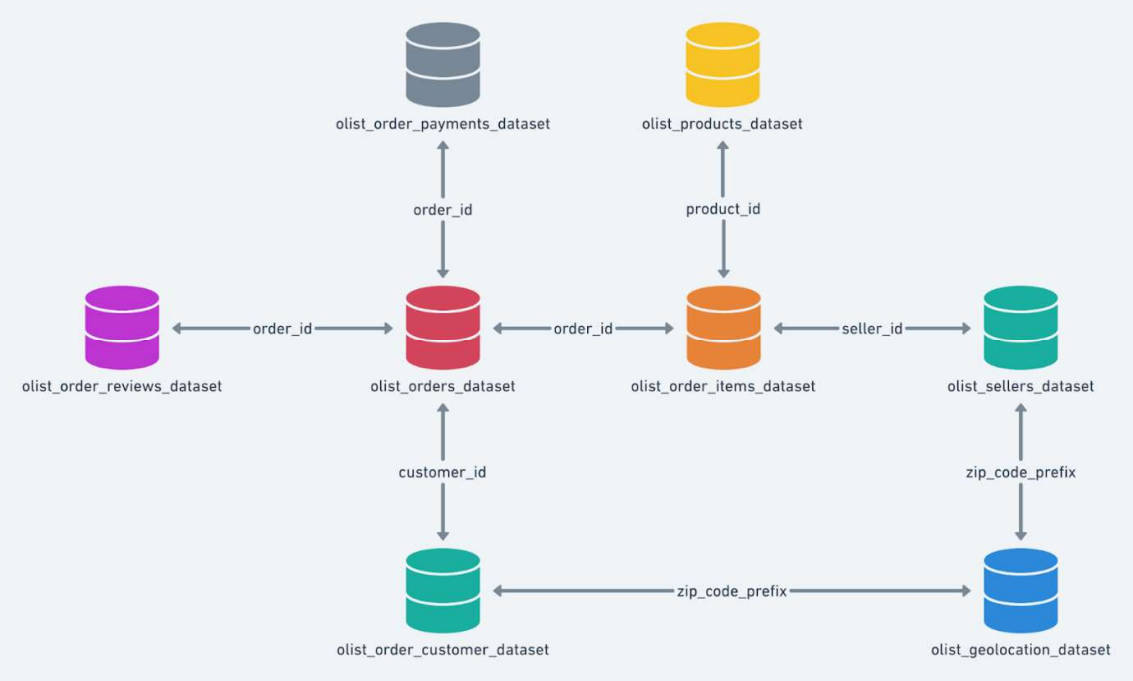
SQL Project – TARGET



1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
    1. Data type of columns in a table
    2. Time period for which the data is given
    3. Cities and States of customers ordered during the given period

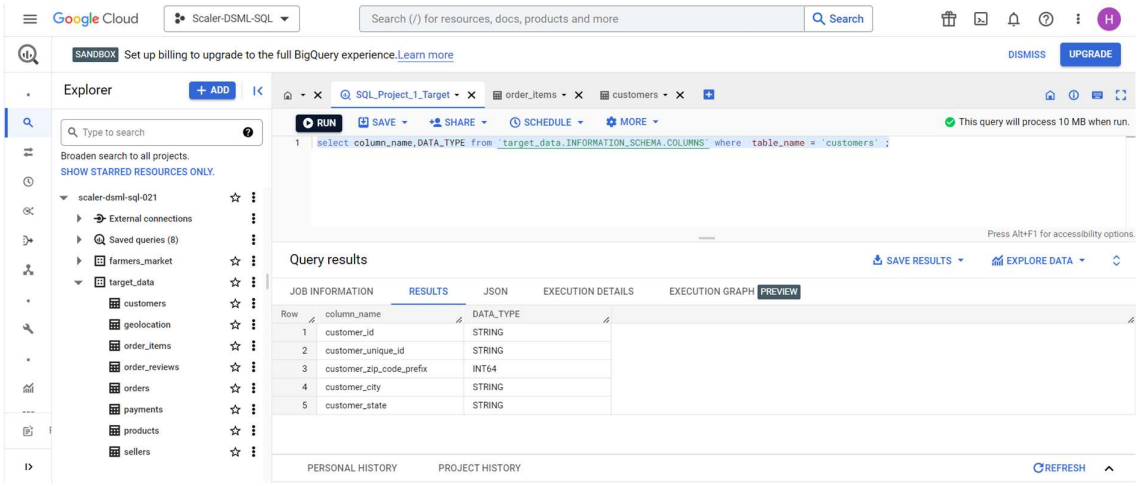
Ans 1.1

- customers Table

Query –

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'customers' ;
```

Output Screenshot -

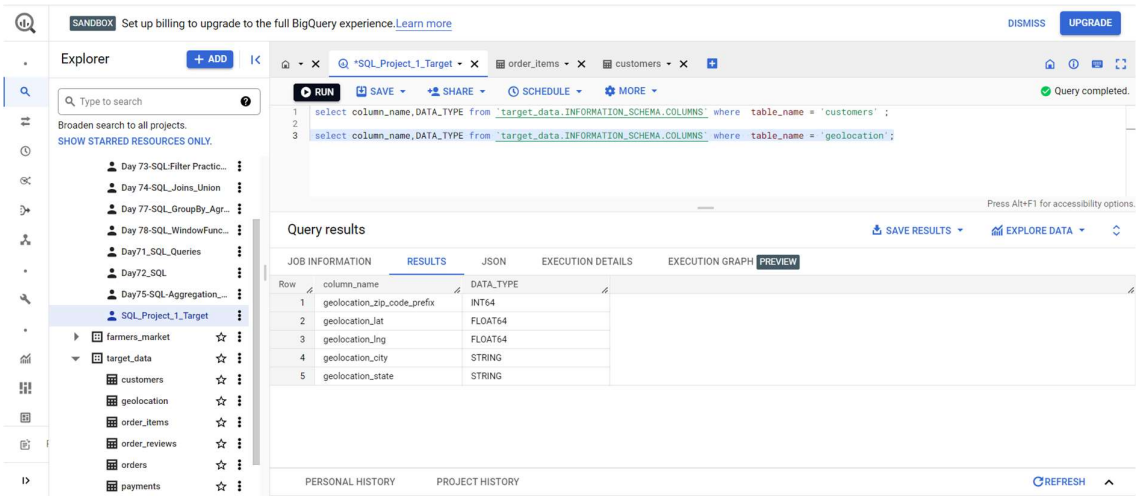


- geolocation Table

Query -

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'geolocation';
```

