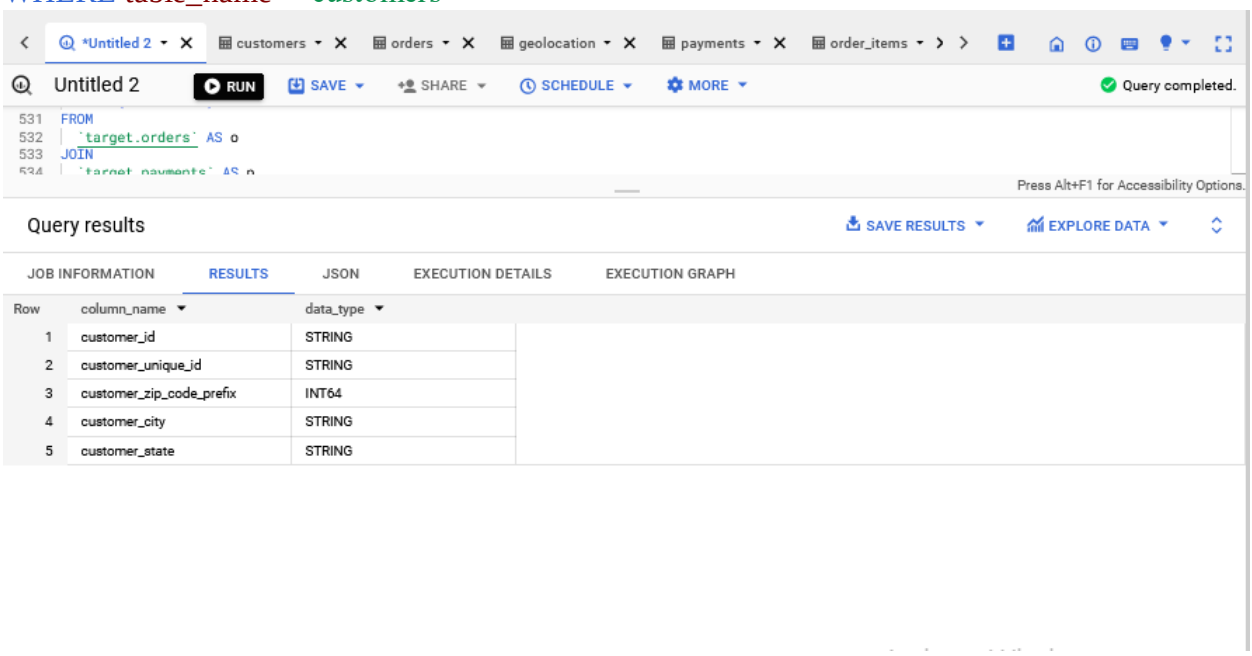
SQL TARGET

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

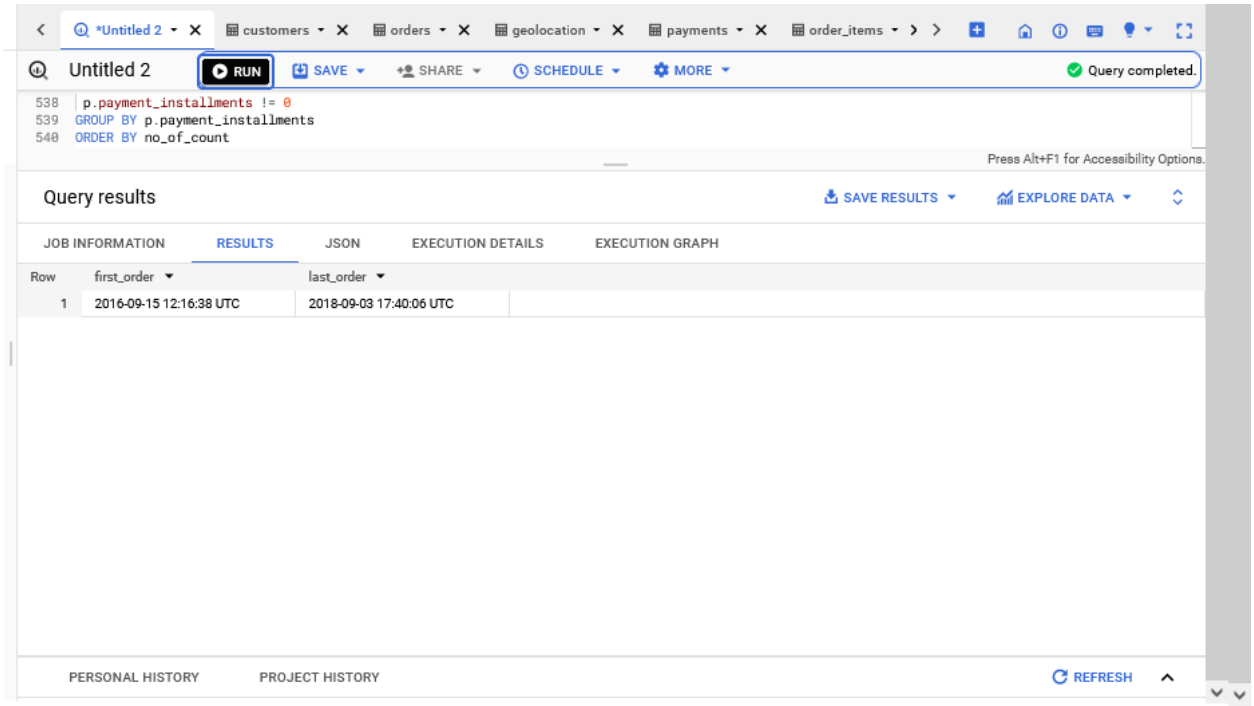
1)Data type of all columns in the "customers" table.

```
SELECT
column_name,
data_type
FROM `target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers'
```



