Name: ALOYSIUS LOBO

Batch: DSML MORNING (TTS), 2023

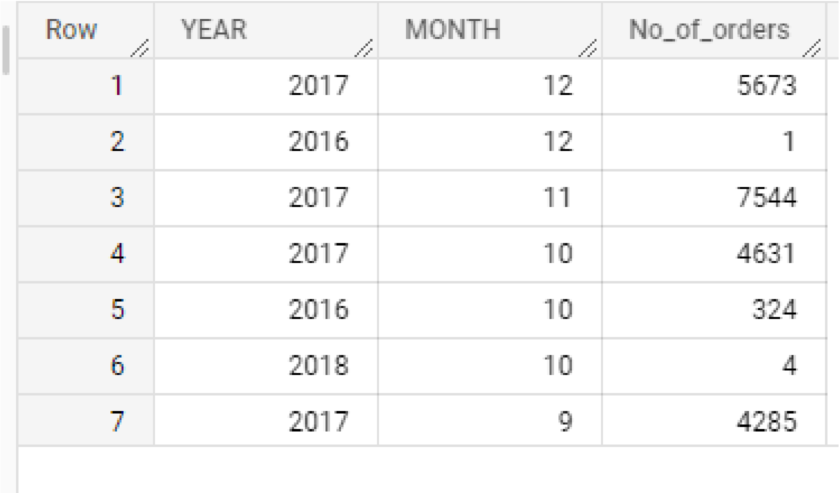
TARGET BUSINESS CASE STUDY

2.1 --Is there a growing trend on e-commerce in Brazil? –

Query

```
select extract(year from order_purchase_timestamp) as YEAR,
       extract(month from order_purchase_timestamp) as MONTH,
       count(*) as No_of_orders
from case_study_1.orders
group by month,year
order by month desc
```

Result



Row	YEAR	MONTH	No_of_orders
1	2017	12	5673
2	2016	12	1
3	2017	11	7544
4	2017	10	4631
5	2016	10	324
6	2018	10	4
7	2017	9	4285

Conclusion

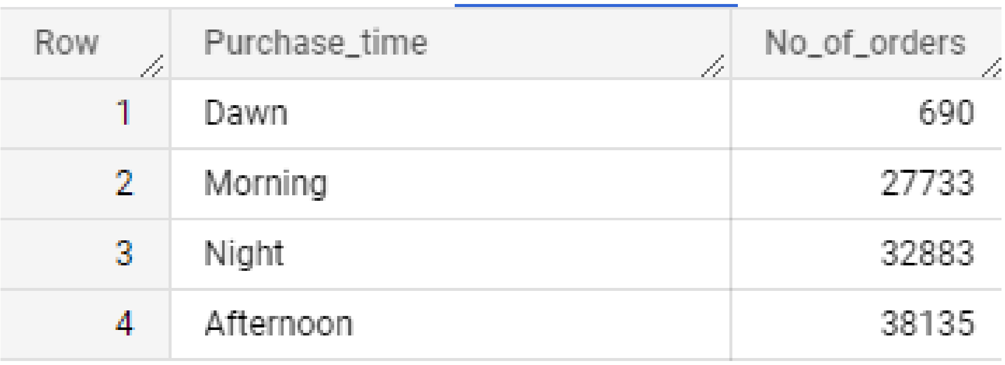
- orders increased from 2016 to 2017 with max orders in Nov 2017 i.e. 7544. After that a decline i n the orders was observed from 2017 to 2018 --
- a high order rate was seen from Nov 2017 to Jan 2018 which decreased thereby --

2.2 -- What time do Brazilian customers tend to buy --

Query

```
select
  case when extract(hour from order_purchase_timestamp) between 5 and 6 then 'Dawn'
       when extract(hour from order_purchase_timestamp) between 6 and 12 then 'Morning'
       when extract(hour from order_purchase_timestamp) between 12 and 18 then 'Afternoon'
       else 'Night'
  end as Purchase_time,
  count(*) as No_of_orders
from case_study_1.orders
group by Purchase_time
order by No_of_orders asc
```

Result



Row	Purchase_time	No_of_orders
1	Dawn	690
2	Morning	27733
3	Night	32883
4	Afternoon	38135