Output Screenshot -

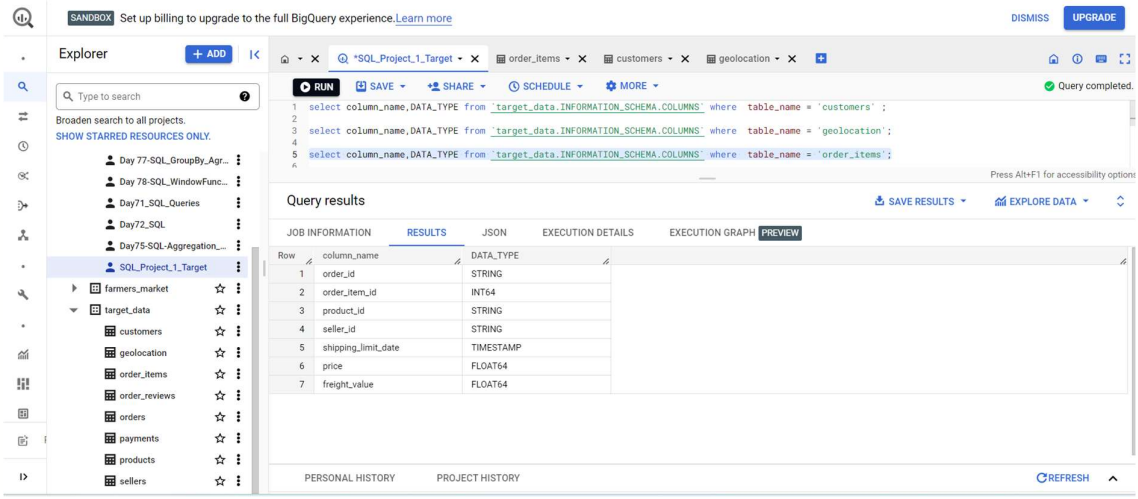


- order_items Table

Query -

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'order_items';
```

Output Screenshot –

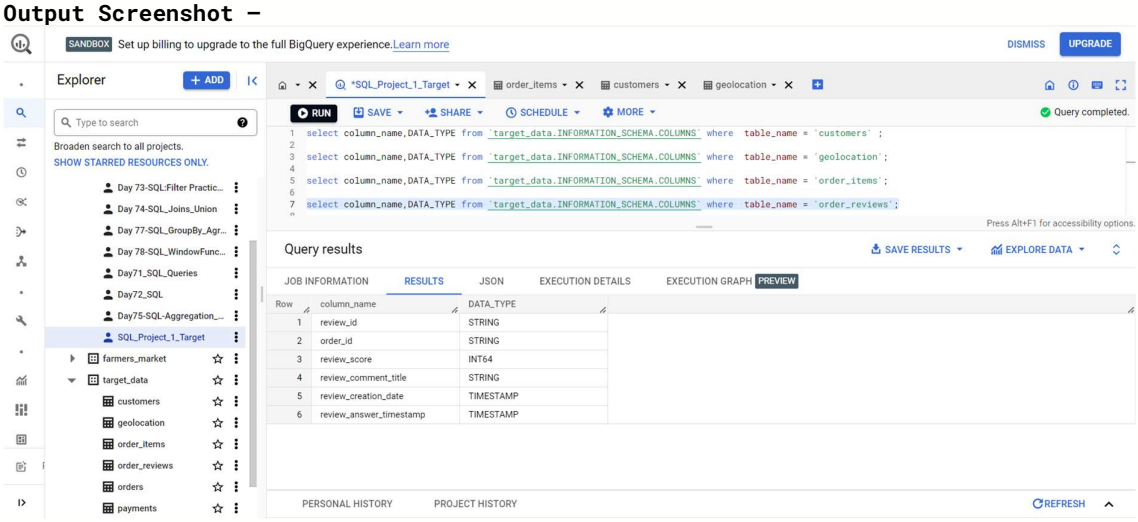


- order_reviews Table

Query –

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'order_reviews';
```

Output Screenshot –

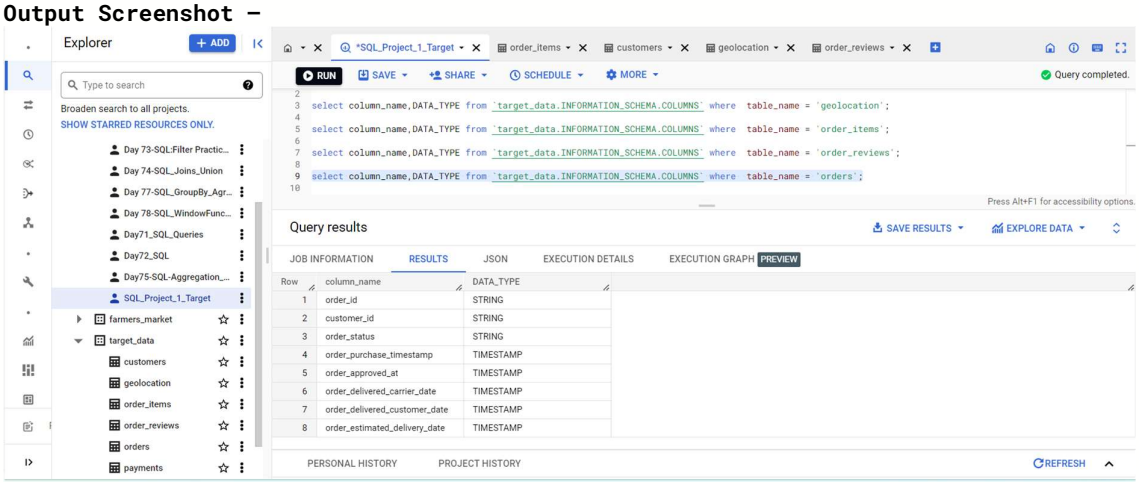


- orders Table

Query –

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'orders';
```

Output Screenshot –

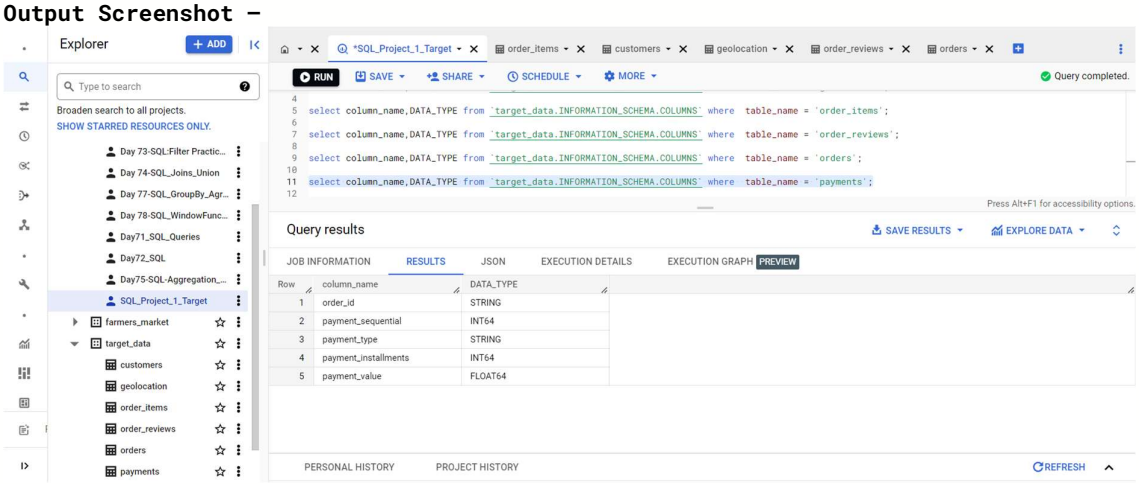


- payments Table

Query –

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'payments';
```

