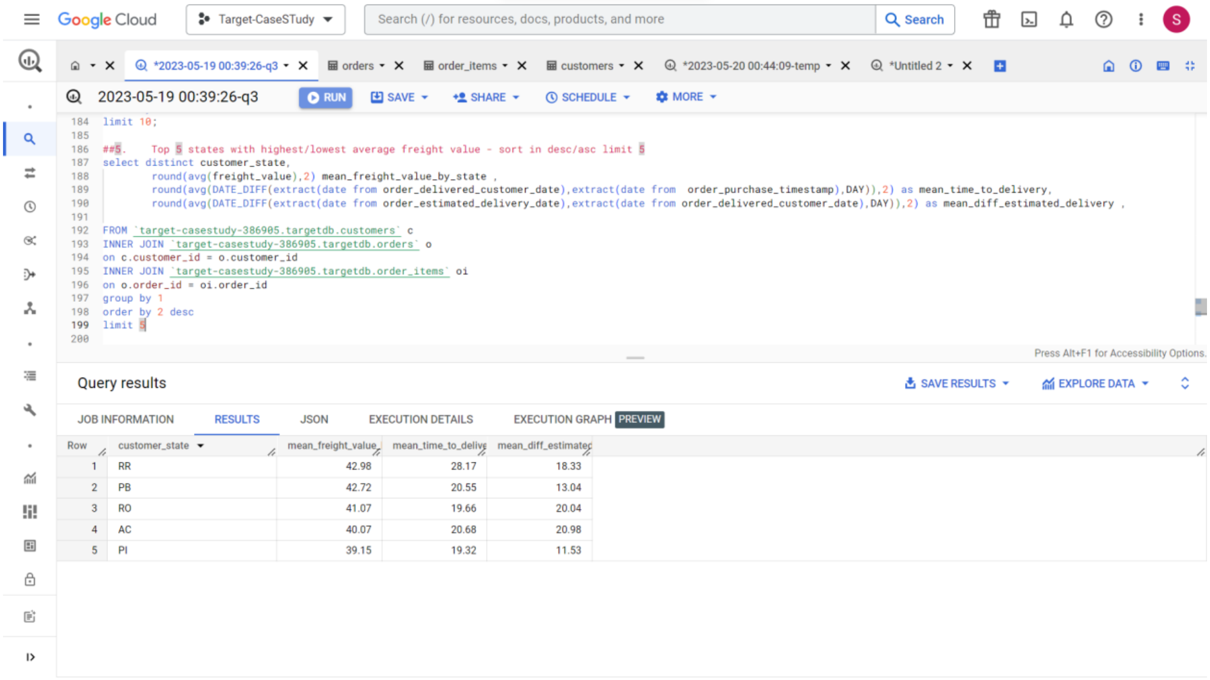
```
order by 1;

##3.    Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 1
limit 10;
```

Row	customer_state	mean_freight_value_	mean_time_to_delive	mean_diff_estimated
1	AC	40.07	20.68	20.98
2	AL	35.84	24.45	8.74
3	AM	33.21	26.34	19.93
4	AP	34.01	28.22	18.4
5	BA	26.36	19.19	10.98
6	CE	32.71	20.92	11.1
7	DF	21.04	12.89	12.2
8	ES	22.06	15.59	10.65
9	GO	22.77	15.34	12.29
10	MA	38.26	21.59	9.91

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

Top 5 states with highest average freight value

```
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 2 desc
limit 5
```



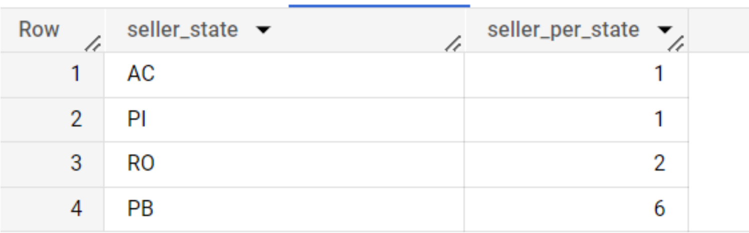
```
limit 10;

##5.    Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 2 desc
limit 5
```

Row	customer_state	mean_freight_value_	mean_time_to_delive	mean_diff_estimated
1	RR	42.98	28.17	18.33
2	PB	42.72	20.55	13.04
3	RO	41.07	19.66	20.04
4	AC	40.07	20.68	20.98
5	PI	39.15	19.32	11.53

Actionable Insights: The state with highest mean freight value does not have any direct seller for that state, further states have either one seller per state.

Numbers of seller should be increase across these state , and new sellers should be onboarded in state with no seller

