Business Case-Study – ShopEase

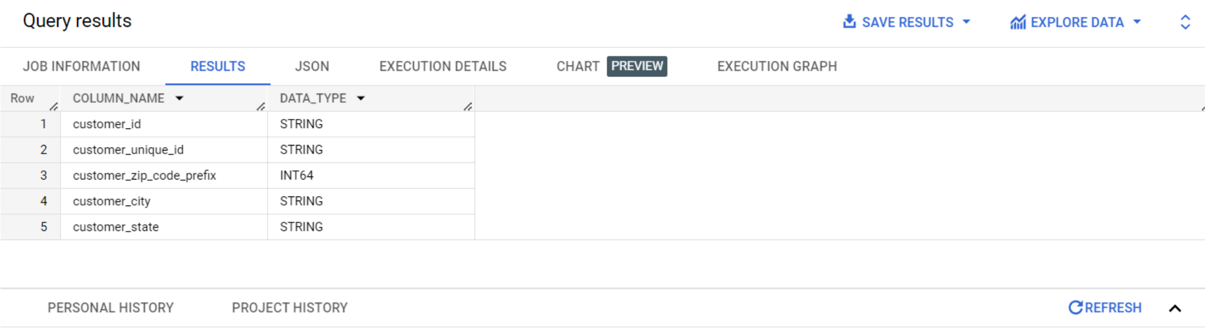
(SQL)

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

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

Query:

```
SELECT
  COLUMN_NAME, DATA_TYPE
FROM
  Ecommerce.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'customers'
```



Row	COLUMN_NAME	DATA_TYPE
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Inference:

There are 5 columns in customer table and the datatype of each as follow:

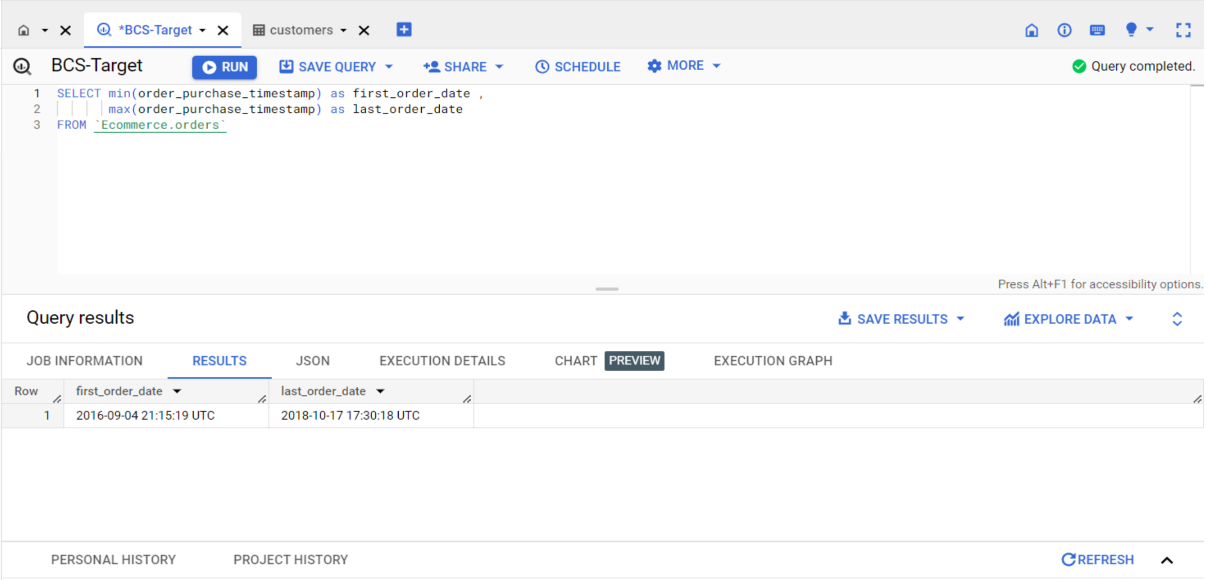
Column Name	Datatype
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

what is time range between which the orders were placed?

Query:

```
SELECT min(order_purchase_timestamp) as first_order_date, max(order_purchase_timestamp) as
last_order_date

FROM `Ecommerce.orders`
```



```
SELECT min(order_purchase_timestamp) as first_order_date ,
       max(order_purchase_timestamp) as last_order_date
FROM `Ecommerce.orders`
```

Row	first_order_date	last_order_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Inference:

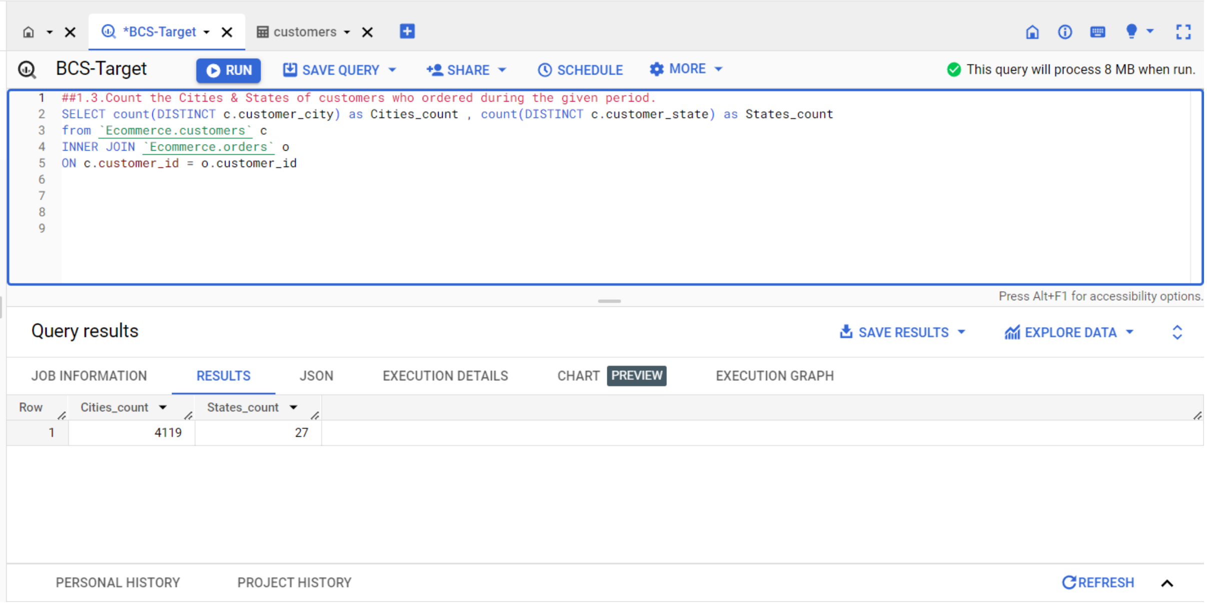
We can clearly see that First Order was placed at 04 September,2016 and Last Order was placed at 17 October,2018. And Difference between both orders in Day is 773 days.

