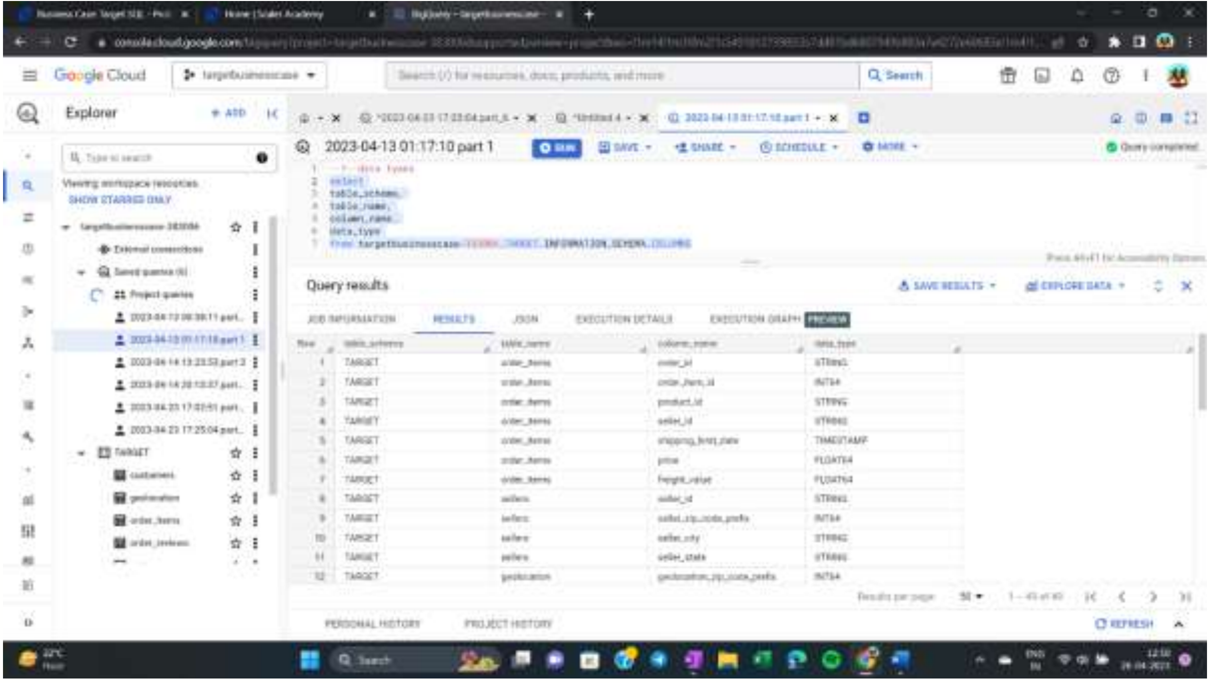
1.

```
--1--data types
select
table_schema,
table_name,
column_name,
data_type
from targetbusinesscase-383006.TARGET.INFORMATION_SCHEMA.COLUMNS
```



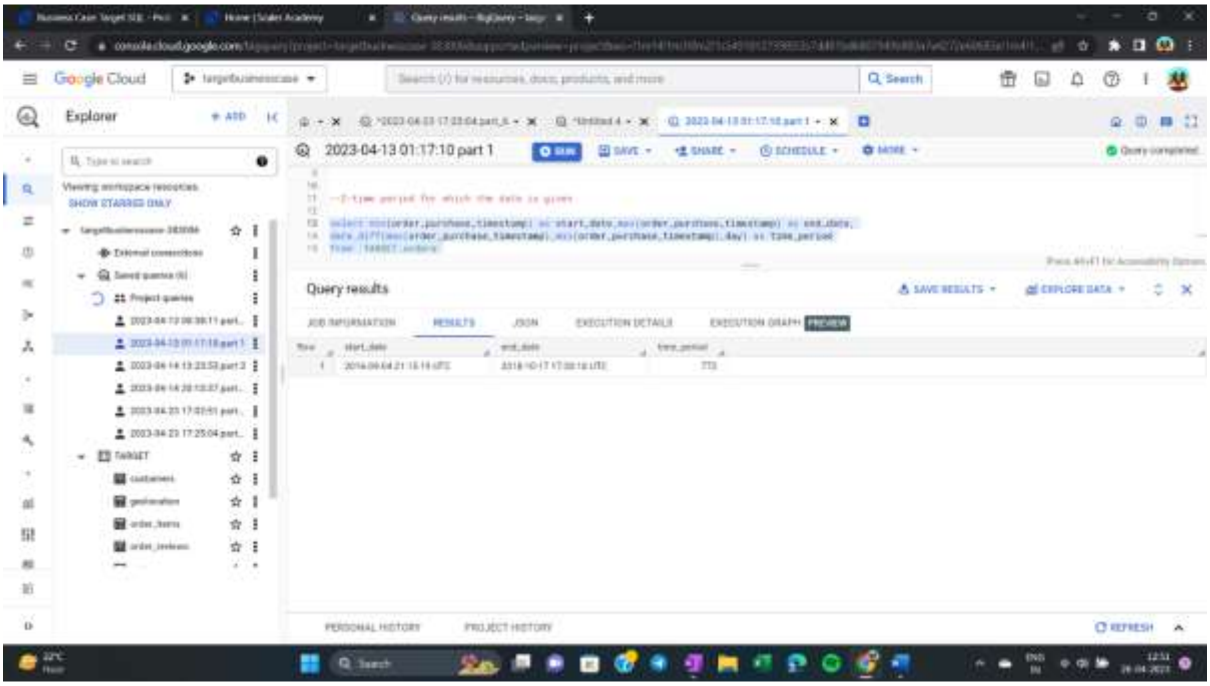
2023-04-13 01:17:10 part 1

```
--1--data types
select
table_schema,
table_name,
column_name,
data_type
from targetbusinesscase-383006.TARGET.INFORMATION_SCHEMA.COLUMNS
```

Query results

| Row | table_schema | table_name | column_name | data_type |
|---|---|---|---|---|
| 1 | TARGET | order_items | order_id | STRING |
| 2 | TARGET | order_items | order_item_id | INT64 |
| 3 | TARGET | order_items | product_id | STRING |
| 4 | TARGET | order_items | seller_id | STRING |
| 5 | TARGET | order_items | shipping_limit_date | TIMESTAMP |
| 6 | TARGET | order_items | price | FLOAT64 |
| 7 | TARGET | order_items | freight_value | FLOAT64 |
| 8 | TARGET | sellers | seller_id | STRING |
| 9 | TARGET | sellers | seller_zip_code_prefix | INT64 |
| 10 | TARGET | sellers | seller_city | STRING |
| 11 | TARGET | sellers | seller_state | STRING |
| 12 | TARGET | geolocation | geolocation_zip_code_prefix | INT64 |

```
--2-time period for which the data is given

select min(order_purchase_timestamp) as start_date,max(order_purchase_timestamp) as
 end_date,
date_diff(max(order_purchase_timestamp),min(order_purchase_timestamp),day) as time_
period
from `TARGET.orders`
```



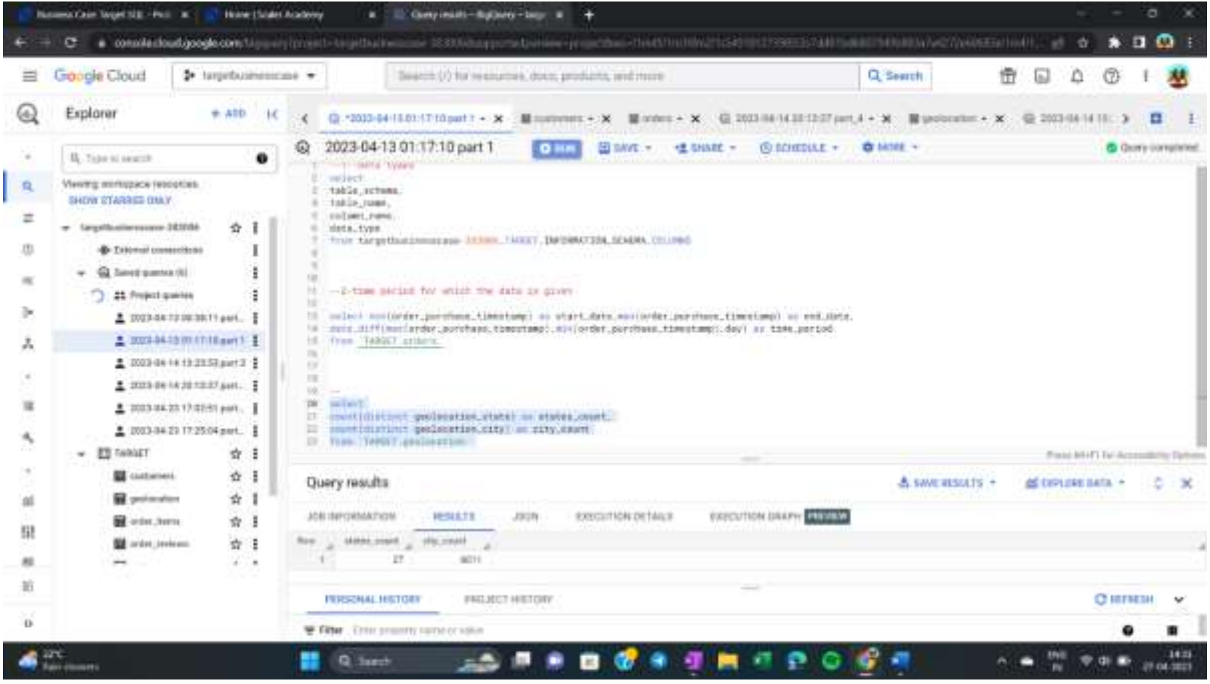
2023-04-13 01:17:10 part 1

```
--2-time period for which the data is given

select min(order_purchase_timestamp) as start_date,max(order_purchase_timestamp) as end_date,
date_diff(max(order_purchase_timestamp),min(order_purchase_timestamp),day) as time_period
from `TARGET.orders`
```