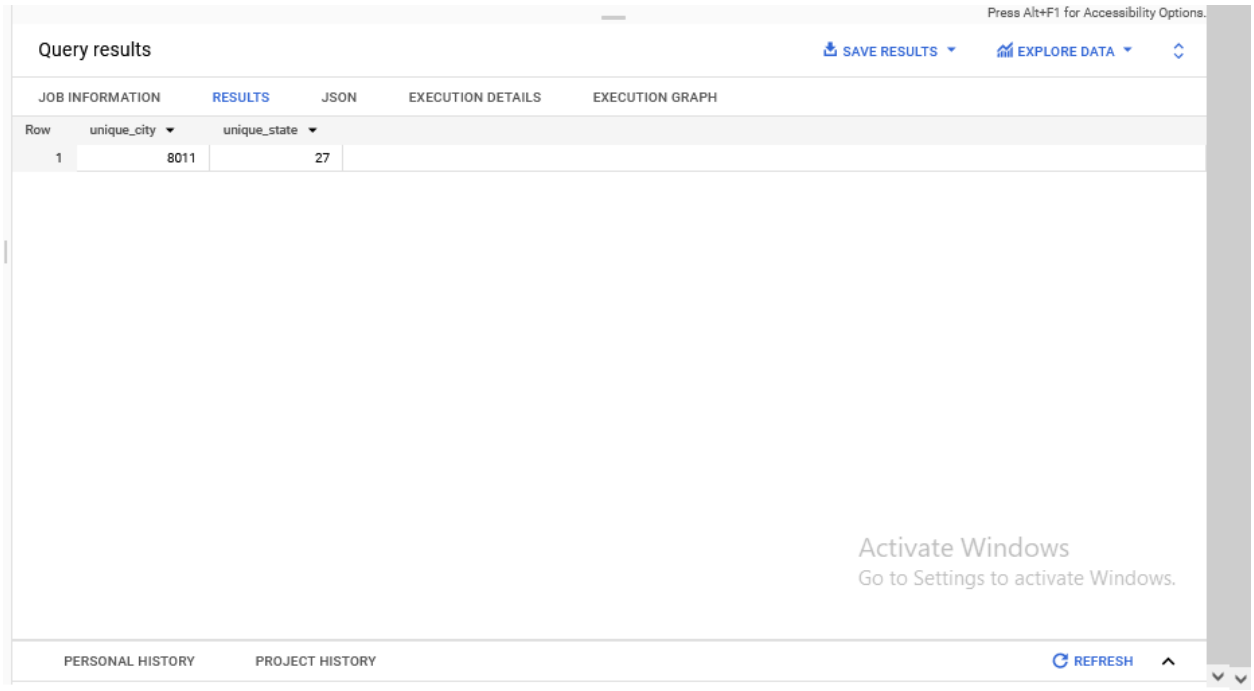
2) Get the time range between which the orders were placed.

```
SELECT
  min(order_approved_at) as first_order,
  max(order_approved_at) as last_order
  from `target.orders`
```



3)Count the number of Cities and States in our dataset.