What is the count of Cities & States of customers who ordered during the given period?

Query:

```
SELECT count(DISTINCT c.customer_city) as Cities_count, count(DISTINCT c.customer_state) as States_count

from `Ecommerce.customers` c
INNER JOIN `Ecommerce.orders` o
ON c.customer_id = o.customer_id
```



This query will process 8 MB when run.

```
##1.3.Count the Cities & States of customers who ordered during the given period.
SELECT count(DISTINCT c.customer_city) as Cities_count , count(DISTINCT c.customer_state) as States_count
from `Ecommerce.customers` c
INNER JOIN `Ecommerce.orders` o
ON c.customer_id = o.customer_id
```

Press Alt+F1 for accessibility options.

Query results

Row	Cities_count	States_count
1	4119	27

Inference:

There are 4119 and 27 unique cities and states of customer who ordered during given time. It means company reached most of the regions to fulfil customer's need and capturing more cities and states.

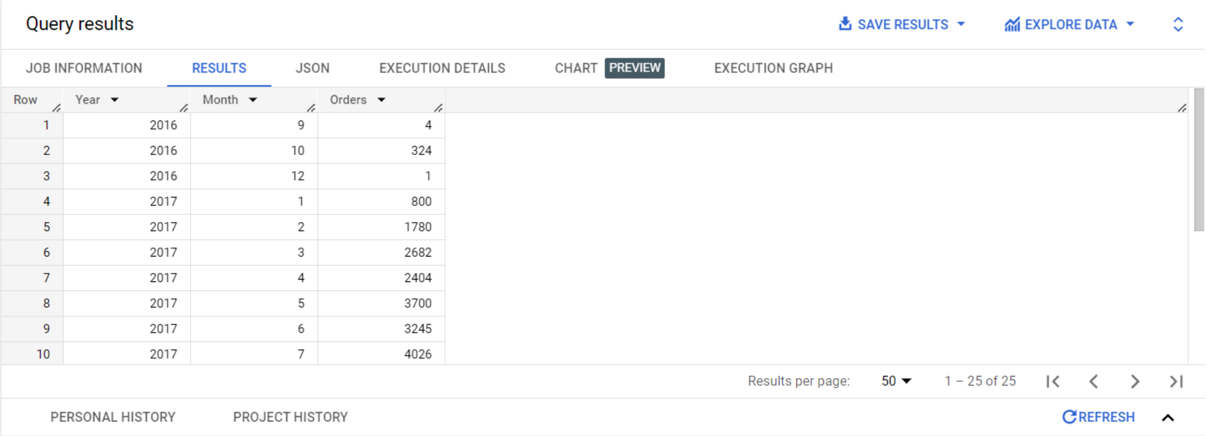
2.In-depth Exploration:

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

Query:

```
SELECT extract(Year from order_purchase_timestamp) as Year, extract(month from
order_purchase_timestamp) as Month, count(order_id) as Orders

FROM `Ecommerce.orders`
group by 1, 2
Order by 1, 2
```



Row	Year	Month	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

Results per page: 50 1 – 25 of 25

Inference:

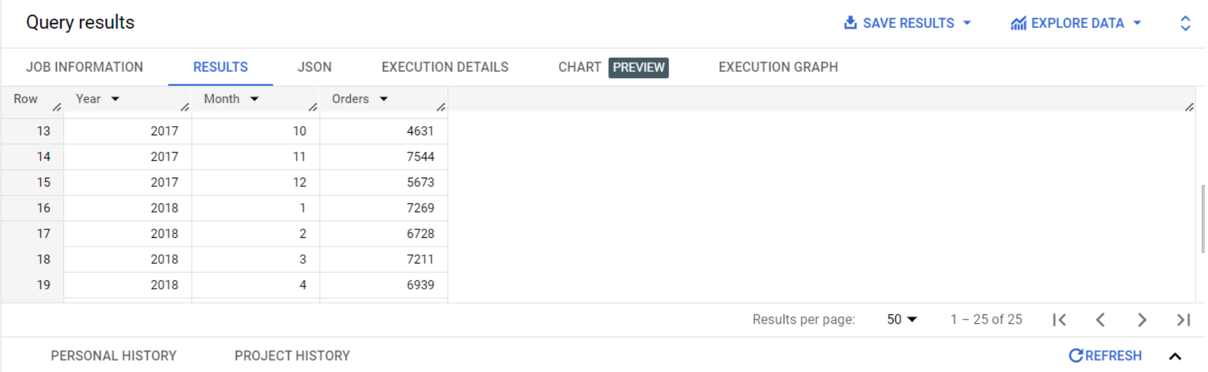
Initial 3 months orders were not recorded properly by operation teams of targets but after end of 2016 there was increase in trend of orders from January, 2016. There is highest number of orders were placed in month of November, 2017, it was the peak of growing trend. In December there was slight downfall and in January 2018, it again increased. After January 2018, sideways trend was started till August 2018. By the end of August there was massive downfall in orders may be because of Target was wrapping up their operations.

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

Query:

```
SELECT extract(Year from order_purchase_timestamp) as Year, extract(month from
order_purchase_timestamp) as Month, count(order_id) as Orders

FROM `Ecommerce.orders`
group by 1, 2
Order by 1, 2
```



Row	Year	Month	Orders
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728
18	2018	3	7211
19	2018	4	6939

Inference:

Yes, we can definitely see that there was a monthly seasonality in three Month in which number of orders placed were highest.

November, 2017: In this month, highest number of orders were placed and it was the peak may be because of Year Ending Sales, Christmas Offers provided by Target to the customers.

January, 2018: This month holds second number in terms of highest orders placed. It may be happened because of New Year Shopping were done by customers and New Year Sales offer provided by Target.

March, 2018: This month holds third number in terms of highest orders placed. As we know march month is a starting month of festivals so may be because of this reason there were highest number of orders were placed in this month.

