Business Case: Ecommerce company SQL

Description :

The Ecommerce company is a globally renowned brand and a prominent retailer in the United States. They make themselves a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

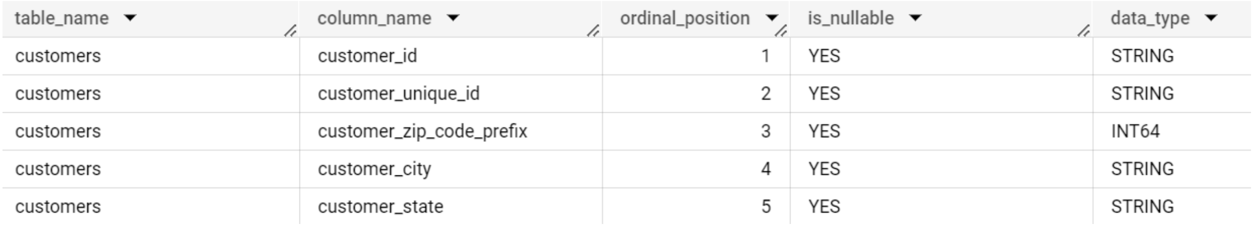
This particular business case focuses on the operations of the company in LatAm and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into the company's operations in LatAm. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

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.**
    2. **Get the time range between which the orders were placed.**
    3. **Count the number of Cities and States in our dataset.**

Query 1:

```
SELECT
  *
FROM
  `Ecomm_BusinessCase`.INFORMATION_SCHEMA.COLUMNS
WHERE
  TABLE_NAME = 'customers'
```

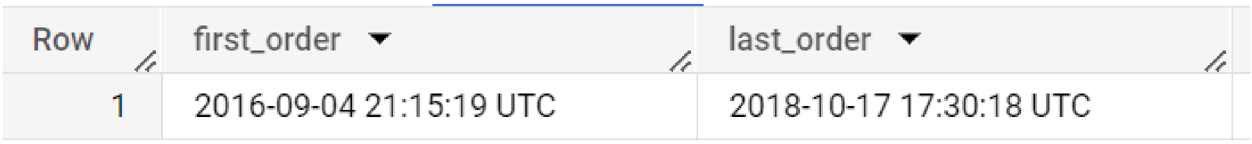


table_name	column_name	ordinal_position	is_nullable	data_type
customers	customer_id	1	YES	STRING
customers	customer_unique_id	2	YES	STRING
customers	customer_zip_code_prefix	3	YES	INT64
customers	customer_city	4	YES	STRING
customers	customer_state	5	YES	STRING

Insights: String data type count : 4
Int64 data type count : 1

Query 2 :

```
SELECT min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order
FROM `Ecomm_BusinessCase.orders`
```

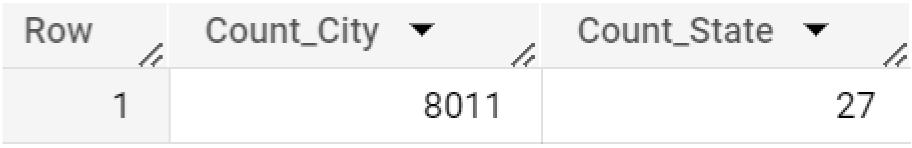


Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights: First order was place on 4th September at night and the Last order on 17th October evening

Query 3:

```
SELECT count(distinct geolocation_city) as Count_City, count(distinct
geolocation_state) as Count_State
FROM `Ecomm_BusinessCase.geolocation`
```



Row	Count_City	Count_State
1	8011	27

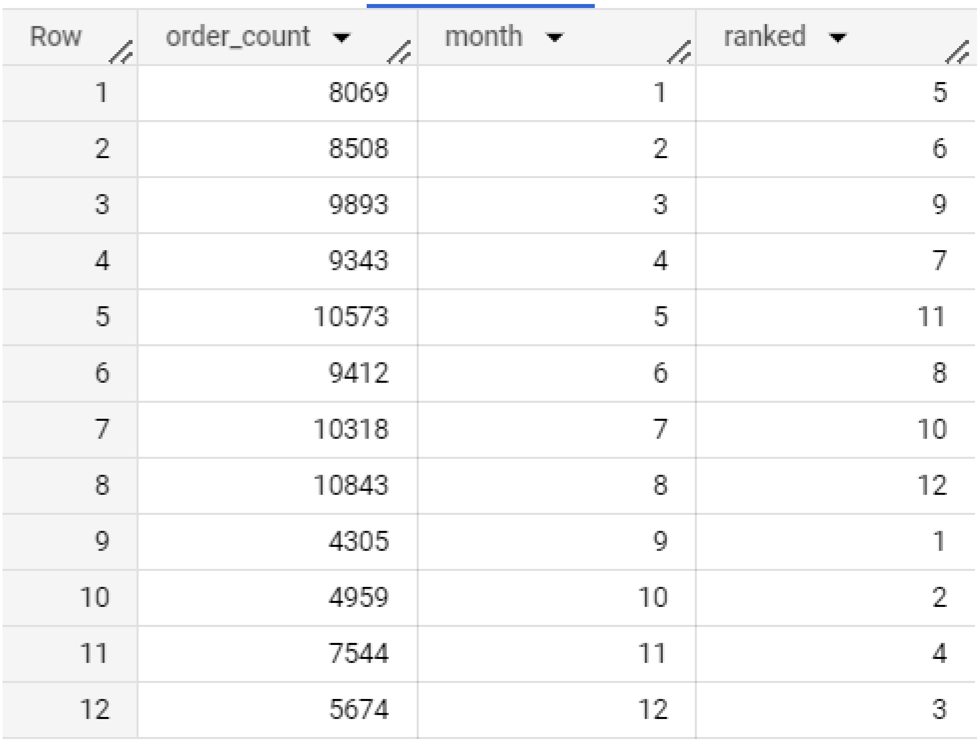
Insights: Number of cities - 8011
Number of States - 27

2. In-depth Exploration:

1. **Is there a growing trend in the no. of orders placed over the past years?**
2. **Can we see some kind of monthly seasonality in terms of the no. of orders being placed?**
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**

Query 1:

```
With trend1 as (SELECT count(order_id) as order_count,extract(month FROM
order_purchase_timestamp) as month,
FROM `Ecomm_BusinessCase.orders`
Group by extract(month FROM order_purchase_timestamp))
SELECT distinct trend1.order_count,trend1.month, dense_rank()over(Order by
order_count) as ranked
FROM trend1
order by trend1.month
```