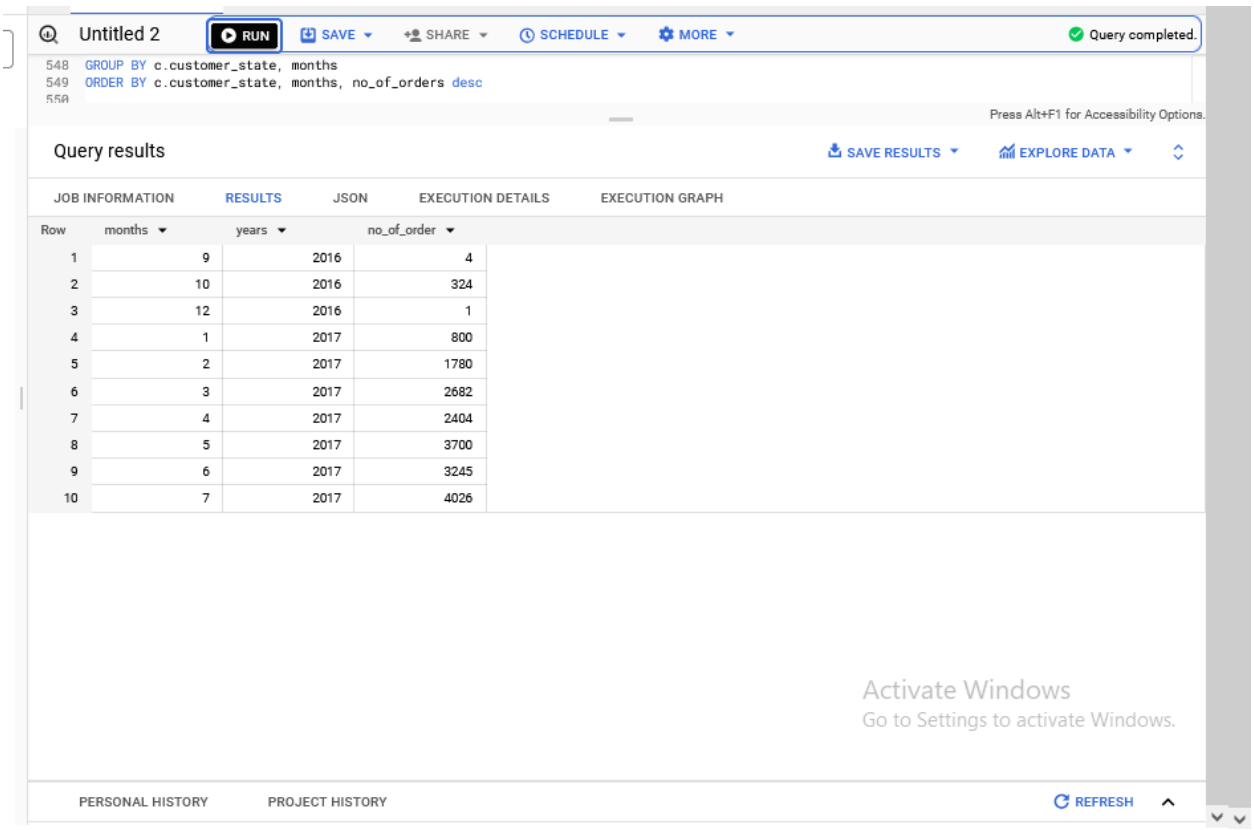
```
SELECT
 count(distinct geolocation_city) as unique_city,
 count(distinct geolocation_state) as unique_state
 from `target.geolocation
```