During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

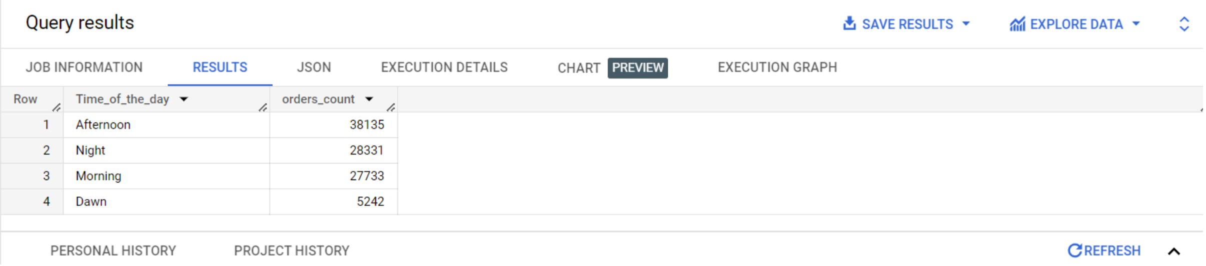
- 0-6 hrs: Dawn
- 7-12 hrs: Mornings
- 13-18 hrs: Afternoon
- 19-23 hrs: Night

Query:

```
With temp as

(
SELECT order_id, EXTRACT(hour FROM (order_purchase_timestamp)) AS time_
FROM `Ecommerce.orders`
)

SELECT ( CASE WHEN time_ between 0 and 6 then 'Dawn'
              WHEN time_ between 7 and 12 then 'Morning'
              WHEN time_ between 13 and 18 then 'Afternoon'
              WHEN time_ between 19 and 23 then 'Night'
              END) as Time_of_the_day, count(order_id) as orders_count
FROM temp
GROUP BY 1
ORDER BY 2 desc
```



Row	Time_of_the_day	orders_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Inference:

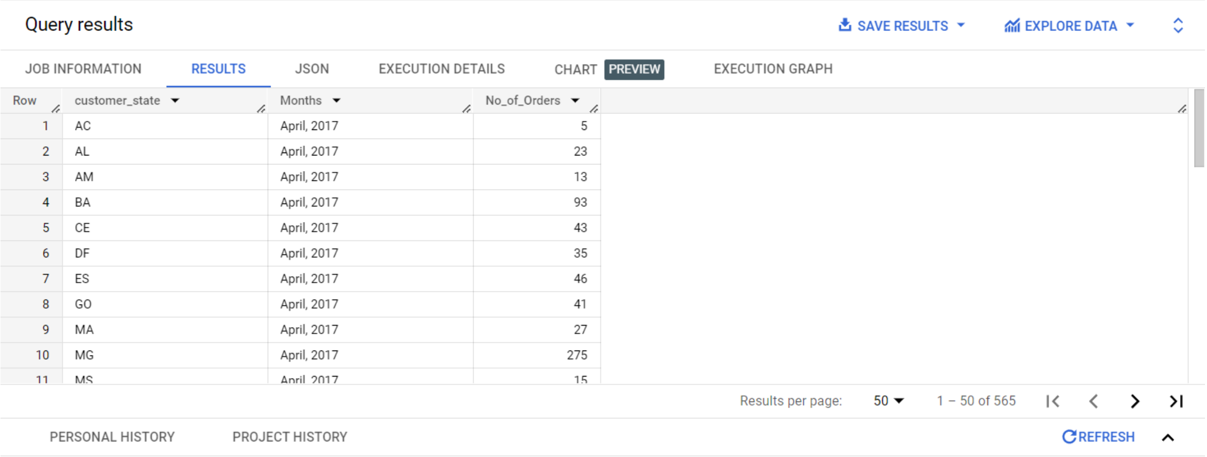
We can clearly see that The Brazilian customers mostly place their orders in Afternoon and Night rather than morning and dawn. And least orders place in dawn. Target has to make sure their websites are very fast and smooth during Afternoon and Night.

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

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

Query:

```
SELECT c.customer_state, FORMAT_TIMESTAMP('%B, %Y',o.order_purchase_timestamp) as Months,
count(o.order_id) as No_of_Orders
FROM `Ecommerce.customers` c
INNER JOIN `Ecommerce.orders` o
ON c.customer_id = o.customer_id
GROUP BY 1, 2
ORDER BY 2, 1
```



Row	customer_state	Months	No_of_Orders
1	AC	April, 2017	5
2	AL	April, 2017	23
3	AM	April, 2017	13
4	BA	April, 2017	93
5	CE	April, 2017	43
6	DF	April, 2017	35
7	ES	April, 2017	46
8	GO	April, 2017	41
9	MA	April, 2017	27
10	MG	April, 2017	275
11	MS	April, 2017	15

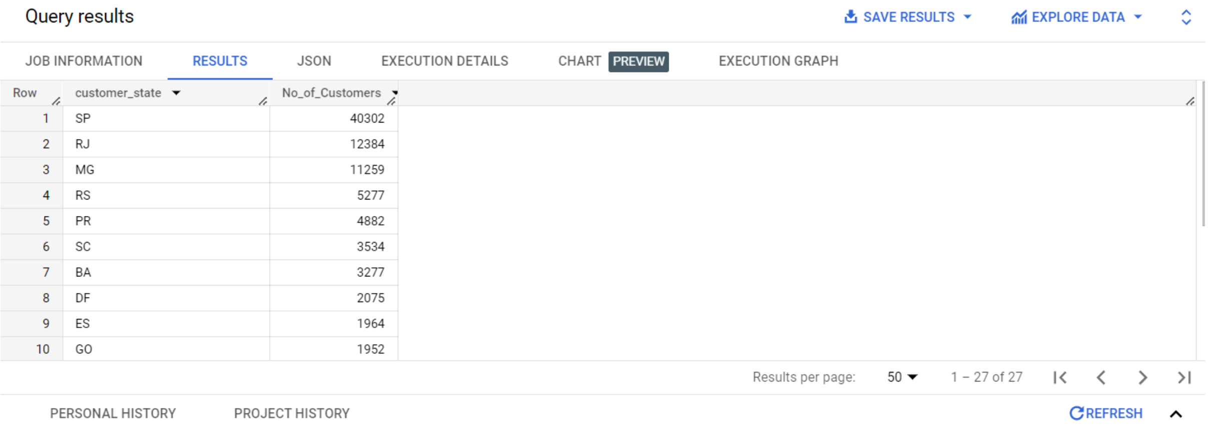
Inference:

There are 26 states in Brazil. As per data Sao Paulo state have highest number of Order placed followed by Rio De Janeiro and Minas Gerais.

How are the customers distributed across all the states?