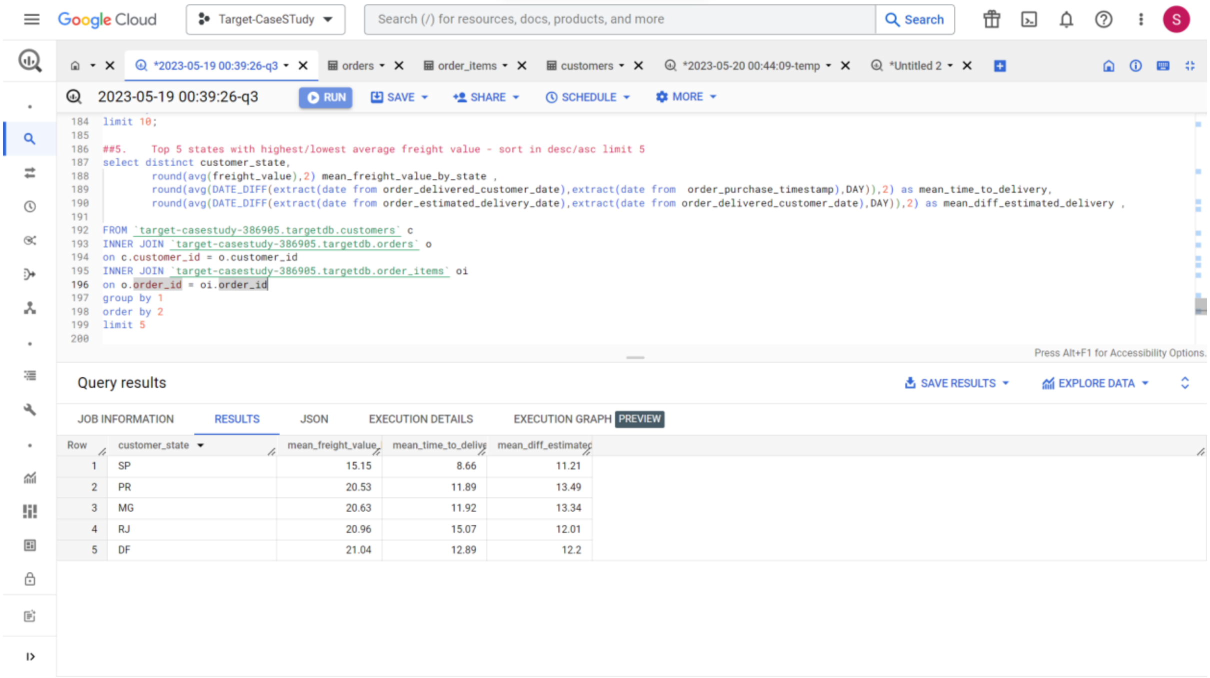
```
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('AC','PB','RO','RR','PI')
group by 1
order by 2;
```



Row	seller_state	seller_per_state
1	AC	1
2	PI	1
3	RO	2
4	PB	6

Top 5 states with lowest average freight value

```
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 2
limit 5
```



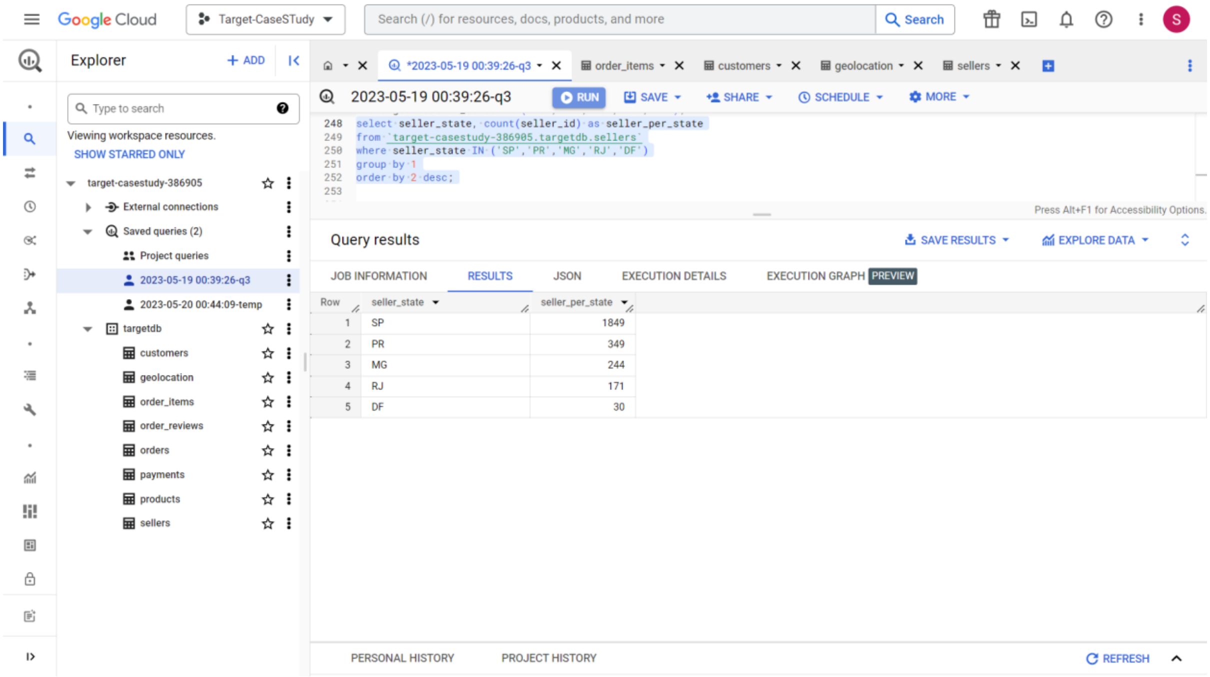
```
limit 10;

##5.   Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 2
limit 5
```

Row	customer_state	mean_freight_value_	mean_time_to_delive	mean_diff_estimatec
1	SP	15.15	8.66	11.21
2	PR	20.53	11.89	13.49
3	MG	20.63	11.92	13.34
4	RJ	20.96	15.07	12.01
5	DF	21.04	12.89	12.2