Conclusion

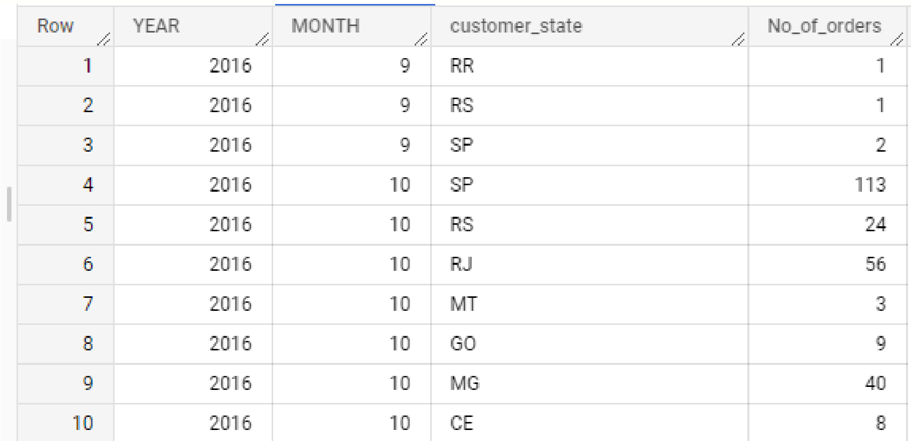
- A majority of the Brazilian customers tend to buy in the Afternoon during 12pm to 6pm.
- While most do prefer purchasing during night ie. between 6pm to 12am.
- As it is evident from above that the No_of_orders purchased in the afternoon is highest at 38135 followed by the orders purchased during night which is 32883.

-- 3.1. Get month on month orders by states –

Query

```
select
      extract(year from o.order_purchase_timestamp) YEAR,
      extract(month from o.order_purchase_timestamp) MONTH,
      c.customer_state,
      count(*) as No_of_orders
from case_study_1.orders o join case_study_1.customers c on o.customer_id = c.customer_id
group by c.customer_state, YEAR, MONTH
order by YEAR asc, MONTH asc
```

Result



Row	YEAR	MONTH	customer_state	No_of_orders
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	SP	113
5	2016	10	RS	24
6	2016	10	RJ	56
7	2016	10	MT	3
8	2016	10	GO	9
9	2016	10	MG	40
10	2016	10	CE	8

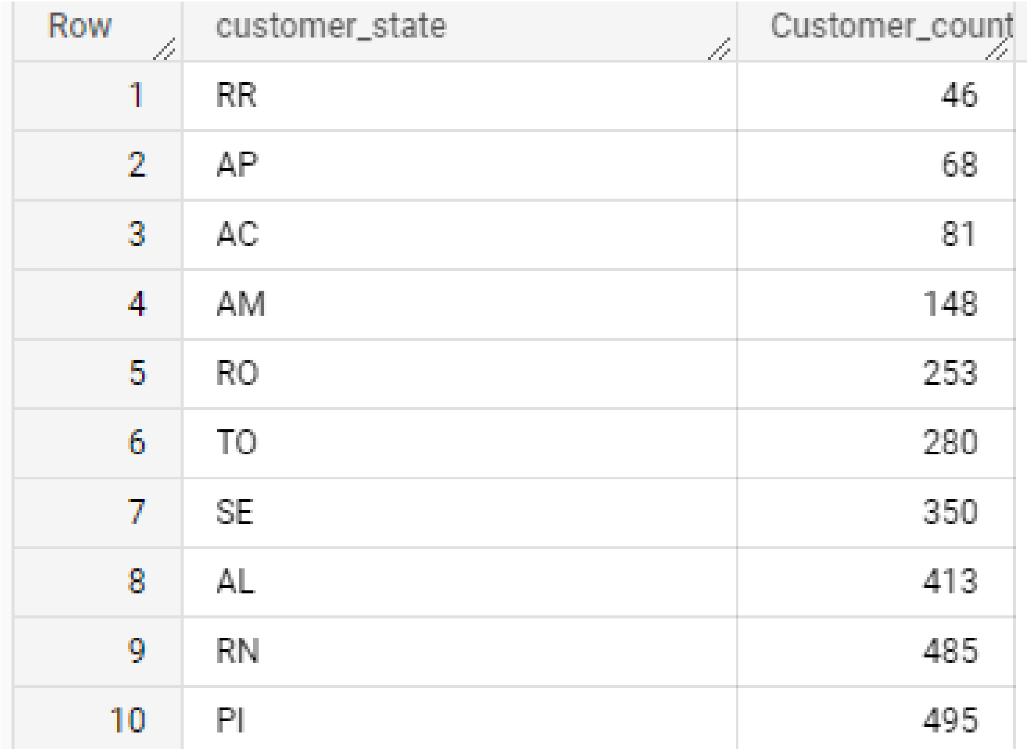
\

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

Ouery

```
select distinct customer_state, count(customer_unique_id) as Customer_count
from case_study_1.customers
group by customer_state
order by Customer_count asc
```

Result



Row	customer_state	Customer_count
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

Conclusion

From the result table it is observed that maximum customers with a count of 41746 are in the state of SP while minimum customers with a count of 46 are in the state of RR.