Query:

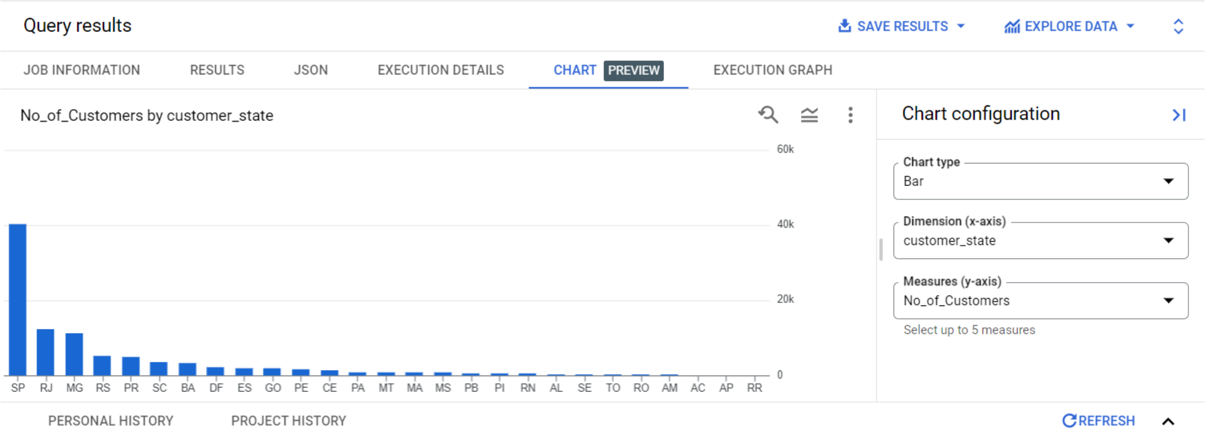
```
SELECT customer_state, count(distinct customer_unique_id) as No_of_Customers
FROM `Ecommerce.customers`
GROUP BY 1
ORDER BY 2 desc
```



Query results

Row	customer_state	No_of_Customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

Results per page: 50 1 – 27 of 27



Query results

Chart configuration

Chart type: Bar

Dimension (x-axis): customer_state

Measures (y-axis): No_of_Customers

Select up to 5 measures

Inference:

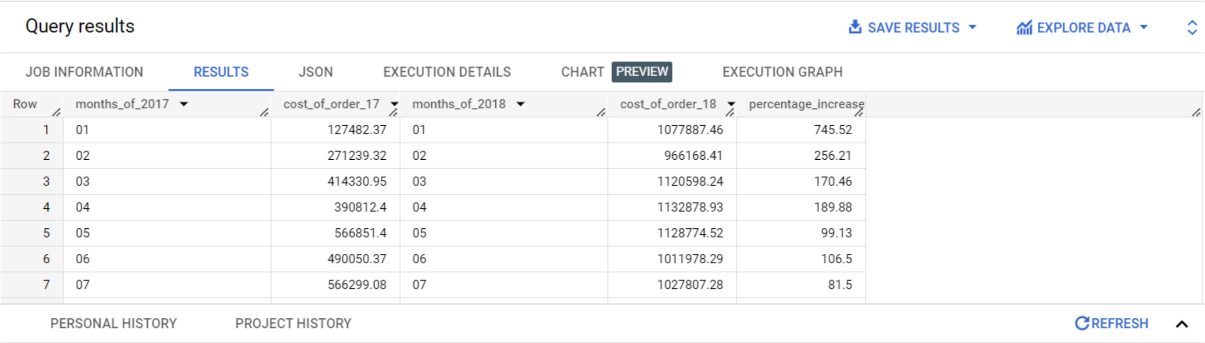
With the help of Bar Chart, we can see that highest number of unique customers are in present in Sao Paulo State and in Rio De Janeiro, and Minas Gerais state 10k+ unique customers are present. Least Customers are present in Roraima State may be because of lack of marketing strategies or unable to reach out to the Customers that that region by Target. Thus, Target should focus on that region by Social Media Platforms and Advertisements.

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

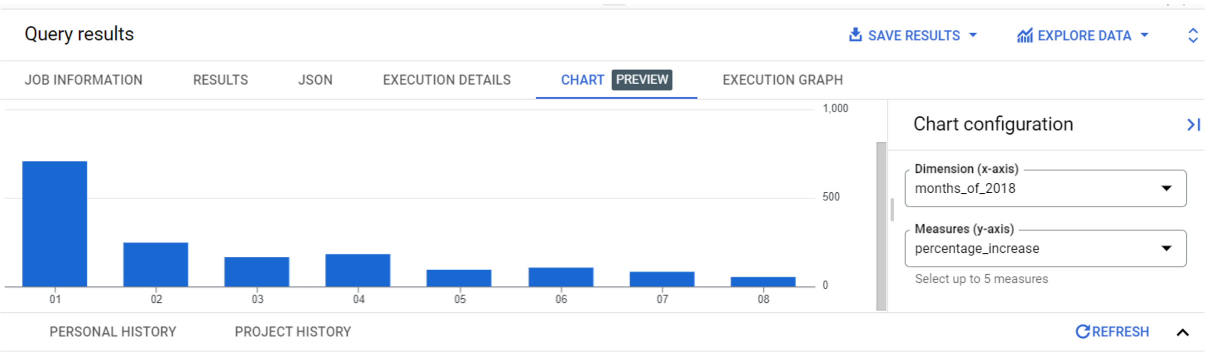
Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

Query:

```
WITH a as
(
SELECT format_timestamp('%m',o.order_purchase_timestamp) as
months_of_2017,format_timestamp('%Y',o.order_purchase_timestamp) as Year_2017,
round(sum(oi.price+oi.freight_value),2) as cost_of_order_17
FROM `Ecommerce.order_items` oi
INNER JOIN `Ecommerce.orders` o
ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY 1,2
HAVING Year_2017 like '%2017' and months_of_2017 between '01' and '08'
ORDER BY 1,2
),
b as
(
SELECT format_timestamp('%m',o.order_purchase_timestamp) as months_of_2018
,format_timestamp('%Y',o.order_purchase_timestamp) as Year_2018,
round(sum(oi.price+oi.freight_value),2) as cost_of_order_18
FROM `Ecommerce.order_items` oi
INNER JOIN `Ecommerce.orders` o
ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY 1 ,2
HAVING Year_2018 like '%2018' and months_of_2018 between '01' and '08'
ORDER BY 1 ,2
)
SELECT A.months_of_2017,A.cost_of_order_17, B.months_of_2018, B.cost_of_order_18,
round(((B.cost_of_order_18-A.cost_of_order_17)/(A.cost_of_order_17))*100,2) as
percentage_increase
FROM a A
JOIN b B
ON A.months_of_2017= B.months_of_2018
ORDER BY A.months_of_2017
```



Row	months_of_2017	cost_of_order_17	months_of_2018	cost_of_order_18	percentage_increase
1	01	127482.37	01	1077887.46	745.52
2	02	271239.32	02	966168.41	256.21
3	03	414330.95	03	1120598.24	170.46
4	04	390812.4	04	1132878.93	189.88
5	05	566851.4	05	1128774.52	99.13
6	06	490050.37	06	1011978.29	106.5
7	07	566299.08	07	1027807.28	81.5