Actionable Insights: The state with lowest mean freight value have multiple sellers for that state.

```
select seller_state, count(seller_id) as seller_per_state
```

```
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('SP','PR','MG','RJ','DF')
group by 1
order by 2 desc;
```



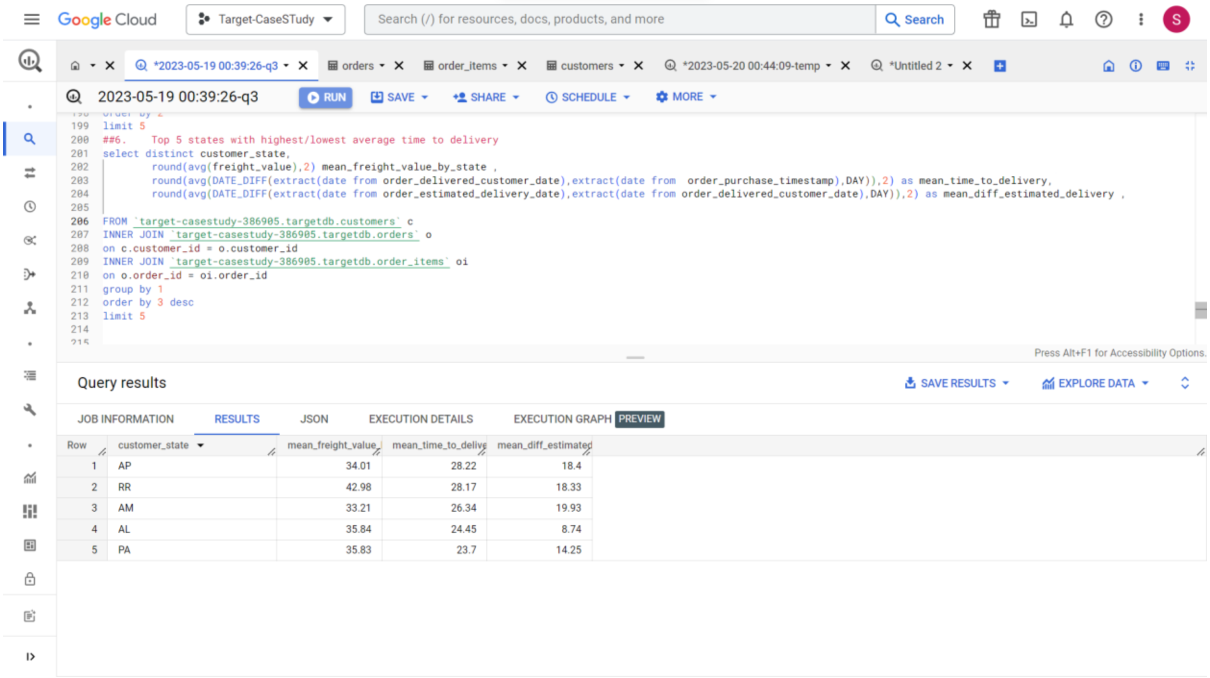
```
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('SP','PR','MG','RJ','DF')
group by 1
order by 2 desc;
```

Row	seller_state	seller_per_state
1	SP	1849
2	PR	349
3	MG	244
4	RJ	171
5	DF	30

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

Highest

```
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 3 desc
limit 5
```



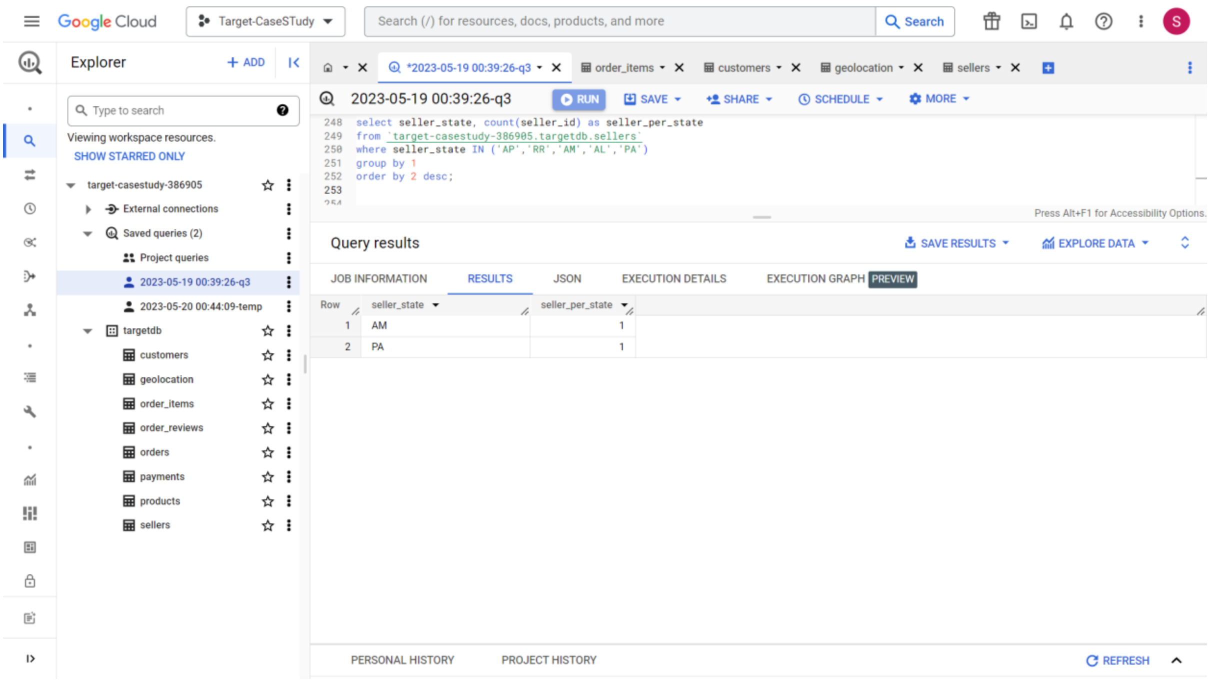
```
limit 5
##6.    Top 5 states with highest/lowest average time to delivery
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 3 desc
limit 5
```