Output Screenshot –

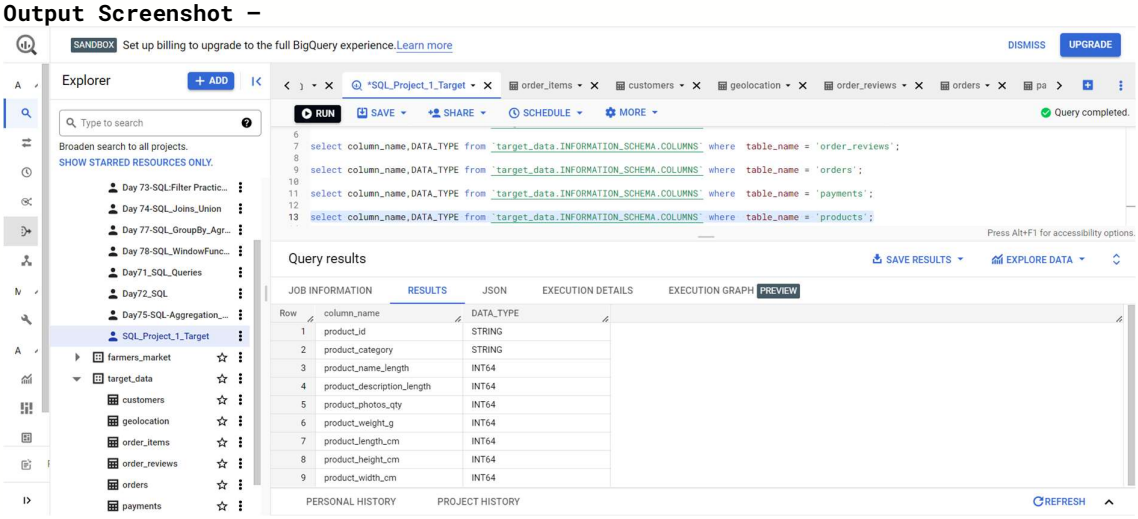


- products Table

Query –

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'products';
```

Output Screenshot –

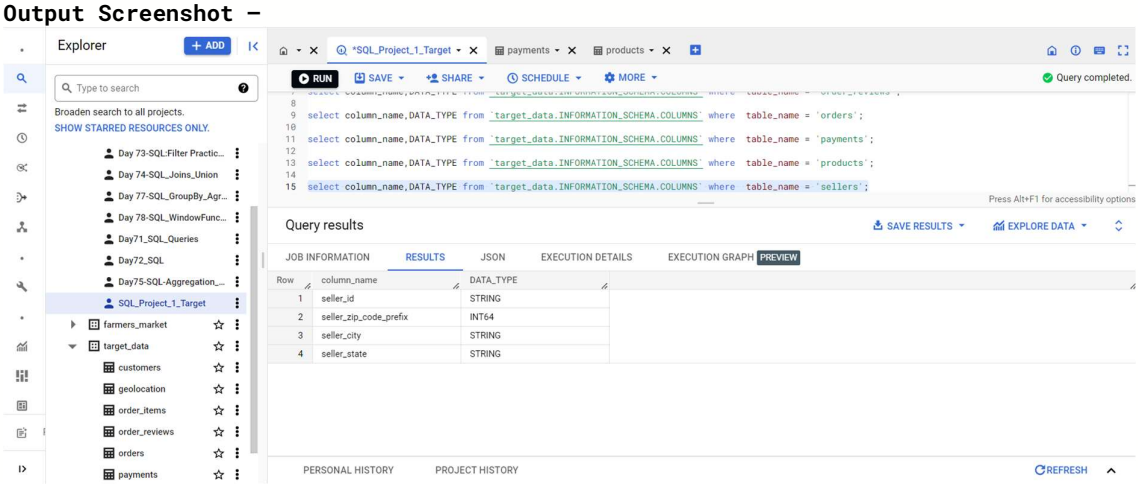


- sellers Table

Query –

```
select column_name,DATA_TYPE from `target_data.INFORMATION_SCHEMA.COLUMNS` where  t
able_name = 'sellers';
```

Output Screenshot –

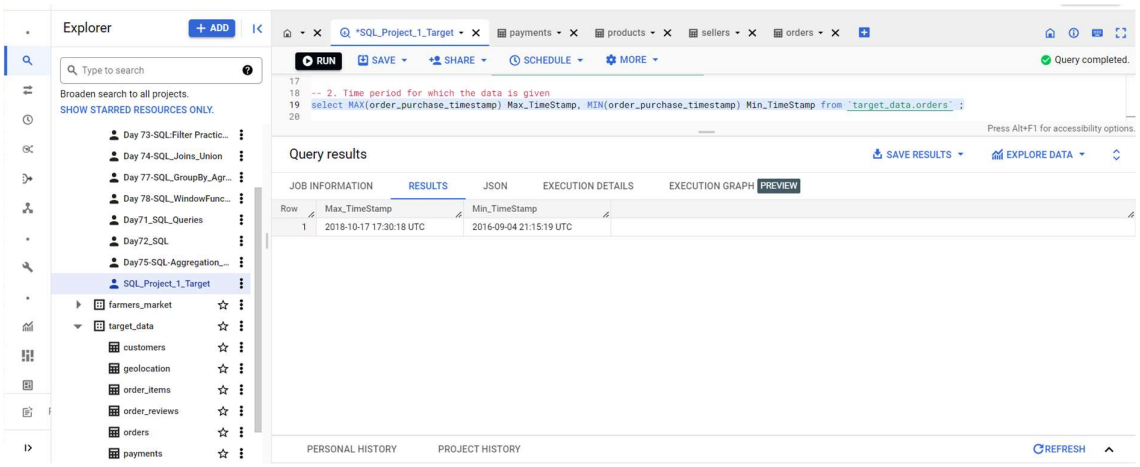


Ans 1.2 Time period for which the data is given

Query –

```
select MAX(order_purchase_timestamp) Max_TimeStamp, MIN(order_purchase_timestamp) M
in_TimeStamp from `target_data.orders` ;
```

Output Screenshot –



Observation :

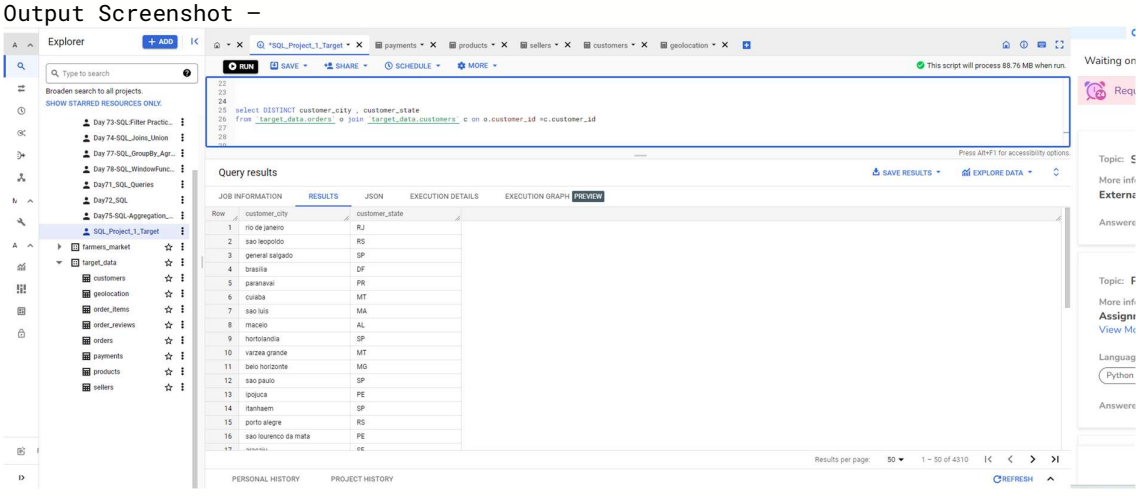
The above date is from time period 4th September,2016 to 17th October,2018 .

Ans 1. 3 Cities and States of customers ordered during the given period

Query –

