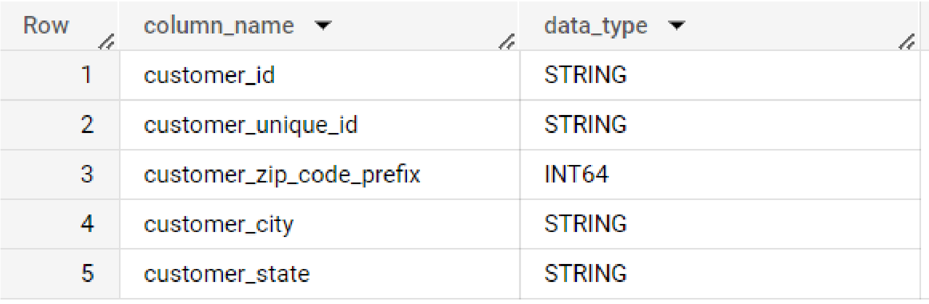
TARGET: CASESTUDY

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.

ANS:

```
select column_name,data_type from Target.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

Insights:

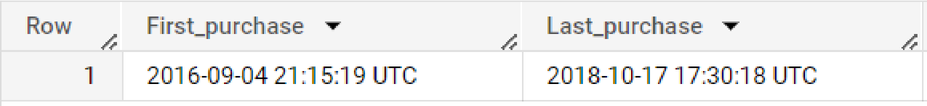
String type – These are the columns with array/list of characters stored.

Integer type – These are the columns with numbers where both positive and negative stored without any decimal value.

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

ANS:

```
select
min(order_purchase_timestamp) as First_purchase,
max(order_purchase_timestamp) as Last_purchase
from `Target.orders`
```



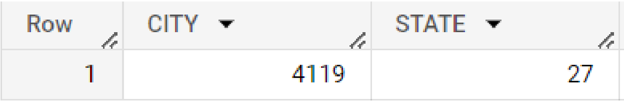
Row	First_purchase	Last_purchase
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights: First Purchase was made on 2016 and last purchase on 2018

3. Count the Cities & States of customers who ordered during the given period

ANS:

```
select
count (distinct c.customer_city ) as CITY,
count (distinct c.customer_state ) as STATE
from `Target.customers` c
join `Target.orders` o on c.customer_id = o.customer_id
```



Row	CITY	STATE
1	4119	27

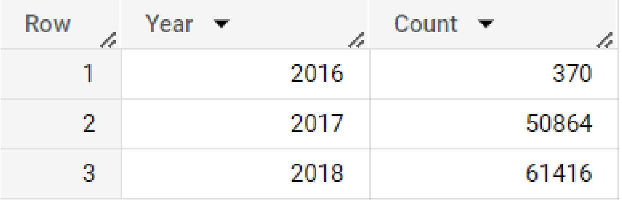
INSIGHTS: Most customer orders come from rio de janerio, Brasilia, Curitiba, belo horizonte, sao paulo

2.In-depth Exploration:

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

ANS:

```
select EXTRACT (YEAR from o.order_purchase_timestamp) as Year,count(*) as Count
from `Target.orders` o
join `Target.order_items` oi on o.order_id = oi.order_id
group by Year
order by Year
```



Row	Year	Count
1	2016	370
2	2017	50864
3	2018	61416

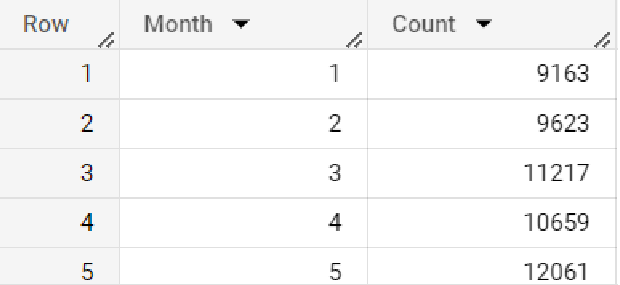
INSIGHTS - It can be noticed that there is a considerable increase in the number of orders year by year from 2016 to 2018.

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

ANS:

```
select EXTRACT (MONTH from o.order_purchase_timestamp) as Month,count(*) as Count

from `Target.orders` o
join `Target.order_items` oi on o.order_id = oi.order_id
group by Month
order by Month
```



Row	Month	Count
1	1	9163
2	2	9623
3	3	11217
4	4	10659
5	5	12061

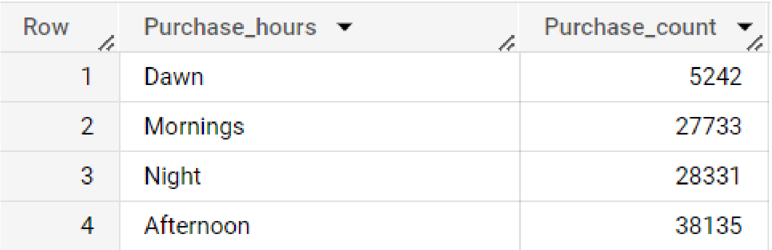
INSIGHTS: As seen in the table, the order count peaked in the month of August .It can be clearly observed that from the month of May till the month of August, the sales were on peak and this is mostly from end of summer till the end of rainy season.

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

ANS:

```
select
CASE
  WHEN EXTRACT (HOUR from order_purchase_timestamp) between 0 and 6 then 'Dawn'
  WHEN EXTRACT (HOUR from order_purchase_timestamp) between 7 and 12 then
'Mornings'
  WHEN EXTRACT (HOUR from order_purchase_timestamp) between 13 and 18 then
'Afternoon'
  WHEN EXTRACT (HOUR from order_purchase_timestamp) between 19 and 23 then 'Night'
END as Purchase_hours,
count (*) as Purchase_count
from
`Target.orders`
group by Purchase_hours
order by Purchase_count
```



Row	Purchase_hours	Purchase_count
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

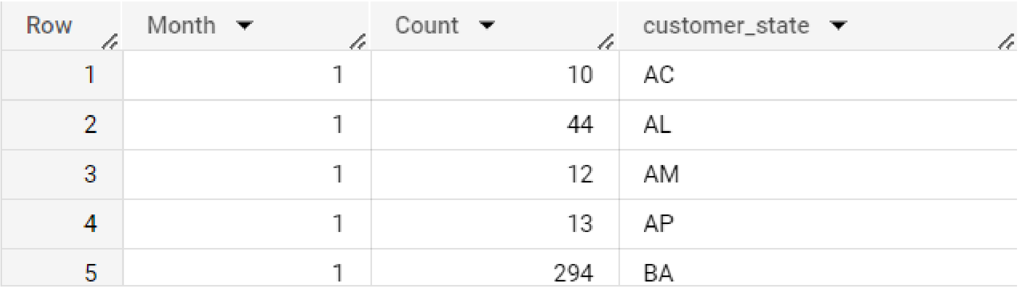
INSIGHTS: From the table it can clearly be said that people tend to place most orders in the afternoon followed by night.

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

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

ANS:

```
select EXTRACT (MONTH from o.order_purchase_timestamp) as Month,count(*) as
Count,c.customer_state,
from `Target.orders` o
join `Target.order_items` oi on o.order_id = oi.order_id
join `Target.customers` c on o.customer_id = c.customer_id
group by Month,c.customer_state
order by Month,c.customer_state
```



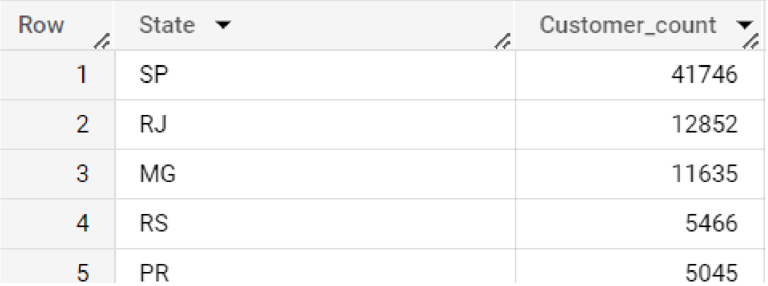
Row	Month	Count	customer_state
1	1	10	AC
2	1	44	AL
3	1	12	AM
4	1	13	AP
5	1	294	BA

INSIGHTS: Most orders were placed in the state SP during the month of August

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

ANS:

```
select
customer_state as State,
count (*) as Customer_count
from
`Target.customers`
group by customer_state
order by Customer_count desc
```



Row	State	Customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045

INSIGHTS: A good portion of customers are located in the states SP,RJ AND MG

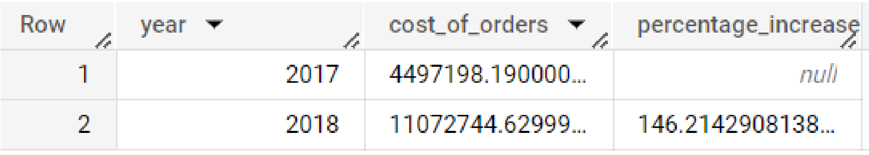
4.Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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.

ANS:

```
WITH order_cost_by_year AS (
    SELECT
        EXTRACT(YEAR FROM ors.order_purchase_timestamp) AS year,
        SUM(tp.payment_value) AS cost_of_orders
    FROM
        `Target.payments` tp
    JOIN
        `Target.order_items` oi ON oi.order_id = tp.order_id
    JOIN
        `Target.orders` ors ON ors.order_id = tp.order_id
    WHERE
        EXTRACT(MONTH FROM ors.order_purchase_timestamp) BETWEEN 1 AND 8
    GROUP BY
        year
)
```

```
SELECT
    ocy.year,
    ocy.cost_of_orders,
    (ocy.cost_of_orders - LAG(ocy.cost_of_orders) OVER (ORDER BY ocy.year)) /
LAG(ocy.cost_of_orders) OVER (ORDER BY ocy.year) * 100 AS percentage_increase
FROM
    order_cost_by_year ocy
WHERE
    ocy.year IN (2017, 2018)
ORDER BY
    ocy.year;
```



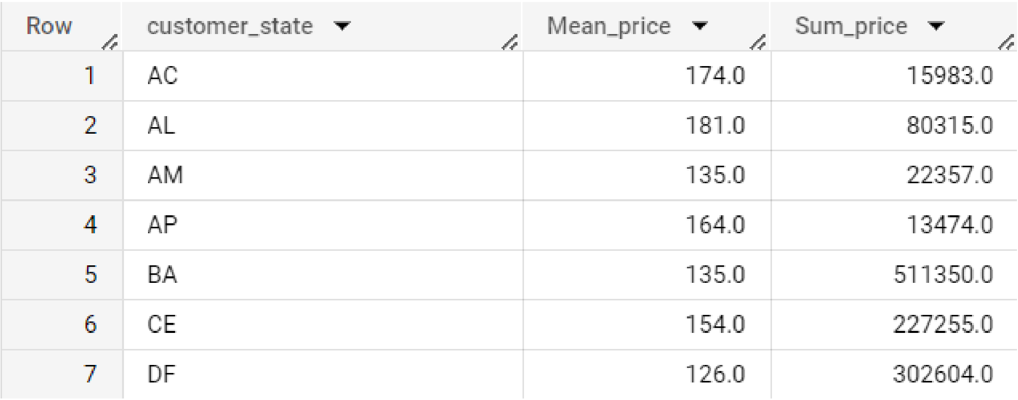
Row	year	cost_of_orders	percentage_increase
1	2017	4497198.190000...	null
2	2018	11072744.62999...	146.2142908138...

INSIGHTS: Thus, there is an increase of 146% in cost of orders considering the months from January to August.

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

ANS:

```
SELECT
    c.customer_state,
    ROUND(AVG(oi.price)) AS Mean_price,
    ROUND(SUM(oi.price)) AS Sum_price
FROM
    `Target.customers` c
INNER JOIN
    `Target.orders` o ON c.customer_id = o.customer_id
INNER JOIN
    `Target.order_items` oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_state
ORDER BY
    c.customer_state;
```



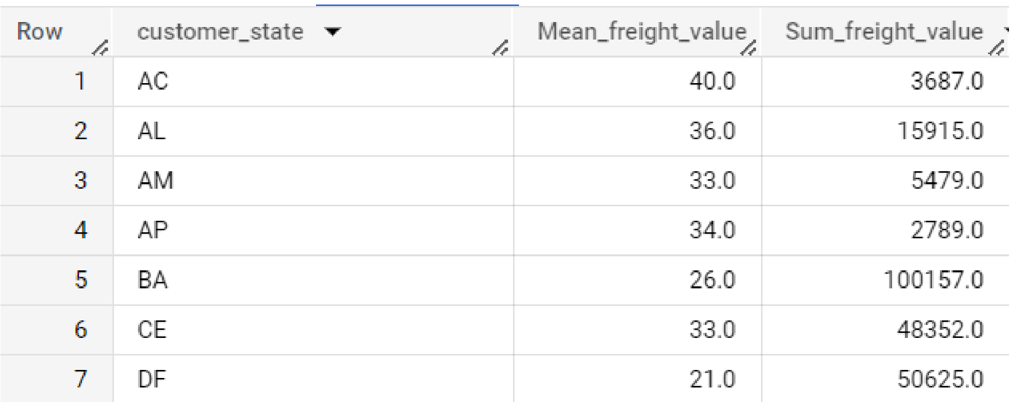
Row	customer_state	Mean_price	Sum_price
1	AC	174.0	15983.0
2	AL	181.0	80315.0
3	AM	135.0	22357.0
4	AP	164.0	13474.0
5	BA	135.0	511350.0
6	CE	154.0	227255.0
7	DF	126.0	302604.0

INSIGHTS: Mean price is highest for state with code PB. Sum price is highest for state with code SP.

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

ANS:

```
SELECT
    c.customer_state,
    ROUND(AVG(oi.freight_value)) AS Mean_freight_value,
    ROUND(SUM(oi.freight_value)) AS Sum_freight_value
FROM
    `Target.customers` c
INNER JOIN
    `Target.orders` o ON c.customer_id = o.customer_id
INNER JOIN
    `Target.order_items` oi ON o.order_id = oi.order_id
GROUP BY
    c.customer_state
ORDER BY
    c.customer_state;
```



Row	customer_state	Mean_freight_value	Sum_freight_value
1	AC	40.0	3687.0
2	AL	36.0	15915.0
3	AM	33.0	5479.0
4	AP	34.0	2789.0
5	BA	26.0	100157.0
6	CE	33.0	48352.0
7	DF	21.0	50625.0

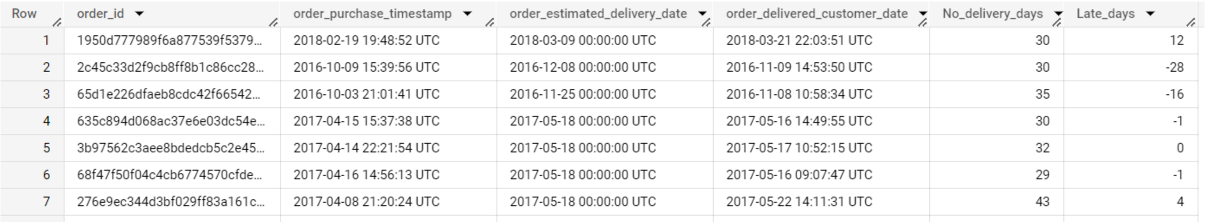
INSIGHTS: Mean freight value is highest for state with code PB. Sum freight value is highest for state with code SP

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.

ANS:

```
SELECT
    order_id,
    order_purchase_timestamp,
    order_estimated_delivery_date,
    order_delivered_customer_date,
    TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
No_delivery_days,
    TIMESTAMP_DIFF(order_delivered_customer_date, order_estimated_delivery_date,
DAY) AS Late_days
FROM
    `Target.orders`;
```



Row	order_id	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	No_delivery_days	Late_days
1	1950d777989f6a877539f5379...	2018-02-19 19:48:52 UTC	2018-03-09 00:00:00 UTC	2018-03-21 22:03:51 UTC	30	12
2	2c45c33d2f9cb8ff8b1c86cc28...	2016-10-09 15:39:56 UTC	2016-12-08 00:00:00 UTC	2016-11-09 14:53:50 UTC	30	-28
3	65d1e226dfaeb8cdc42f66542...	2016-10-03 21:01:41 UTC	2016-11-25 00:00:00 UTC	2016-11-08 10:58:34 UTC	35	-16
4	635c894d068ac37e6e03dc54e...	2017-04-15 15:37:38 UTC	2017-05-18 00:00:00 UTC	2017-05-16 14:49:55 UTC	30	-1
5	3b97562c3aee8bdedcb5c2e45...	2017-04-14 22:21:54 UTC	2017-05-18 00:00:00 UTC	2017-05-17 10:52:15 UTC	32	0
6	68f47f50f04c4cb6774570cfde...	2017-04-16 14:56:13 UTC	2017-05-18 00:00:00 UTC	2017-05-16 09:07:47 UTC	29	-1
7	276e9ec344d3bf029ff83a161c...	2017-04-08 21:20:24 UTC	2017-05-18 00:00:00 UTC	2017-05-22 14:11:31 UTC	43	4

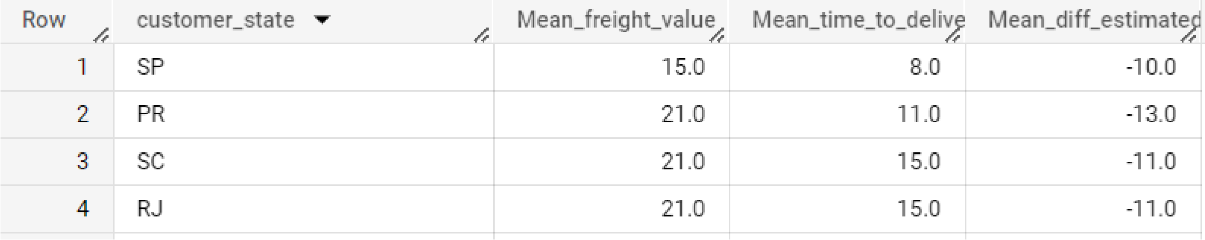
INSIGHTS: In some of the cases, delivery delay was close to 6 months than the expected delivery days while some had fastest delivery time within a minimum of 3 days or so.

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

ANS:

A. LOWEST AVG FREIGHT VALUE

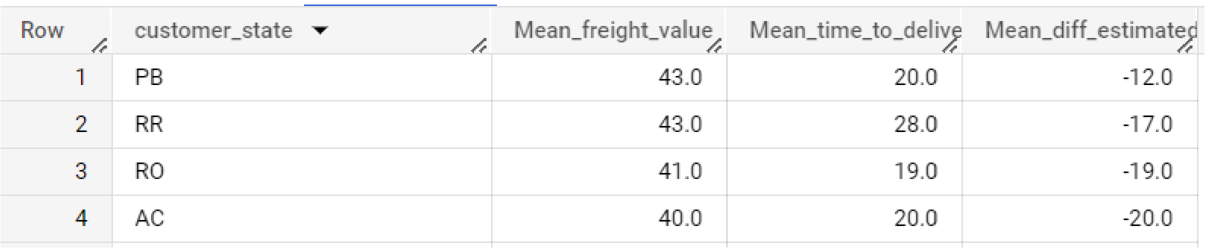
```
select
t.customer_state,
round(avg(t.freight_value)) Mean_freight_value,
round(avg(t.time_to_delivery)) Mean_time_to_delivery,
round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from(select*,
timestamp_diff(order_delivered_customer_date , order_purchase_timestamp, day)
as time_to_delivery,
timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date,
day) as diff_estimated_delivery
from `Target.customers` c
inner join `Target.orders` o on c.customer_id = o.customer_id
inner join`Target.order_items` oi on o.order_id = oi.order_id)t
group by t.customer_state
order by Mean_freight_value
LIMIT 5
```



Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	SP	15.0	8.0	-10.0
2	PR	21.0	11.0	-13.0
3	SC	21.0	15.0	-11.0
4	RJ	21.0	15.0	-11.0

B. HIGHEST AVG FREIGHT VALUE