Row	customer_state	mean_freight_value_	mean_time_to_delive	mean_diff_estimated
1	AP	34.01	28.22	18.4
2	RR	42.98	28.17	18.33
3	AM	33.21	26.34	19.93
4	AL	35.84	24.45	8.74
5	PA	35.83	23.7	14.25

Actionable Insights : The state with highest mean time to delivery does not have any direct seller for that state, further states have either one seller per state.

Numbers of seller should be increase across these state, and new sellers should be onboarded in state with no seller

```
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('AP','RR','AM','AL','PA')
group by 1
order by 2 desc;
```

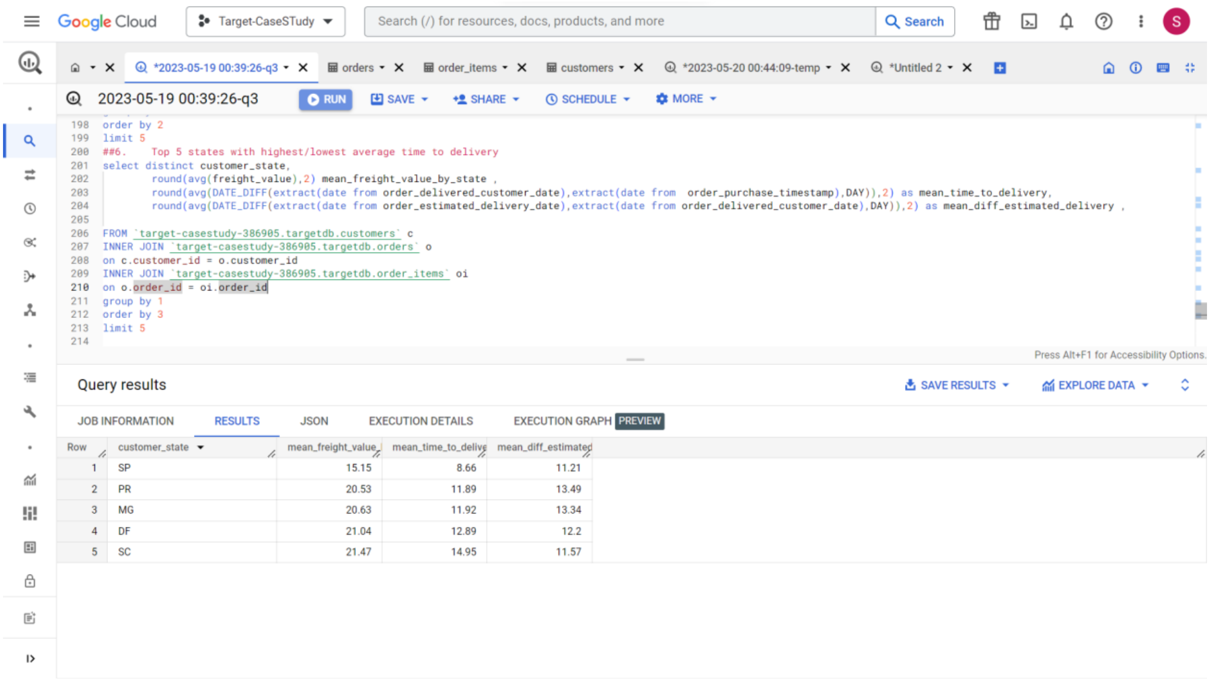


```
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('AP','RR','AM','AL','PA')
group by 1
order by 2 desc;
```