```
select DISTINCT customer_city , customer_state
from `target_data.orders` o join `target_data.customers` c on o.customer_id =c.cust
omer_id
```

Output Screenshot –



Observation :

The above date is shows all State and City names of customers placing order.

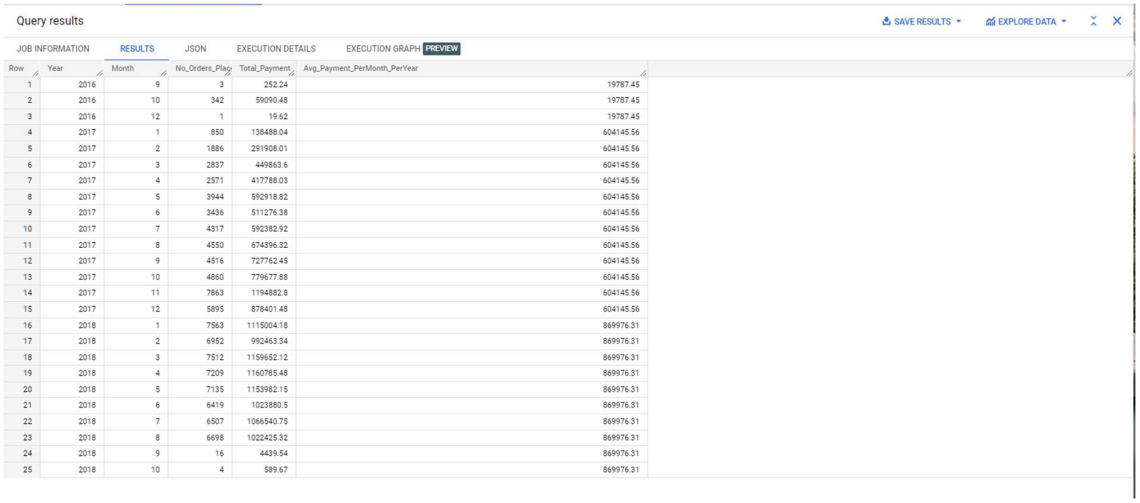
2. In-depth Exploration:
    1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?
    2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Ans 2.1

Query :

```
select Year , Month , No_Orders_Placed , Total_Payment,
ROUND(SUM(Total_Payment) OVER(partition by Year)/COUNT(MONTH) OVER(partition by Yea
r),2)  Avg_Payment_PerMonth_PerYear
from
(
select Year , Month , Count(order_id) No_Orders_Placed ,  ROUND(SUM(payment_value),
3)  Total_Payment  from
(select  o.order_id ,order_purchase_timestamp,
EXTRACT(YEAR from order_purchase_timestamp ) as Year,
EXTRACT(MONTH from order_purchase_timestamp ) as Month,
payment_value
from  `target_data.orders`  o join `target_data.payments` p on  o.order_id = p.orde
r_id )
group by Year,Month
)
order by Year,Month
```

Output Screenshot :



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Year	Month	No_Orders_Plac	Total_Payment	Avg_Payment_PerMonth_PerYear
1	2016	9	3	252.24	19787.45
2	2016	10	342	59090.48	19787.45
3	2016	12	1	19.62	19787.45
4	2017	1	850	138488.04	604145.56
5	2017	2	1886	291908.01	604145.56
6	2017	3	2837	449863.6	604145.56
7	2017	4	2571	417788.03	604145.56
8	2017	5	3944	592918.82	604145.56
9	2017	6	3436	511276.38	604145.56
10	2017	7	4317	592382.92	604145.56
11	2017	8	4550	674396.32	604145.56
12	2017	9	4516	727762.45	604145.56
13	2017	10	4860	779677.88	604145.56
14	2017	11	7863	1194882.8	604145.56
15	2017	12	5895	878401.48	604145.56
16	2018	1	7563	1115004.18	869976.31
17	2018	2	6952	992463.34	869976.31
18	2018	3	7512	1159652.12	869976.31
19	2018	4	7209	1160785.48	869976.31
20	2018	5	7135	1153982.15	869976.31
21	2018	6	6419	1023880.5	869976.31
22	2018	7	6507	1066540.75	869976.31
23	2018	8	6698	1022425.32	869976.31
24	2018	9	16	4439.54	869976.31
25	2018	10	4	589.67	869976.31

Observation :

- **Insights :**

Order purchase data for Year 2016 is only there for 3 months (September, October and December) and Year 2018 also has no data for (Novemeber and December) . So we have calculated the Average Payment done for orders per month for each year . Seeing Avg_Payment_PerMonth_PerYear we can see it **is increasing** for each year from 2016 to 2018 ,this signifies **growing trend on e-commerce in Brazil**.

- In Year 2016 the maximum payment (= 59090.48) on orders was done in October Month where in total order count was 342.
- In Year 2017 the maximum payment (= 1194882.8) on orders was done in November Month where in total order count was 7863 .
- In Year 2018 the maximum payment (= 1160785.48) on orders was done in April Month where in total order count was 7209.

- **Recommendations :**

We can see the No. of orders Placed in September and October of 2018 had drastically reduced due to which average sales per month of year also reduced , else it would be much higher .

Extra discounts must be given near end of year and events like stock clearance sale must be held in end months of year to increase count of orders.

Loyalty program reward program can also be introduced to award the customers placing most number of orders . This will intern also increase customer base and increase order count.

Ans 2.2 What time do Brazilian customers tend to buy

Query :

```
with customer_buy_time as
(
select order_id , customer_id ,order_purchase_timestamp, EXTRACT(HOUR FROM order_pu
rchase_timestamp) Order_Hour ,
CASE
When  EXTRACT(HOUR FROM order_purchase_timestamp) >=0  and EXTRACT(HOUR FROM order_
purchase_timestamp) <=6 Then "Dawn"
```

```
When  EXTRACT(HOUR FROM order_purchase_timestamp) >6  and EXTRACT(HOUR FROM order_p
urchase_timestamp) <=12 Then "Morning"
When  EXTRACT(HOUR FROM order_purchase_timestamp) >12 and EXTRACT(HOUR FROM order_p
urchase_timestamp) <=18 Then "Afternoon"
When  EXTRACT(HOUR FROM order_purchase_timestamp) >18 and EXTRACT(HOUR FROM order_p
urchase_timestamp)  <24 Then "Night"
END as Day_Period
from  `target_data.orders`
)
select Day_Period , Count(Customer_id) Count_Customers
from  customer_buy_time
group by Day_Period
```

Output Screenshot :

Row	Day_Period	Count_Customers
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

Observation :

- Insights :

Seeing above data we can see Brazilian customers tend to buy most in “Afternoon” time and seond highest in “Night” time . Least orders are placed at Dawn time .

- Recommendations :

We can increase the count of Customer in other Day Period Like Morning and Night by introducing special limited time period offers on products which will be valid only for Morning and Night time period .

Seasonal sales of limited products can be scheduled to start in “Dawn Time Period”, which will encourage more customers to buy in Dawn Time , due to risk of all limited products getting bought in Morning / Afternoon .

3. Evolution of E-commerce orders in the Brazil region:
    1. Get month on month orders by states
    2. Distribution of customers across the states in Brazil

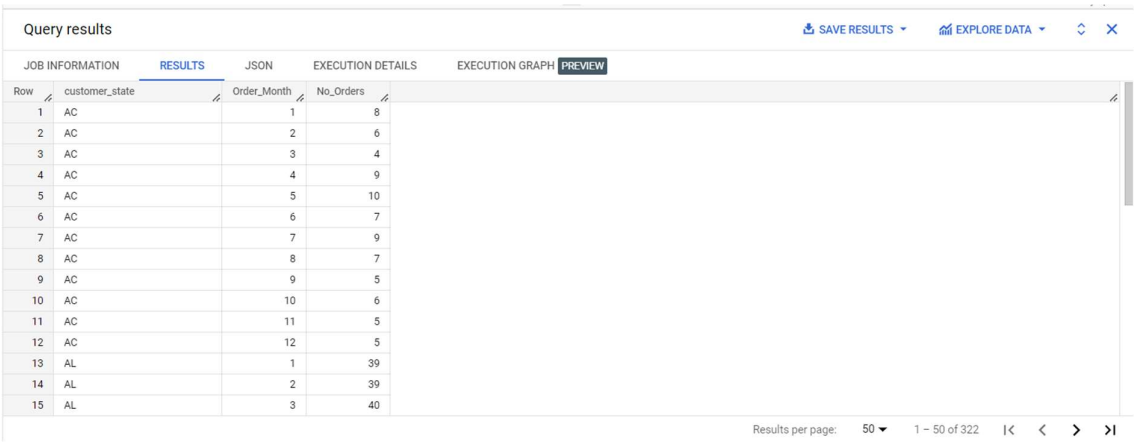
Ans 3.1 Get month on month orders by states

Query :