- 4.1. Get % increase in the cost of orders from 2017 to 2018 (including months between Jan to Aug only) . You can use "payment_value" column in the payment table. –

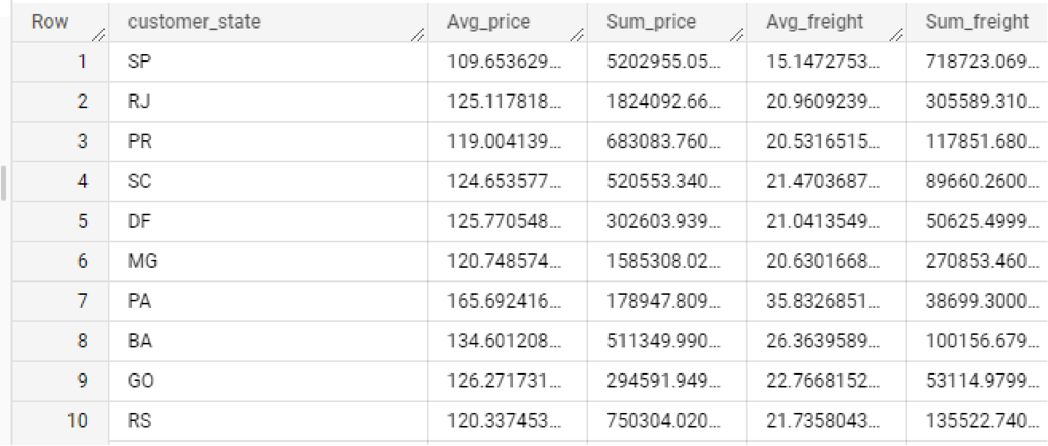
```
select distinct YEAR, sum(payment_value) as Total_Cost from
( select extract(year from o.order_purchase_timestamp) YEAR,extract(month from o.order_purchase_timestamp) MONTH, p.payment_value
  from case_study_1.orders o join case_study_1.payments p on o.order_id = p.order_id
  group by YEAR, MONTH
)
where YEAR in (2017,2018) and MONTH between 1 and 8
group by YEAR
order by YEAR asc
```

-- 4.2. Mean and sum of price and freight value by customer state --

Query

```
select distinct c.customer_state, avg(oi.price) Avg_price, sum(oi.price) Sum_price, Avg(oi.freight_value) Avg_freight, sum(oi.freight_value) Sum_freight
from case_study_1.order_items oi
join case_study_1.orders o on oi.order_id = o.order_id
join case_study_1.customers c on o.customer_id = c.customer_id
group by c.customer_state
```

Result



Row	customer_state	Avg_price	Sum_price	Avg_freight	Sum_freight
1	SP	109.653629...	5202955.05...	15.1472753...	718723.069...
2	RJ	125.117818...	1824092.66...	20.9609239...	305589.310...
3	PR	119.004139...	683083.760...	20.5316515...	117851.680...
4	SC	124.653577...	520553.340...	21.4703687...	89660.2600...
5	DF	125.770548...	302603.939...	21.0413549...	50625.4999...
6	MG	120.748574...	1585308.02...	20.6301668...	270853.460...
7	PA	165.692416...	178947.809...	35.8326851...	38699.3000...
8	BA	134.601208...	511349.990...	26.3639589...	100156.679...
9	GO	126.271731...	294591.949...	22.7668152...	53114.9799...
10	RS	120.337453...	750304.020...	21.7358043...	135522.740...