Row	seller_state	seller_per_state
1	AM	1
2	PA	1

Lowest

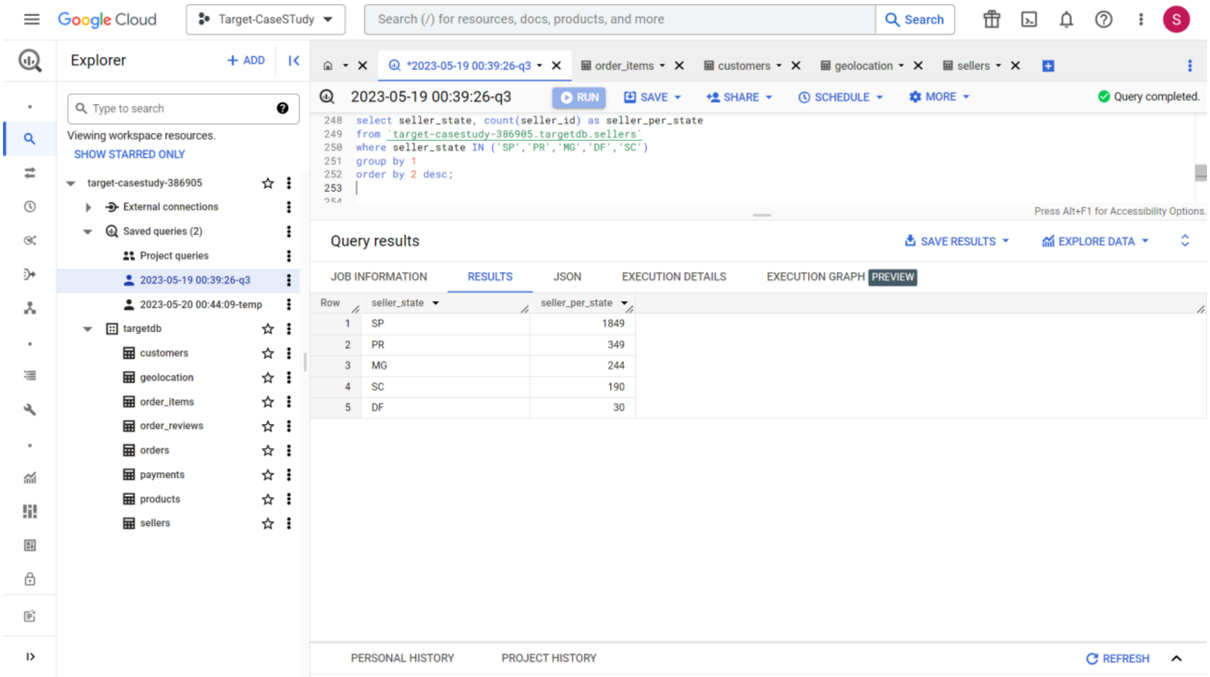
```
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 3
limit 5
```



```
order by 2
limit 5
##6.    Top 5 states with highest/lowest average time to delivery
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 3
limit 5
```

Row	customer_state	mean_freight_value_	mean_time_to_delive	mean_diff_estimated
1	SP	15.15	8.66	11.21
2	PR	20.53	11.89	13.49
3	MG	20.63	11.92	13.34
4	DF	21.04	12.89	12.2
5	SC	21.47	14.95	11.57

Actionable Insights: The state with lowest mean freight value have multiple sellers for that state.



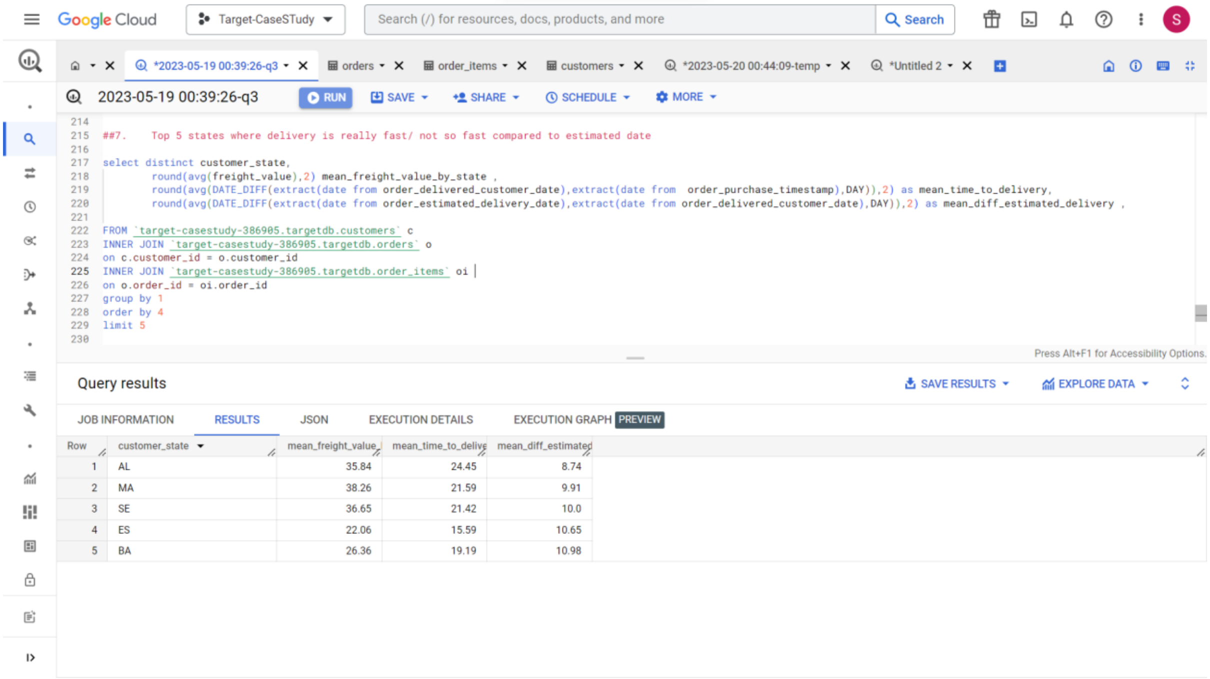
```
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('SP','PR','MG','DF','SC')
group by 1
order by 2 desc;
```