Query results

| Row | start_date | end_date | time_period |
|---|---|---|---|
| 1 | 2016-09-04 21:15:19 UTC | 2018-10-17 17:30:18 UTC | 773 |

```
--3-count of states and cities presnent.
select
count(distinct geolocation_state) as states_count,
count(distinct geolocation_city) as city_count
from `TARGET.geolocation`
```



2023-04-13 01:17:10 part 1

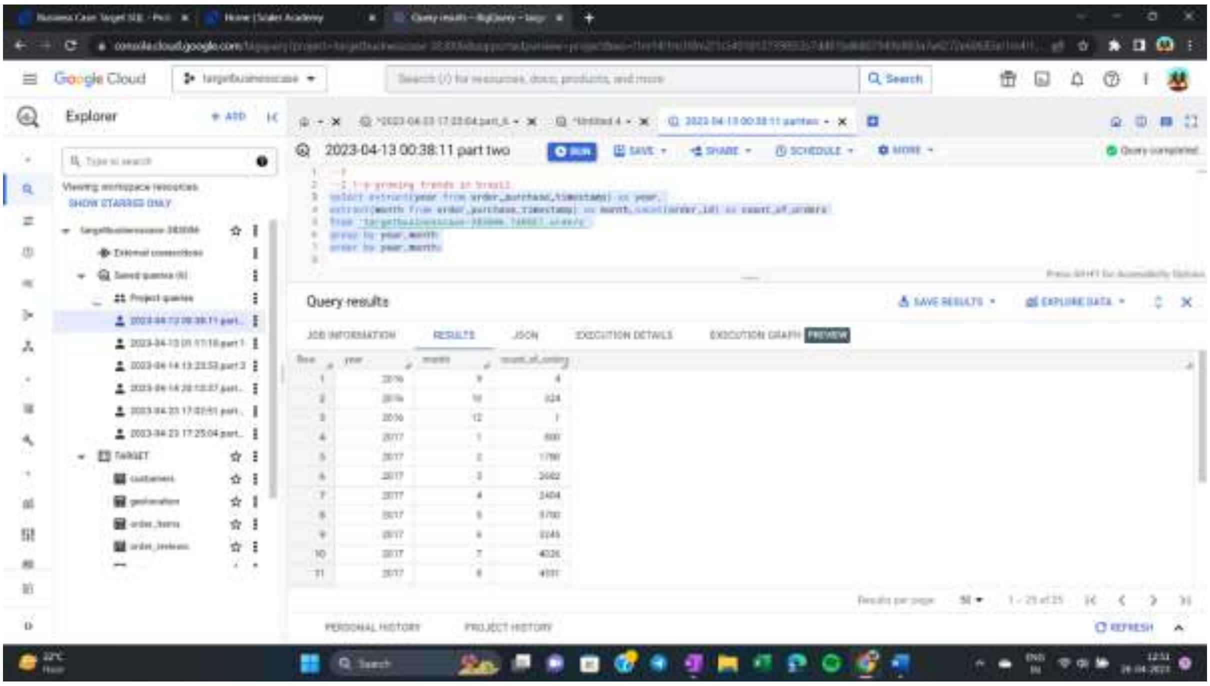
```
--1-data types
select
table_schema,
table_name,
column_name,
data_type
from targetbusinesscase-383006.TARGET.INFORMATION_SCHEMA.COLUMNS



--2-time period for which the data is given

select min(order_purchase_timestamp) as start_date,max(order_purchase_timestamp) as end_date,
date_diff(max(order_purchase_timestamp),min(order_purchase_timestamp),day) as time_period
from `TARGET.orders`



--
select
count(distinct geolocation_state) as states_count,
count(distinct geolocation_city) as city_count
from `TARGET.geolocation`
```

Query results

| Row | states_count | city_count |
|---|---|---|
| 1 | 27 | 8011 |

## 2.

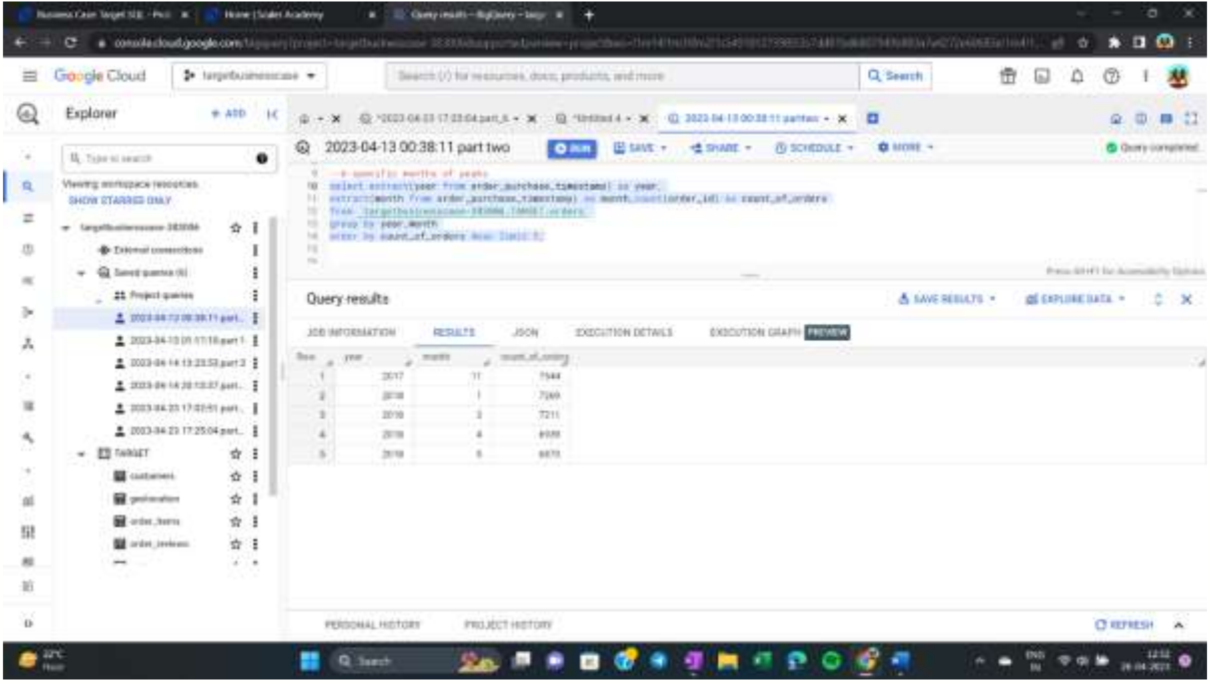
```
--2.1-a-growing trends in brazil
select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,count(order_id) as count_of_orders
from `targetbusinesscase-383006.TARGET.orders`
group by year,month
order by year,month;
```



```
--1
--2.1-a-growing trends in brazil
select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,count(order_id) as count_of_orders
from `targetbusinesscase-383006.TARGET.orders`
group by year,month
order by year,month;
```

| Row | year | month | count_of_orders |
|---|---|---|---|
| 1 | 2016 | 9 | 4 |
| 2 | 2016 | 10 | 324 |
| 3 | 2016 | 12 | 1 |
| 4 | 2017 | 1 | 800 |
| 5 | 2017 | 2 | 1780 |
| 6 | 2017 | 3 | 2682 |
| 7 | 2017 | 4 | 2404 |
| 8 | 2017 | 5 | 3700 |
| 9 | 2017 | 6 | 3245 |
| 10 | 2017 | 7 | 4026 |
| 11 | 2017 | 8 | 4331 |

```
--b-specific months of peaks
select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,count(order_id) as count_of_orders
from `targetbusinesscase-383006.TARGET.orders`
group by year,month
order by count_of_orders desc limit 5;
```