Dimension (x-axis): months_of_2018

Measures (y-axis): percentage_increase

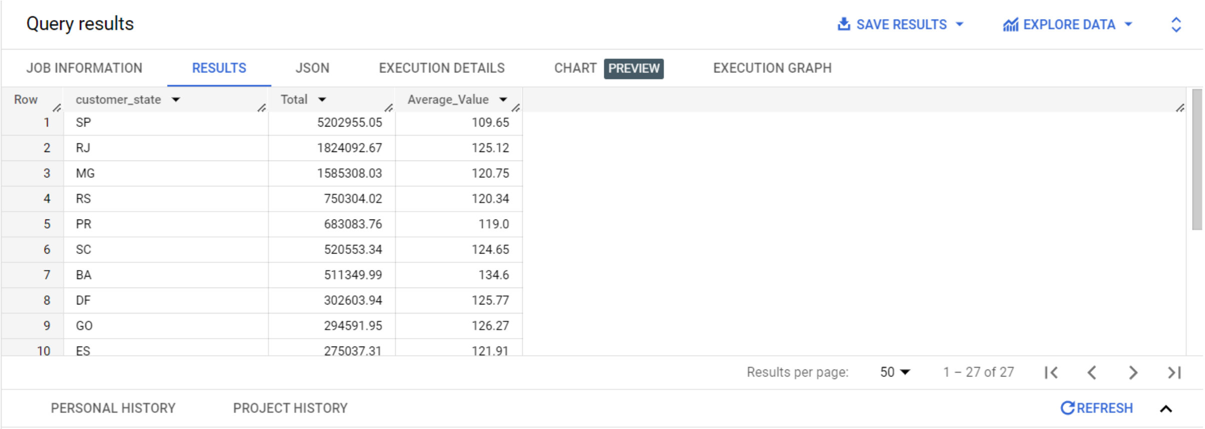
Inference:

The percentage increase in cost of orders in January of 2018 compare to January 2017 is the highest (745.52%) and after that downfall has shown in percentage increase in February month and after that downtrend took place till August which was good for Target.

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

Query:

```
SELECT c.customer_state , round(sum(oi.price),2) as Total ,
round(avg(oi.price),2) as Average_Value
FROM `Ecommerce.order_items` oi
JOIN `Ecommerce.orders` o
ON oi.order_id = o.order_id
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 desc
```



Row	customer_state	Total	Average_Value
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

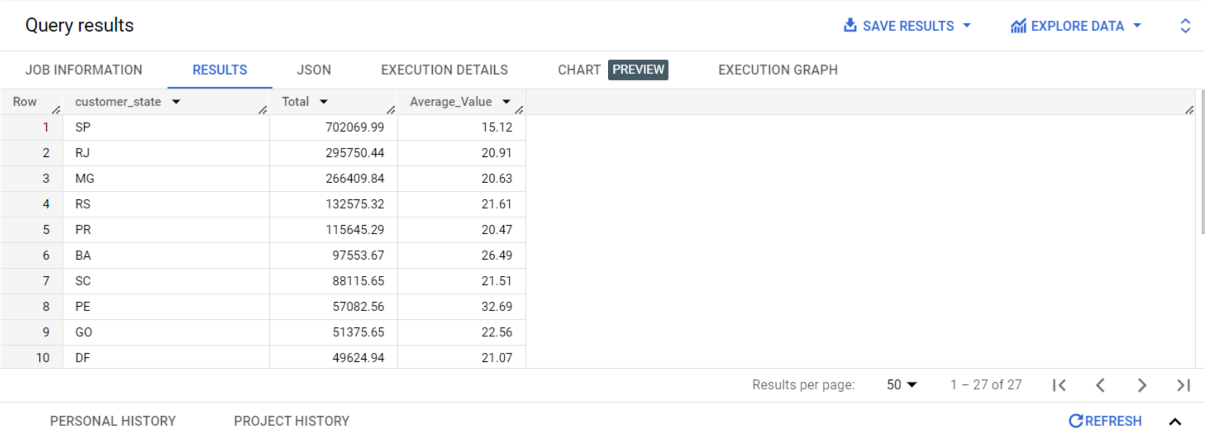
Inference:

Highest total and average of order price is in Sao Paulo state and least total of order price is in Roraima state.

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

Query:

```sql
SELECT c.customer_state, round(sum(oi.freight_value),2) as Total,
round(avg(oi.freight_value),2) as Average_Value
FROM `Ecommerce.order_items` oi
JOIN `Ecommerce.orders` o
ON oi.order_id = o.order_id
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
Where o.order_status = 'delivered'
GROUP BY 1
ORDER BY 2 desc, 3 desc
```



Row	customer_state	Total	Average_Value
1	SP	702069.99	15.12
2	RJ	295750.44	20.91
3	MG	266409.84	20.63
4	RS	132575.32	21.61
5	PR	115645.29	20.47
6	BA	97553.67	26.49
7	SC	88115.65	21.51
8	PE	57082.56	32.69
9	GO	51375.65	22.56
10	DF	49624.94	21.07

Results per page: 50 1 – 27 of 27

Inference:

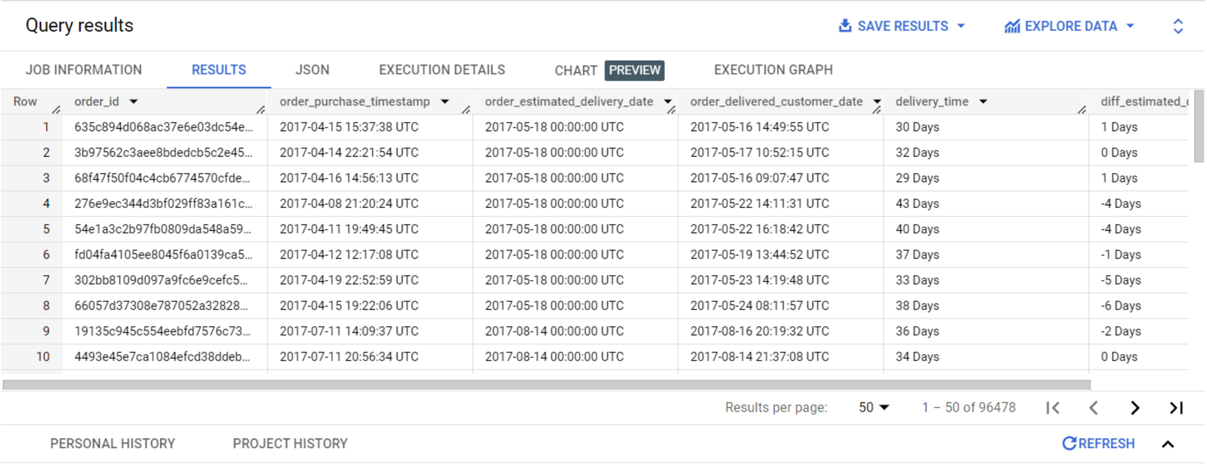
Highest total and average of order freight is in Sao Paulo state and least total of order freight is in Roraima state.

5.Analysis based on sales, freight and delivery time:

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.