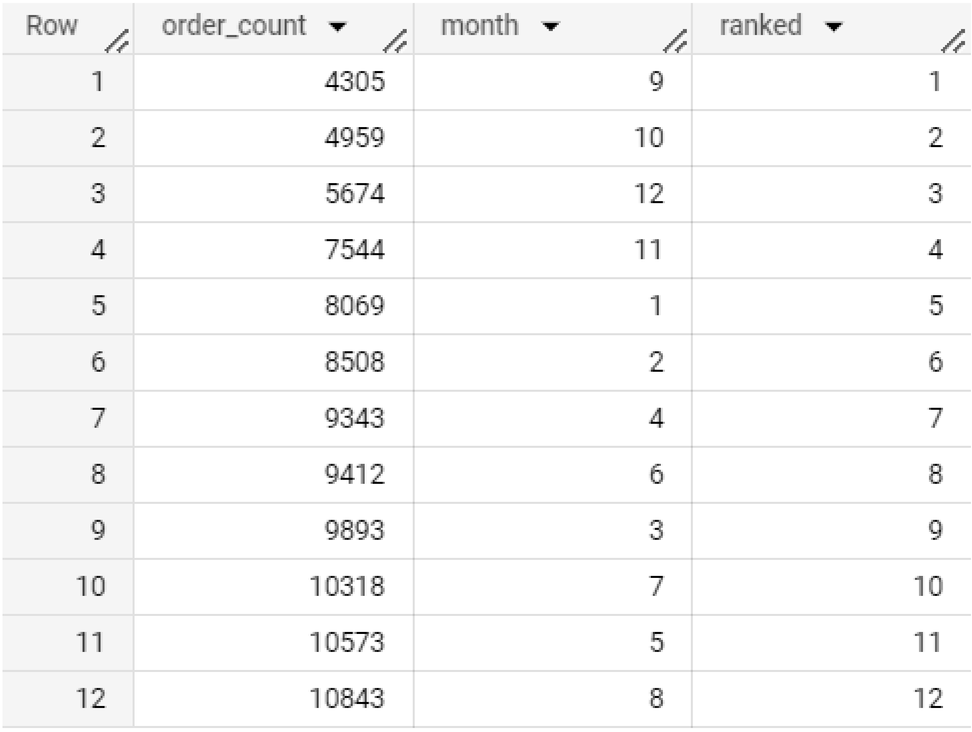
Row	order_count	month	ranked
1	8069	1	5
2	8508	2	6
3	9893	3	9
4	9343	4	7
5	10573	5	11
6	9412	6	8
7	10318	7	10
8	10843	8	12
9	4305	9	1
10	4959	10	2
11	7544	11	4
12	5674	12	3

Insights :

- We can see a growing trend on the number of orders placed. At the same time, there is a decrease in the number of orders during the off season.
- Peak season of sales is observed in the months May, July, August. This can be a peak graph when plotted on number of orders VS month.

Query 2:

```
With trend1 as (SELECT count(order_id) as order_count,extract(month FROM
order_purchase_timestamp) as month,
FROM `Ecomm_BusinessCase.orders`
Group by extract(month FROM order_purchase_timestamp))
SELECT distinct trend1.order_count,trend1.month, dense_rank()over(Order by
order_count) as ranked
FROM trend1
order by ranked
```



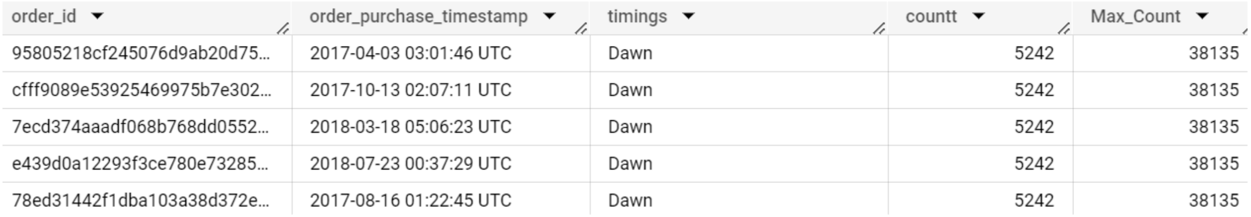
Row	order_count	month	ranked
1	4305	9	1
2	4959	10	2
3	5674	12	3
4	7544	11	4
5	8069	1	5
6	8508	2	6
7	9343	4	7
8	9412	6	8
9	9893	3	9
10	10318	7	10
11	10573	5	11
12	10843	8	12

Insights :

- Since we see a peak graph drawn from the previous scenario, we can rank the number of orders based on month. This provides us an insight that August has seen the greatest number of sales and September has seen minimum sales.
- This difference resulted in a loss of over 60% seen immediately in the next month

Query 3:

```
With
t1 as (SELECT order_id, extract(hour FROM order_purchase_timestamp) as hour,
order_purchase_timestamp
FROM `Ecomm_BusinessCase.orders`),
t2 as
(SELECT order_id, order_purchase_timestamp,t1.hour,
  case when hour between 0 and 6 then "Dawn"
    when hour between 7 and 12 then "Mornings"
    when hour between 13 and 18 then "Afternoon"
    when hour between 19 and 23 then "Night"
    end as timings
  FROM t1),
  t3 as
  (
    SELECT order_id, order_purchase_timestamp, timings, count(timings)over(partition
by timings)countt
    FROM t2
  ),
  t4 as (
    SELECT order_id, order_purchase_timestamp, timings, countt, max(countt)over() as
Max_Count
    FROM t3
  )
SELECT order_id, order_purchase_timestamp, timings, countt,Max_Count
FROM t4
```



order_id	order_purchase_timestamp	timings	countt	Max_Count
95805218cf245076d9ab20d75...	2017-04-03 03:01:46 UTC	Dawn	5242	38135
cfff9089e53925469975b7e302...	2017-10-13 02:07:11 UTC	Dawn	5242	38135
7ecd374aaadf068b768dd0552...	2018-03-18 05:06:23 UTC	Dawn	5242	38135
e439d0a12293f3ce780e73285...	2018-07-23 00:37:29 UTC	Dawn	5242	38135
78ed31442f1dba103a38d372e...	2017-08-16 01:22:45 UTC	Dawn	5242	38135

Insights:

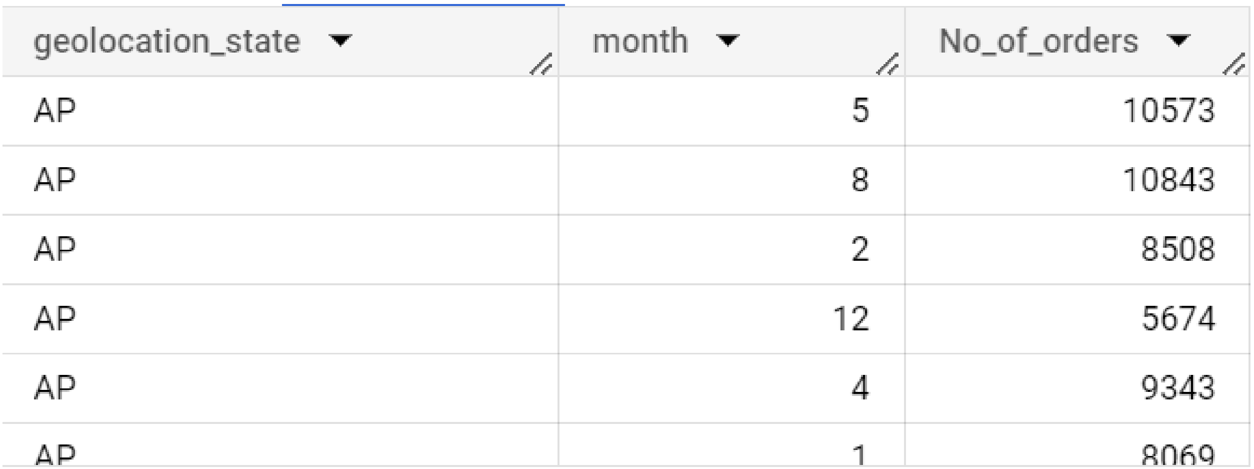
- Most of the customers are observed placing their orders during the Afternoon in a day.
- Timings : Between 1PM to 6PM

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

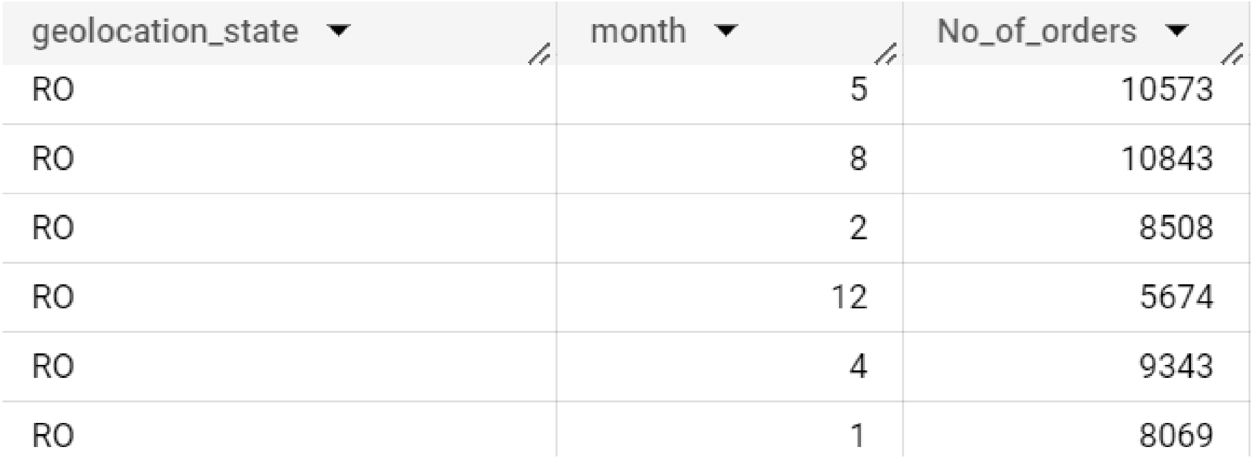
1. **Get the month on month no. of orders placed in each state.**
2. **How are the customers distributed across all the states?**

Query 1:

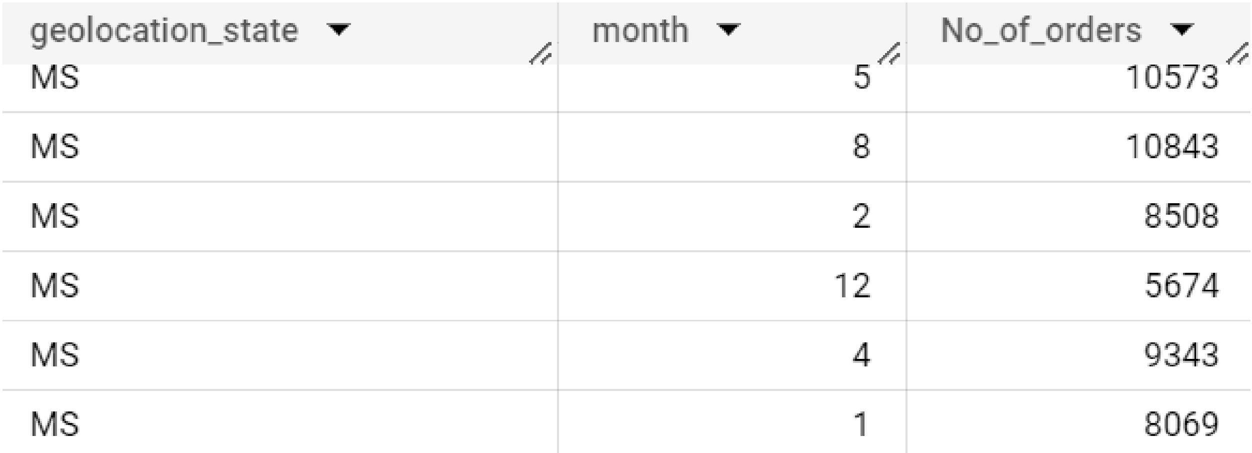
```
With state as (SELECT distinct geolocation_state
FROM `Ecomm_BusinessCase.geolocation`),
orders as
(SELECT
  DISTINCT EXTRACT(month FROM order_purchase_timestamp) AS month,
  COUNT(DISTINCT order_id) AS No_of_orders
FROM `Ecomm_BusinessCase.orders`
GROUP BY
  EXTRACT(month FROM order_purchase_timestamp))
SELECT *
FROM state,orders
```



geolocation_state	month	No_of_orders
AP	5	10573
AP	8	10843
AP	2	8508
AP	12	5674
AP	4	9343
AP	1	8069



geolocation_state	month	No_of_orders
RO	5	10573
RO	8	10843
RO	2	8508
RO	12	5674
RO	4	9343
RO	1	8069



