1. Time period for which the data is given

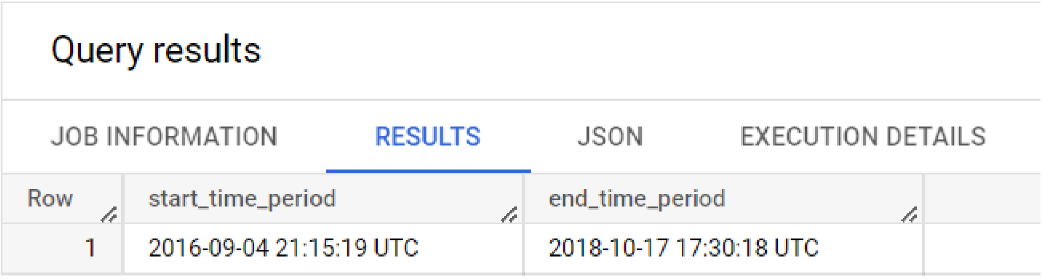
Query:

```
SELECT

MIN(order_purchase_timestamp) AS start_time_period,

MAX(order_purchase_timestamp) AS end_time_period

FROM `target-scaler-360317.customers.orders`
```

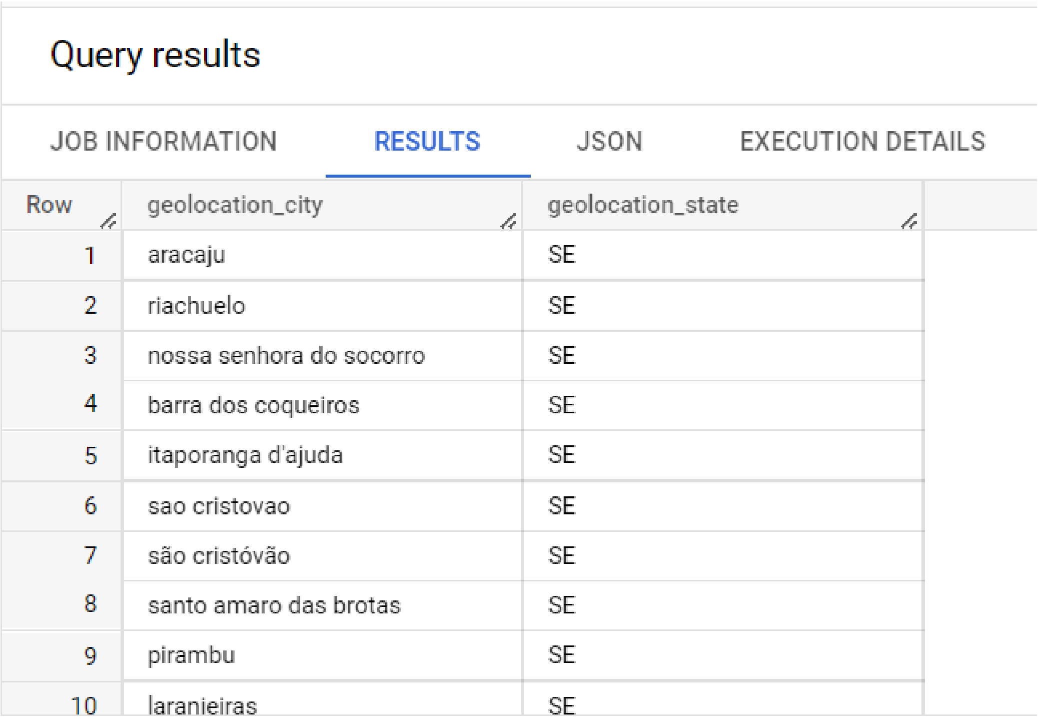


Row	start_time_period	end_time_period
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

2. Cities and States covered in the dataset

Query:

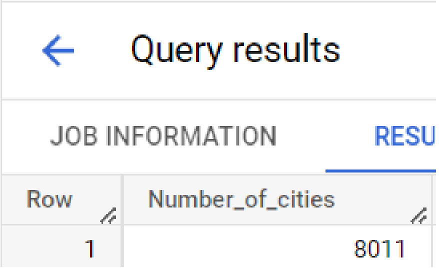
```
SELECT DISTINCT geolocation_city,geolocation_state
FROM  geolocation
```



Row	geolocation_city	geolocation_state
1	aracaju	SE
2	riachuelo	SE
3	nossa senhora do socorro	SE
4	barra dos coqueiros	SE
5	itaporanga d'ajuda	SE
6	sao cristovao	SE
7	são cristóvão	SE
8	santo amaro das brotas	SE
9	pirambu	SE
10	laranjeiras	SE

Query:

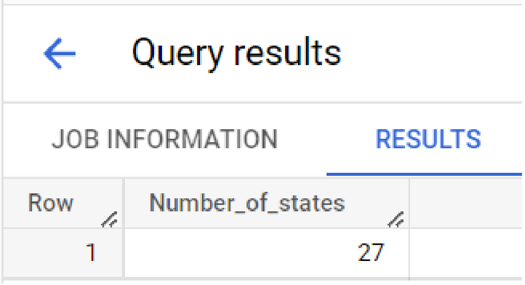
```
SELECT COUNT(DISTINCT geolocation_city) AS Number_of_cities
FROM geolocation;
```



Row	Number_of_cities
1	8011

Query:

```
SELECT COUNT(DISTINCT geolocation_state) AS Number_of_states
FROM .geolocation
```



Row	Number_of_states
1	27

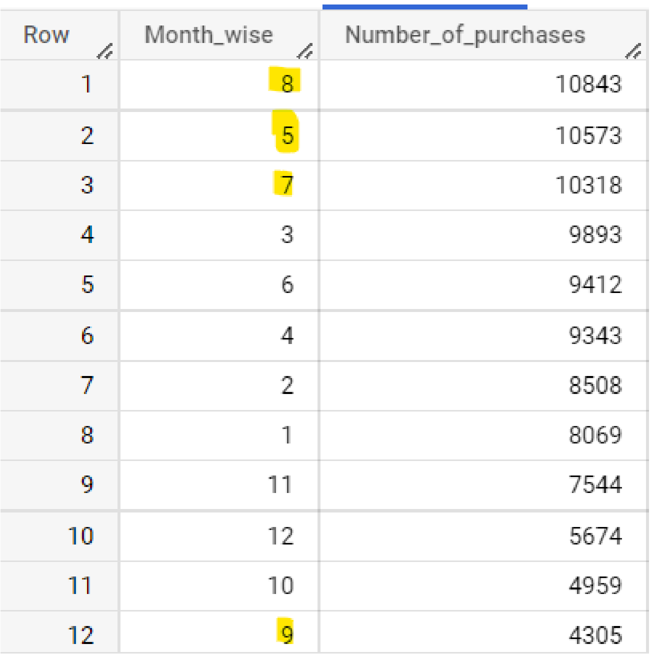
In-depth Exploration:

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?**

Month- Wise

Query:

```
WITH month_purchase AS
(
SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS Month_wise
FROM orders
)
SELECT Month_wise,COUNT(Month_wise) AS Number_of_purchases
from month_purchase
GROUP BY Month_wise
ORDER BY COUNT(Month_wise) DESC;
```



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

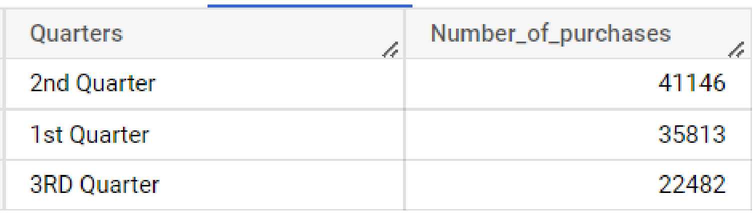
- **As we see in the Screenshot 8,5,7 month have more number of purchases compares to other months and 9th month have lowest purchases.**
- **if we observer it seems like there is pattern according to different quarters in the year.**

Quarter – wise

Query:

```
WITH month_purchase AS
(
SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS Month_wise
FROM orders
),
quarter_purchse AS
(
SELECT CASE
WHEN Month_wise IN (1,2,3,4) THEN "1st Quarter"
WHEN Month_wise IN (5,6,7,8) THEN "2nd Quarter"
ELSE "3RD Quarter"
END AS Quarters
FROM month_purchase
)

SELECT Quarters,COUNT(Quarters) AS Number_of_purchases
from quarter_purchse
GROUP BY Quarters
ORDER BY COUNT(Quarters) DESC;
```



Quarters	Number_of_purchases
2nd Quarter	41146
1st Quarter	35813
3RD Quarter	22482

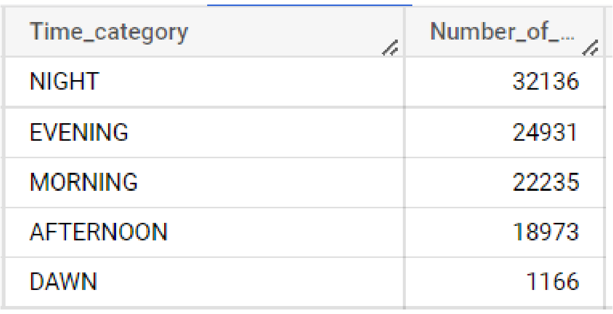
3rd Quarter purchases are very low compares to other quarters ,2nd quarter have high purchases counts.

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query:

```
WITH time_purchase AS
(
SELECT EXTRACT(TIME FROM order_purchase_timestamp) AS time_wise
FROM orders
),
category AS (
SELECT
CASE
WHEN time_wise BETWEEN "02:01:00" AND "06:00:00"
THEN "DAWN"
WHEN time_wise BETWEEN "06:01:00" AND "12:00:00"
THEN "MORNING"
WHEN time_wise BETWEEN "12:01:00" AND "15:00:00"
THEN "AFTERNOON"
WHEN time_wise BETWEEN "15:01:00" AND "19:00:00"
THEN "EVENING"
ELSE "NIGHT"
END AS Time_category
FROM time_purchase
)

SELECT Time_category,
COUNT(Time_category) AS Number_of_purchases
FROM category
GROUP BY Time_category
ORDER BY COUNT(Time_category) DESC;
```



Time_category	Number_of_...
NIGHT	32136
EVENING	24931
MORNING	22235
AFTERNOON	18973
DAWN	1166

It clearly makes sense as most of the people will be free at evening and night times that’s why at that time we see increase in number of purchases.

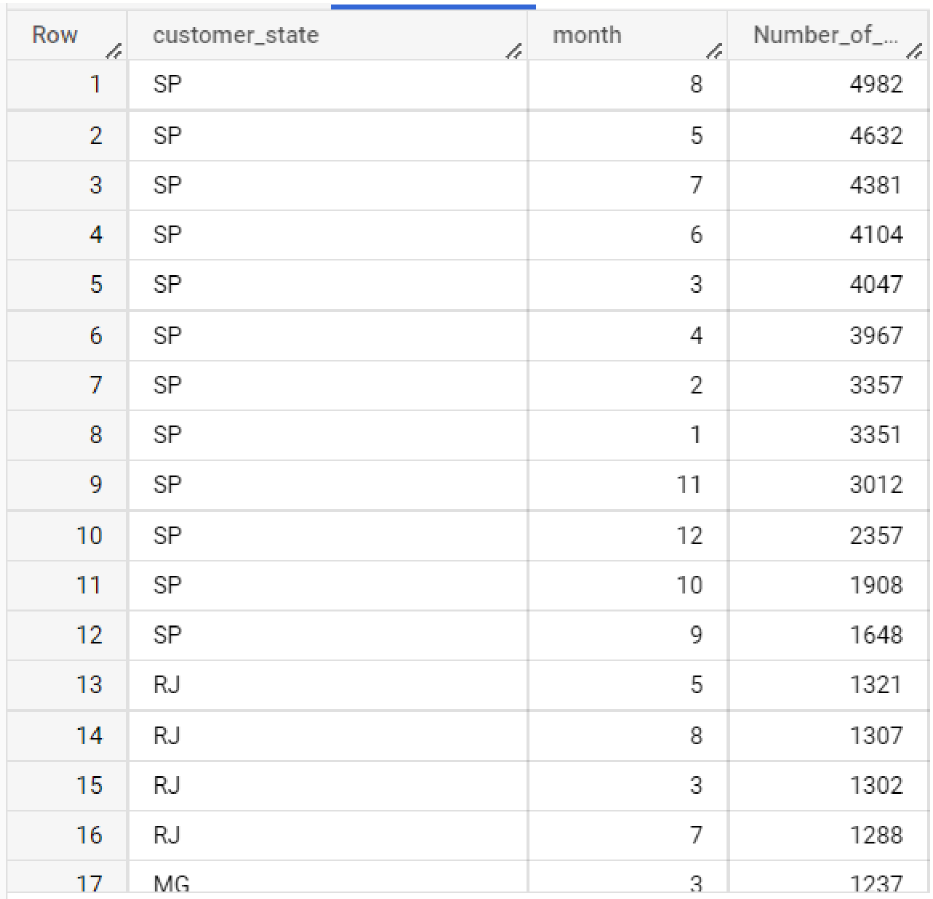
3.Evolution of E-commerce orders in the Brazil region:

1. Get month on month orders by states

Query:

```
WITH monthadd AS(
 SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS month
 FROM  orders
)
```

```
SELECT c.customer_state,o.month,COUNT(o.order_id) AS Number_of_purchases
FROM monthadd AS o
JOIN customers AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state,o.month
ORDER BY COUNT(o.order_id) DESC ,c.customer_state;
```



Row	customer_state	month	Number_of_...
1	SP	8	4982
2	SP	5	4632
3	SP	7	4381
4	SP	6	4104
5	SP	3	4047
6	SP	4	3967
7	SP	2	3357
8	SP	1	3351
9	SP	11	3012
10	SP	12	2357
11	SP	10	1908
12	SP	9	1648
13	RJ	5	1321
14	RJ	8	1307
15	RJ	3	1302
16	RJ	7	1288
17	MG	3	1237

SP – state have clearly very high number of purchases in all months

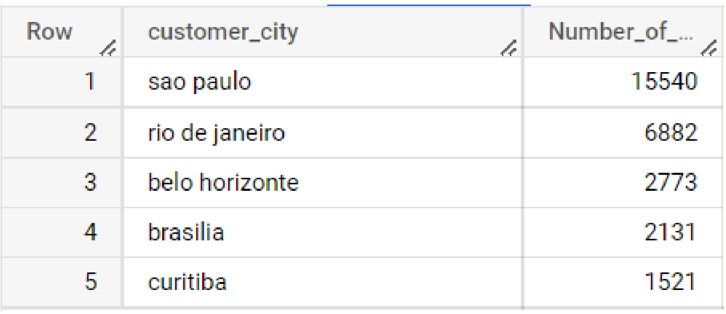
City Wise – Highest and Lowest number of purchases , 5 Cities in each case.

Query:

TOP 5:

```
SELECT c.customer_city,COUNT(o.order_id) AS Number_of_purchases
FROM orders AS o
JOIN  customers AS c
ON o.customer_id = c.customer_id
```

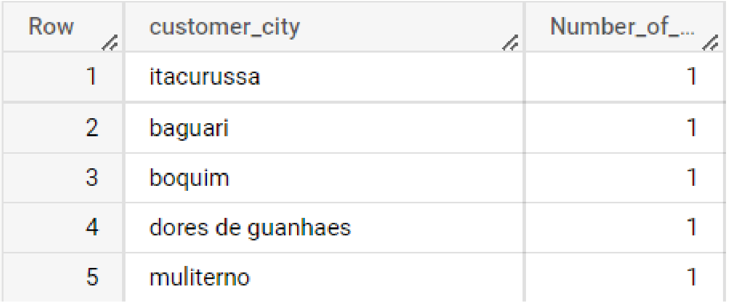
```sql
GROUP BY c.customer_city
ORDER BY COUNT(o.order_id) DESC
LIMIT 5;
```



Row	customer_city	Number_of_…
1	sao paulo	15540
2	rio de janeiro	6882
3	belo horizonte	2773
4	brasilia	2131
5	curitiba	1521

Least 5

```sql
SELECT c.customer_city,COUNT(o.order_id) AS Number_of_purchases
FROM orders AS o
JOIN customers AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_city
ORDER BY COUNT(o.order_id)
LIMIT 5;
```



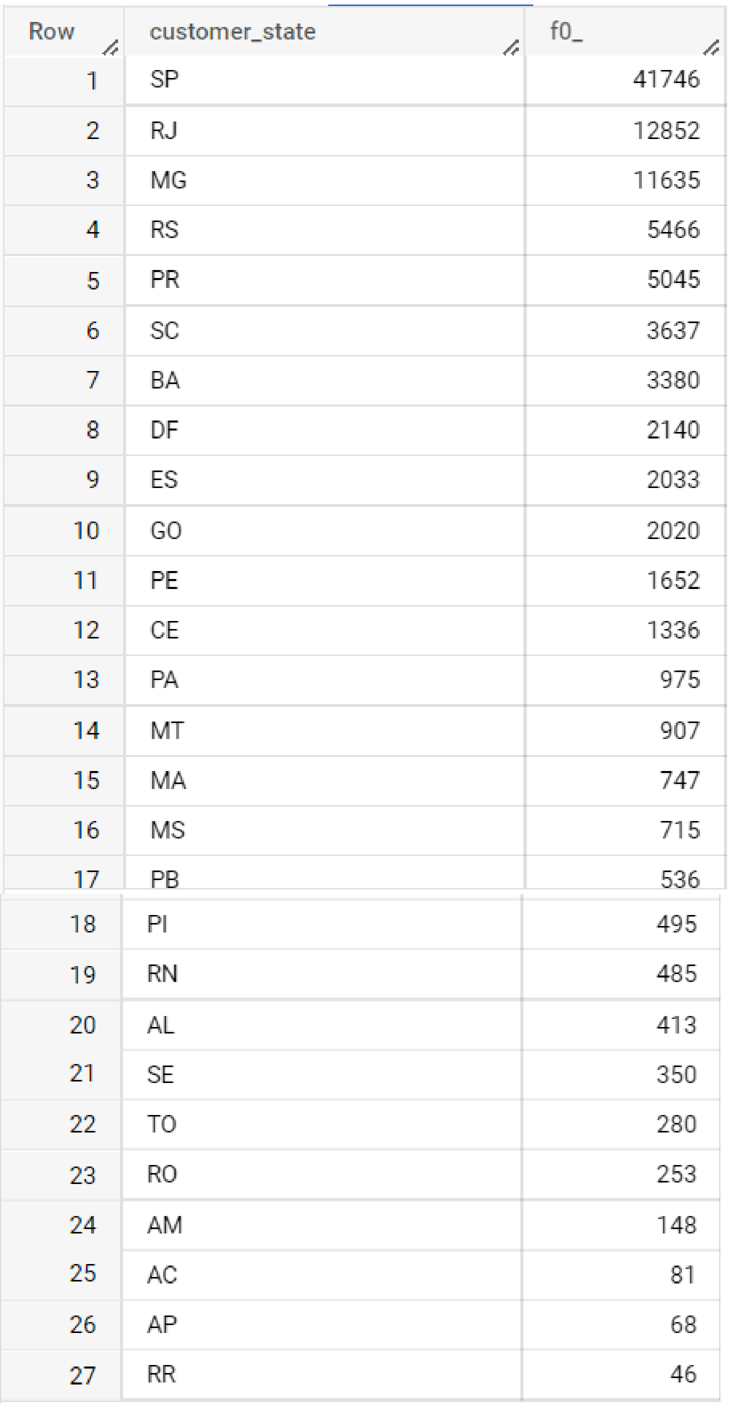
Row	customer_city	Number_of_…
1	itacurussa	1
2	baguari	1
3	boquim	1
4	dores de guanhaes	1
5	muliterno	1