Row	seller_state	seller_per_state
1	SP	1849
2	PR	349
3	MG	244
4	SC	190
5	DF	30

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

FAST

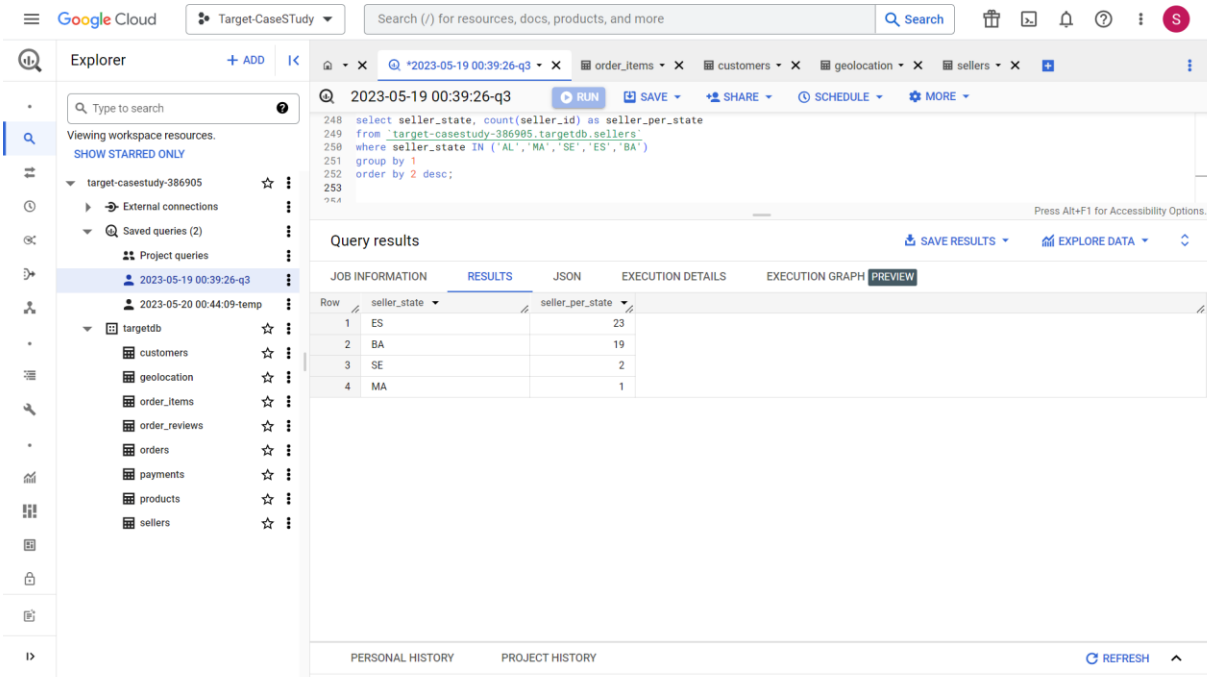
```
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 4
limit 5
```



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

select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 4
limit 5
```

Row	customer_state	mean_freight_value_	mean_time_to_delive	mean_diff_estimated
1	AL	35.84	24.45	8.74
2	MA	38.26	21.59	9.91
3	SE	36.65	21.42	10.0
4	ES	22.06	15.59	10.65
5	BA	26.36	19.19	10.98

Fast delivery with multiple sellers across the state

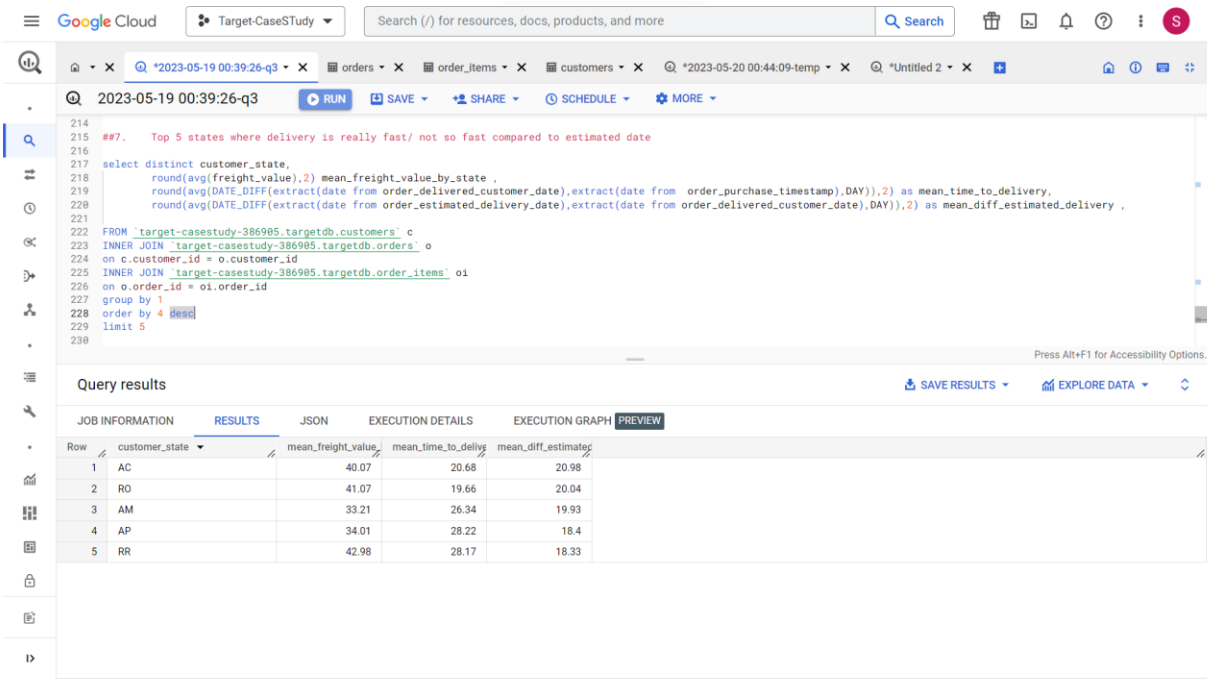