```
select
t.customer_state,
round(avg(t.freight_value)) Mean_freight_value,
round(avg(t.time_to_delivery)) Mean_time_to_delivery,
round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from(select*,
timestamp_diff(order_delivered_customer_date , order_purchase_timestamp, day)
as time_to_delivery,
timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date,
day) as diff_estimated_delivery
from `Target.customers` c
inner join `Target.orders` o on c.customer_id = o.customer_id
inner join`Target.order_items` oi on o.order_id = oi.order_id)t
group by t.customer_state
order by Mean_freight_value DESC
LIMIT 5;
```



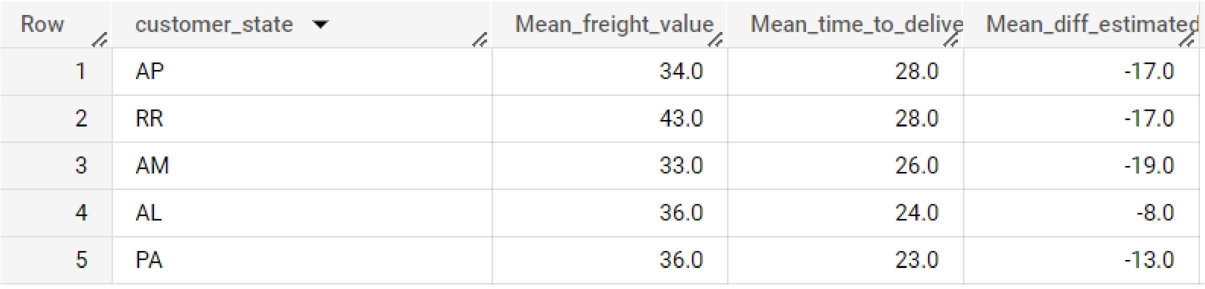
Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	PB	43.0	20.0	-12.0
2	RR	43.0	28.0	-17.0
3	RO	41.0	19.0	-19.0
4	AC	40.0	20.0	-20.0

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

ANS:

A. HIGHEST AVG DELIVERY TIME

```
SELECT
    t.customer_state,
    ROUND(AVG(t.freight_value)) AS Mean_freight_value,
    ROUND(AVG(t.time_to_delivery)) AS Mean_time_to_delivery,
    ROUND(AVG(t.diff_estimated_delivery)) AS Mean_diff_estimated_delivery
FROM
    (SELECT *,
            TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) AS time_to_delivery,
            TIMESTAMP_DIFF(order_delivered_customer_date,
order_estimated_delivery_date, DAY) AS diff_estimated_delivery
    FROM
        `Target.customers` c
    INNER JOIN
        `Target.orders` o ON c.customer_id = o.customer_id
    INNER JOIN
         `Target.order_items` oi ON o.order_id = oi.order_id) t
GROUP BY
    t.customer_state
ORDER BY
    Mean_time_to_delivery DESC
LIMIT 5;
```



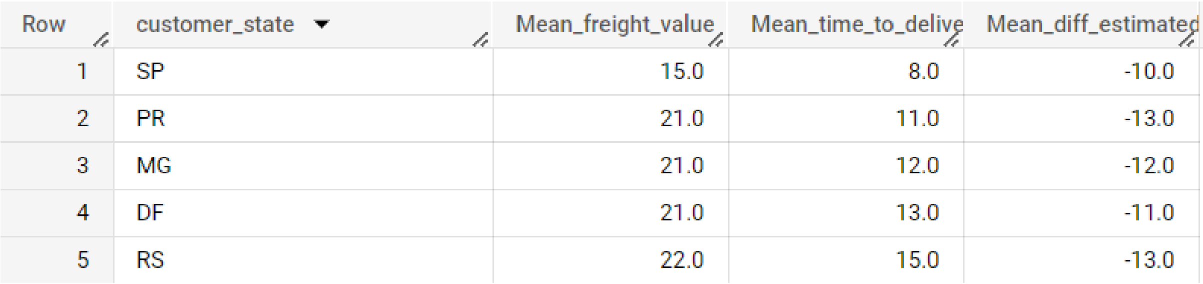
Row	customer_state	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	AP	34.0	28.0	-17.0
2	RR	43.0	28.0	-17.0
3	AM	33.0	26.0	-19.0
4	AL	36.0	24.0	-8.0
5	PA	36.0	23.0	-13.0

B. LOWEST AVG DELIVERY TIME:

ANS:

```
select

t.customer_state,
round(avg(t.freight_value)) Mean_freight_value,
round(avg(t.time_to_delivery)) Mean_time_to_delivery,
round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
FROM(select*,
timestamp_diff(order_delivered_customer_date , order_purchase_timestamp, day)
as time_to_delivery,
timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date, day)
as diff_estimated_delivery
from `Target.customers` c
inner join `Target.orders` o on c.customer_id = o.customer_id
inner join `Target.order_items` oi on o.order_id = oi.order_id )t
group by t.customer_state
order by Mean_time_to_delivery
LIMIT 5;
```



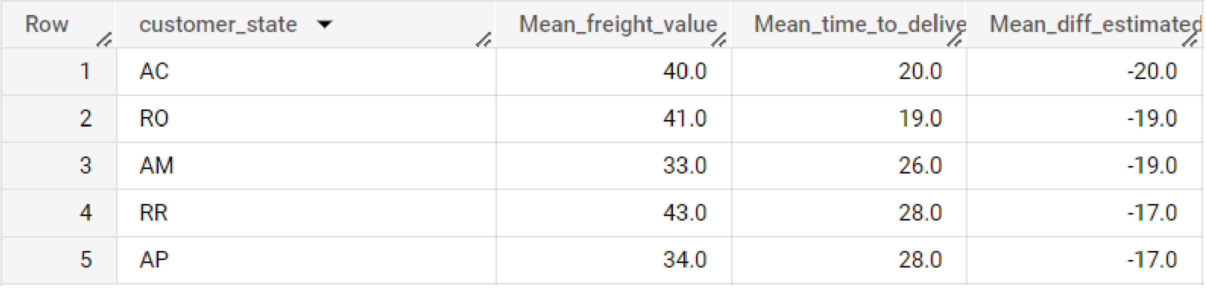
Row	customer_state ▼	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	SP	15.0	8.0	-10.0
2	PR	21.0	11.0	-13.0
3	MG	21.0	12.0	-12.0
4	DF	21.0	13.0	-11.0
5	RS	22.0	15.0	-13.0

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.

ANS:

```
select

t.customer_state,
round(avg(t.freight_value)) Mean_freight_value,
round(avg(t.time_to_delivery)) Mean_time_to_delivery,
round(avg(t.diff_estimated_delivery)) Mean_diff_estimated_delivery
from
(select*,
timestamp_diff(order_delivered_customer_date , order_purchase_timestamp, day) as
time_to_delivery,
timestamp_diff(order_delivered_customer_date , order_estimated_delivery_date, day)
as diff_estimated_delivery
from `Target.customers` c
inner join `Target.orders` o on c.customer_id = o.customer_id
inner join `Target.order_items` oi on o.order_id =oi.order_id)t
group by t.customer_state
order by Mean_diff_estimated_delivery
LIMIT 5
```



Row	customer_state ▼	Mean_freight_value	Mean_time_to_delive	Mean_diff_estimated
1	AC	40.0	20.0	-20.0
2	RO	41.0	19.0	-19.0
3	AM	33.0	26.0	-19.0
4	RR	43.0	28.0	-17.0
5	AP	34.0	28.0	-17.0

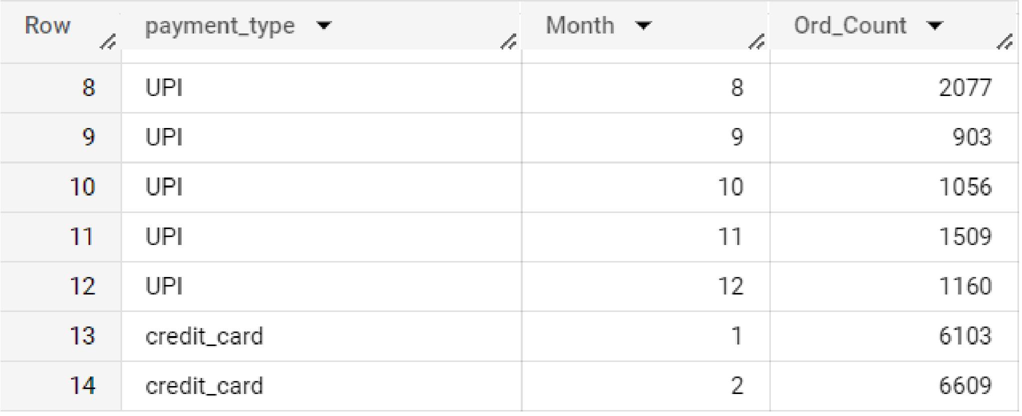
6.Analysis based on the payments:

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

ANS:

```
select t.payment_type, t.Month, count(*) as Ord_Count

from
(
select*, extract(month from o.order_purchase_timestamp) as Month
from `Target.orders` o
inner join `Target.payments` p on o.order_id = p.order_id
)t
group by t.payment_type, t.Month order by t.payment_type, t.Month;
```



Row	payment_type	Month	Ord_Count
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056
11	UPI	11	1509
12	UPI	12	1160
13	credit_card	1	6103
14	credit_card	2	6609

INSIGHTS: The payment type credit card had the greatest number of transactions followed by UPI.

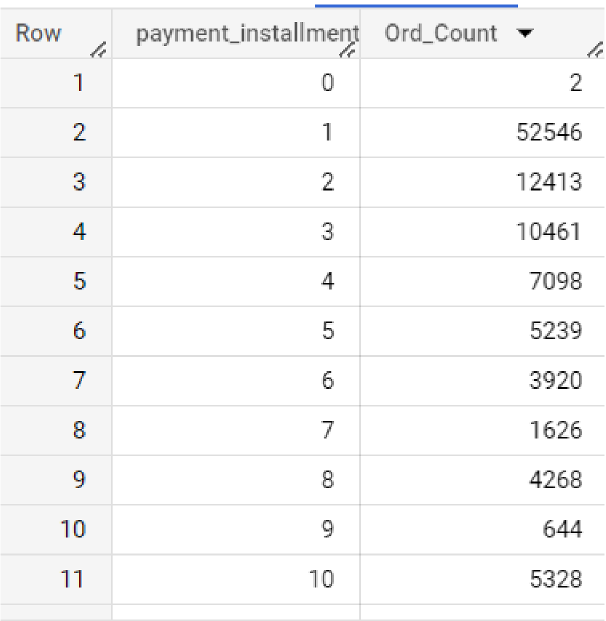
2. Find the no. of orders placed on the basis of the payment instalments that have been paid.

ANS:

```
select p.payment_installments, count(*) as Ord_Count

from `Target.orders` o
inner join `Target.payments` p on o.order_id = p.order_id
group by p.payment_installments order by p.payment_installments;
```

ANS:



Row	payment_installment	Ord_Count
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328

INSIGHTS: Instalments periods of 22, 23, 21 months seem to be the least preferred choices as shown below.

RECOMMENDATIONS BASED ON INSIGHTS:

1. Open More Stores Where People Order Most: Mostly folks in Sao Paulo and Rio de Janeiro buy stuff from us. So, let's put more stores in these cities where lots of people shop.

2. Invest More in Brazil: From 2016 to 2018, more and more people ordered from us. So, we should spend more money on our business in Brazil.

3. Sell Popular Summer Stuff: When its hot outside, people buy a lot from us. We should find out what they like in the summer and make sure we always have it to sell.

4. Busy Afternoons Online: Most people order things in the afternoon. If our website takes orders, we need to make sure it works really well during those times so customers don't get mad.

5. Big Growth in Brazil: Our business in Brazil grew a whole lot—137%—between 2017 and 2018. That means there's a big opportunity for us there.

6. Sell Fancy Stuff in Paraiba: People in Paraiba like expensive things. So, let's try selling more fancy stuff there.

7. Figure Out Late Deliveries: Some orders took almost six months longer to arrive than we promised. We need to find out why. Is it normal for some things, or did our supply chain mess up?

8. Fast Deliveries Make Customers Happy: Customers love it when their orders come quickly. To do well in the market, we need to make sure all deliveries are fast.

9. Speed Up Deliveries in Roraima: It takes a long time to deliver orders in Roraima. Maybe we should set up a warehouse with the most popular items there to make things faster.

10. Solve Alagoas Delivery Delays: In Alagoas, orders are taking a lot longer than we said they would. We need to find out what's going wrong between our main warehouse, local warehouse, and the customer's door.