Sao Paulo have the highest number of purchases by far – This also makes sense as Sao Paulo is populous city in Brazil

2. How are customers distributed in Brazil

Query:

```sql
SELECT customer_state, COUNT(customer_unique_id)
FROM customers
GROUP BY customer_state
ORDER BY COUNT(customer_unique_id) DESC;
```



Row	customer_state	f0_
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336
13	PA	975
14	MT	907
15	MA	747
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

As we saw above SP had more purchases it links well here as more number of customers, more number of purchases.

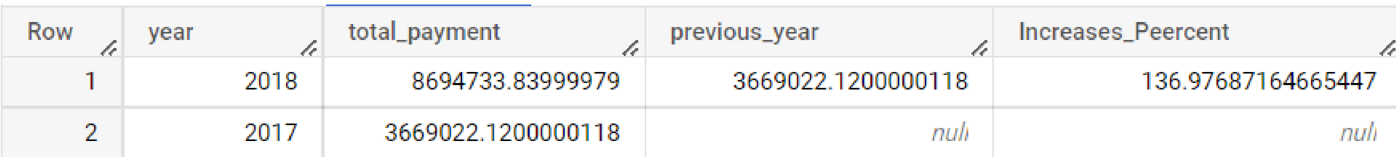
Impact on Economy: Analyze the money movemented by e-commerce by looking at order prices, freight and others.

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

QUERY:

```
WITH monthyearadd AS(
  SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  EXTRACT(YEAR FROM order_purchase_timestamp) AS year
  FROM orders
),
year_wise AS(
SELECT m.year,SUM(p.payment_value) AS total_payment
FROM monthyearadd AS m
JOIN payments AS p
ON m.order_id = p.order_id
WHERE m.year  in  (2017,2018) AND m.month BETWEEN 1 AND 8
GROUP BY m.year
)

SELECT *,LAG(total_payment) OVER(order by year) AS previous_year,
(total_payment-
(LAG(total_payment) OVER(order by year)))/(LAG(total_payment) OVER(order by year))*100
 AS Increased_Percentage
FROM year_wise;
```



Row	year	total_payment	previous_year	Increases_Peercent
1	2018	8694733.83999979	3669022.1200000118	136.97687164665447
2	2017	3669022.1200000118	null	null

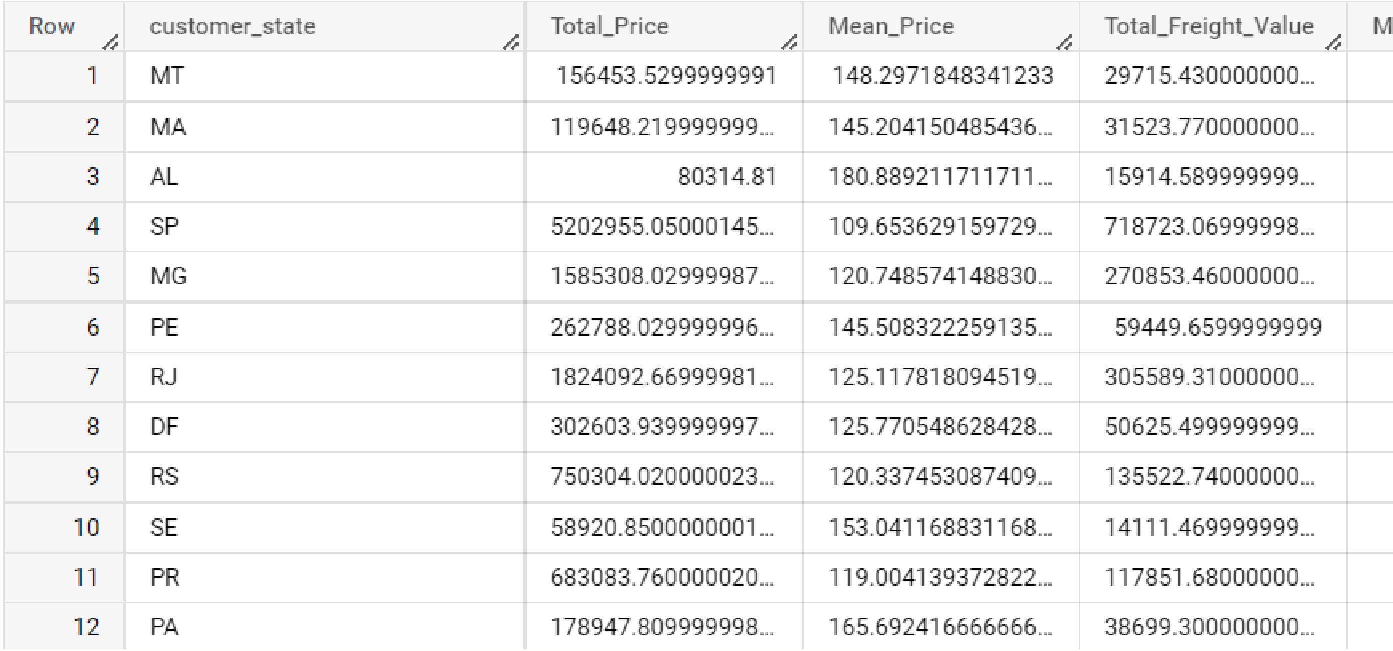
We have 136% Increase in payments from 2017 to 2018 (include months between Jan to Aug only)

2. **Mean & Sum of price and freight value by customer state**

QUERY:

```
SELECT c.customer_state,
SUM(ot.price) AS Total_Price ,
AVG(ot.price) AS Mean_Price ,
SUM(ot.freight_value) AS Total_Freight_Value ,
AVG(ot.freight_value) AS Mean_Freight_Value
FROM order_items AS ot
JOIN orders AS o
```

```
ON ot.order_id = o.order_id
JOIN customers AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
```