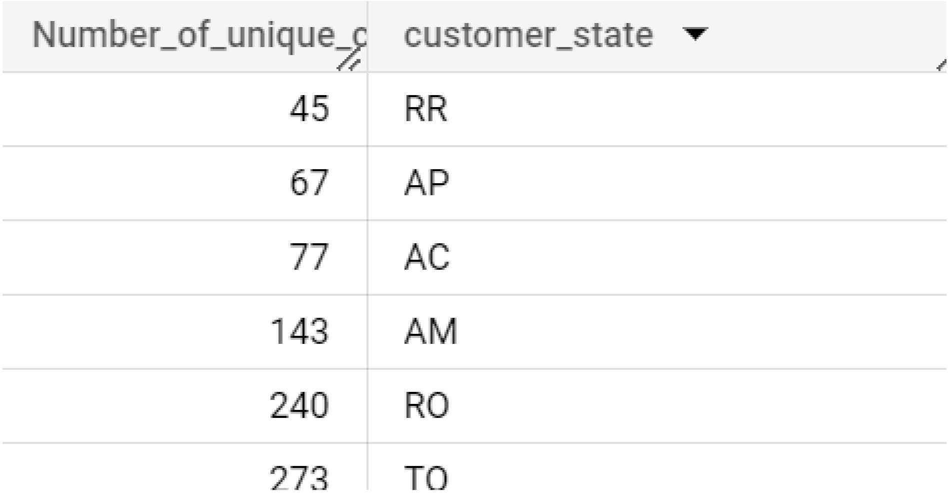
geolocation_state	month	No_of_orders
MS	5	10573
MS	8	10843
MS	2	8508
MS	12	5674
MS	4	9343
MS	1	8069

Insights:

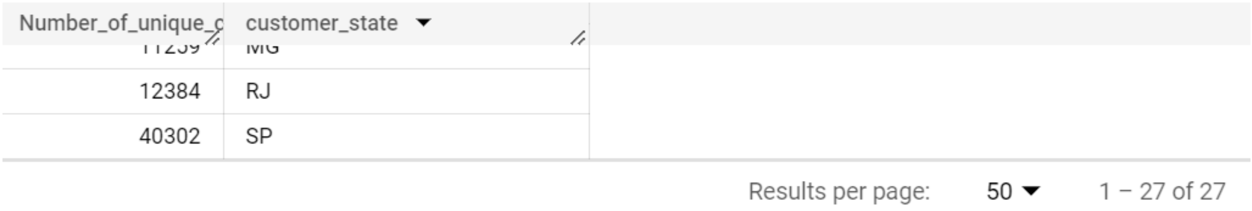
- Number of orders placed in each state in each month implies that August is the peak season for sales in all the month

Query 2:

```
select count(distinct customer_unique_id) Number_of_unique_customers , customer_state
FROM `Ecomm_BusinessCase.customers`
group by customer_state
Order by Number_of_unique_customers, customer_state
```



Number_of_unique_c	customer_state
45	RR
67	AP
77	AC
143	AM
240	RO
273	TO



Number_of_unique_c	customer_state
11259	MG
12384	RJ
40302	SP

Results per page: 50 1 – 27 of 27

Insights:

- Number of unique customers are maximum at SP and minimum at RR. Hence increased maximum profit can be concentrated on SP and more variety of products can be leveraged on this area.
- States having comparatively less number of unique customers can be surveyed and based on the survey results of the customers, products can be focused on the particular geolocation.

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

Query 1:

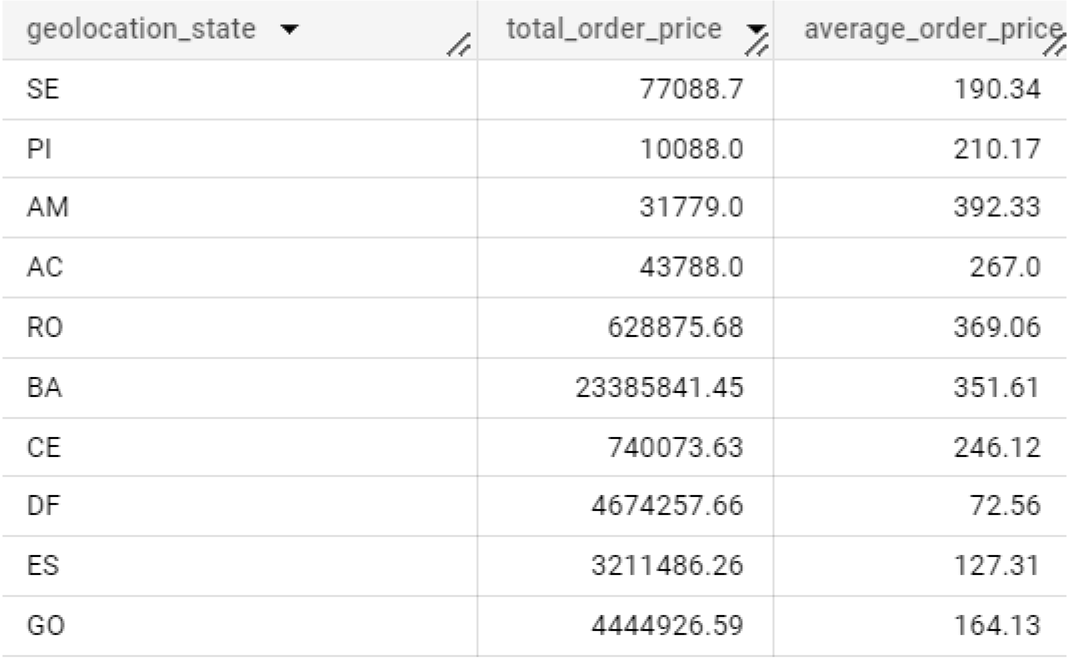
```
SELECT
EXTRACT(month FROM o.order_purchase_timestamp) month,
extract(year from o.order_purchase_timestamp) year, p.payment_value -
LAG(p.payment_value) OVER (PARTITION BY EXTRACT(year FROM order_purchase_timestamp)
ORDER BY EXTRACT(month FROM o.order_purchase_timestamp)) AS payment_difference
FROM `Ecomm_BusinessCase.orders` o inner join `Ecomm_BusinessCase.payments` p on
o.order_id=p.order_id
WHERE EXTRACT(year FROM order_purchase_timestamp) in (2017, 2018)
and EXTRACT(month FROM order_purchase_timestamp) between 1 and 8
GROUP BY year,month

Review: WITH t1 as (
  SELECT *
  FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.payments` p
  On o.order_id = p.order_id
  WHERE extract(year From o.order_purchase_timestamp) Between 2017 and 2018
```

```
    And extract(month From o.order_purchase_timestamp) Between 1 and 8
), t2 as (
  SELECT
    extract(year From order_purchase_timestamp) as year,
    SUM(payment_value) as cost
  FROM t1
  GROUP BY year
)
SELECT *,(cost-lag(cost,1)over(ORDER BY t2.year))*100/lag(cost,1)over(Order by
t2.year) as percentage_increase
FROM t2
```