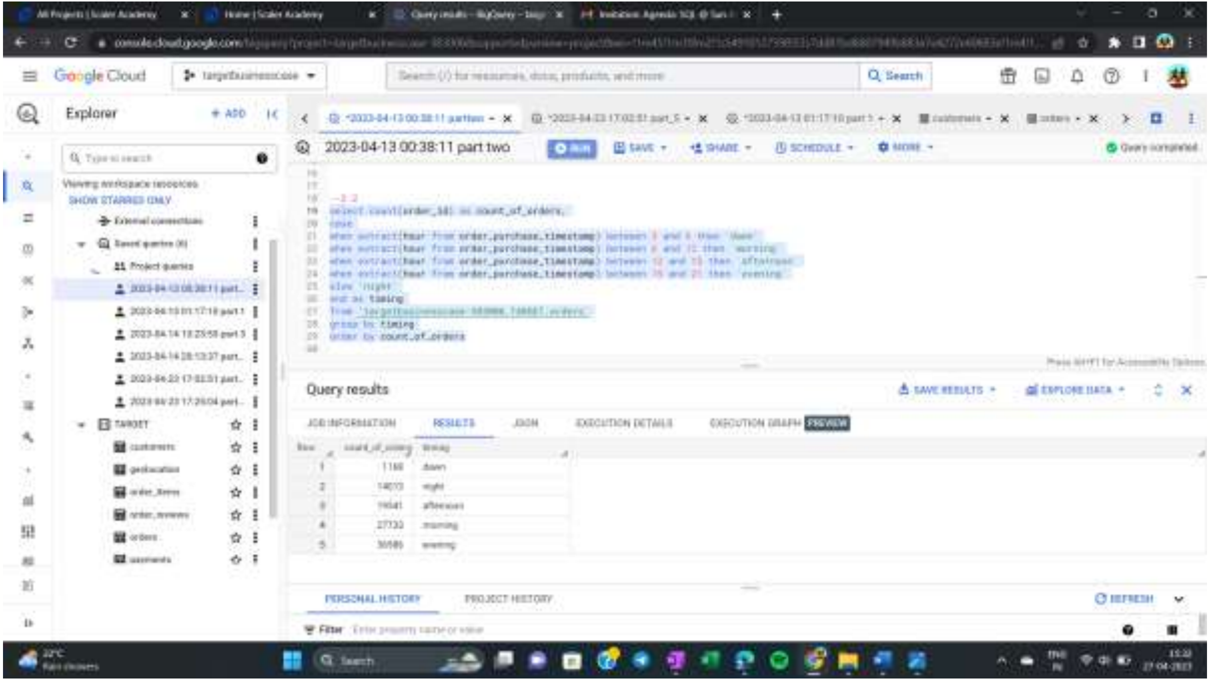


```
--b-specific months of peaks
select extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,count(order_id) as count_of_orders
from `targetbusinesscase-383006.TARGET.orders`
group by year,month
order by count_of_orders desc limit 5;
```

| Row | year | month | count_of_orders |
|---|---|---|---|
| 1 | 2017 | 11 | 7544 |
| 2 | 2018 | 1 | 7269 |
| 3 | 2018 | 3 | 7211 |
| 4 | 2018 | 4 | 6939 |
| 5 | 2018 | 5 | 6873 |

```
--2.2-Different timings of a day
select count(order_id) as count_of_orders,
case
when extract(hour from order_purchase_timestamp) between 3 and 6 then 'dawn'
when extract(hour from order_purchase_timestamp) between 6 and 12 then 'morning'
when extract(hour from order_purchase_timestamp) between 12 and 15 then 'afternoon'
when extract(hour from order_purchase_timestamp) between 15 and 21 then 'evening'
```

```
else 'night'
end as timing
from `targetbusinesscase-383006.TARGET.orders`
group by timing
order by 1 desc
```



2023-04-13 00:38:11 part two

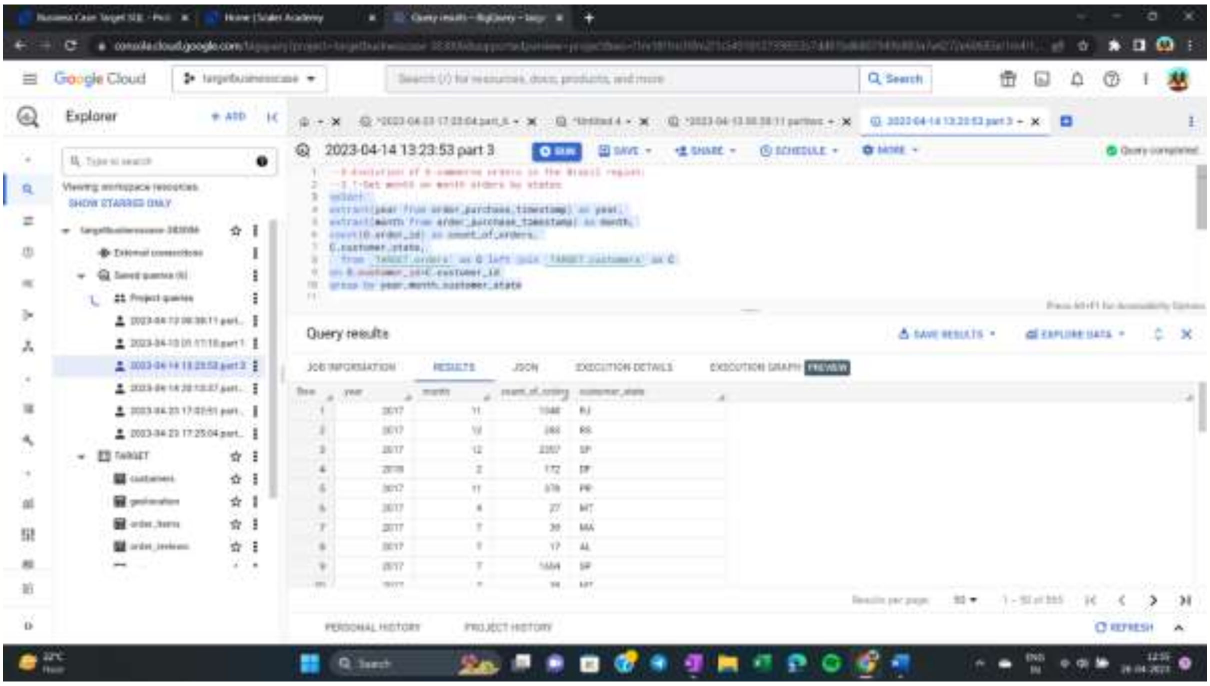
```
--3.2
select count(order_id) as count_of_orders,
case
when extract(hour from order_purchase_timestamp) between 0 and 6 then 'dawn'
when extract(hour from order_purchase_timestamp) between 6 and 11 then 'morning'
when extract(hour from order_purchase_timestamp) between 12 and 15 then 'afternoon'
when extract(hour from order_purchase_timestamp) between 16 and 21 then 'evening'
else 'night'
end as timing
from `targetbusinesscase-383006.TARGET.orders`
group by timing
order by count_of_orders
```

Query results

| Row | count_of_orders | timing |
|---|---|---|
| 1 | 1168 | dawn |
| 2 | 14073 | night |
| 3 | 19341 | afternoon |
| 4 | 27733 | morning |
| 5 | 30585 | evening |

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

```
--3.1-Get month on month orders by states
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(O.order_id) as count_of_orders,
C.customer_state,
  from `TARGET.orders` as O left join `TARGET.customers` as C
on O.customer_id=C.customer_id
group by year,month,customer_state
```



2023-04-14 13:23:53 part 3

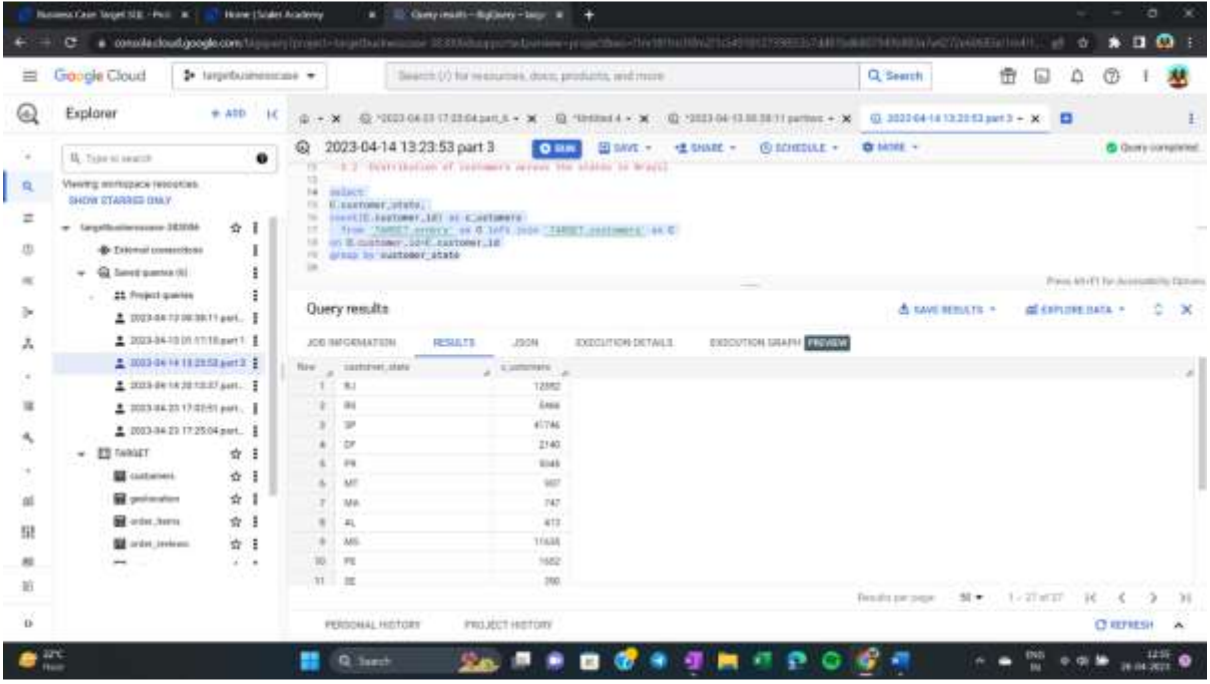
```
--3-Evolution of E-commerce orders in the Brazil region:
--3.1-Get month on month orders by states
select
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month,
count(O.order_id) as count_of_orders,
C.customer_state,
  from `TARGET.orders` as O left join `TARGET.customers` as C
on O.customer_id=C.customer_id
group by year,month,customer_state
```