```
with Order_Customer as
```

```
(
select o.customer_id , o.order_id , EXTRACT(MONTH FROM order_purchase_timestamp) Or
der_Month ,customer_state
from `target_data.orders`  o join `target_data.customers` c on  o.customer_id= c.cu
stomer_id
)
select  customer_state ,Order_Month ,Count(order_id) No_Orders
from Order_Customer
group by customer_state ,Order_Month
order by customer_state ,Order_Month
```

Output Screenshot :



Observation :

- Insights :

On sorting the above query result based on “No_orders” in descending order , we can see all months largest Number of Orders are placed from Customer State “SP” .

On sorting the above query result based on “No_orders” in ascending order ,we can see Smallest number of orders are placed from customer state “RR” in month (January , September and November) and customer state “AP” in month (September)

- Recommendations :

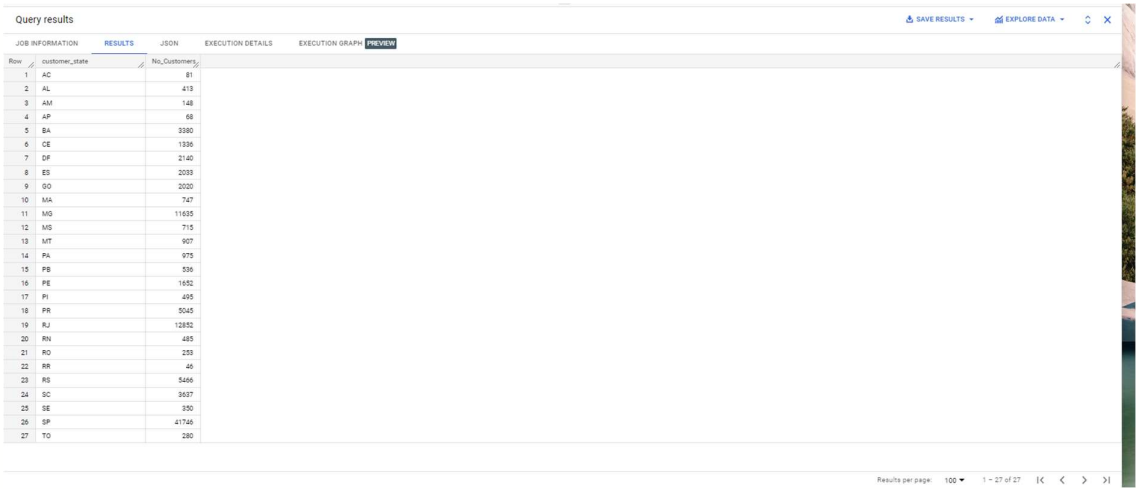
Number of Orders in State “RR” and “AP” can be increased by increasing their customer base doing more advertisements and giving more discount on products (items) in these states.

Ans 3.2 Distribution of customers across the states in Brazil

Query :

```
with Order_Customer as
(
select o.customer_id , o.order_id , EXTRACT(MONTH FROM order_purchase_timestamp) Or
der_Month ,customer_state
from `target_data.orders`  o join `target_data.customers` c on  o.customer_id= c.cu
stomer_id
)
select  customer_state ,Count(DISTINCT customer_id) No_Customers
from Order_Customer
group by customer_state
order by customer_state
```

Output Screenshot :



Observation :

- Insights :

Most number of customer are from “SP” State in Brazil . Least number of customers are in States RR, AP,AC .

- Recommendations :

The number of customers in other regions of Brazil where its really low like States RR, AP ,AC can be increased by doing more advertisements and giving more discount on products (items) in these states to increase their customer base.

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
    1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table
    2. Mean & Sum of price and freight value by customer state

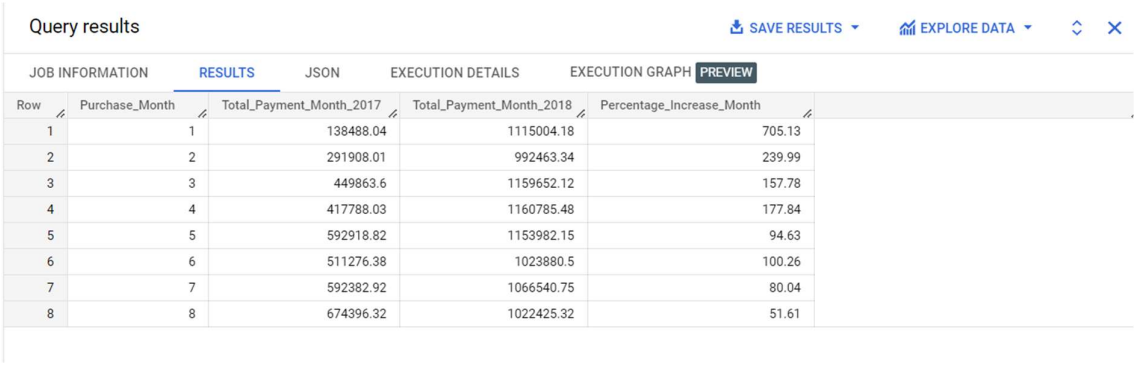
Ans 4.1 Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

Query :

```
with d_2017 as
(
select ROUND(SUM(payment_value),2) Total_Payment_Month_2017,
EXTRACT(MONTH from order_purchase_timestamp) Purchase_Month ,
from `target_data.payments` p join `target_data.orders`  o on o.order_id = p.order_
id
where EXTRACT(MONTH from order_purchase_timestamp) <=8 and EXTRACT(YEAR from order_
purchase_timestamp) = 2017
group by  Purchase_Month
) ,
d_2018 as
(
select ROUND(SUM(payment_value),2) Total_Payment_Month_2018,
EXTRACT(MONTH from order_purchase_timestamp) Purchase_Month ,
from `target_data.payments` p join `target_data.orders`  o on o.order_id = p.order_
id
where EXTRACT(MONTH from order_purchase_timestamp) <=8 and EXTRACT(YEAR from order_
purchase_timestamp) = 2018
group by Purchase_Month
)