Query 2:

```
SELECT geo.geolocation_state,round(SUM(OI.price),2) AS total_order_price,
round((SUM(OI.price)/count(distinct OI.order_id)),2) AS average_order_price
FROM `Ecomm_BusinessCase.geolocation` as geo join `Ecomm_BusinessCase.sellers` as sell
On geo.geolocation_zip_code_prefix = sell.seller_zip_code_prefix
join `Ecomm_BusinessCase.order_items` OI
On sell.seller_id=OI.seller_id
Group by geo.geolocation_state
```



geolocation_state	total_order_price	average_order_price
SE	77088.7	190.34
PI	10088.0	210.17
AM	31779.0	392.33
AC	43788.0	267.0
RO	628875.68	369.06
BA	23385841.45	351.61
CE	740073.63	246.12
DF	4674257.66	72.56
ES	3211486.26	127.31
GO	4444926.59	164.13

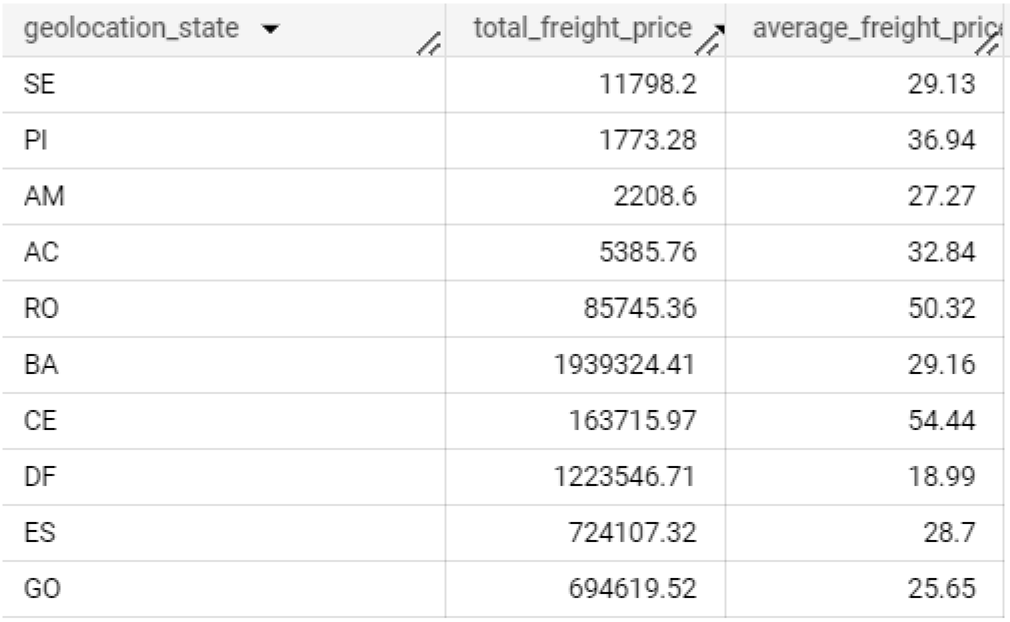
Insights:

- Average order price is maximum at RO though the total order_price is comparatively less

- BA has a maximum total order price with comparatively a better average. Can make a profitable turn over.

Query 3:

```
SELECT geo.geolocation_state,round(SUM(OI.freight_value),2) AS
total_freight_price,round(AVG(OI.freight_value),2) AS average_freight_price
FROM `Ecomm_BusinessCase.geolocation` as geo join `Ecomm_BusinessCase.sellers` as sell
On geo.geolocation_zip_code_prefix = sell.seller_zip_code_prefix
join `Ecomm_BusinessCase.order_items` OI
On sell.seller_id=OI.seller_id
Group by geo.geolocation_state
```



geolocation_state	total_freight_price	average_freight_pric
SE	11798.2	29.13
PI	1773.28	36.94
AM	2208.6	27.27
AC	5385.76	32.84
RO	85745.36	50.32
BA	1939324.41	29.16
CE	163715.97	54.44
DF	1223546.71	18.99
ES	724107.32	28.7
GO	694619.52	25.65

Insights:

- The rate at which the product is delivered from one point to another is observed significantly high at BA but CE has a high average freight price.
- This can probably infer that the transportation distance is comparatively high at CE and more number of shipment travels are seen in BA

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

1. **Find the no. of days taken to deliver each order from the order's purchase date as delivery time.**
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
   Do this in a single query.

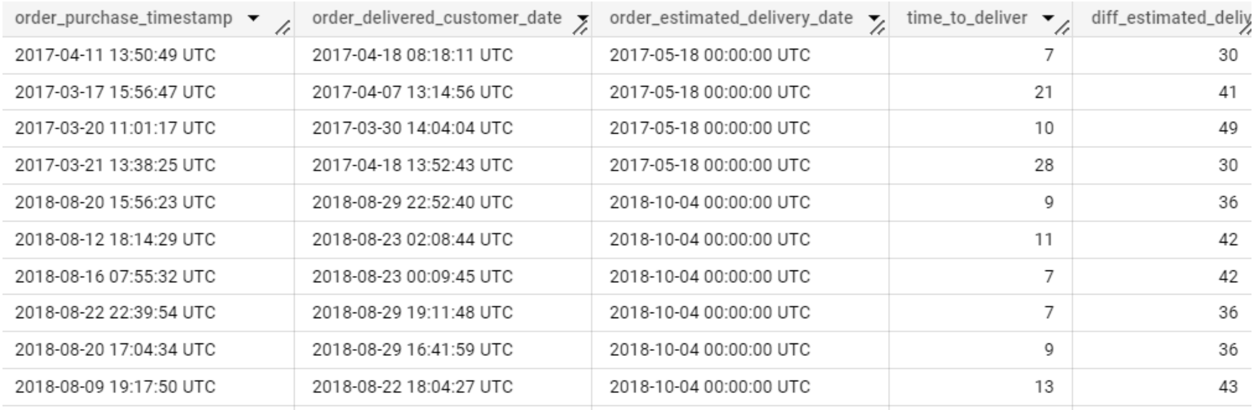
You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver = order_delivered_customer_date - order_purchase_timestamp**
- **diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date**

2. **Find out the top 5 states with the highest & lowest average freight value.**
3. **Find out the top 5 states with the highest & lowest average delivery time.**
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.