Query results

| Row | year | month | count_of_orders | customer_state |
|---|---|---|---|---|
| 1 | 2017 | 11 | 1048 | RJ |
| 2 | 2017 | 12 | 288 | RS |
| 3 | 2017 | 12 | 2357 | SP |
| 4 | 2018 | 2 | 172 | DF |
| 5 | 2017 | 11 | 378 | PR |
| 6 | 2017 | 4 | 27 | MT |
| 7 | 2017 | 7 | 39 | MA |
| 8 | 2017 | 7 | 17 | AL |
| 9 | 2017 | 7 | 1664 | SP |

--3.2--Distribution of customers across the states in Brazil

```
select
C.customer_state,
count(C.customer_id) as c_ustomers
  from `TARGET.orders` as O left join `TARGET.customers` as C
on O.customer_id=C.customer_id
group by customer_state
```



2023-04-14 13:23:53 part 3

Query results

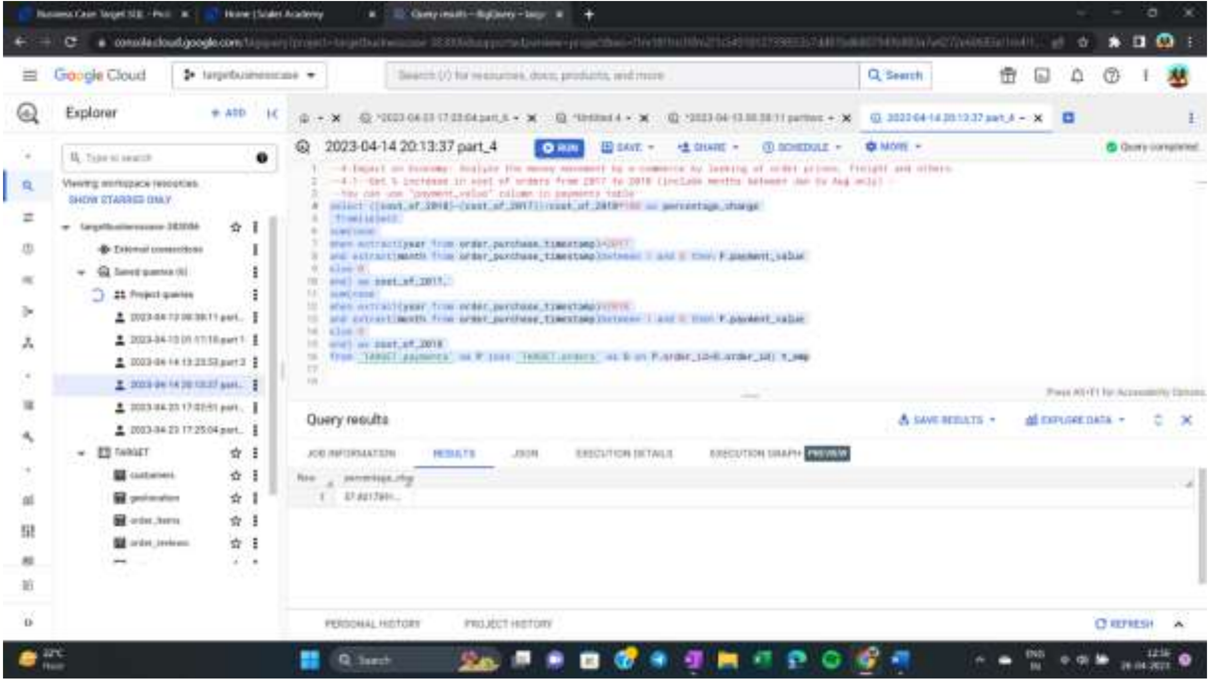
| Row | customer_state | c_ustomers |
|---|---|---|
| 1 | RJ | 12852 |
| 2 | RS | 5466 |
| 3 | SP | 41746 |
| 4 | DF | 2140 |
| 5 | PR | 5045 |
| 6 | MT | 907 |
| 7 | MA | 747 |
| 8 | AL | 413 |
| 9 | MG | 11635 |
| 10 | PE | 1652 |
| 11 | SE | 350 |

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

--4.1--

Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) -

```
select ((cost_of_2018)-(cost_of_2017))/cost_of_2018*100 as percentage_change
 from(select
sum(case
when extract(year from order_purchase_timestamp)=2017
and extract(month from order_purchase_timestamp)between 1 and 8 then P.payment_valu
e
else 0
end) as cost_of_2017,
sum(case
when extract(year from order_purchase_timestamp)=2018
and extract(month from order_purchase_timestamp)between 1 and 8 then P.payment_valu
e
else 0
end) as cost_of_2018
from `TARGET.payments` as P join `TARGET.orders` as O on P.order_id=O.order_id) t_e
mp
```



2023-04-14 20:13:37 part_4

```
--4.Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
--4.1- 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
```

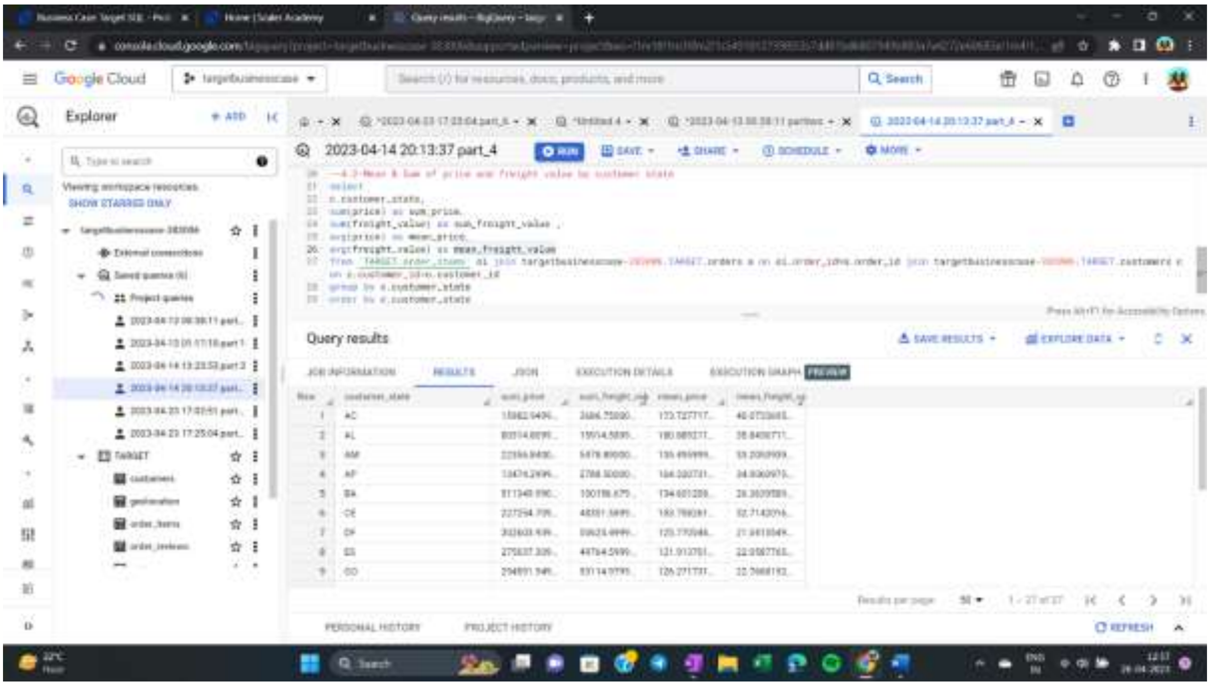
```
select ((cost_of_2018)-(cost_of_2017))/cost_of_2018*100 as percentage_change
 from(select
sum(case
when extract(year from order_purchase_timestamp)=2017
and extract(month from order_purchase_timestamp)between 1 and 8 then P.payment_value
else 0
end) as cost_of_2017,
sum(case
when extract(year from order_purchase_timestamp)=2018
and extract(month from order_purchase_timestamp)between 1 and 8 then P.payment_value
else 0
end) as cost_of_2018
from `TARGET.payments` as P join `TARGET.orders` as O on P.order_id=O.order_id) t_emp
```

Query results

| Row | percentage_chg |
|---|---|
| 1 | 37.021789... |

--4.2-Mean & Sum of price and freight value by customer state

```
Select

c.customer_state,

sum(price) as sum_price,

sum(freight_value) as sum_freight_value ,
avg(price) as mean_price,
avg(freight_value) as mean_freight_value
from `TARGET.order_items` oi join targetbusinesscase-
383006.TARGET.orders o on oi.order_id=o.order_id join targetbusinesscase-
383006.TARGET.customers c on c.customer_id=o.customer_id
group by c.customer_state
order by c.customer_state
```