```sql
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('AL','MA','SE','ES','BA')
group by 1
order by 2 desc;
```

Row	seller_state	seller_per_state
1	ES	23
2	BA	19
3	SE	2
4	MA	1

NOT SO FAST

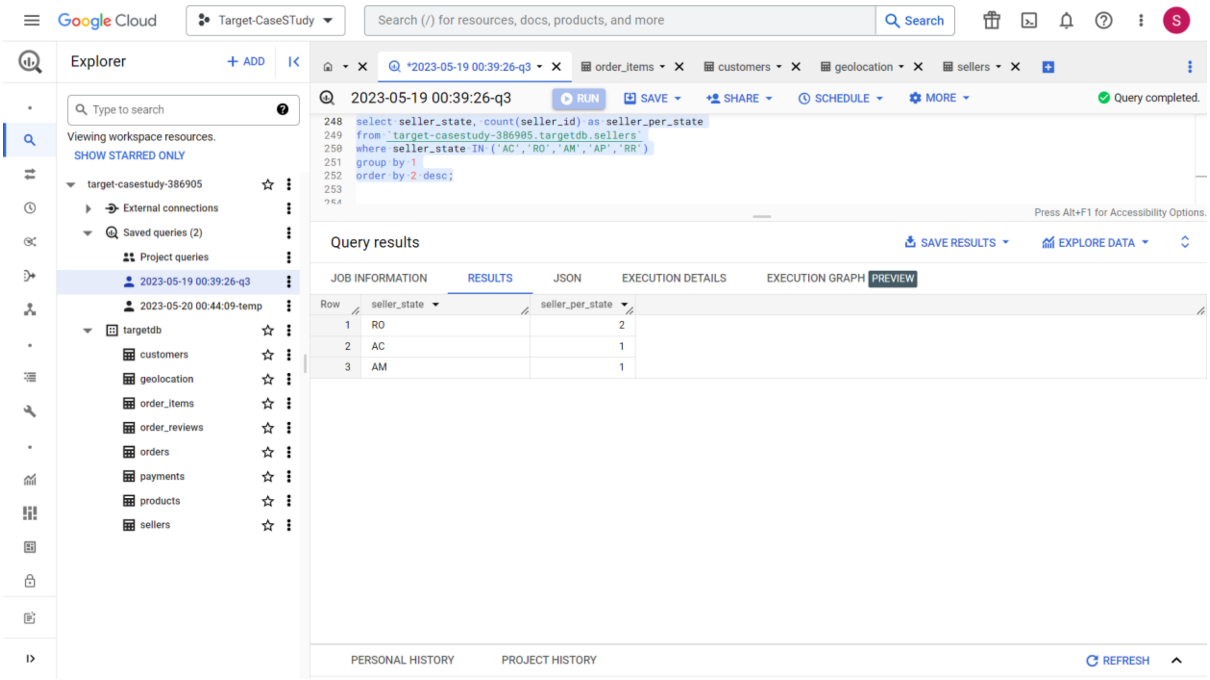
```sql
select distinct customer_state,
        round(avg(freight_value),2) mean_freight_value_by_state ,
        round(avg(DATE_DIFF(extract(date from
order_delivered_customer_date),extract(date
from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
        round(avg(DATE_DIFF(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 4 desc
limit 5
```



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

select distinct customer_state,
       round(avg(freight_value),2) mean_freight_value_by_state ,
       round(avg(DATE_DIFF(extract(date from order_delivered_customer_date),extract(date from  order_purchase_timestamp),DAY)),2) as mean_time_to_delivery,
       round(avg(DATE_DIFF(extract(date from order_estimated_delivery_date),extract(date from order_delivered_customer_date),DAY)),2) as mean_diff_estimated_delivery ,

FROM `target-casestudy-386905.targetdb.customers` c
INNER JOIN `target-casestudy-386905.targetdb.orders` o
on c.customer_id = o.customer_id
INNER JOIN `target-casestudy-386905.targetdb.order_items` oi
on o.order_id = oi.order_id
group by 1
order by 4 desc
limit 5
```

Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimatec
1	AC	40.07	20.68	20.98
2	RO	41.07	19.66	20.04
3	AM	33.21	26.34	19.93
4	AP	34.01	28.22	18.4
5	RR	42.98	28.17	18.33

Slower delivery with less seller across the customer state



```
select seller_state, count(seller_id) as seller_per_state
from `target-casestudy-386905.targetdb.sellers`
where seller_state IN ('AC','RO','AM','AP','RR')
group by 1
order by 2 desc;
```