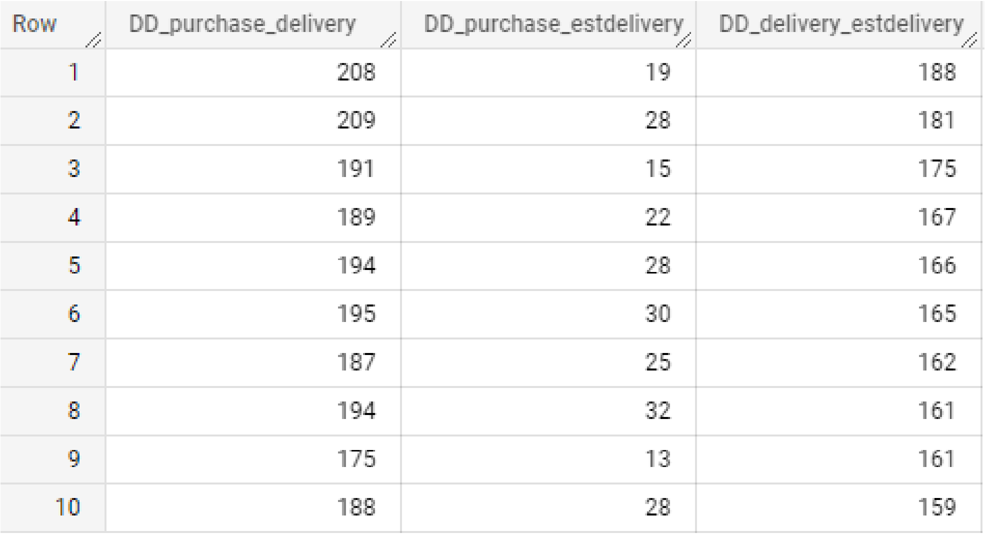
Conclusion

-- 5.1. Calculate days between purchasing, delivering and estimated_delivery

Query

```
select timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as DD_purchase_delivery,
timestamp_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as DD_purchase_estdelivery,
timestamp_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as DD_delivery_estdelivery
from case_study_1.orders
where order_delivered_customer_date is not null
order by DD_delivery_estdelivery desc
```

Result



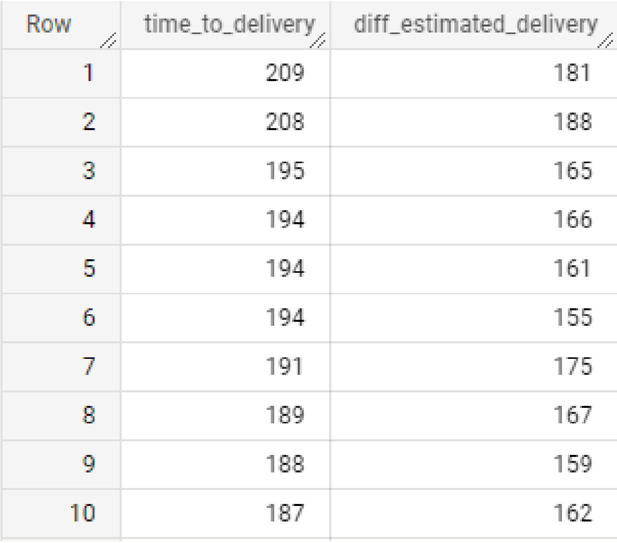
Row	DD_purchase_delivery	DD_purchase_estdelivery	DD_delivery_estdelivery
1	208	19	188
2	209	28	181
3	191	15	175
4	189	22	167
5	194	28	166
6	195	30	165
7	187	25	162
8	194	32	161
9	175	13	161
10	188	28	159

-- 5.2. Find time_to_delivery and diff_estimated_delivery --

Query

```
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 case_study_1.orders
where order_delivered_customer_date is not null
order by time_to_delivery desc, diff_estimated_delivery desc
```

Result



Row	time_to_delivery	diff_estimated_delivery
1	209	181
2	208	188
3	195	165
4	194	166
5	194	161
6	194	155
7	191	175
8	189	167
9	188	159
10	187	162

- 5.3. Group data by state. Take mean of freight_value, time_to_delivery, diff_estimated_delivery --

Query

```
select avg(oi.freight_value) Avg_freight_value, avg(timestamp_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) over(partition by c.customer_state) as time_to_delivery,
      avg(timestamp_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)) over(partition by c.customer_state) as diff_estimated_delivery,c.customer_state
from case_study_1.order_items oi
join  case_study_1.orders o  on o.order_id = oi.order_id
join case_study_1.customers c on o.customer_id = c.customer_id
where o.order_delivered_customer_date is not null
group by c.customer_state
order by time_to_delivery desc, diff_estimated_delivery desc
```