2) **In-depth Exploration:**

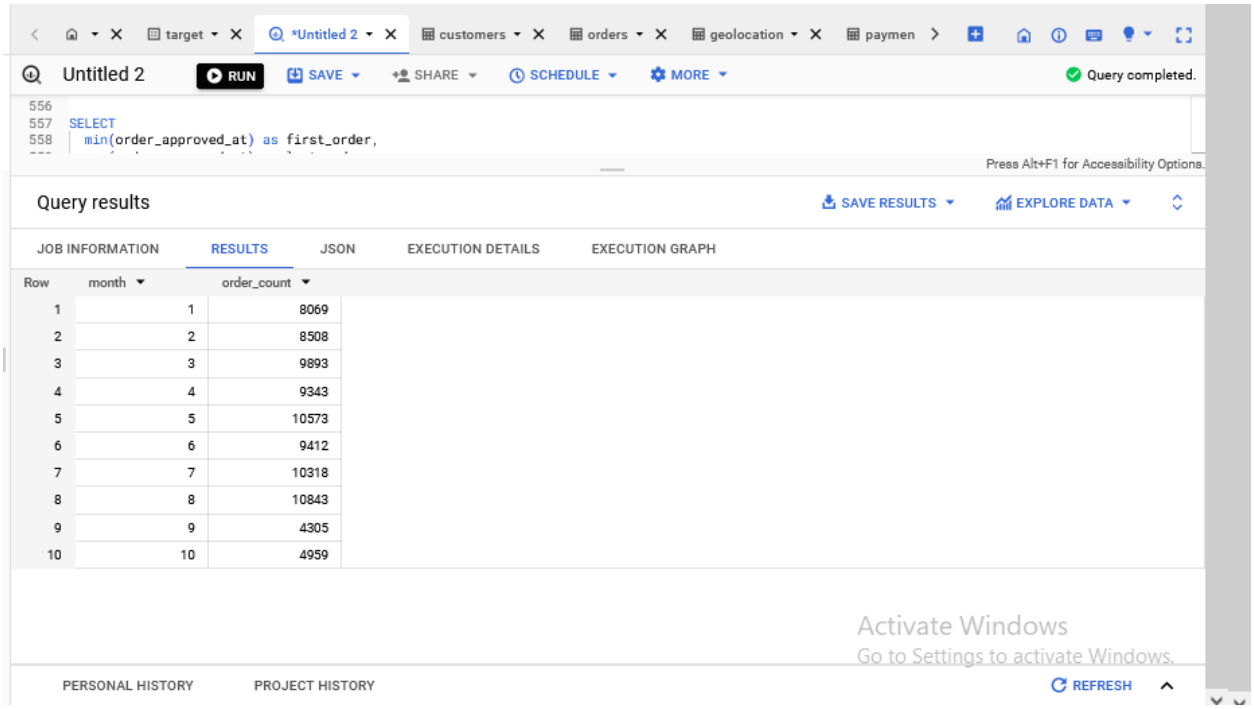
1) Is there a growing trend in the no. of orders placed over the past years?

```
SELECT
EXTRACT(month FROM order_purchase_timestamp) AS months,
EXTRACT(year FROM order_purchase_timestamp) AS years,
COUNT(DISTINCT order_id) AS no_of_order
FROM `target.orders`
GROUP BY months,years
ORDER BY years,months
LIMIT 10
```



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

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
COUNT(DISTINCT order_id) AS order_count,
FROM `target.orders`
GROUP BY month
ORDER BY month
LIMIT 10
```



Row	month	order_count
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959

3) 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 : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night
- SELECT
- COUNT(order_id) as no_of_orders,
- CASE
- WHEN
- EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 0 AND 5
- THEN 'Dawn'
- WHEN
- EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 6 AND 11
- THEN 'Mornings'
- WHEN
- EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 12 AND 17
- THEN 'Afternoon'
- WHEN
- EXTRACT(Hour FROM order_purchase_timestamp) BETWEEN 18 AND 23
- THEN 'Night'
- END AS time_period,
- FROM target.orders
- GROUP BY time_period
- ORDER BY no_of_orders

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	no_of_orders	time_period
1	4740	Dawn
2	22240	Mornings
3	34100	Night
4	38361	Afternoon

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

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

1) Get the month on month no. of orders placed in each state.

```
SELECT
c.customer_state as customer_state,
EXTRACT(month FROM o.order_purchase_timestamp) as months,
COUNT(o.order_id) as no_of_orders
FROM target.orders o
JOIN target.customers c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state, months
ORDER BY c.customer_state, months
Limit 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	months	no_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