select d_2017.Purchase_Month, Total_Payment_Month_2017, Total_Payment_Month_2018 ,
ROUND( ((Total_Payment_Month_2018 - Total_Payment_Month_2017) / Total_Payment_Month
_2017)*100,2) as Percentage_Increase_Month
from d_2017 join d_2018 on  d_2017.Purchase_Month =  d_2018.Purchase_Month
order by  d_2017.Purchase_Month
```

Output Screenshot :



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	Purchase_Month	Total_Payment_Month_2017	Total_Payment_Month_2018	Percentage_Increase_Month
1	1	138488.04	1115004.18	705.13
2	2	291908.01	992463.34	239.99
3	3	449863.6	1159652.12	157.78
4	4	417788.03	1160785.48	177.84
5	5	592918.82	1153982.15	94.63
6	6	511276.38	1023880.5	100.26
7	7	592382.92	1066540.75	80.04
8	8	674396.32	1022425.32	51.61

Observation :

- **Insights :**

We can see highest increase in % of Payment from 2017 to 2018 is in Month of January , it can be due to start of New Year the customer purchase increases every Year due to new products in market and also due to sale on products in Start of New Year (New Year Sale) .

- **Recommendations :**

Target can increase their sales further in middle months of the year (July , August) by

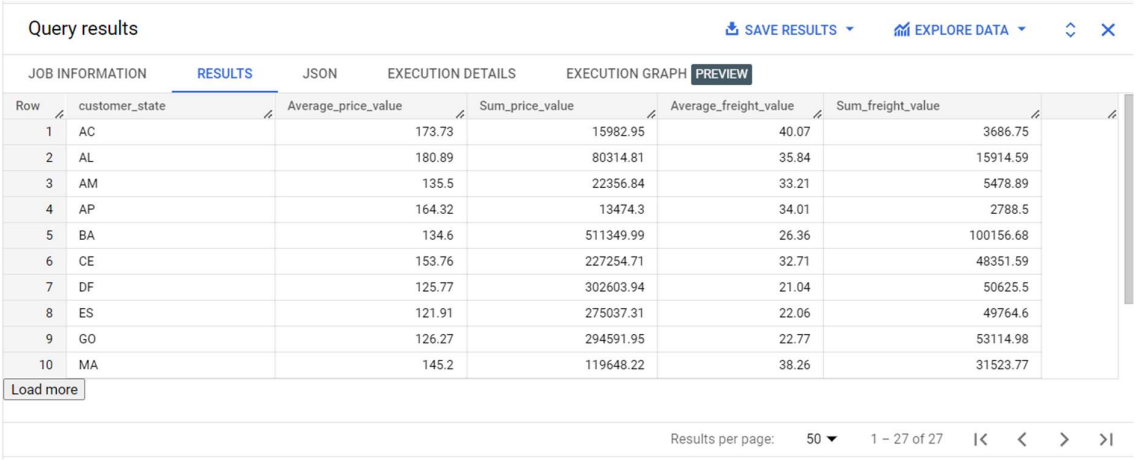
- Introducing Mid-Year Sale , Stock Clearance Sale
- Introducing EMI payment options with 0 or very less interest on products
- Arranging Lucky Draw contest in Middle of Year to award random frequent consumers

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

Query :

```
select  c.customer_state ,
ROUND(AVG(oi.price),2) Average_price_value ,
ROUND(SUM(oi.price),2) Sum_price_value ,
ROUND(AVG(oi.freight_value),2) Average_freight_value,
ROUND(SUM(oi.freight_value),2) Sum_freight_value
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
join `target_data.order_items` oi on  oi.order_id = o.order_id
group by  c.customer_state
order by c.customer_state
```

Output Screenshot :



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Average_price_value	Sum_price_value	Average_freight_value	Sum_freight_value
1	AC	173.73	15982.95	40.07	3686.75
2	AL	180.89	80314.81	35.84	15914.59
3	AM	135.5	22356.84	33.21	5478.89
4	AP	164.32	13474.3	34.01	2788.5
5	BA	134.6	511349.99	26.36	100156.68
6	CE	153.76	227254.71	32.71	48351.59
7	DF	125.77	302603.94	21.04	50625.5
8	ES	121.91	275037.31	22.06	49764.6
9	GO	126.27	294591.95	22.77	53114.98
10	MA	145.2	119648.22	38.26	31523.77

Load more

Results per page: 50 1 – 27 of 27

Observation :

- **Insights :**

By seeing the output of above query we can conclude that average fright value (product transportation cost) is lowest in “SP state” , which is good. Whereas there are states where average freight cost is more than double the average freight cost of SP state .

- **Recommendations :**

One can try to reduce Total Fright (Sum_Fright_value) or Average Fright Cost by following better product transportation strategies like :-

- Carrying and delivering Multiple orders in same area together (will reduce freight cost of individual products)
- Using Electrical Vehicles for transportation (if not very long distance to customer) , will save fuel cost .
- By setting up new warehouses in Cities where count of orders is very large. This will allow products to be shipped faster and reduce transportation cost (freight cost)

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
    - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
    - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date
3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
6. Top 5 states with highest/lowest average time to delivery
7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Ans 5.1 Calculate days between purchasing, delivering and estimated delivery

Query :

```
 select   order_id,
 DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) as Days_Deli
vered_Purchase,
 DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp ,DAY) as Days_Est
imatedDeliver_Purchase,
 DATE_DIFF( order_delivered_customer_date, order_estimated_delivery_date ,DAY) as D
ays_Delivered_EstimatedDeliver
 from `target_data.orders`
 where order_delivered_customer_date is not NULL and
 order_purchase_timestamp is not NULL  and
 order_delivered_customer_date is not Null
```

Output Screenshot :

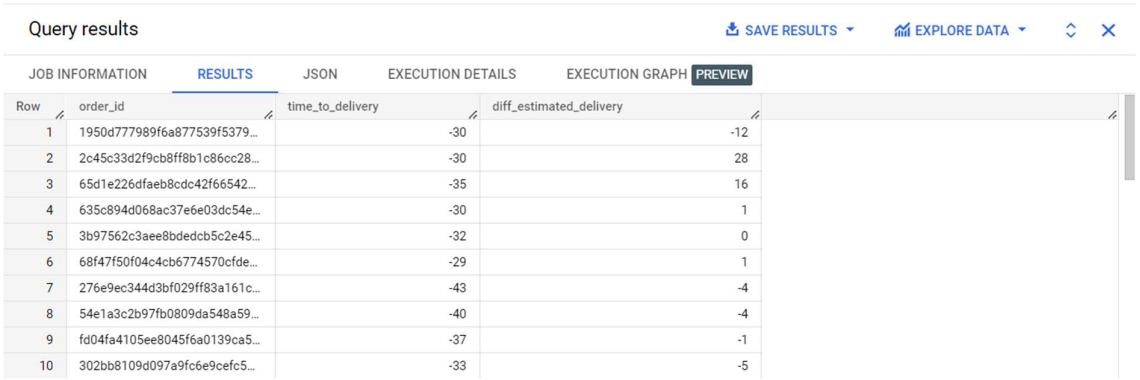
Row	order_id	Days_Delivered_Purchase	Days_EstimatedDeliver_Purchase	Days_Delivered_EstimatedDeliver
1	1950d777989f6a877539f5379...	30	17	12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	59	-28
3	65d1e226dfaeb8cdc42f66542...	35	52	-16
4	635c894d068ac37e6e03dc54e...	30	32	-1
5	3b97562c3aee8bdedcb5c2e45...	32	33	0
6	68f47f50f04c4cb6774570cfde...	29	31	-1
7	276e9ec344d3bf029ff83a161c...	43	39	4
8	54e1a3c2b97fb0809da548a59...	40	36	4
9	fd04fa4105ee8045f6a0139ca5...	37	35	1
10	302bb8109d097a9fc6e9cefc5...	33	28	5

Ans 5.2 Find time_to_delivery & diff_estimated_delivery.

Query :

```
select  order_id,
 DATE_DIFF(order_purchase_timestamp,order_delivered_customer_date,DAY) as time_to_d
elivery,
 DATE_DIFF( order_estimated_delivery_date ,order_delivered_customer_date, DAY) as d
iff_estimated_delivery
 from `target_data.orders`
 where order_delivered_customer_date is not NULL and
  order_delivered_customer_date is not Null
```

Output Screenshot :



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	order_id	time_to_delivery	diff_estimated_delivery
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

- **Note :**
  time_to_delivery is negative as per the question the formula of “time_to_delivery “ is time_to_delivery = order_purchase_timestamp - order_delivered_customer_date
  But purchase date will always be Less than delivered date , so that is why it was showing negative . We can remove negative sign by using “ABS() “ function on “time_to_delivery” column or by changing the formula of time_to_delivery to “order_delivered_customer_date - order_purchase_timestamp “

ANS 5.3

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

Query :