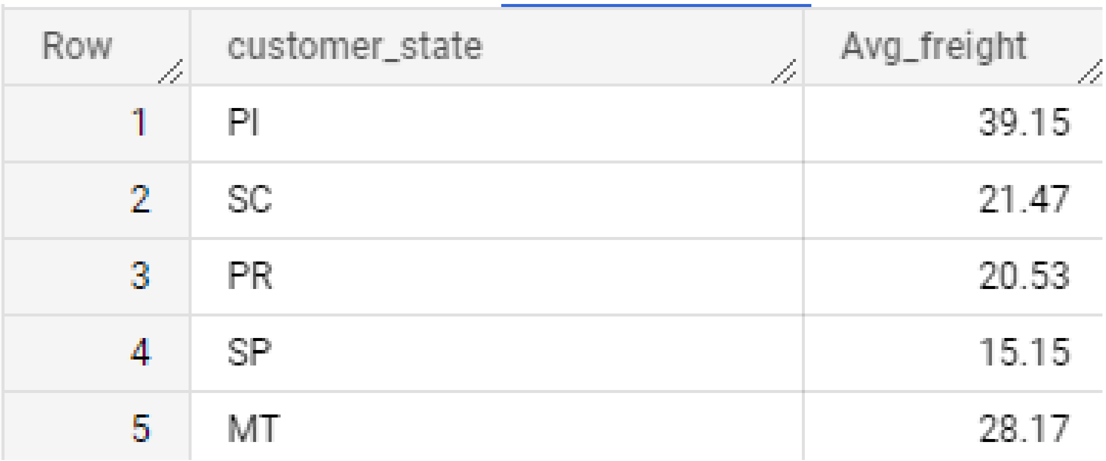
Result

--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) over(partition by c.customer_state order by c.customer_state desc),2) as Avg_freight
from case_study_1.order_items oi
join  case_study_1.orders o  on o.order_id = oi.order_id
join case_study_1.customers c on o.customer_id = c.customer_id
order by oi.freight_value desc
limit 5
```

Result



Row	customer_state	Avg_freight
1	PI	39.15
2	SC	21.47
3	PR	20.53
4	SP	15.15
5	MT	28.17

Conclusion

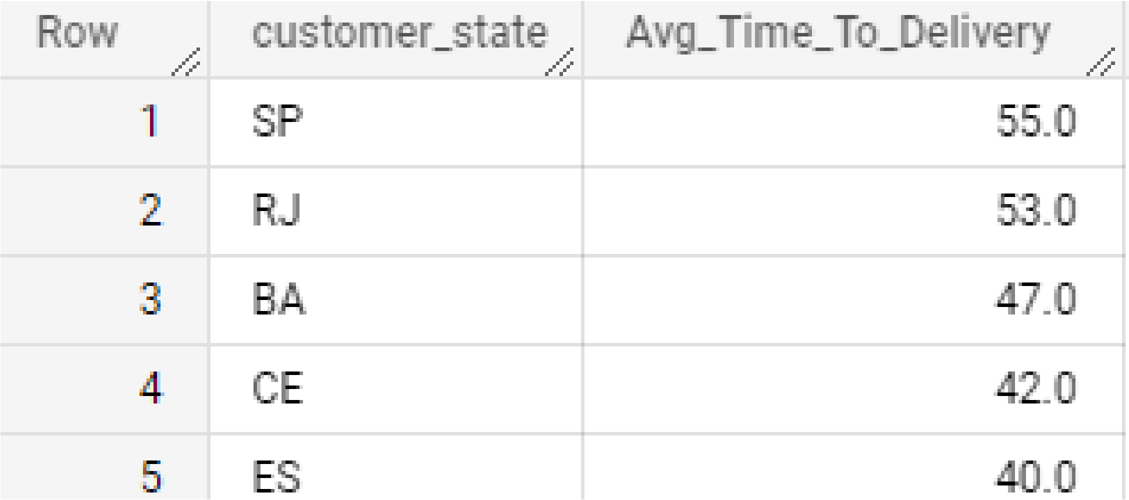
The top 5 states with highest average freight value are PI, SC. PR, SP and MT with state PI showing the highest average freight value i.e. 39.15.

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

Query