Query 1:

```
SELECT order_purchase_timestamp, order_delivered_customer_date,
order_estimated_delivery_date, DATETIME_DIFF(DATETIME(order_delivered_customer_date),
DATETIME (order_purchase_timestamp), day) as time_to_deliver,
abs(DATETIME_DIFF(DATETIME(order_estimated_delivery_date), DATETIME
(order_delivered_customer_date), day)) as diff_estimated_delivery
FROM `Ecomm_BusinessCase.orders`
WHERE lower(order_status) like "delivered"
```



order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_deliv
2017-04-11 13:50:49 UTC	2017-04-18 08:18:11 UTC	2017-05-18 00:00:00 UTC	7	30
2017-03-17 15:56:47 UTC	2017-04-07 13:14:56 UTC	2017-05-18 00:00:00 UTC	21	41
2017-03-20 11:01:17 UTC	2017-03-30 14:04:04 UTC	2017-05-18 00:00:00 UTC	10	49
2017-03-21 13:38:25 UTC	2017-04-18 13:52:43 UTC	2017-05-18 00:00:00 UTC	28	30
2018-08-20 15:56:23 UTC	2018-08-29 22:52:40 UTC	2018-10-04 00:00:00 UTC	9	36
2018-08-12 18:14:29 UTC	2018-08-23 02:08:44 UTC	2018-10-04 00:00:00 UTC	11	42
2018-08-16 07:55:32 UTC	2018-08-23 00:09:45 UTC	2018-10-04 00:00:00 UTC	7	42
2018-08-22 22:39:54 UTC	2018-08-29 19:11:48 UTC	2018-10-04 00:00:00 UTC	7	36
2018-08-20 17:04:34 UTC	2018-08-29 16:41:59 UTC	2018-10-04 00:00:00 UTC	9	36
2018-08-09 19:17:50 UTC	2018-08-22 18:04:27 UTC	2018-10-04 00:00:00 UTC	13	43

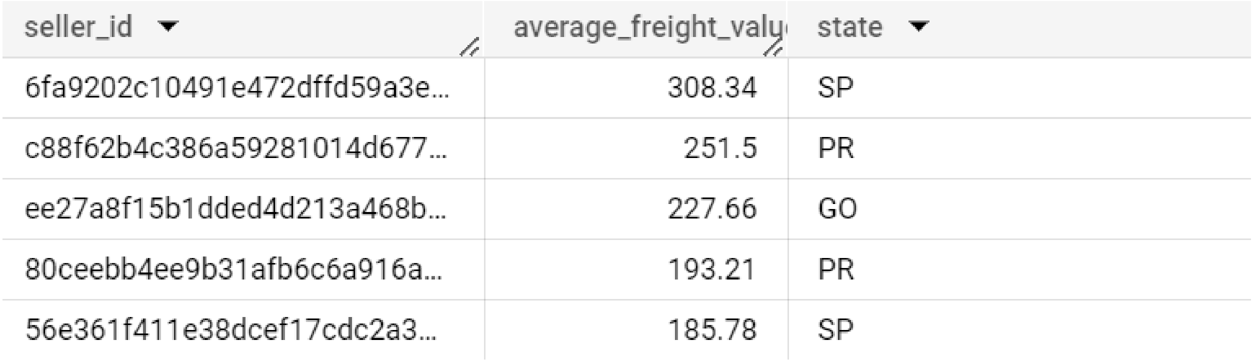
Insights:

- The orders are delivered before time to the customers
- More orders can be incorporated during the time based on man-power or estimated time can be reduced with a few customers.

Query 2:

Highest

```
SELECT o.seller_id, round(avg(o.freight_value),2) as average_freight_value,
s.seller_state as state
FROM `Ecomm_BusinessCase.order_items` o join `Ecomm_BusinessCase.sellers` s
On o.seller_id = s.seller_id
Group by seller_id,s.seller_state
Order by average_freight_value DESC
```



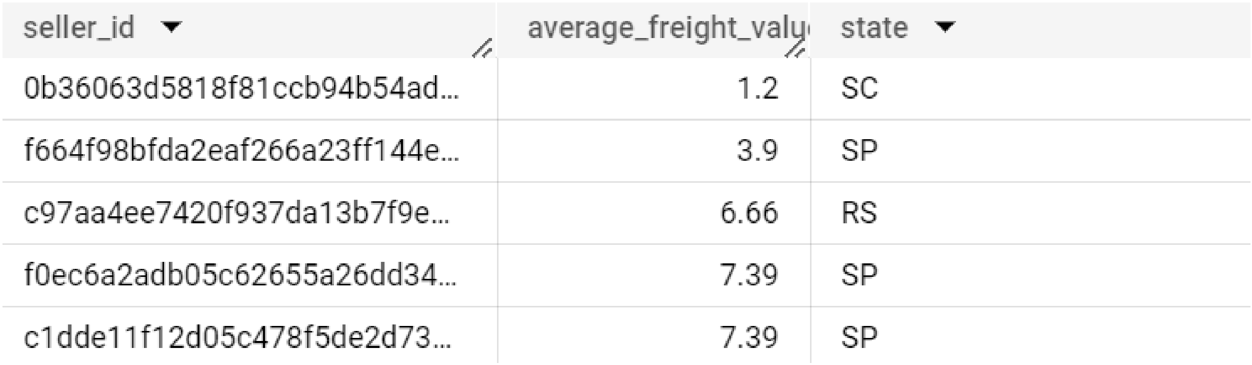
seller_id	average_freight_valu	state
6fa9202c10491e472dffd59a3e...	308.34	SP
c88f62b4c386a59281014d677...	251.5	PR
ee27a8f15b1dded4d213a468b...	227.66	GO
80ceebb4ee9b31afb6c6a916a...	193.21	PR
56e361f411e38dcef17cdc2a3...	185.78	SP

Insights:

- Top 5 states with highest average freight value - SP, PR, GO, MG, SE

Lowest

```
SELECT o.seller_id, round(avg(o.freight_value),2) as average_freight_value,
s.seller_state as state
FROM `Ecomm_BusinessCase.order_items` o join `Ecomm_BusinessCase.sellers` s
On o.seller_id = s.seller_id
Group by seller_id,s.seller_state
Order by average_freight_value
```



seller_id	average_freight_valu	state
0b36063d5818f81ccb94b54ad...	1.2	SC
f664f98bfda2eaf266a23ff144e...	3.9	SP
c97aa4ee7420f937da13b7f9e...	6.66	RS
f0ec6a2adb05c62655a26dd34...	7.39	SP
c1dde11f12d05c478f5de2d73...	7.39	SP