```
with ct as (
select  c.customer_state ,o.order_id ,oi.freight_value,
DATE_DIFF(order_purchase_timestamp,order_delivered_customer_date,DAY) as time_to_de
livery,
DATE_DIFF( order_estimated_delivery_date ,order_delivered_customer_date, DAY) as di
ff_estimated_delivery
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
join `target_data.order_items` oi on  oi.order_id = o.order_id
order by  o.order_id )

select customer_state ,ROUND(AVG(freight_value),2) Mean_freight_value ,
ROUND(AVG(time_to_delivery),2) Mean_time_to_delivery ,
ROUND(AVG(diff_estimated_delivery),2) Mean_diff_estimated_delivery
from ct
group by customer_state
order by customer_state
```

Output Screenshot :

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Mean_freight_value	Mean_time_to_delivery	Mean_diff_estimated_delivery
1	AC	40.07	-20.33	20.01
2	AL	35.84	-23.99	7.98
3	AM	33.21	-25.96	18.98
4	AP	34.01	-27.75	17.44
5	BA	26.36	-18.77	10.12
6	CE	32.71	-20.54	10.26
7	DF	21.04	-12.5	11.27
8	ES	22.06	-15.19	9.77
9	GO	22.77	-14.95	11.37
10	MA	38.26	-21.2	9.11

Results per page: 50 | 1 – 27 of 27

Ans 5.4 Sort the data to get the following:

Ans 5.5

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

Query :

```
(select  c.customer_state ,ROUND(AVG(oi.freight_value),2) Average_freight_value,"Hi
ghest Avg Freight" Highest_OR_Lowest_AvgFreight ,
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
join `target_data.order_items` oi on  oi.order_id = o.order_id
group by  c.customer_state
order by  Average_freight_value DESC
LIMIT 5 )
UNION ALL
(select  c.customer_state ,ROUND(AVG(oi.freight_value),2) Average_freight_value, "L
owest Avg Freight" Highest_OR_Lowest_AvgFreight,
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
join `target_data.order_items` oi on  oi.order_id = o.order_id
group by  c.customer_state
order by  Average_freight_value ASC
LIMIT 5 )
```

Output Screenshot :

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Average_freight_value	Highest_OR_Lowest_AvgFreight
1	RR	42.98	Highest Avg Freight
2	PB	42.72	Highest Avg Freight
3	RO	41.07	Highest Avg Freight
4	AC	40.07	Highest Avg Freight
5	PI	39.15	Highest Avg Freight
6	SP	15.15	Lowest Avg Freight
7	PR	20.53	Lowest Avg Freight
8	MG	20.63	Lowest Avg Freight
9	RJ	20.96	Lowest Avg Freight
10	DF	21.04	Lowest Avg Freight

Note: Column “Highest_OR_Lowest_AvgFreight” was added to differentiate Between Top 5 (Highest Avg. Freight) and Top5 (Lowest Avg. Freight) **Insights :**

- **Insights :**

Above date shows top 5 sates having highest and lowest freight cost .

By seeing the output of above query we can conclude that average fright value (product transportation cost) is lowest in “SP state” , which is good and highest in RR state . Whereas there are states where average freight cost is more than double the average freight cost of SP state .

- **Recommendations :**

One can try to reduce Average Fright Cost by following better product transportation strategies like :-

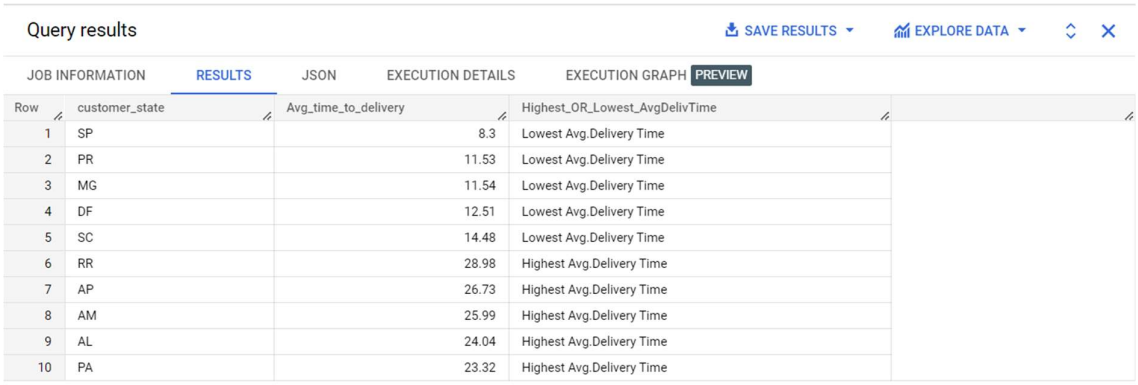
- Carrying and delivering Multiple orders in same area together (will reduce freight cost of individual products)
- Using Electrical Vehicles for transportation (if not very long distance to customer), will save fuel cost .
- By setting up new warehouses in Cities where count of orders is very large. This will allow products to be shipped faster and reduce transportation cost (freight cost)

Ans 5.6 Top 5 states with highest/lowest average time to delivery

Query :

```
(select  c.customer_state ,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)
 as Avg_time_to_delivery,"Highest Avg.Delivery Time " Highest_OR_Lowest_AvgDelivTim
e
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
group by  c.customer_state
order by  Avg_time_to_delivery DESC
LIMIT 5 )
UNION ALL
(select  c.customer_state ,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)
 as Avg_time_to_delivery,"Lowest Avg.Delivery Time " Highest_OR_Lowest_AvgDelivTime
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
group by  c.customer_state
order by  Avg_time_to_delivery ASC
LIMIT 5 )
```

Output Screenshot :



Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Avg_time_to_delivery	Highest_OR_Lowest_AvgDelivTime
1	SP	8.3	Lowest Avg.Delivery Time
2	PR	11.53	Lowest Avg.Delivery Time
3	MG	11.54	Lowest Avg.Delivery Time
4	DF	12.51	Lowest Avg.Delivery Time
5	SC	14.48	Lowest Avg.Delivery Time
6	RR	28.98	Highest Avg.Delivery Time
7	AP	26.73	Highest Avg.Delivery Time
8	AM	25.99	Highest Avg.Delivery Time
9	AL	24.04	Highest Avg.Delivery Time
10	PA	23.32	Highest Avg.Delivery Time

Note: Column “Highest_OR_Lowest_AvgDelivTime” was added to differentiate Between Top 5 (Highest Avg. Delivery Time) and Top5 (Lowest Avg. Delivery Time)

Observation : -

- Insights :

We can see States SP , PR, MG,DF,SC have least average delivery time to customers from date of purchase , least being 8 days (approx.).

Whereas states like RR,AP,AM,AL,PA have average highest delivery time , largest being 29 days (approx.)

- Recommendations :

Target should try to reduce further the delivery time and try to achieve single digit days (1 to 9 days) in average highest delivery time in all states first .This can be done by setting up new warehouses major customer base states , which will reduce days to deliver a product to customer . The good transportation service can be improved further by giving contracts to specialized good transportation companies (in each state) who have better expertise in goods transportation.

Strategies can be devised to deliver few products on same day as Purchase Day –“1 day delivery” for few products .

Ans 5.7 Top 5 states where delivery is really fast/ not so fast compared to estimated date

Query :