2023-04-14 20:13:37 part_4

```
--4.2-Mean & Sum of price and freight value by customer state
select
c.customer_state,
sum(price) as sum_price,
sum(freight_value) as sum_freight_value ,
avg(price) as mean_price,
avg(freight_value) as mean_freight_value
from `TARGET.order_items` oi join targetbusinesscase-[illegible].TARGET.orders o on oi.order_id=o.order_id join targetbusinesscase-[illegible].TARGET.customers c
on o.customer_id=c.customer_id
group by c.customer_state
order by c.customer_state
```

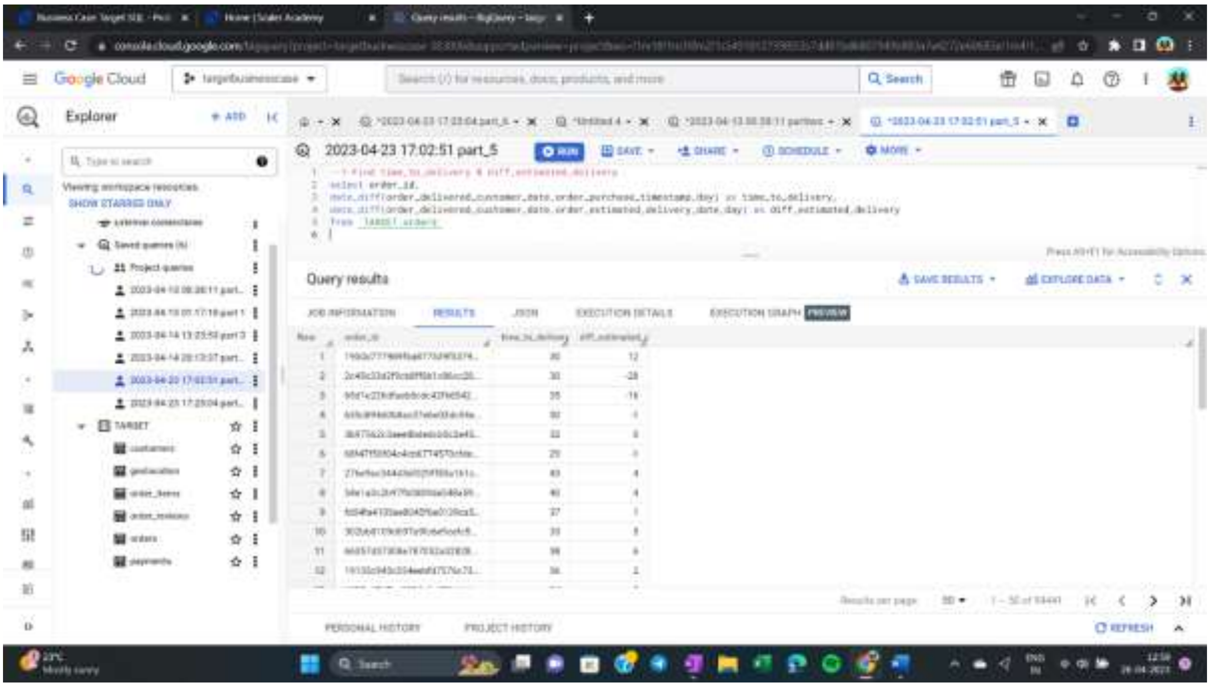
Query results

| Row | customer_state | sum_price | sum_freight_value | mean_price | mean_freight_value |
|---|---|---|---|---|---|
| 1 | AC | 15982.9499... | 3686.75000... | 173.727717... | 40.0733695... |
| 2 | AL | 80314.8099... | 15914.5899... | 180.889211... | 35.8436711... |
| 3 | AM | 22356.8400... | 5478.89000... | 135.495999... | 33.2053939... |
| 4 | AP | 13474.2999... | 2788.50000... | 164.320731... | 34.0060975... |
| 5 | BA | 511349.990... | 100156.679... | 134.601208... | 26.3639589... |
| 6 | CE | 227254.709... | 48351.5899... | 153.758261... | 32.7142016... |
| 7 | DF | 302603.939... | 50625.4999... | 125.770548... | 21.0413549... |
| 8 | ES | 275037.309... | 49764.5999... | 121.913701... | 22.0587765... |
| 9 | GO | 294591.949... | 53114.9799... | 126.271731... | 22.7668152... |

Results per page 50 ▾ 1 – 27 of 27

## 5 Analysis on sales, freight and delivery time

```
--1-Find time_to_delivery & diff_estimated_delivery
select order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as diff_estimated_delivery
from `TARGET.orders`
```



2023-04-23 17:02:51 part_5

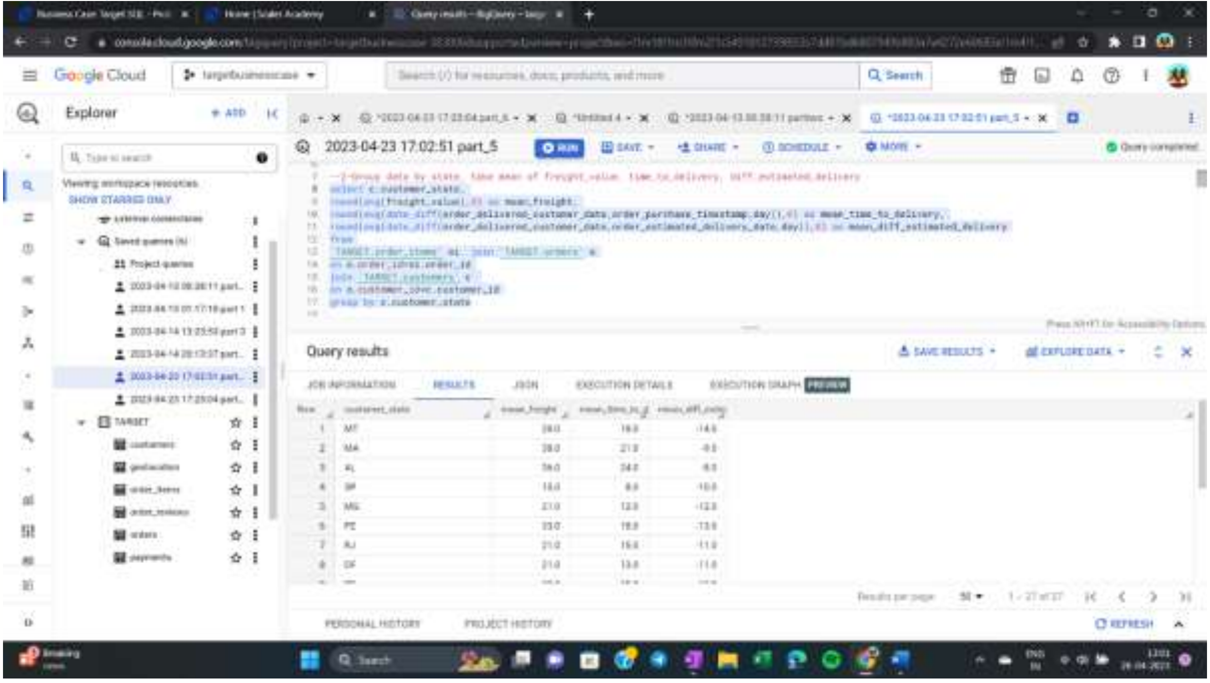
```
--1-Find time_to_delivery & diff_estimated_delivery
select order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as diff_estimated_delivery
from `TARGET.orders`
```

Query results

| Row | order_id | time_to_delivery | diff_estimated_d |
|---|---|---|---|
| 1 | 1950d777989f6a877539f5379... | 30 | 12 |
| 2 | 2c45c33d2f9cb8ff8b1c86cc28... | 30 | -28 |
| 3 | 65d1e226dfaeb8cdc42f66542... | 35 | -16 |
| 4 | 635c894d068ac37e6e03dc54e... | 30 | -1 |
| 5 | 3b97562c3aee8bdedcb5c2e45... | 32 | 0 |
| 6 | 68f47f50f04c4cb6774570cfde... | 29 | -1 |
| 7 | 276e9ec344d3bf029ff83a161c... | 43 | 4 |
| 8 | 54e1a3c2b97fb0809da548a59... | 40 | 4 |
| 9 | fd04fa4105ee8045f6a0139ca5... | 37 | 1 |
| 10 | 302bb8109d097a9fc6e9cefc5... | 33 | 5 |
| 11 | 66057d37308e787052a32828... | 38 | 6 |
| 12 | 19135c945c554eebfd7576c73... | 36 | 2 |

Results per page 50 ▾ 1 – 50 of 96441

```
--2-
Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
select c.customer_state,
round(avg(freight_value),0) as mean_freight,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),0)
 as mean_time_to_delivery,
round(avg(date_diff(order_delivered_customer_date,order_estimated_delivery_date,day
)),0) as mean_diff_estimated_delivery
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
```