Row	customer_state	Total_Price	Mean_Price	Total_Freight_Value	M
1	MT	156453.5299999991	148.2971848341233	29715.430000000...	
2	MA	119648.219999999...	145.204150485436...	31523.770000000...	
3	AL	80314.81	180.889211711711...	15914.589999999...	
4	SP	5202955.05000145...	109.653629159729...	718723.06999998...	
5	MG	1585308.02999987...	120.748574148830...	270853.46000000...	
6	PE	262788.029999996...	145.508322259135...	59449.6599999999	
7	RJ	1824092.66999981...	125.117818094519...	305589.31000000...	
8	DF	302603.939999997...	125.770548628428...	50625.499999999...	
9	RS	750304.020000023...	120.337453087409...	135522.74000000...	
10	SE	58920.8500000001...	153.041168831168...	14111.469999999...	
11	PR	683083.760000020...	119.004139372822...	117851.68000000...	
12	PA	178947.809999998...	165.692416666666...	38699.300000000...	

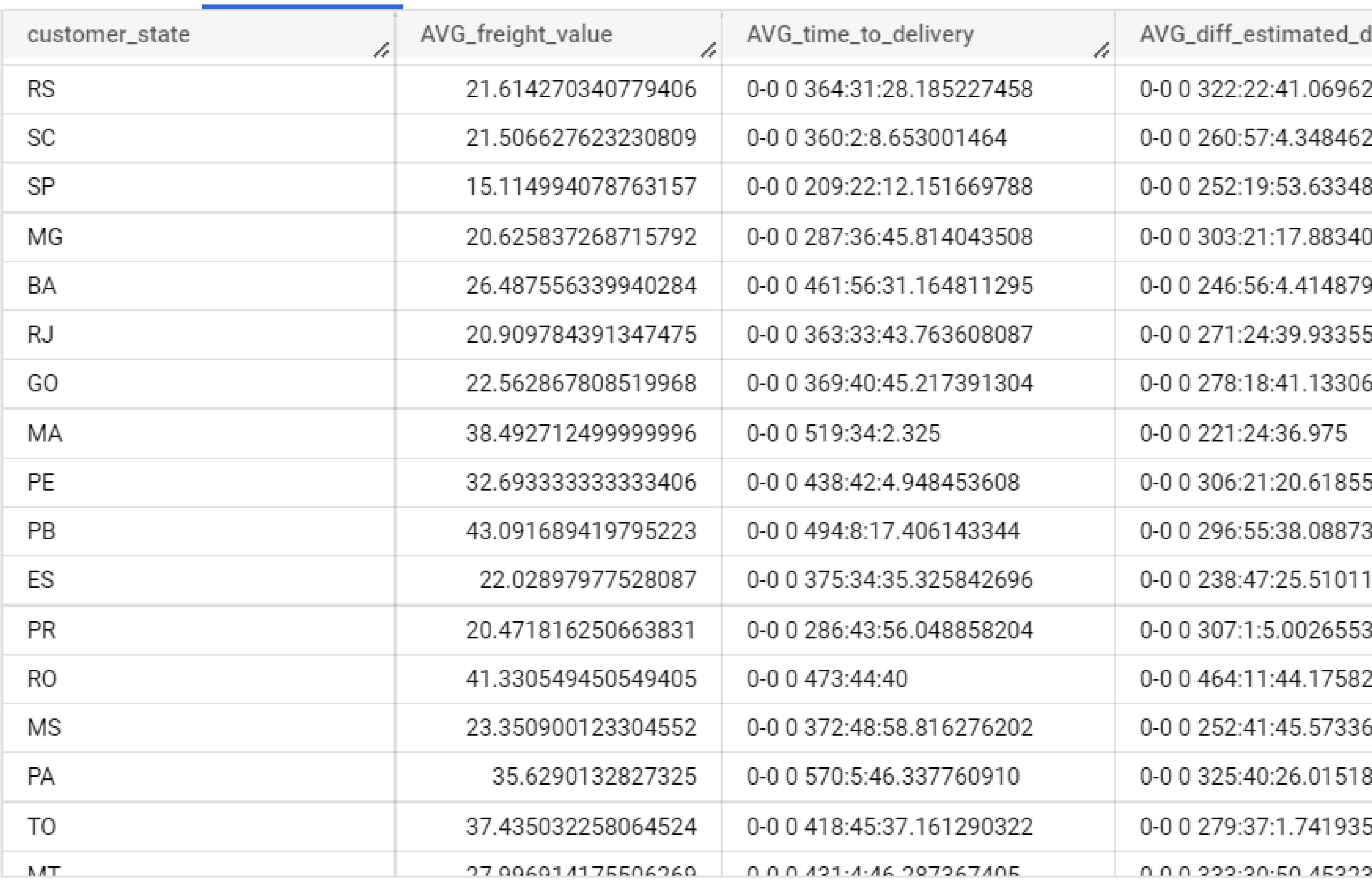
Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

Query:

```
WITH diff AS(
SELECT  *,order_delivered_customer_date-order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS NOT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,AVG(ot.freight_value) AS AVG_freight_value,
AVG(o.time_to_delivery) AS AVG_time_to_delivery, AVG(o.diff_estimated_delivery) AS AVG_diff_estimated_delivery
FROM customers AS c
JOIN diff AS o
```

```
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
```