2) How are the customers distributed across all the states?

```
SELECT
customer_state,
COUNT(customer_id) AS no_of_customers
FROM `target.customers`
GROUP BY customer_state
ORDER BY no_of_customers
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	no_of_customers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350

Load more

Results per page: 50 1 – 27 of 27

Activate Windows
Go to Settings to activate Windows.

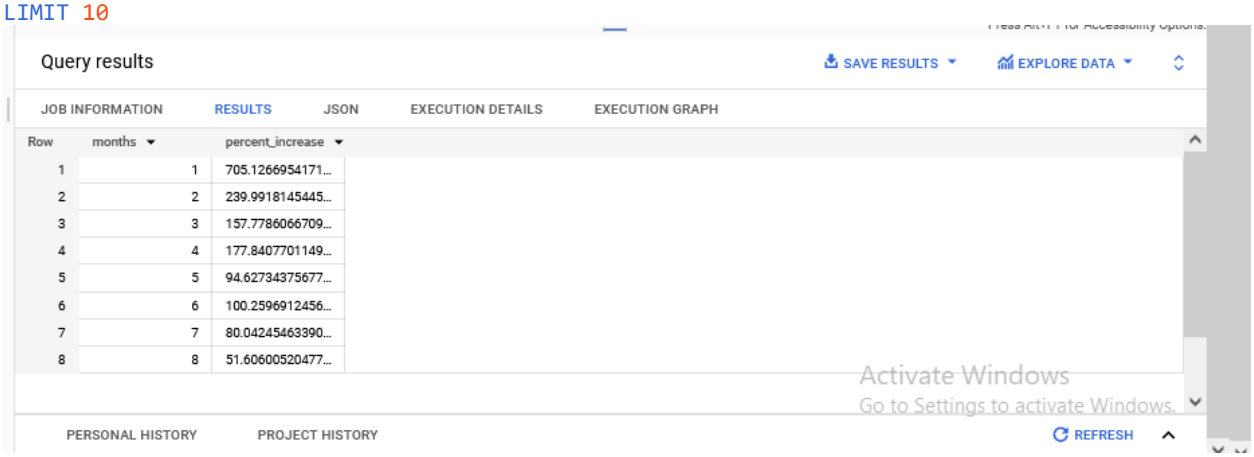
PERSONAL HISTORY PROJECT HISTORY REFRESH

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

1) 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.

```
SELECT
```

```
EXTRACT(MONTH FROM o.order_purchase_timestamp) as months,
(( SUM
(CASE
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 AND EXTRACT(MONTH FROM
o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value
END) - SUM(
CASE
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM
o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value END) ) / SUM(CASE
WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 AND EXTRACT(MONTH FROM
o.order_purchase_timestamp) BETWEEN 1 AND 8
THEN p.payment_value END) )*100 as percent_increase
FROM `target.orders` o
JOIN `target.payments` p ON o.order_id = p.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 months
ORDER BY months
LIMIT 10
```



Row	months	percent_increase
1	1	705.1266954171...
2	2	239.9918145445...
3	3	157.7786066709...
4	4	177.8407701149...
5	5	94.62734375677...
6	6	100.2596912456...
7	7	80.04245463390...
8	8	51.60600520477...

2) Calculate the Total & Average value of order price for each state.

```
SELECT
round(avg(o.price), 2) as average_price,
round(sum(o.price), 2) as total_price ,
c.customer_state as customer_state
from `target.order_items`as o
cross join `target.customers`as c
group by c.customer_state
order by c.customer_state, total_price
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	average_price	total_price	customer_state
1	120.65	1100923139.7	AC
2	120.65	5613348848.1	AL
3	120.65	2011563267.6	AM
4	120.65	924231771.6	AP
5	120.65	45939755705.81	BA
6	120.65	18158435983.23	CE
7	120.65	29086117518.01	DF
8	120.65	27631811642.12	ES
9	120.65	27455120274.02	GO
10	120.65	10152957843.91	MA