```
select distinct c.customer_state, round(avg(x.time_to_delivery) over(partition by c.customer_state)) as Avg_Time_To_Delivery
from
    (select customer_id, timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery
    from case_study_1.orders
    ) x join case_study_1.customers c on x.customer_id = c.customer_id
group by c.customer_state, x.time_to_delivery
order by Avg_Time_To_Delivery desc
limit 5
```

Result



Row	customer_state	Avg_Time_To_Delivery
1	SP	55.0
2	RJ	53.0
3	BA	47.0
4	CE	42.0
5	ES	40.0

Conclusion

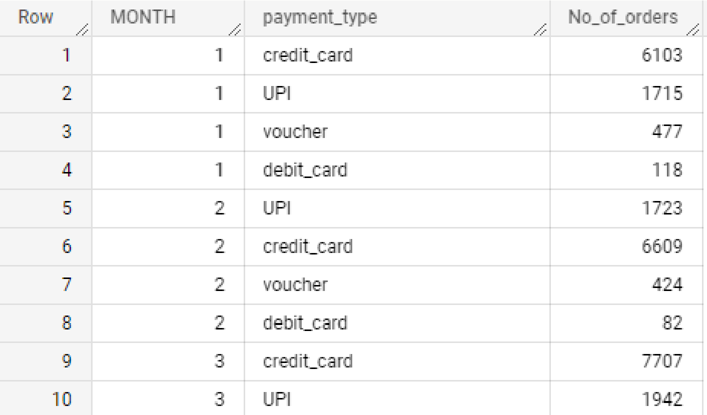
The top 5 states with the highest time to delivery are SP, RJ, BA, CE and ES with state SP taking the highest time to delivery with 55 days.

-- 6.1. Month over month count of orders for different payment types --

Query

```
select distinct MONTH, payment_type, No_of_orders
from
(
select distinct p.payment_type,
extract(month from o.order_purchase_timestamp) as MONTH, count(o.order_id) as No_of_orders
from case_study_1.orders o join case_study_1.payments p on o.order_id = p.order_id
group by MONTH, p.payment_type
)
order by MONTH asc
```

Result



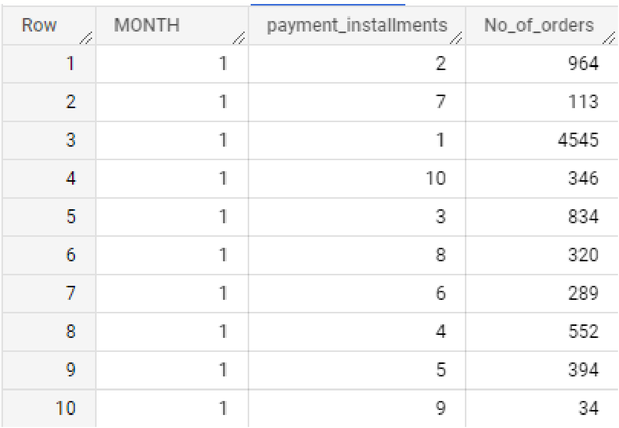
Row	MONTH	payment_type	No_of_orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6609
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942

--6.2 Count of orders based on the number of payment installments --

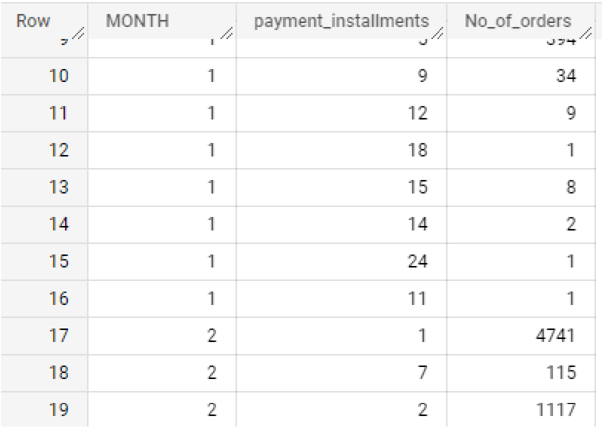
Query

```
select distinct MONTH, payment_installments, No_of_orders
from
(
select distinct p.payment_installments,
extract(month from o.order_purchase_timestamp) as MONTH, count(o.order_id) as No_of_orders
from case_study_1.orders o join case_study_1.payments p on o.order_id = p.order_id
group by p.payment_installments, MONTH )
order by MONTH asc
```

Result



Row	MONTH	payment_installments	No_of_orders
1	1	2	964
2	1	7	113
3	1	1	4545
4	1	10	346
5	1	3	834
6	1	8	320
7	1	6	289
8	1	4	552
9	1	5	394
10	1	9	34



Row	MONTH	payment_installments	No_of_orders
10	1	9	34
11	1	12	9
12	1	18	1
13	1	15	8
14	1	14	2
15	1	24	1
16	1	11	1
17	2	1	4741
18	2	7	115
19	2	2	1117