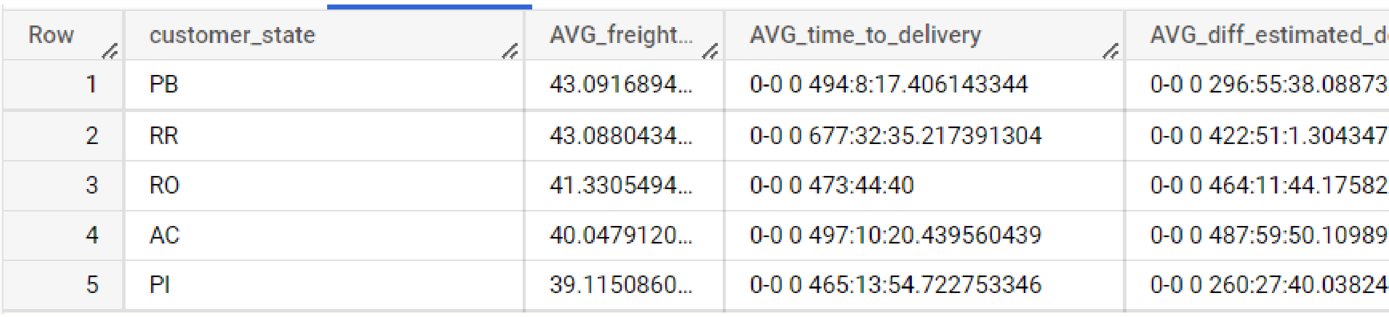
customer_state	AVG_freight_value	AVG_time_to_delivery	AVG_diff_estimated_d
RS	21.614270340779406	0-0 0 364:31:28.185227458	0-0 0 322:22:41.06962
SC	21.506627623230809	0-0 0 360:2:8.653001464	0-0 0 260:57:4.348462
SP	15.114994078763157	0-0 0 209:22:12.151669788	0-0 0 252:19:53.63348
MG	20.625837268715792	0-0 0 287:36:45.814043508	0-0 0 303:21:17.88340
BA	26.487556339940284	0-0 0 461:56:31.164811295	0-0 0 246:56:4.414879
RJ	20.909784391347475	0-0 0 363:33:43.763608087	0-0 0 271:24:39.93355
GO	22.562867808519968	0-0 0 369:40:45.217391304	0-0 0 278:18:41.13306
MA	38.492712499999996	0-0 0 519:34:2.325	0-0 0 221:24:36.975
PE	32.693333333333406	0-0 0 438:42:4.948453608	0-0 0 306:21:20.61855
PB	43.091689419795223	0-0 0 494:8:17.406143344	0-0 0 296:55:38.08873
ES	22.02897977528087	0-0 0 375:34:35.325842696	0-0 0 238:47:25.51011
PR	20.471816250663831	0-0 0 286:43:56.048858204	0-0 0 307:1:5.0026553
RO	41.330549450549405	0-0 0 473:44:40	0-0 0 464:11:44.17582
MS	23.350900123304552	0-0 0 372:48:58.816276202	0-0 0 252:41:45.57336
PA	35.6290132827325	0-0 0 570:5:46.337760910	0-0 0 325:40:26.01518
TO	37.435032258064524	0-0 0 418:45:37.161290322	0-0 0 279:37:1.741935
MT	27.996914175506269	0-0 0 431:4:46.287367405	0-0 0 333:30:50.45323

Top 5 states with highest/lowest average freight value:

Highest:

```
WITH diff AS(
SELECT  *,order_delivered_customer_date-
order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS N
OT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,AVG(ot.freight_value) AS AVG_freight_value,
AVG(o.time_to_delivery) AS AVG_time_to_delivery, AVG(o.diff_estimated_delivery) AS AVG
_diff_estimated_delivery
```

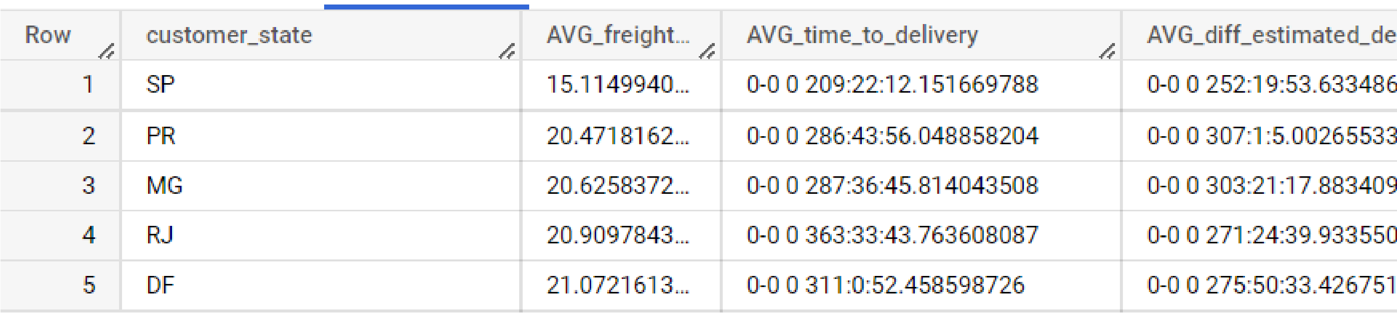
```
FROM customers AS c
JOIN diff AS o
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY AVG(ot.freight_value) DESC
LIMIT 5
```



Row	customer_state	AVG_freight...	AVG_time_to_delivery	AVG_diff_estimated_d
1	PB	43.0916894...	0-0 0 494:8:17.406143344	0-0 0 296:55:38.08873
2	RR	43.0880434...	0-0 0 677:32:35.217391304	0-0 0 422:51:1.304347
3	RO	41.3305494...	0-0 0 473:44:40	0-0 0 464:11:44.17582
4	AC	40.0479120...	0-0 0 497:10:20.439560439	0-0 0 487:59:50.10989
5	PI	39.1150860...	0-0 0 465:13:54.722753346	0-0 0 260:27:40.03824

Lowest:

```
WITH diff AS(
SELECT  *,order_delivered_customer_date-
order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS N
OT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,AVG(ot.freight_value) AS AVG_freight_value,
AVG(o.time_to_delivery) AS AVG_time_to_delivery, AVG(o.diff_estimated_delivery) AS AVG
_diff_estimated_delivery
FROM customers AS c
JOIN diff AS o
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY AVG(ot.freight_value)
LIMIT 5
```

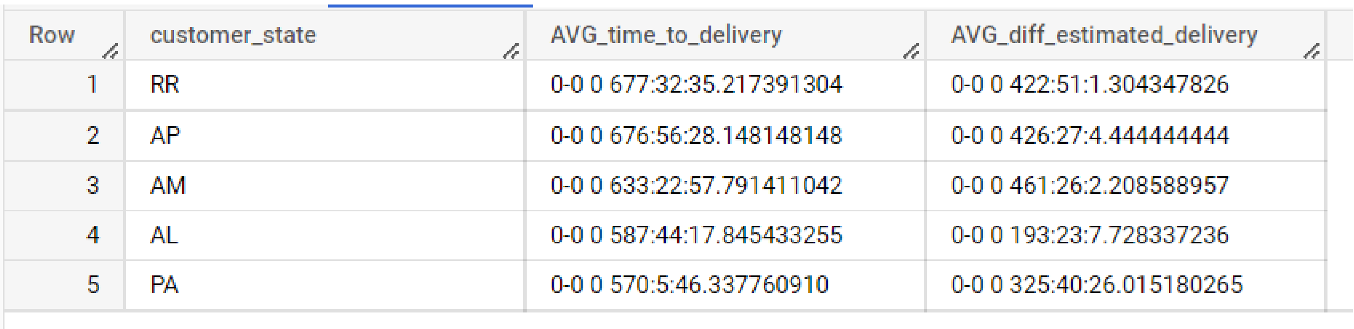


Row	customer_state	AVG_freight...	AVG_time_to_delivery	AVG_diff_estimated_de
1	SP	15.1149940...	0-0 0 209:22:12.151669788	0-0 0 252:19:53.633486
2	PR	20.4718162...	0-0 0 286:43:56.048858204	0-0 0 307:1:5.00265533
3	MG	20.6258372...	0-0 0 287:36:45.814043508	0-0 0 303:21:17.883409
4	RJ	20.9097843...	0-0 0 363:33:43.763608087	0-0 0 271:24:39.933550
5	DF	21.0721613...	0-0 0 311:0:52.458598726	0-0 0 275:50:33.426751