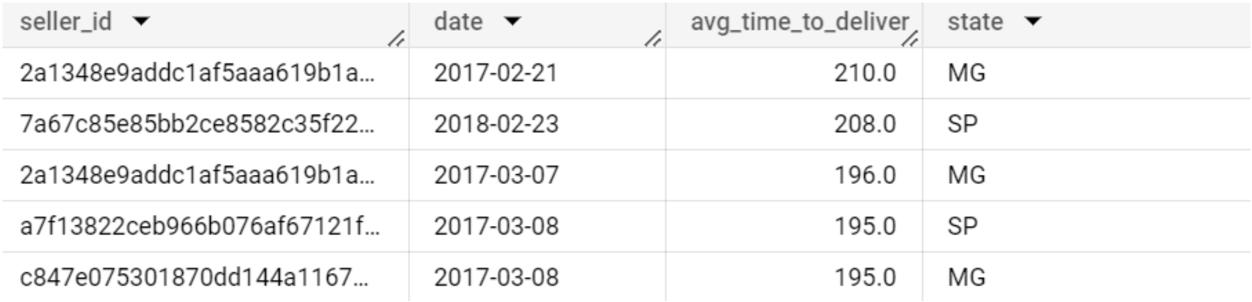
Insights:

- Top 5 states with lowest average freight value - SC, SP, RS, MG, RJ

Query 3:

Highest

```
WITH t1 as
(Select distinct s.seller_id, extract(date from o.order_purchase_timestamp)date,
avg(DATETIME_DIFF(DATETIME(o.order_delivered_customer_date), DATETIME
(o.order_purchase_timestamp), day)) over(partition by o.order_id,o.customer_id) as
avg_time_to_deliver,s.seller_state as state
FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.order_items` oi
On o.order_id = oi.order_id join `Ecomm_BusinessCase.sellers` s On oi.seller_id =
s.seller_id)
Select *
From t1 Where avg_time_to_deliver is not null
Order by avg_time_to_deliver desc
```



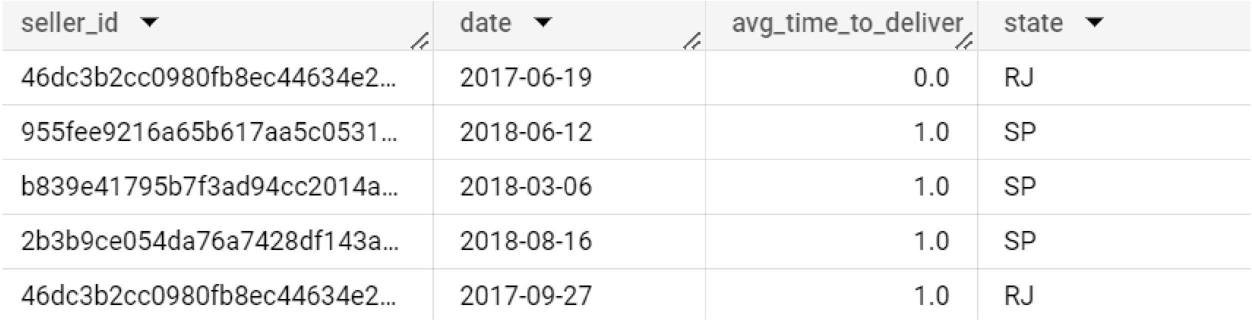
seller_id	date	avg_time_to_deliver	state
2a1348e9addc1af5aaa619b1a...	2017-02-21	210.0	MG
7a67c85e85bb2ce8582c35f22...	2018-02-23	208.0	SP
2a1348e9addc1af5aaa619b1a...	2017-03-07	196.0	MG
a7f13822ceb966b076af67121f...	2017-03-08	195.0	SP
c847e075301870dd144a1167...	2017-03-08	195.0	MG

Insights:

- Top 5 states with highest average delivery time - MG, SP, RJ, CE, PR

Lowest

```
WITH t1 as
(Select distinct s.seller_id, extract(date from o.order_purchase_timestamp)date,
avg(DATETIME_DIFF(DATETIME(o.order_delivered_customer_date), DATETIME
(o.order_purchase_timestamp), day)) over(partition by o.order_id,o.customer_id) as
avg_time_to_deliver,s.seller_state as state
FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.order_items` oi
On o.order_id = oi.order_id join `Ecomm_BusinessCase.sellers` s On oi.seller_id =
s.seller_id)
Select *
From t1 Where avg_time_to_deliver is not null
Order by avg_time_to_deliver
```



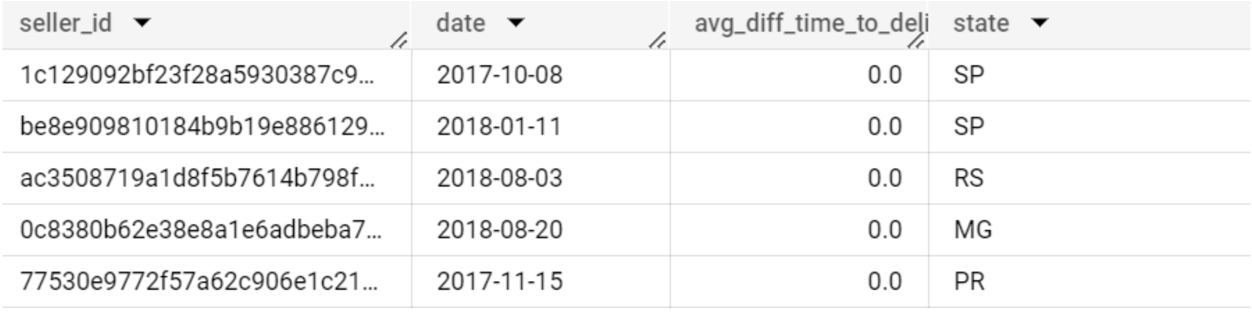
seller_id	date	avg_time_to_deliver	state
46dc3b2cc0980fb8ec44634e2...	2017-06-19	0.0	RJ
955fee9216a65b617aa5c0531...	2018-06-12	1.0	SP
b839e41795b7f3ad94cc2014a...	2018-03-06	1.0	SP
2b3b9ce054da76a7428df143a...	2018-08-16	1.0	SP
46dc3b2cc0980fb8ec44634e2...	2017-09-27	1.0	RJ

Insights:

- Top 5 states with lowest average delivery time - RJ,MG, SP, PR,GO

Query 4:

```
WITH t1 as
(Select distinct s.seller_id, extract(date from o.order_purchase_timestamp)date,
avg(abs(DATETIME_DIFF(DATETIME(order_estimated_delivery_date), DATETIME
(order_delivered_customer_date), day)))over(partition by o.order_id,o.customer_id) as
avg_diff_time_to_deliver,s.seller_state as state
FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.order_items` oi
On o.order_id = oi.order_id join `Ecomm_BusinessCase.sellers` s On oi.seller_id =
s.seller_id)
Select *
From t1 Where avg_diff_time_to_deliver is not null
Order by avg_diff_time_to_deliver
limit 5
```



seller_id	date	avg_diff_time_to_deli	state
1c129092bf23f28a5930387c9...	2017-10-08	0.0	SP
be8e909810184b9b19e886129...	2018-01-11	0.0	SP
ac3508719a1d8f5b7614b798f...	2018-08-03	0.0	RS
0c8380b62e38e8a1e6adbeba7...	2018-08-20	0.0	MG
77530e9772f57a62c906e1c21...	2017-11-15	0.0	PR