Query:

```
SELECT order_id, order_purchase_timestamp, order_estimated_delivery_date,
order_delivered_customer_date ,
concat(datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day),'
','Days') as delivery_time,
concat(datetime_diff(order_estimated_delivery_date,order_delivered_customer_date,day),'
','Days') as diff_estimated_delivery
FROM `Ecommerce.orders`
where order_status = 'delivered'
```



Row	order_id	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	delivery_time	diff_estimated_(
1	635c894d068ac37e6e03dc54e...	2017-04-15 15:37:38 UTC	2017-05-18 00:00:00 UTC	2017-05-16 14:49:55 UTC	30 Days	1 Days
2	3b97562c3aee8bdedcb5c2e45...	2017-04-14 22:21:54 UTC	2017-05-18 00:00:00 UTC	2017-05-17 10:52:15 UTC	32 Days	0 Days
3	68f47f50f04c4cb6774570cfde...	2017-04-16 14:56:13 UTC	2017-05-18 00:00:00 UTC	2017-05-16 09:07:47 UTC	29 Days	1 Days
4	276e9ec344d3bf029ff83a161c...	2017-04-08 21:20:24 UTC	2017-05-18 00:00:00 UTC	2017-05-22 14:11:31 UTC	43 Days	-4 Days
5	54e1a3c2b97fb0809da548a59...	2017-04-11 19:49:45 UTC	2017-05-18 00:00:00 UTC	2017-05-22 16:18:42 UTC	40 Days	-4 Days
6	fd04fa4105ee8045f6a0139ca5...	2017-04-12 12:17:08 UTC	2017-05-18 00:00:00 UTC	2017-05-19 13:44:52 UTC	37 Days	-1 Days
7	302bb8109d097a9fc6e9cefc5...	2017-04-19 22:52:59 UTC	2017-05-18 00:00:00 UTC	2017-05-23 14:19:48 UTC	33 Days	-5 Days
8	66057d37308e787052a32828...	2017-04-15 19:22:06 UTC	2017-05-18 00:00:00 UTC	2017-05-24 08:11:57 UTC	38 Days	-6 Days
9	19135c945c554eebfd7576c73...	2017-07-11 14:09:37 UTC	2017-08-14 00:00:00 UTC	2017-08-16 20:19:32 UTC	36 Days	-2 Days
10	4493e45e7ca1084efcd38ddeb...	2017-07-11 20:56:34 UTC	2017-08-14 00:00:00 UTC	2017-08-14 21:37:08 UTC	34 Days	0 Days

Results per page: 50 1 – 50 of 96478

Inference:

Delivery time should be less so that customers will not have longer. And it will also attract more customers.

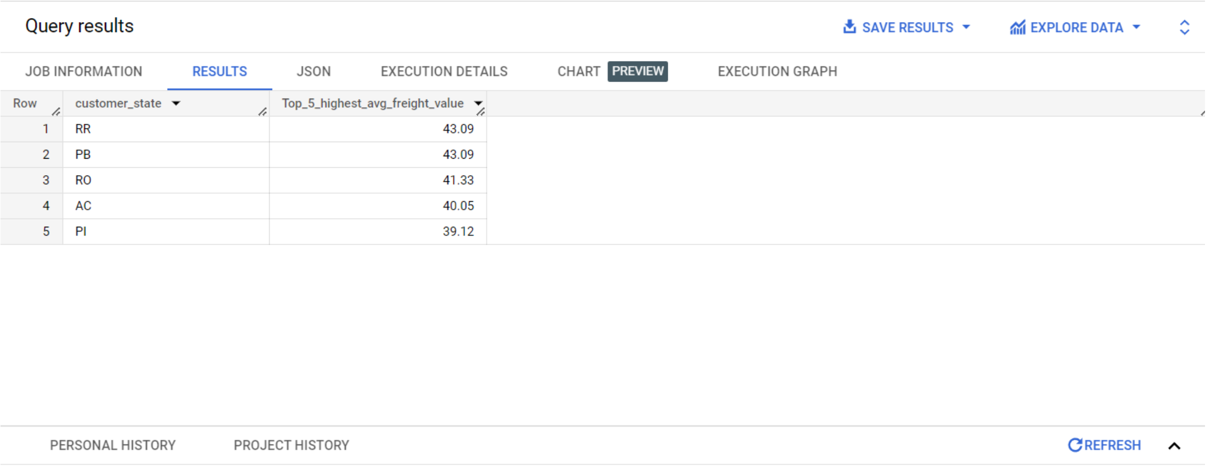
Find out the top 5 states with the highest & lowest average freight value.

Query:

```
#QUERY for top 5 highest average freight value:
SELECT c.customer_state, round(avg(oi.freight_value),2) as Top_5_highest_avg_freight_value
FROM `Ecommerce.order_items` oi
JOIN `Ecommerce.orders` o
ON oi.order_id = o.order_id
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
where o.order_status = 'delivered'
GROUP BY 1
ORDER BY 2 desc
LIMIT 5


#QUERY for top 5 lowest average freight value:
SELECT c.customer_state, round(avg(oi.freight_value),2) as Top_5_lowest_avg_freight_value
FROM `Ecommerce.order_items` oi
JOIN `Ecommerce.orders` o
ON oi.order_id = o.order_id
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
where o.order_status = 'delivered'
GROUP BY 1
ORDER BY 2 asc
LIMIT 5
```

Top 5 Highest:

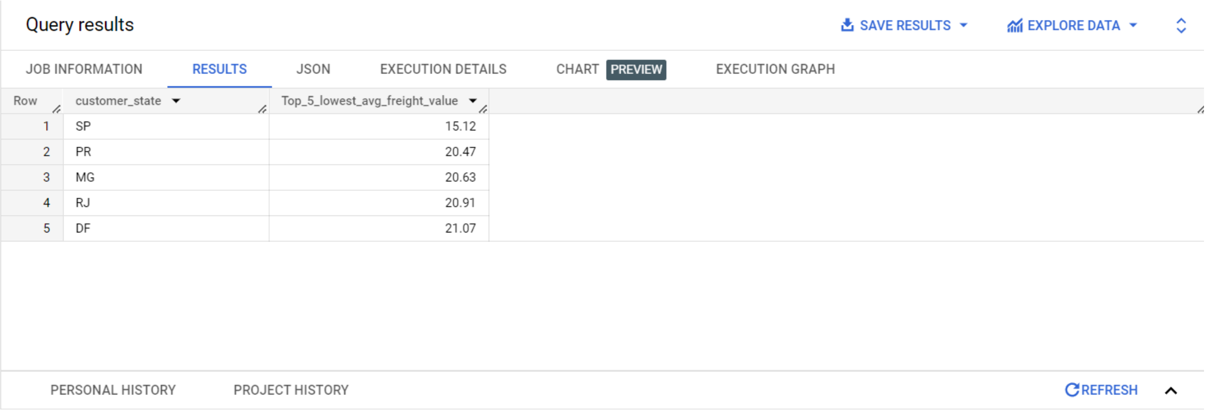


Row	customer_state	Top_5_highest_avg_freight_value
1	RR	43.09
2	PB	43.09
3	RO	41.33
4	AC	40.05
5	PI	39.12

Inference:

Highest average freight value is in (43.09) and in Paraba State (43.09) followed by Rondonia (41.33), Acre (40.05) and Piaui (39.12). It can be Minimize in this region by Target. Because of this more customers will start to order.