Top 5 states with highest/lowest average time to delivery

Highest:

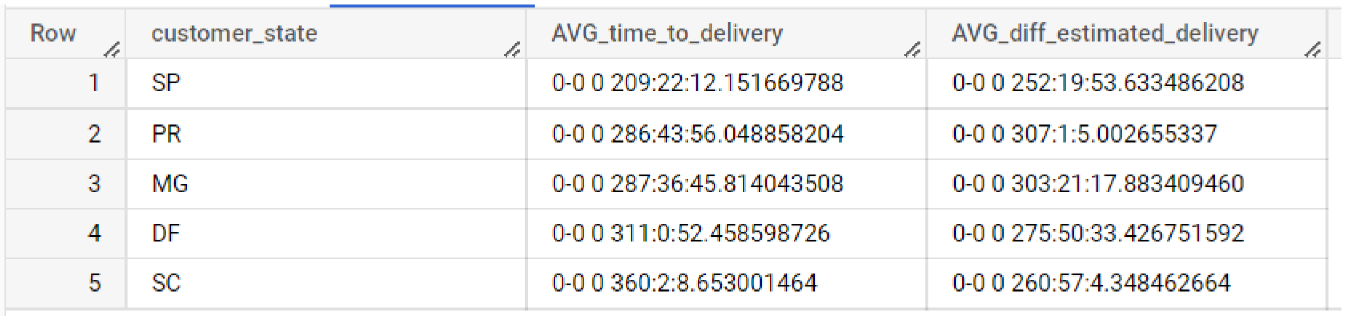
```
WITH diff AS(
SELECT  *,order_delivered_customer_date-
order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS N
OT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,
AVG(o.time_to_delivery) AS AVG_time_to_delivery, AVG(o.diff_estimated_delivery) AS AVG
_diff_estimated_delivery
FROM customers AS c
JOIN diff AS o
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY  AVG(o.time_to_delivery) DESC
LIMIT 5
```



Row	customer_state	AVG_time_to_delivery	AVG_diff_estimated_delivery
1	RR	0-0 0 677:32:35.217391304	0-0 0 422:51:1.304347826
2	AP	0-0 0 676:56:28.148148148	0-0 0 426:27:4.444444444
3	AM	0-0 0 633:22:57.791411042	0-0 0 461:26:2.208588957
4	AL	0-0 0 587:44:17.845433255	0-0 0 193:23:7.728337236
5	PA	0-0 0 570:5:46.337760910	0-0 0 325:40:26.015180265

Lowest:

```
WITH diff AS(
SELECT  *,order_delivered_customer_date-
order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS N
OT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,
AVG(o.time_to_delivery) AS AVG_time_to_delivery, AVG(o.diff_estimated_delivery) AS AVG
_diff_estimated_delivery
FROM customers AS c
JOIN diff AS o
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY  AVG(o.time_to_delivery)
LIMIT 5
```



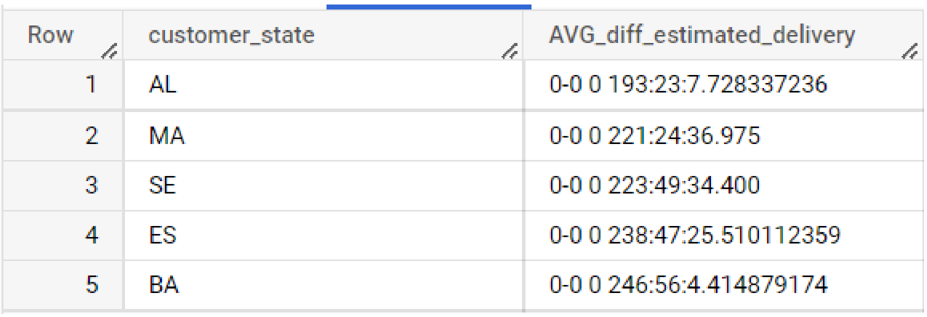
Row	customer_state	AVG_time_to_delivery	AVG_diff_estimated_delivery
1	SP	0-0 0 209:22:12.151669788	0-0 0 252:19:53.633486208
2	PR	0-0 0 286:43:56.048858204	0-0 0 307:1:5.002655337
3	MG	0-0 0 287:36:45.814043508	0-0 0 303:21:17.883409460
4	DF	0-0 0 311:0:52.458598726	0-0 0 275:50:33.426751592
5	SC	0-0 0 360:2:8.653001464	0-0 0 260:57:4.348462664

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

Really Fast:

```
WITH diff AS(
SELECT  *,order_delivered_customer_date-
order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS N
OT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,
AVG(o.diff_estimated_delivery) AS AVG_diff_estimated_delivery
```

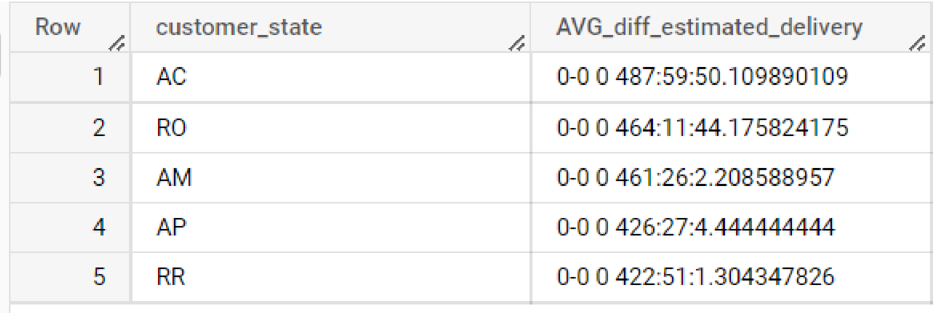
```
FROM customers AS c
JOIN diff AS o
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY  AVG(o.diff_estimated_delivery)
LIMIT 5
```