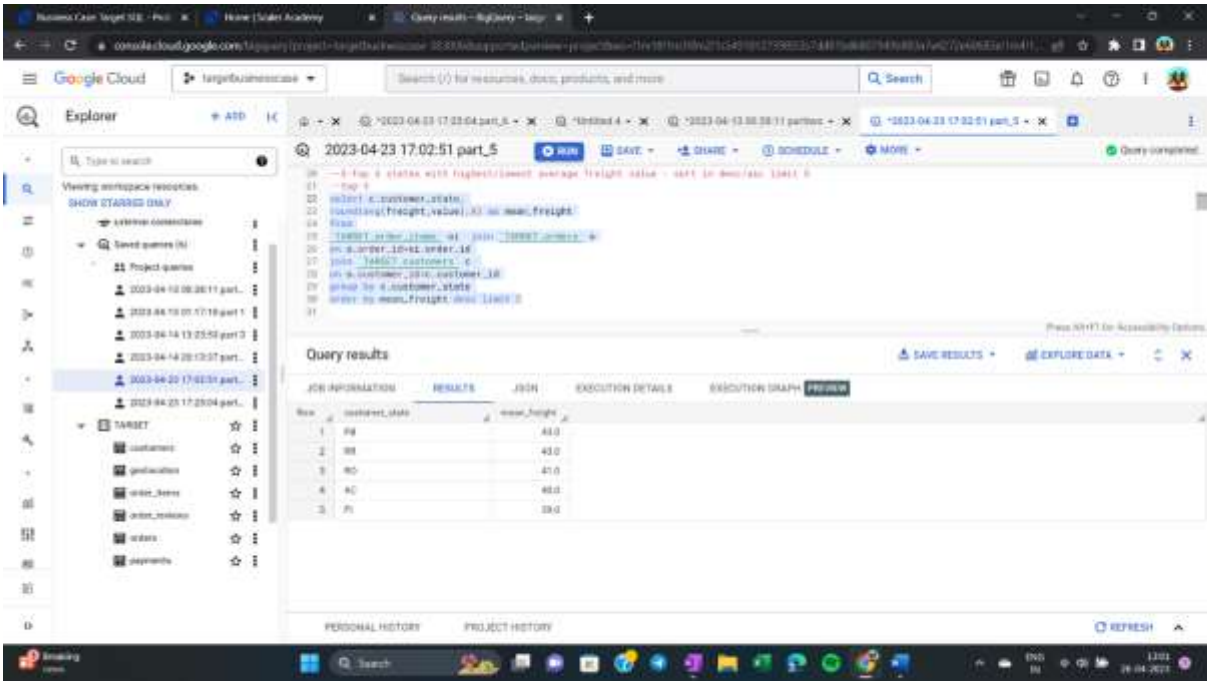
```
--2-Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
select c.customer_state,
round(avg(freight_value),0) as mean_freight,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),0) as mean_time_to_delivery,
round(avg(date_diff(order_delivered_customer_date,order_estimated_delivery_date,day)),0) as mean_diff_estimated_delivery
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
```

Query results

| Row | customer_state | mean_freight | mean_time_to_d | mean_diff_estim |
|---|---|---|---|---|
| 1 | MT | 28.0 | 19.0 | -14.0 |
| 2 | MA | 28.0 | 21.0 | -9.0 |
| 3 | AL | 36.0 | 24.0 | -8.0 |
| 4 | SP | 15.0 | 8.0 | -10.0 |
| 5 | MG | 21.0 | 12.0 | -12.0 |
| 6 | PE | 33.0 | 18.0 | -13.0 |
| 7 | RJ | 21.0 | 15.0 | -11.0 |
| 8 | DF | 21.0 | 13.0 | -11.0 |

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

```
with A as

(
select c.customer_state,
round(avg(freight_value),0) as mean_freight
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by c.customer_state

)
--Top 5 states with highest average freight value
Select A.customer_state, A.mean_freight
from A
order by mean_freight desc limit 5;
```



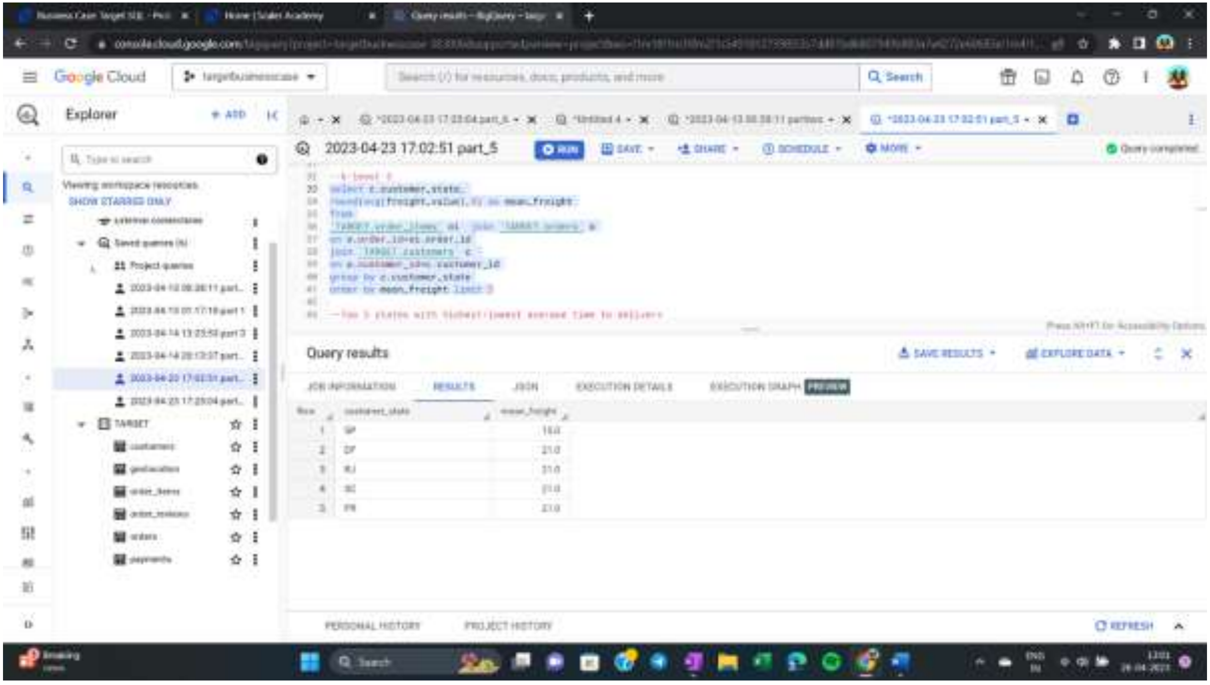
2023-04-23 17:02:51 part_5

```
--5 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
--top 5
select c.customer_state,
round(avg(freight_value),0) as mean_freight
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by mean_freight desc limit 5
```

Query results

| Row | customer_state | mean_freight |
|---|---|---|
| 1 | PB | 43.0 |
| 2 | RR | 43.0 |
| 3 | RO | 41.0 |
| 4 | AC | 40.0 |
| 5 | PI | 39.0 |

```
--Top 5 states with lowest average freight value
Select A.customer_state, A.mean_freight
from A
order by 2
limit 5;
```



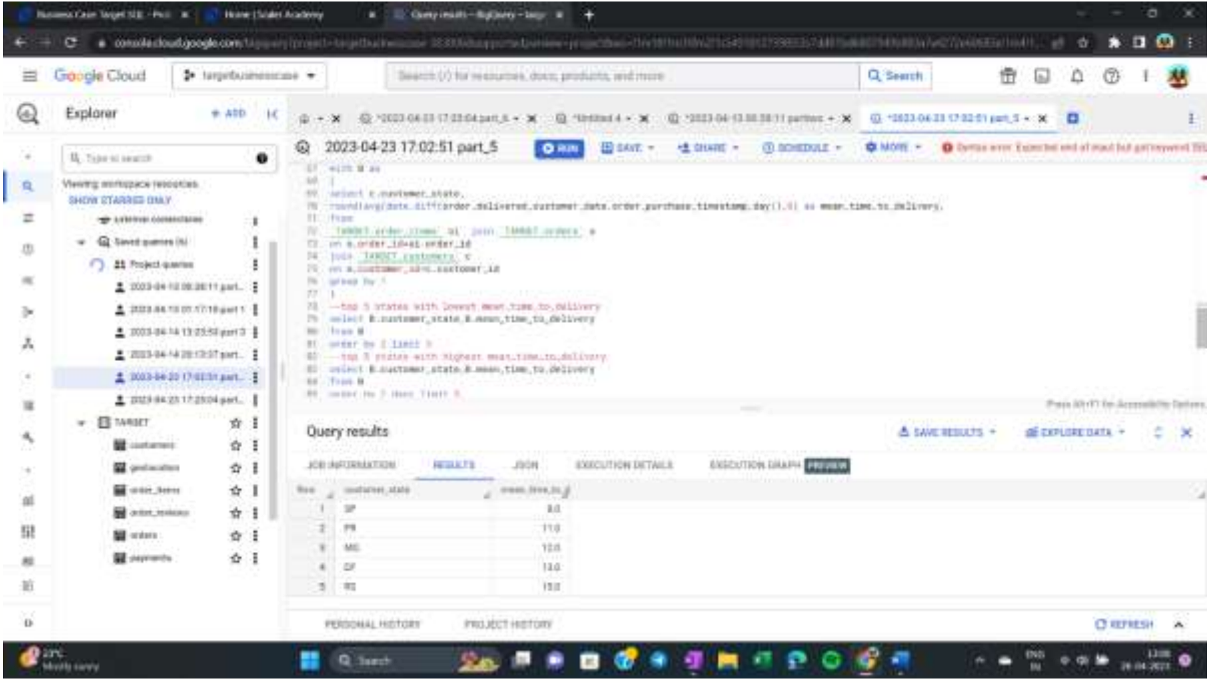
2023-04-23 17:02:51 part_5

```
--5 lowest 5
select c.customer_state,
round(avg(freight_value),0) as mean_freight
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by c.customer_state
order by mean_freight limit 5

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

Query results

| Row | customer_state | mean_freight |
|---|---|---|
| 1 | SP | 15.0 |
| 2 | DF | 21.0 |
| 3 | RJ | 21.0 |
| 4 | SC | 21.0 |
| 5 | PR | 21.0 |