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

3) Calculate the Total & Average value of order freight for each state.

```
SELECT
round(avg(o.freight_value), 2) as average_freight_price,
round(sum(o.freight_value), 2) as total_freight_price,
c.customer_state as customer_state
from `target.order_items`as o
cross join `target.customers`as c
group by c.customer_state
order by c.customer_state, total_freight_price
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	average_freight_pric	total_freight_price	customer_state
1	19.99	182404672.74	AC
2	19.99	930038640.02	AL
3	19.99	333282611.92	AM
4	19.99	153129848.72	AP
5	19.99	7611454245.2	BA
6	19.99	3008551145.44	CE
7	19.99	4819086415.6	DF
8	19.99	4578132094.82	ES

Load more

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

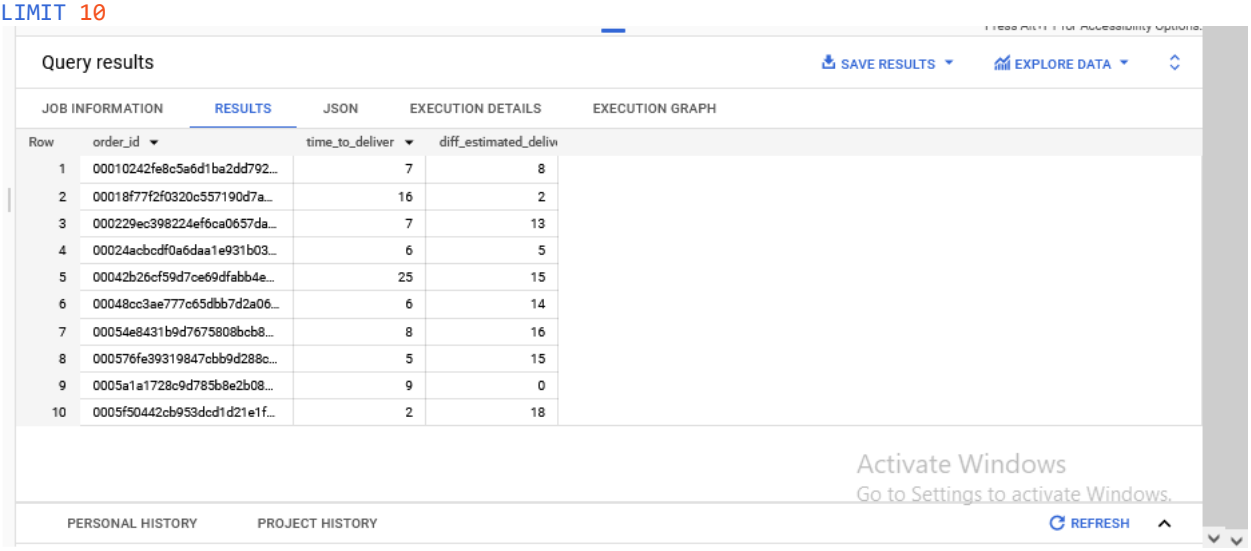
5) **Analysis based on sales, freight and delivery time.**

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_estimated_delivery_date - order_delivered_customer_date

```
SELECT
 order_id,
 date_diff(order_delivered_customer_date, order_purchase_timestamp, day) AS time_to_deliver,
 date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) AS
diff_estimated_delivery ,
from `target.orders`
where
date_diff(order_delivered_customer_date, order_purchase_timestamp, day) is not null and
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) is not null
order by  order_id, time_to_deliver, diff_estimated_delivery
LIMIT 10
```

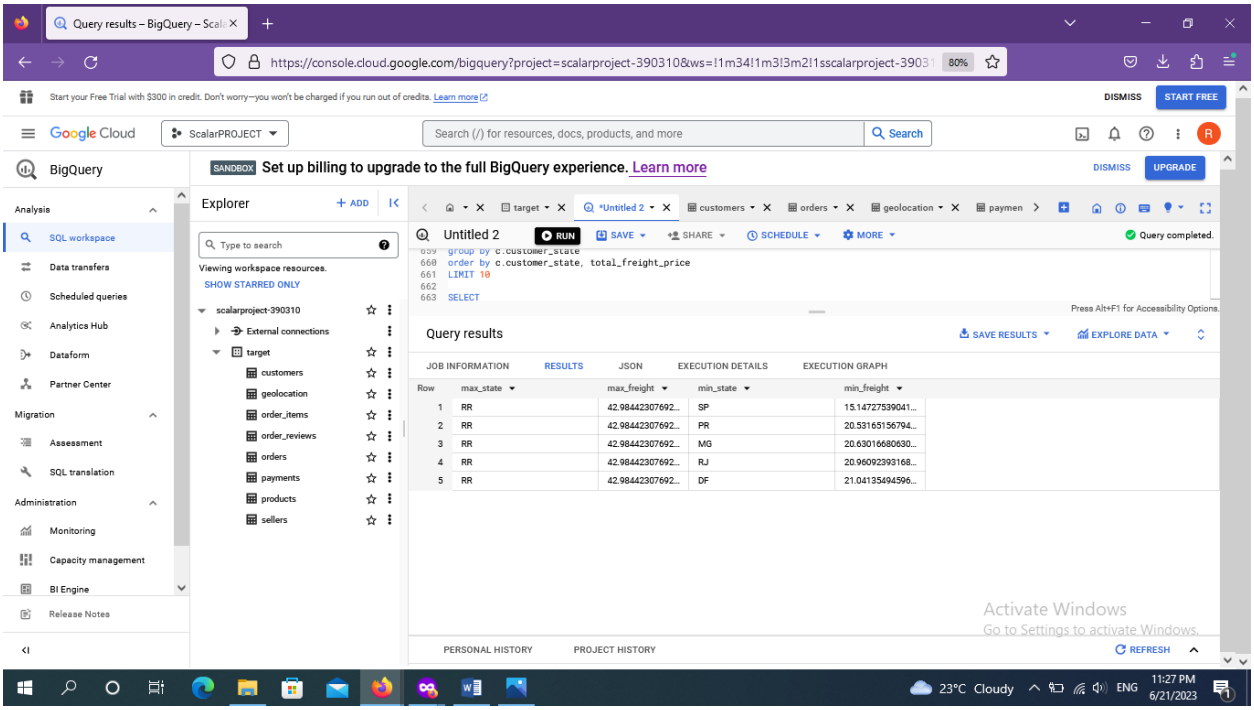


Row	order_id	time_to_deliver	diff_estimated_deliv
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

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