Top 5 Lowest:



Row	customer_state	Top_5_lowest_avg_freight_value
1	SP	15.12
2	PR	20.47
3	MG	20.63
4	RJ	20.91
5	DF	21.07

Inference:

Lowest average freight value is in Sau Paulo State (15.12) followed by Parana (20.47), Minas Gerais (20.63), Rio De Janeiro (20.91) and Distrito Federal State (21.07).

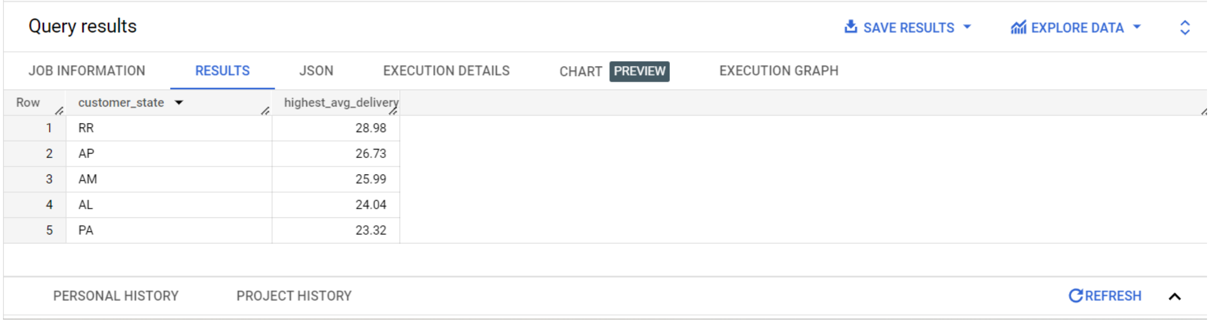
Find out the top 5 states with the highest & lowest average delivery time.

Query:

```
#QUERY for top 5 highest average delivery time:
SELECT c.customer_state, round(avg(o.delivery_time),2) as highest_avg_delivery_time
FROM (
      SELECT *, timestamp_diff(order_delivered_customer_date, order_purchase_timestamp,day)
as delivery_time
      FROM `Ecommerce.orders`
      where order_status = 'delivered'
     ) o
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 desc
LIMIT 5


#QUERY for top 5 lowest average delivery time:
SELECT c.customer_state, round(avg(o.delivery_time),2) as lowest_avg_delivery_time
FROM (
      SELECT *, timestamp_diff(order_delivered_customer_date, order_purchase_timestamp,day)
as delivery_time
      FROM `Ecommerce.orders`
      where order_status = 'delivered'
     ) o
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 asc
LIMIT 5
```

Top 5 Highest:

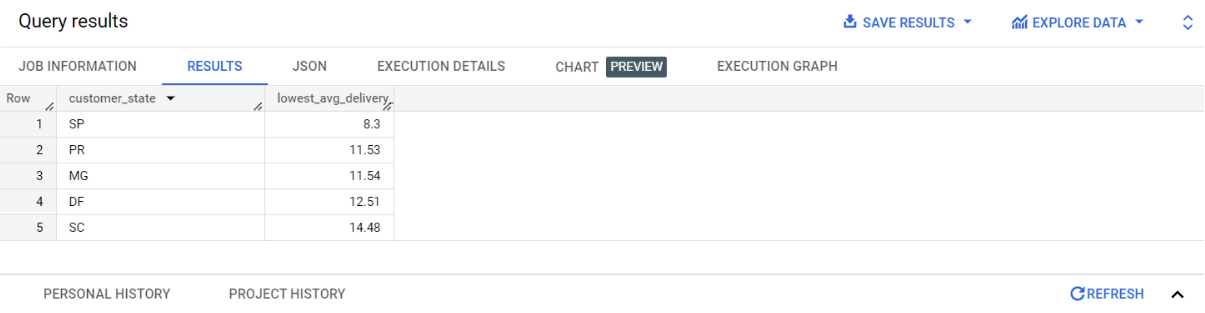


Row	customer_state	highest_avg_delivery
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

Inference:

Highest average delivery time taken to deliver orders are in Roraima State (28.98) followed by Amapa (26.73), Amazonas (25.99), Alagoas (24.04) and Para State (23.32). Delivery time should have to be minimize in these regions by Target. In these Regions customers have to wait longer compare to other regions.

Top 5 Lowest:



Row	customer_state	lowest_avg_delivery
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

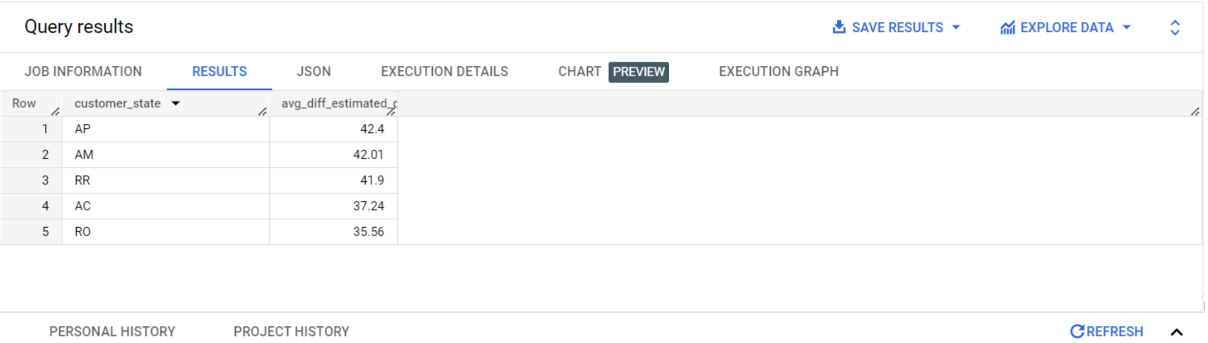
Inference:

Lowest average delivery time have in Sao Paulo (8.3) followed by Parana (11.53), Minas Gerais (11.54), Distrito Federal (12.51), Santa Catarina State (14.48). It means more orders come from these regions compare to other region because the delivery time is less. And customers don’t have wait longer.

Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Query:

```
SELECT c.customer_state, round(avg(o.diff_estimated_del_actual_del),2) as
avg_diff_estimated_del_actual_del
FROM (
      SELECT *, timestamp_diff(order_estimated_delivery_date,
order_delivered_carrier_date,day) as diff_estimated_del_actual_del
      FROM `Ecommerce.orders`
      where order_status = 'delivered'
     ) o
JOIN `Ecommerce.customers` c
ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 desc
LIMIT 5
```



Row	customer_state	avg_diff_estimated_c
1	AP	42.4
2	AM	42.01
3	RR	41.9
4	AC	37.24
5	RO	35.56

Inference:

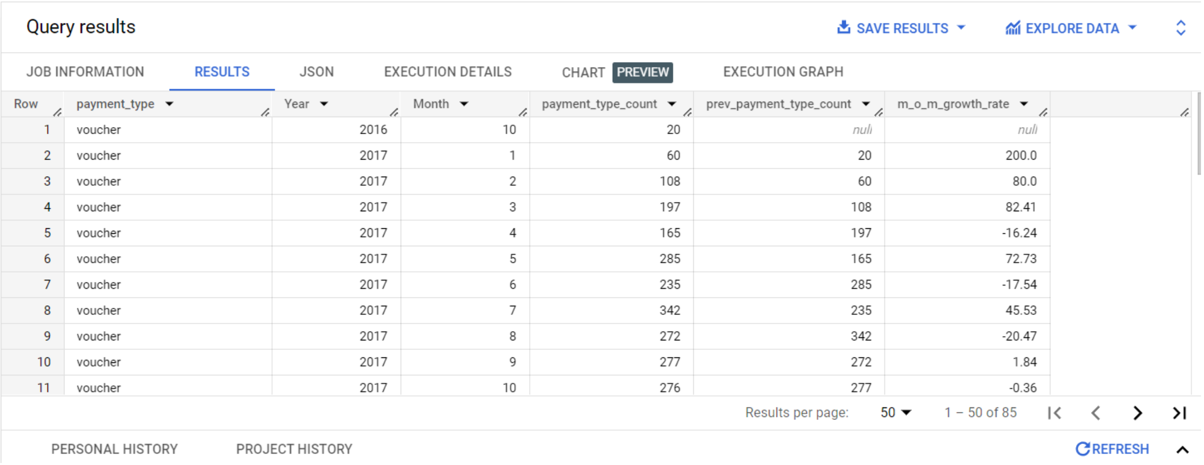
There are top 5 states where Average delivery time is fast compared to estimated delivery time. Amapa State (42.4) has fastest Average delivery time compared to estimated delivery time followed by Amazonas (42.01), Roraima (41.9), Acre (37.24) and Rondonia (35.56). It is green flag to Target and customers will increase in these regions.

6.Analysis based on the payments:

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

Query:

```
SELECT temp2.*, round(((temp2.payment_type_count -
temp2.prev_payment_type_count)/temp2.prev_payment_type_count)*100,2) as m_o_m_growth_rate
FROM (SELECT temp1.*, lag(temp1.payment_type_count)over(partition by temp1.payment_type
order by temp1.Year,temp1.Month) as prev_payment_type_count
FROM (SELECT distinct p.payment_type, o.Year, o.Month, count(p.payment_type)over(partition
by p.payment_type, o.Year, o.Month order by o.Year, o.
      Month) as payment_type_count
FROM `Ecommerce.payments` p
JOIN (SELECT order_id, EXTRACT(month from order_purchase_timestamp) as Month, EXTRACT(year
from order_purchase_timestamp) as Year
      FROM `Ecommerce.orders`
      where order_status='delivered'
     ) o
ON p.order_id = o.order_id ) temp1 ) temp2
```



Query results

Row	payment_type	Year	Month	payment_type_count	prev_payment_type_count	m_o_m_growth_rate
1	voucher	2016	10	20	null	null
2	voucher	2017	1	60	20	200.0
3	voucher	2017	2	108	60	80.0
4	voucher	2017	3	197	108	82.41
5	voucher	2017	4	165	197	-16.24
6	voucher	2017	5	285	165	72.73
7	voucher	2017	6	235	285	-17.54
8	voucher	2017	7	342	235	45.53
9	voucher	2017	8	272	342	-20.47
10	voucher	2017	9	277	272	1.84
11	voucher	2017	10	276	277	-0.36

Results per page: 50 1 – 50 of 85

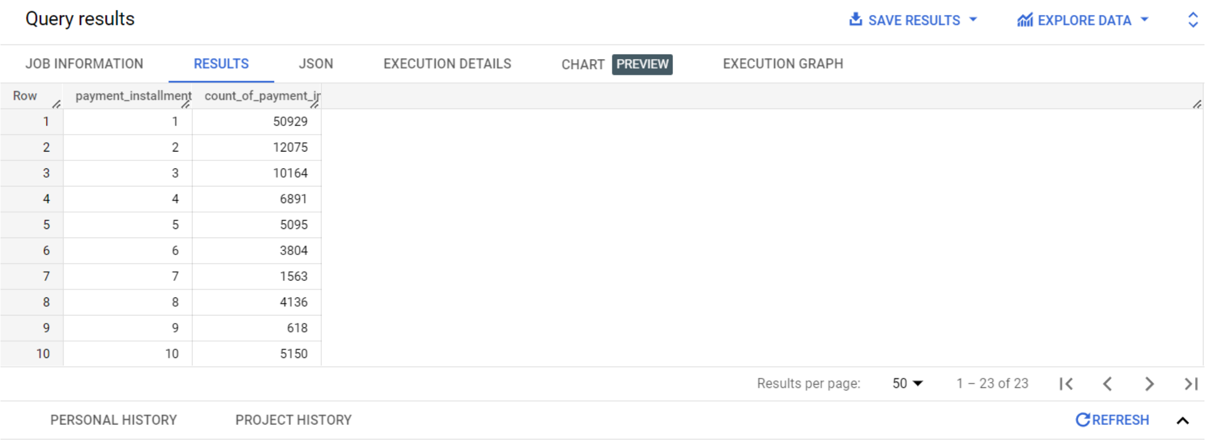
Inference:

Credit Card users are more in Brazil therefore, the most preferred payment type by customers is Credit card with UPI and Vouchers. In Brazil ‘Buy Now, Pay Later’ scheme is very popular that’s why most customers preferred to pay their amounts in installments through Credit Cards. This is the reason that most customers use Credit Cards for payments. Digital Transactions are very easy and fast and Brazil’s customers prefer E-wallets like Google Pay, Apple Pay, PayPal etc. is the reason that UPI payments are also a good option for the Brazilians customers.

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

Query:

```
SELECT payment_installments, count(payment_installments) as count_of_orders
FROM `Ecommerce.payments` p
JOIN (SELECT *
      FROM `Ecommerce.orders`
      WHERE order_status = 'delivered') o
ON p.order_id = o.order_id
where payment_installments != 0
GROUP BY 1
```



Query results

Row	payment_installment	count_of_payment_ir
1	1	50929
2	2	12075
3	3	10164
4	4	6891
5	5	5095
6	6	3804
7	7	1563
8	8	4136
9	9	618
10	10	5150

Results per page: 50   1 – 23 of 23

Inference:

We can see that most of the payments are made in small installments by customers.