```
--5 states with lowest/highest mean_time_to_delivery
with B as
(
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),0)
 as mean_time_to_delivery,
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by 1
)
--top 5 states with lowest mean_time_to_delivery
select B.customer_state,B.mean_time_to_delivery
from B
order by 2 limit 5
```

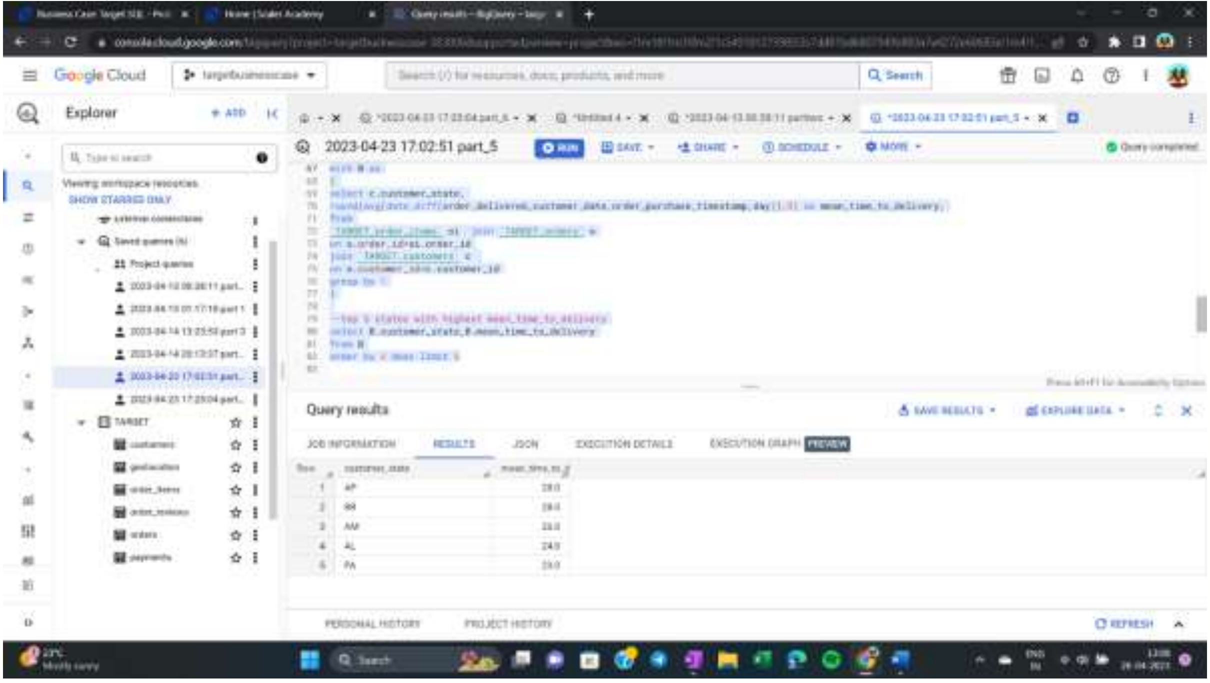


2023-04-23 17:02:51 part_5

Query results

| Row | customer_state | mean_time_to_... |
|---|---|---|
| 1 | SP | 8.0 |
| 2 | PR | 11.0 |
| 3 | MG | 12.0 |
| 4 | DF | 13.0 |
| 5 | RS | 15.0 |

```
--top 5 states with highest mean_time_to_delivery
select B.customer_state,B.mean_time_to_delivery
from B
order by 2 desc limit 5
```



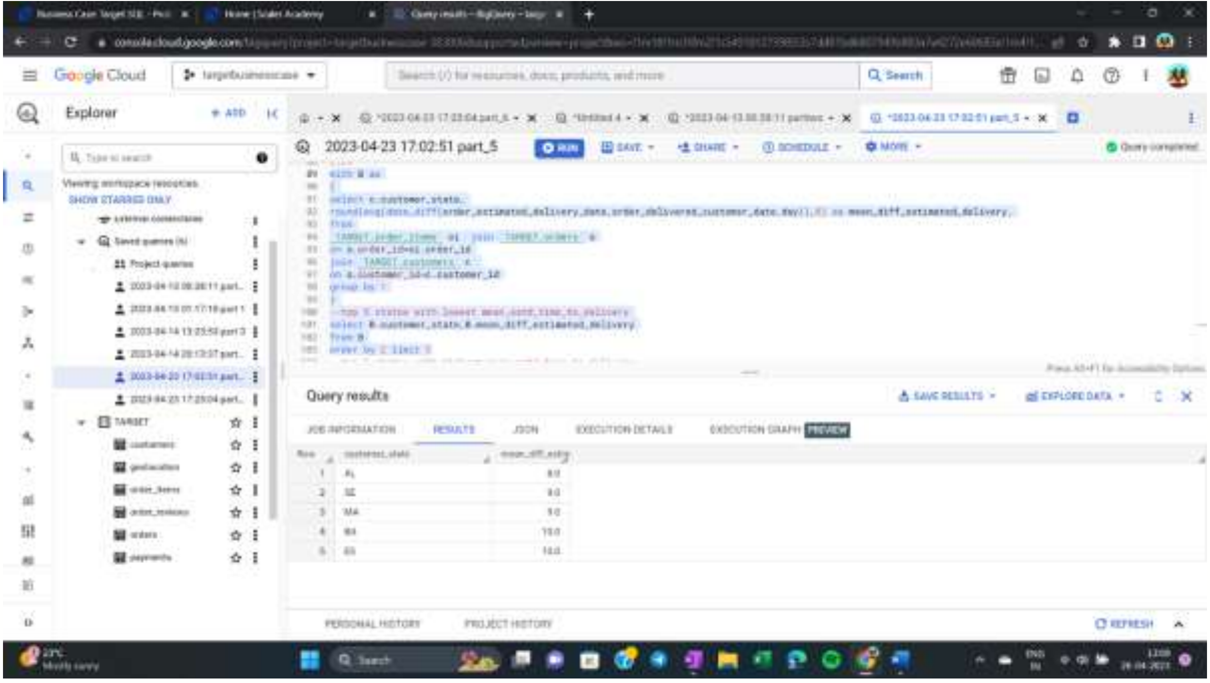
2023-04-23 17:02:51 part_5

Query results

| Row | customer_state | mean_time_to_... |
|---|---|---|
| 1 | AP | 28.0 |
| 2 | RR | 28.0 |
| 3 | AM | 26.0 |
| 4 | AL | 24.0 |
| 5 | PA | 23.0 |

**Top 5 states where delivery is really fast/ not so fast compared to the estimated date**

```
with B as
(
select c.customer_state,
round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day
)),0) as mean_diff_estimated_delivery,
from
`TARGET.order_items` oi  join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by 1
)
--top 5 states with lowest mean_estd_time_to_delivery
select B.customer_state,B.mean_diff_estimated_delivery
from B
order by 2 limit 5
```

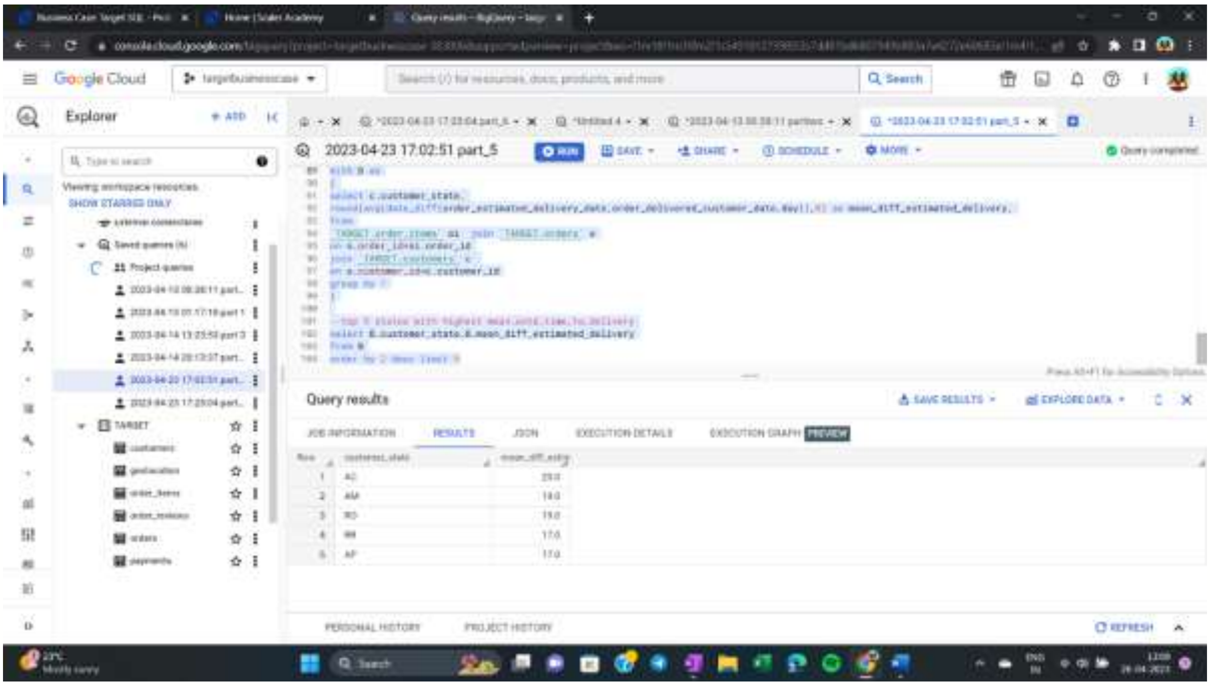


2023-04-23 17:02:51 part_5

Query results

| Row | customer_state | mean_diff_estin |
|---|---|---|
| 1 | AL | 8.0 |
| 2 | SE | 8.0 |
| 3 | MA | 9.0 |
| 4 | BA | 10.0 |
| 5 | ES | 10.0 |