```
WITH sara as
(
SELECT c.customer_state as max_state, avg(ori.freight_value) AS max_freight
FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
INNER JOIN `target.order_items` ori ON o.order_id = ori.order_id
GROUP BY c.customer_state
ORDER BY avg(ori.freight_value) DESC
LIMIT 5),
lara as
(
SELECT c.customer_state as min_state, avg(ori.freight_value) AS min_freight
FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
INNER JOIN `target.order_items` ori ON o.order_id = ori.order_id
GROUP BY c.customer_state
```

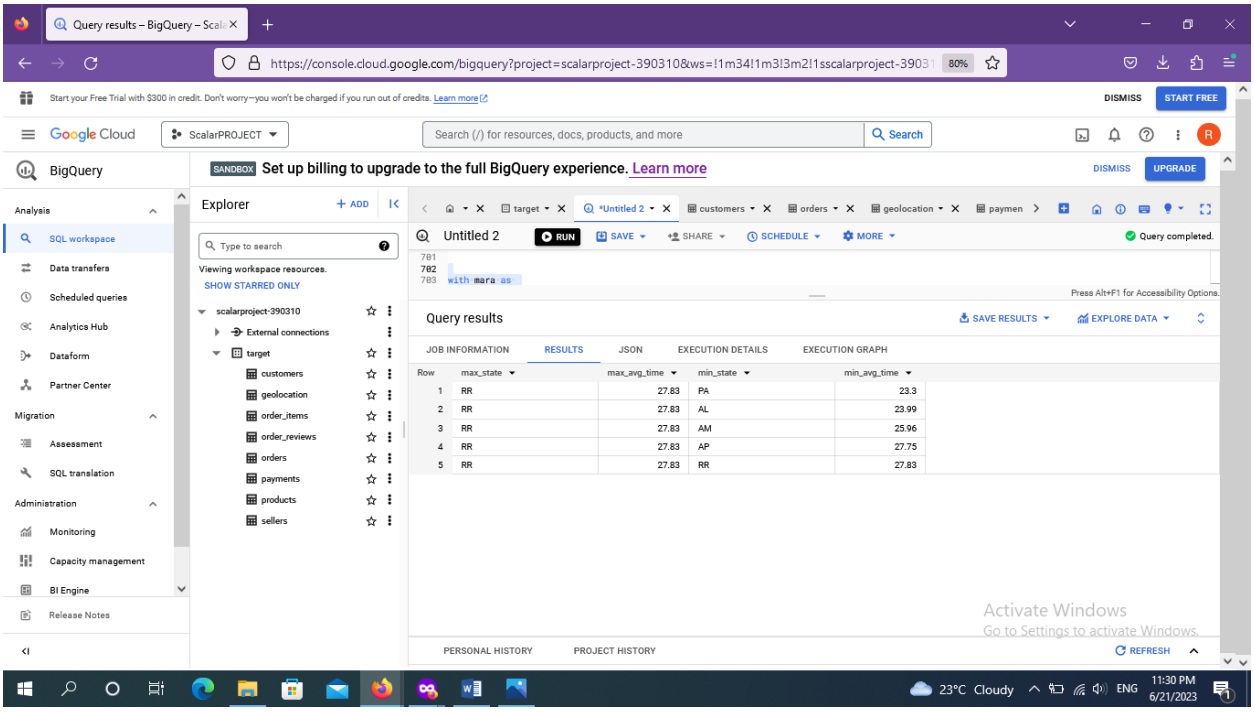
```
ORDER BY avg(ori.freight_value)
LIMIT 5
)
SELECT
s.max_state,
s.max_freight,
l.min_state,
l.min_freight
FROM sara as s
CROSS JOIN lara as l
ORDER BY s.max_freight desc, l.min_freight
LIMIT 5
```



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