```
(select  c.customer_state ,
ROUND(AVG(DATE_DIFF(order_estimated_delivery_date ,order_delivered_customer_date, D
AY)),2) as Avg_diff_estimated_delivery,"Really Fast.Delivery" ReallyFast_OR_NotFast
_Delivery
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
group by  c.customer_state
order by  Avg_diff_estimated_delivery DESC
LIMIT 5 )
UNION ALL
(select  c.customer_state ,
ROUND(AVG(DATE_DIFF(order_estimated_delivery_date ,order_delivered_customer_date, D
AY)),2) as Avg_diff_estimated_delivery,"Not So Fast Delivery " ReallyFast_OR_NotFas
t_Delivery
from `target_data.orders`  o join `target_data.customers` c on o.customer_id=c.cust
omer_id
group by  c.customer_state
order by  Avg_diff_estimated_delivery ASC
LIMIT 5 )
```

Output Screenshot :

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	Avg_diff_estimated_delivery	ReallyFast_OR_NotFast_Delivery
1	AC	19.76	Really Fast.Delivery
2	RO	19.13	Really Fast.Delivery
3	AP	18.73	Really Fast.Delivery
4	AM	18.61	Really Fast.Delivery
5	RR	16.41	Really Fast.Delivery
6	AL	7.95	Not So Fast Delivery
7	MA	8.77	Not So Fast Delivery
8	SE	9.17	Not So Fast Delivery
9	ES	9.62	Not So Fast Delivery
10	BA	9.93	Not So Fast Delivery

Column "ReallyFast_OR_NotFast_Delivery" was added to differentiate Between Top 5 (Really Fast Delivery compared to estimated date) and Top5 (Not So Fast Delivery compared to estimated date)

Observation : -

- Insights :

We can see states AC , RO , AP,AM,RR have really fast delivery . Whereas states AL,MA,SE,ES,BA have not so fast delivery .

- Recommendations :

We can reduce the time to deliver products in “Not So fast delivery States “ by setting up warehouses in these states which will stock items and improve time to deliver the products and save freight cost as well.

6. Payment type analysis:

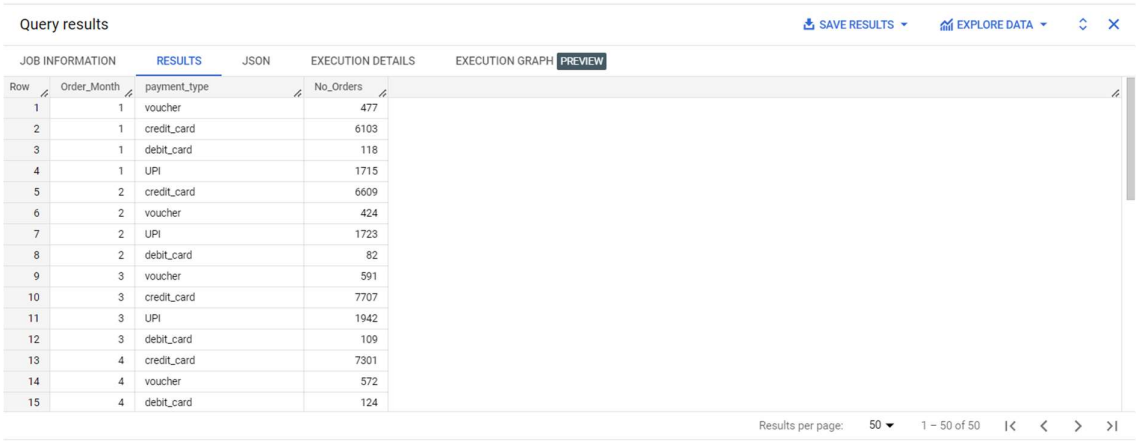
1. Month over Month count of orders for different payment types
2. Count of orders based on the no. of payment installments

Ans 6.1 Month over Month count of orders for different payment types

Query :

```
with Order_PaymentType as
(
select o.order_id ,  EXTRACT(MONTH FROM order_purchase_timestamp) Order_Month ,paym
ent_type , payment_installments
from `target_data.orders`  o join `target_data.payments` p on  o.order_id = p.order
_id
)
select Order_Month , payment_type , Count(order_id) No_Orders
from Order_PaymentType
group by Order_Month , payment_type
order by Order_Month
```

Output Screenshot :



Observation : -

- **Insights :**

Seeing Month over Month count of orders for different payment types we can conclude most of the orders payment type is “ credit_card” for each month . Second highest is UPI payment type.

- **Recommendations :**

  So one can partner with credit card companies to offer more discounts and to increase sales also .

  Also Digital payment_type like UPI should be encouraged and promoted to use by offering more discounts , which will increase count of customers placing order through these payment type.

  Allowing discounts / cashback on products bought from “debit card” will further encourage users to pay from debit card

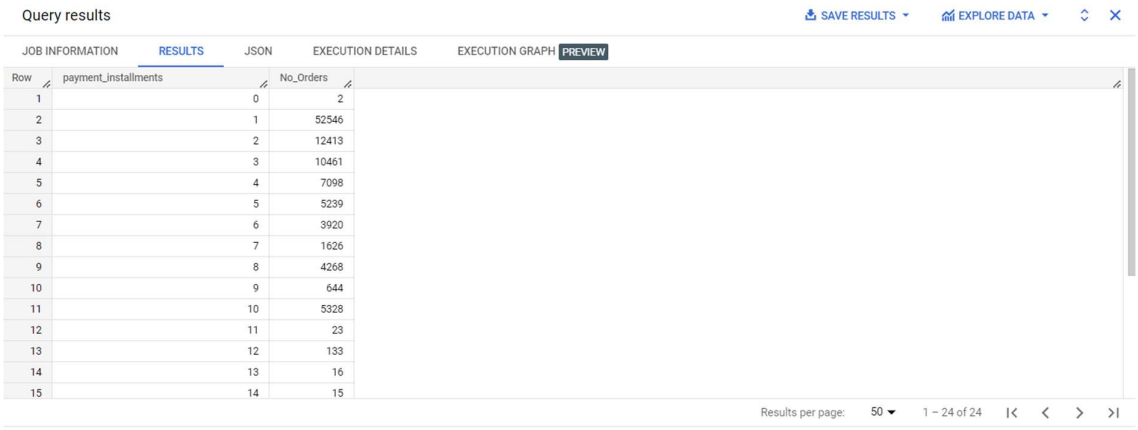
Ans 6.2 Count of orders based on the no. of payment installments

Query:

```
with Order_PaymentType as
(
select o.order_id ,  EXTRACT(MONTH FROM order_purchase_timestamp) Order_Month ,paym
ent_type , payment_installments
from `target_data.orders`  o join `target_data.payments` p on  o.order_id = p.order
_id
)
select payment_installments , Count(order_id) No_Orders
from Order_PaymentType
group by payment_installments
order by payment_installments
```

Note : I have reused same CTE used in Answer 6.1

Output Screenshot :



Observation : -

- **Insights :**

We can see that most of the orders are paid with payment installments of “1” .

- **Recommendations :**

We can increase the number of order in other categories of payment installments (>1) by offering NO extra charges and extra discount for customer opting for larger number of payment installments.

We can increase the number of orders further by introducing schemes like

- Giving instant cashback to customers who pay the amount in one go (installements =0) or to those customers who are opting for payment instalments of 12 or more .
- By tying up with banks to reduce charges on installments . Like 0% interest installements
- Target should start an program to award customers who are placing most orders through any payment installments.