```
--top 5 states with highest mean_estd_time_to_delivery
select B.customer_state,B.mean_diff_estimated_delivery
from B
order by 2 desc limit 5
```



2023-04-23 17:02:51 part_5

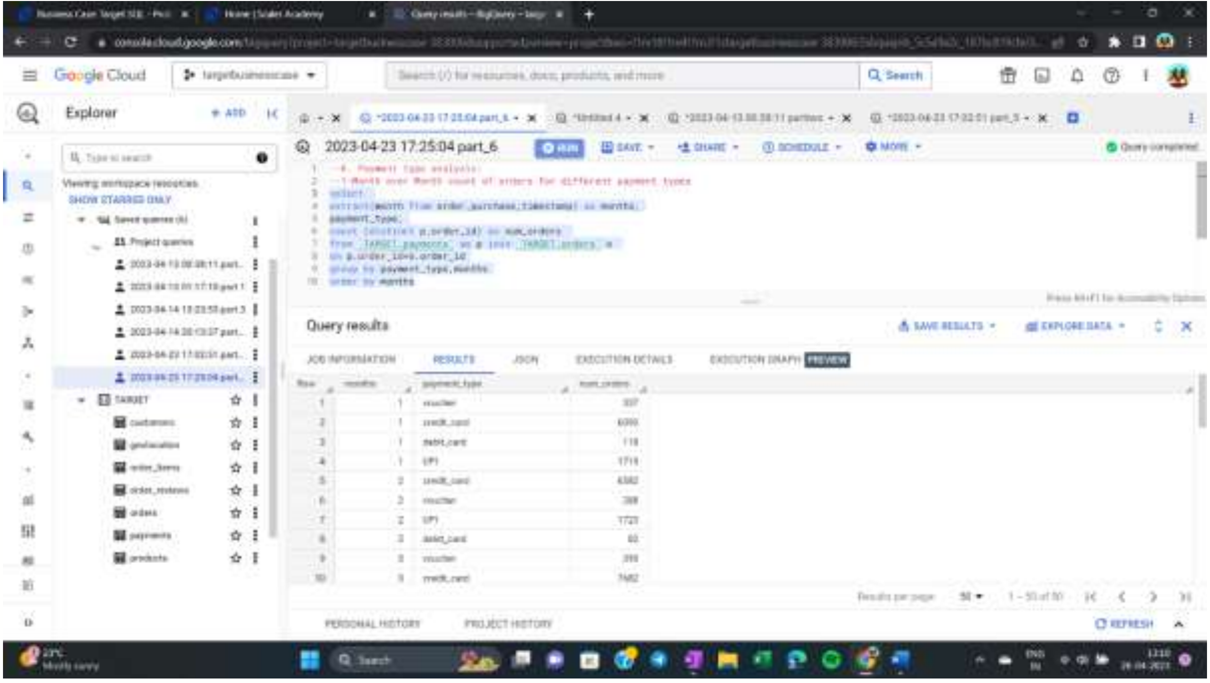
```
with B as
(
select c.customer_state,
round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)),0) as mean_diff_estimated_delivery,
from
`TARGET.order_items` oi join `TARGET.orders` o
on o.order_id=oi.order_id
join `TARGET.customers` c
on o.customer_id=c.customer_id
group by 1
)

--top 5 states with highest mean_esti_time_to_delivery
select B.customer_state,B.mean_diff_estimated_delivery
from B
order by 2 desc limit 5
```

Query results

| Row | customer_state | mean_diff_estd |
|---|---|---|
| 1 | AC | 20.0 |
| 2 | AM | 19.0 |
| 3 | RO | 19.0 |
| 4 | RR | 17.0 |
| 5 | AP | 17.0 |

## 6. Payment type analysis:

```
--1-Month over Month count of orders for different payment types
select
extract(month from order_purchase_timestamp) as months,
payment_type,
count (distinct p.order_id) as num_orders
from `TARGET.payments` as p join `TARGET.orders` o
on p.order_id=o.order_id
group by payment_type,months
order by months
```



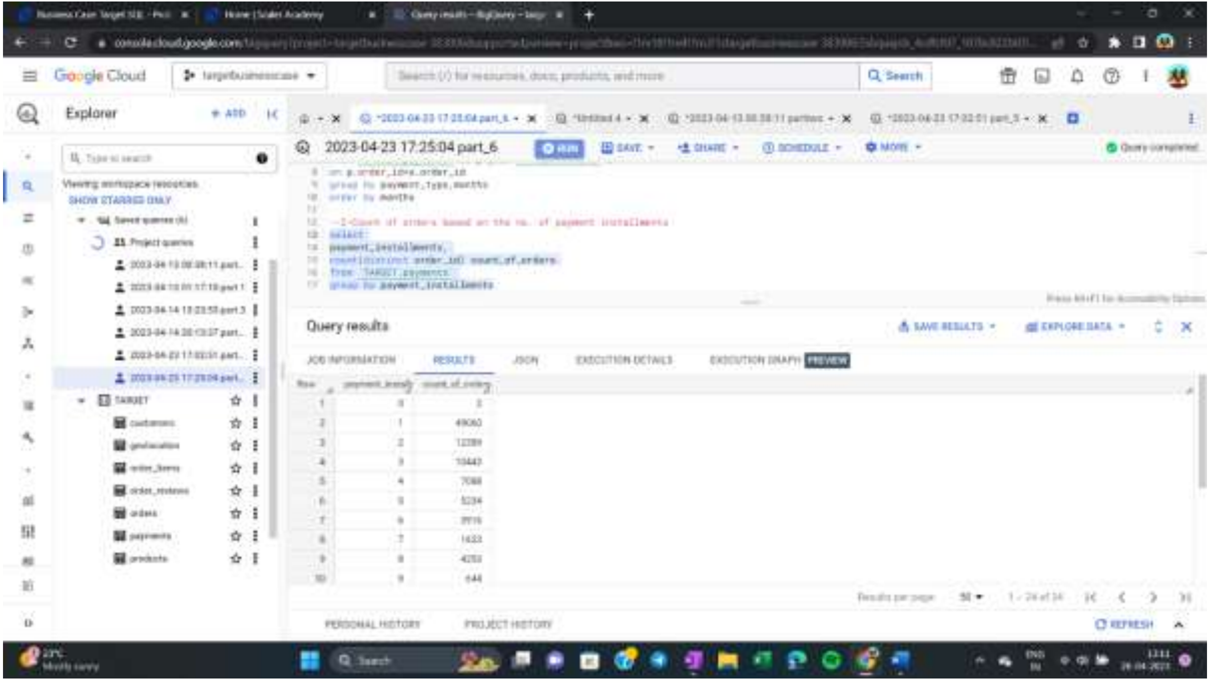
2023-04-23 17:25:04 part_6

Query results

| Row | months | payment_type | num_orders |
|---|---|---|---|
| 1 | 1 | voucher | 337 |
| 2 | 1 | credit_card | 6090 |
| 3 | 1 | debit_card | 118 |
| 4 | 1 | UPI | 1715 |
| 5 | 2 | credit_card | 6582 |
| 6 | 2 | voucher | 288 |
| 7 | 2 | UPI | 1723 |
| 8 | 3 | debit_card | 82 |
| 9 | 3 | voucher | 396 |
| 10 | 3 | credit_card | 7682 |

```
--2-Count of orders based on the no. of payment installments
select
payment_installments,
count(distinct order_id) count_of_orders
from `TARGET.payments`
```

```
group by 1
```



2023-04-23 17:25:04 part_6

```
on p.order_id=o.order_id
group by payment_type,months
order by months

--I-Count of orders based on the no. of payment installments
select
payment_installments,
count(distinct order_id) count_of_orders
from `TARGET.payments`
group by payment_installments
```

Query results

| Row | payment_install | count_of_order |
|---|---|---|
| 1 | 0 | 2 |
| 2 | 1 | 49060 |
| 3 | 2 | 12389 |
| 4 | 3 | 10443 |
| 5 | 4 | 7088 |
| 6 | 5 | 5234 |
| 7 | 6 | 3916 |
| 8 | 7 | 1623 |
| 9 | 8 | 4253 |
| 10 | 9 | 644 |