```
with mara as
(
SELECT c.customer_state as max_state,
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)), 2) as max_avg_time
FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
INNER JOIN `target.order_items` ori ON o.order_id = ori.order_id
GROUP BY c.customer_state
ORDER BY ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)), 2)  desc
LIMIT 5
),dora as
(SELECT c.customer_state as min_state,
```

```
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)), 2) as min_avg_time
FROM `target.orders` o
INNER JOIN `target.customers` c ON o.customer_id = c.customer_id
INNER JOIN `target.order_items` ori ON o.order_id = ori.order_id
GROUP BY c.customer_state
ORDER BY ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)), 2)  desc
LIMIT 5)
SELECT
m.max_state,
m.max_avg_time,
d.min_state,
d.min_avg_time
from mara m,dora d
order by m.max_avg_time desc, d.min_avg_time
limit 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(order_delivered_customer_date, order_purchase_timestamp,
DAY)), 2)
  AS avg_time_to_delivery,
```

```
  ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
DAY)), 2)
  AS avg_diff_estimated_delivery,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)), 2)-ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)), 2) as compare_time
FROM
  `target.orders` o
JOIN
  `target.customers` c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY compare_time desc
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	customer_state	avg_time_to_delivery	avg_diff_estimated_(	compare_time
1	AL	24.04	7.95	16.09
2	RR	28.98	16.41	12.57
3	MA	21.12	8.77	12.35000000000...
4	SE	21.03	9.17	11.86000000000...
5	CE	20.82	9.96	10.86
6	PA	23.32	13.19	10.13
7	BA	18.87	9.93	8.940000000000...
8	PI	18.99	10.47	8.519999999999...
9	AP	26.73	18.73	8.0
10	PB	19.95	12.37	7.58

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

6) **Analysis based on the payments:**

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

```
SELECT
  p.payment_type as payment_type,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  COUNT(DISTINCT o.order_id) AS monthly_order
FROM`target.orders` as o
INNER JOIN `target.payments` p
ON o.order_id = p.order_id
GROUP BY
  payment_type, month
ORDER BY
  payment_type, month
```

```
730
731 SELECT
```

Press Alt+F1 for Accessibility Options.

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	payment_type	month	monthly_order
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6093

Results per page: 50 1 – 50 of 50

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

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

```
SELECT
  p.payment_installments,
  COUNT(o.order_id) AS no_of_count
FROM
  `target.orders` AS o
JOIN
  `target.payments` AS p
ON
  o.order_id = p.order_id
WHERE
 p.payment_installments != 0
GROUP BY p.payment_installments
ORDER BY no_of_count
LIMIT 10
```

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	payment_installment	no_of_count
1	23	1
2	22	1
3	21	3
4	16	5
5	17	8
6	14	15
7	13	16
8	20	17
9	24	18
10	11	23

Activate Windows
Go to Settings to activate Windows.

PERSONAL HISTORY PROJECT HISTORY REFRESH

7)Actionable Insights

1.During monthly seasonality of orders,'MAY', 'AUGUST' and 'SEPTEMBER' have the highest amount of orders placed, which also gives an oppurtunity to improve in other months.
2.During growing trend of no. of orders, we can observe that there is an increase of oreders from 2016 to 2018.
3)From time of the day, we get to know that customers opt to order during afternoon and night most of the times, so we should focus on the market during this time period and helpt improve their experience.
4)% percent on month on month orders placed on each state is on a decreasing trend, with January having the highest percent increase.
5)SP has the highest average in freight value, the market should enhance customer retaining and create brand loyalty.
6)PA has highest average in freight value and value should be sustainable over the coming years.

Recommendations

1) Implement customer retention strategies to encourage repeat purchases especially in RR state where they have lowest average freight value and lowest customer average value.
2) Mornings and dawn are the timings where customers ordering less, we should attract more customers during this time frame to improve the finances.
3) RR also has the lowest customers distributed per state, focusing on improving the advertising and offering discounts may attract new customers.
4) Enhance the customer service experience by offering chat support services and ensuring prompt and effective responses to customer inquiries.
5) Improving the mode of payment is also huge bonus as it helps in increasing the options of payments and lessen the hassle for money.