Insights:

- Order delivery date is quick at SP,RS,MG,PR

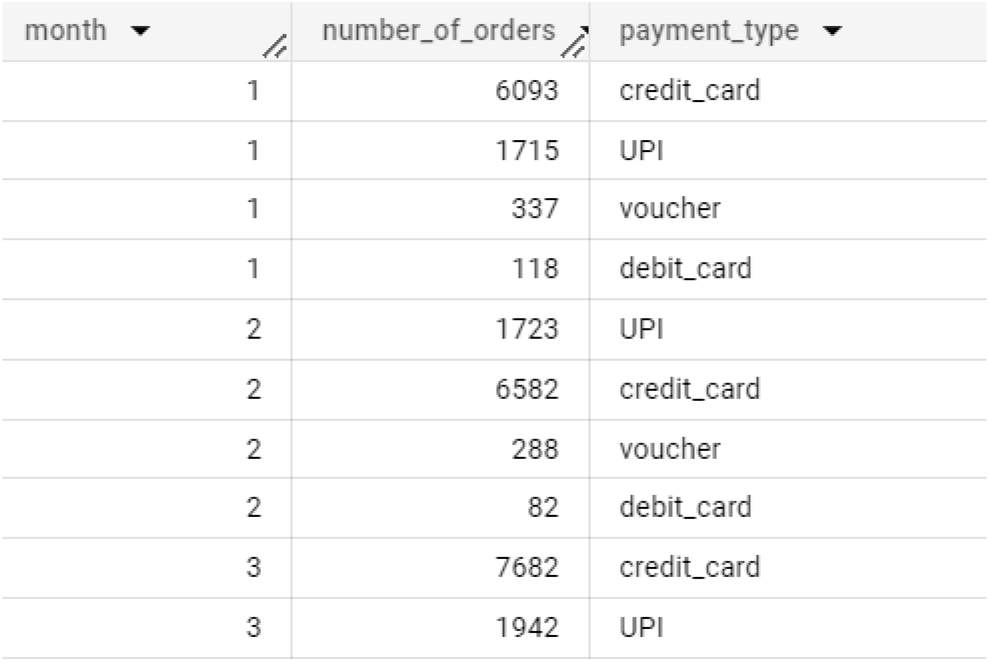
6. Analysis based on the payments:

1. **Find the month on month no. of orders placed using different payment types.**
2. **Find the no. of orders placed on the basis of the payment installments that have been paid.**

Query 1:

```
WITH t1 as (SELECT distinct extract(month from o.order_purchase_timestamp) as month,
count(distinct o.order_id) as number_of_orders,
p.payment_type
FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.payments` p
On o.order_id = p.order_id
Group by month,p.payment_type)
SELECT * FROM t1 Order by t1.month
 Review : SELECT distinct extract(month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year, count(distinct o.order_id) as
number_of_orders,
p.payment_type,dense_rank()over(partition by count(distinct o.order_id)Order by
count(distinct o.order_id))
FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.payments` p
```

```
On o.order_id = p.order_id
Group by month,year, p.payment_type
Order by year,month
```



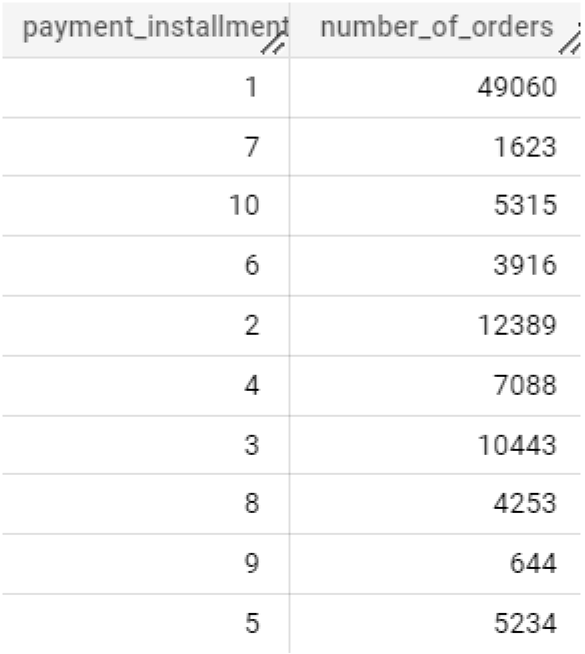
month	number_of_orders	payment_type
1	6093	credit_card
1	1715	UPI
1	337	voucher
1	118	debit_card
2	1723	UPI
2	6582	credit_card
2	288	voucher
2	82	debit_card
3	7682	credit_card
3	1942	UPI

Insights:

- Max number of payments are done through credit cards then the UPI by month
- Credit card payments - max

Query 2:

```
SELECT distinct p.payment_installments, count(distinct o.order_id) as number_of_orders
FROM `Ecomm_BusinessCase.orders` o join `Ecomm_BusinessCase.payments` p
On o.order_id = p.order_id
Group by p.payment_installments
Review:
SELECT payment_installments,count(order_id) as num_of_orders
FROM `Ecomm_BusinessCase.payments`
WHERE payment_installments>1
Group by payment_installments
ORDER BY num_of_orders desc, payment_installments
```



payment_installment	number_of_orders
1	49060
7	1623
10	5315
6	3916
2	12389
4	7088
3	10443
8	4253
9	644
5	5234

Insights:

- Number of orders paid on the first and second third and fourth installments is the highest
- Less number of people consider payments made on 9 installments

Recommendations

- Since peak season of sales is seen in the months of May, July, August, on season discounts can be given to attract more customers.
- In order to overcome the drastic decrease in sales during the off season months, based on the purchases made by each customer, they can be recommended with additional products that can be bought along with the item purchased, during the mid noon, which is the peak time of buying.
- States having comparatively less number of unique customers can be surveyed and based on the survey results of the customers, products can be focused on the particular geolocation - RR in this case.
- Transportation can be seen less frequently in CE hence can increase the sales in the region, since it also has a comparatively good average delivery time.
- Credit card offers can be given to the customers who take up a membership option, thus increasing the income through membership and sales.