Row	seller_state	seller_per_state
1	RO	2
2	AC	1
3	AM	1

Payment type analysis

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

```
with
cte6 as
(select payment_type,
        extract(year from order_purchase_timestamp) as year,
        extract(month from order_purchase_timestamp) as month,
        count(p.order_id) as count_of_orders
from `target-casestudy-386905.targetdb.payments` p
LEFT JOIN `target-casestudy-386905.targetdb.orders` o
ON p.order_id = o.order_id
group by 1,2,3
order by 1,2,3 )

select payment_type,
        year,
        month,
        count_of_orders,

        LAG(count_of_orders) over(partition by payment_type,year order by month
asc ) as previous_month_count_of_orders,
        CASE
        WHEN LAG(count_of_orders) over(partition by payment_type,year order by
month asc ) is null then null
        ELSE
        round(((count_of_orders - LAG(count_of_orders) over(partition by
payment_type,year order by month asc))/LAG(count_of_orders) over(partition by
payment_type,year order by month asc))*100,2)
        END as percent_icrease

from cte6
order by 1,2,3;
```

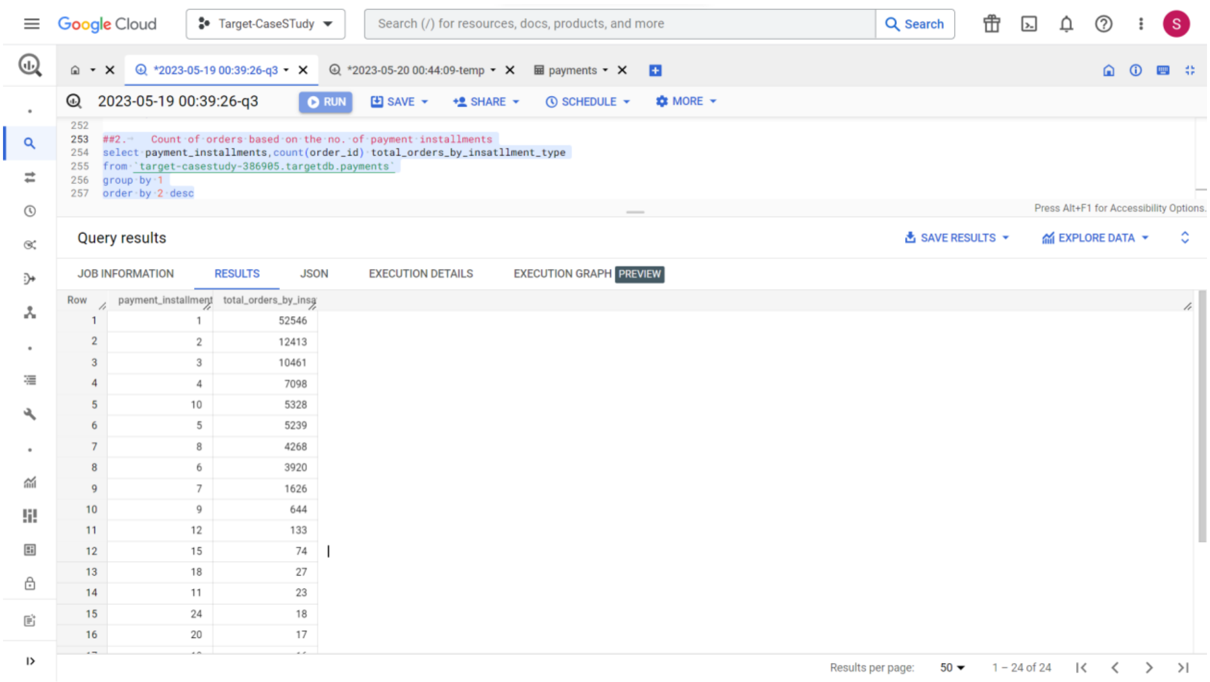
https://docs.google.com/spreadsheets/d/1aVMiD3E1Wqxt3amNfjHJZOu3KaeO3SKRAniQk9yGtpg/edit?usp=sharing

- For payment type UPI, month over month increase was highest in Feb 2017 and lowest in Dec 2017, also year 2017 has more growth in orders over months then 2018.

- For payment type Credit Card , month over month increase was highest in Feb 2017 and lowest in Dec 2017, also year 2017 has more growth in orders over months then 2018.
- For payment type Debit Card, month over month increase was highest in Jun 2018 and lowest in May 2018, also year 2018 has more growth in orders over months then 2017.
- For payment type Voucher, month over month increase was highest in Feb 2017 and lowest in Sept 2018, also year 2017 has more growth in orders over months then 2018.

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

```
##2.    Count of orders based on the no. of payment instalments
select payment_installments,count(order_id) total_orders_by_insatllment_type
from `target-casestudy-386905.targetdb.payments`
group by 1
order by 2 desc
```



```
##2.    Count of orders based on the no. of payment installments
select payment_installments,count(order_id) total_orders_by_insatllment_type
from `target-casestudy-386905.targetdb.payments`
group by 1
order by 2 desc
```

Query results

Row	payment_installment	total_orders_by_insa
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644
11	12	133
12	15	74
13	18	27
14	11	23
15	24	18
16	20	17

Results per page: 50 ▾ 1 – 24 of 24

1. Actionable Insights

- 1 instalment is **most preferred mode** for customers compares to 23 instalments which is **least preferred**.
- Credit is **first most preferred** mode of transaction and UPI is the **second most**
- **Count of orders are highest during each year end of 2017,**

2. Recommendations

- More offers should be given to people to buy in 23 instalments.