Row	customer_state	AVG_diff_estimated_delivery
1	AL	0-0 0 193:23:7.728337236
2	MA	0-0 0 221:24:36.975
3	SE	0-0 0 223:49:34.400
4	ES	0-0 0 238:47:25.510112359
5	BA	0-0 0 246:56:4.414879174

Not So Fast:

```
WITH diff AS(
SELECT  *,order_delivered_customer_date-
order_purchase_timestamp AS time_to_delivery,
order_estimated_delivery_date-order_delivered_customer_date AS diff_estimated_delivery
FROM orders
WHERE (order_delivered_customer_date IS NOT NULL AND order_purchase_timestamp IS N
OT NULL) AND order_estimated_delivery_date IS NOT NULL
)
SELECT c.customer_state,
AVG(o.diff_estimated_delivery) AS AVG_diff_estimated_delivery
FROM customers AS c
JOIN diff AS o
ON c.customer_id = o.customer_id
JOIN order_items AS ot
ON o.order_id = ot.order_id
GROUP BY c.customer_state
ORDER BY  AVG(o.diff_estimated_delivery) DESC
LIMIT 5
```



Row	customer_state	AVG_diff_estimated_delivery
1	AC	0-0 0 487:59:50.109890109
2	RO	0-0 0 464:11:44.175824175
3	AM	0-0 0 461:26:2.208588957
4	AP	0-0 0 426:27:4.444444444
5	RR	0-0 0 422:51:1.304347826

Payment type analysis:

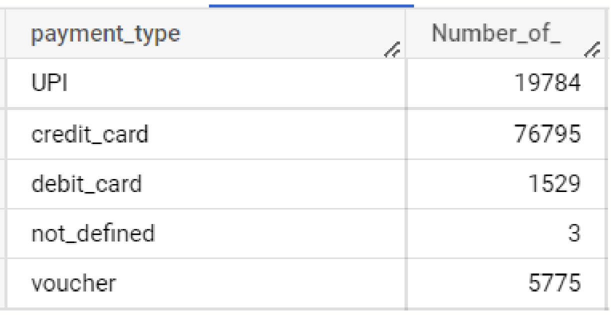
1. Month over Month count of orders for different payment types

Over Payment types

QUERY:

```
WITH monthadd AS(
 SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS month
 FROM orders
)

SELECT p.payment_type,COUNT(m.order_id) AS Number_of_orders
FROM monthadd AS m
JOIN payments AS p
ON m.order_id = p.order_id
GROUP BY p.payment_type
ORDER BY p.payment_type,COUNT(m.order_id);
```

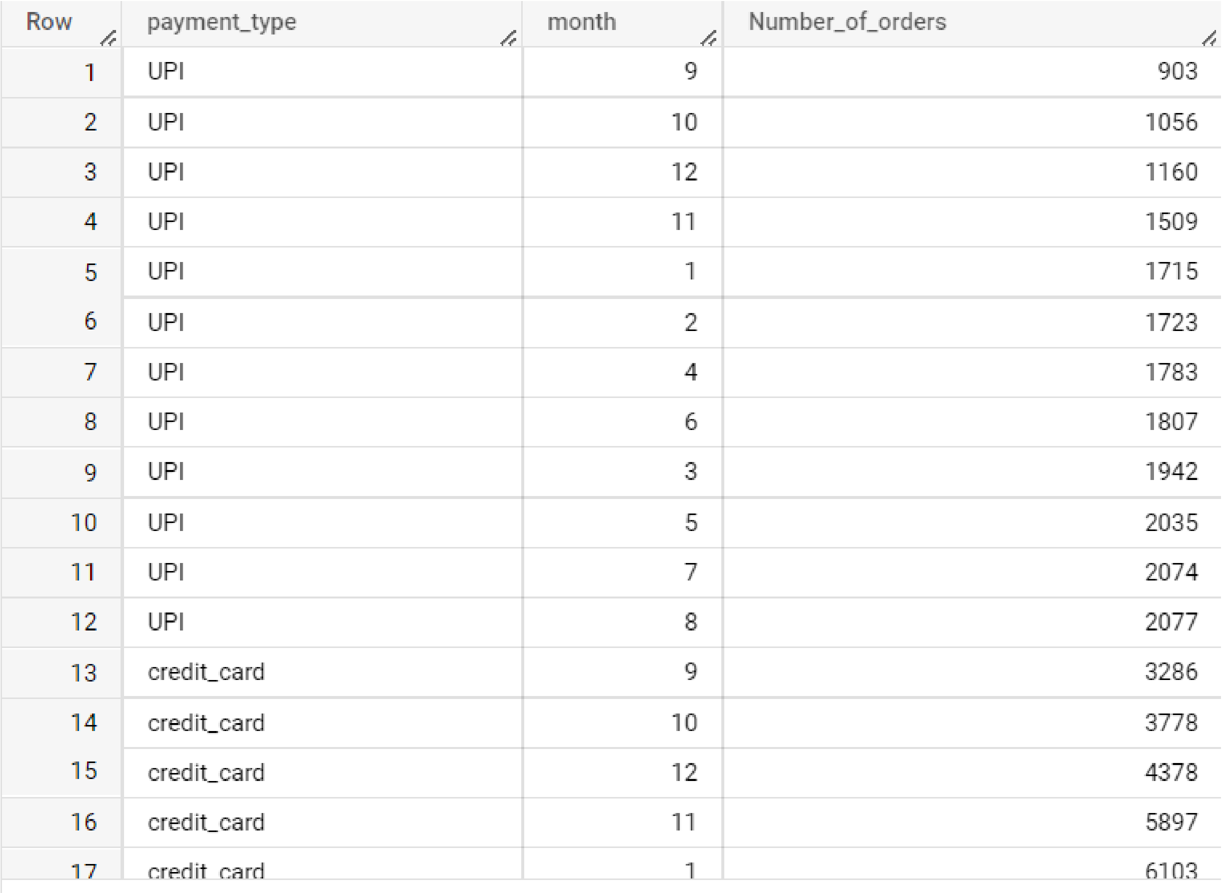


payment_type	Number_of_
UPI	19784
credit_card	76795
debit_card	1529
not_defined	3
voucher	5775

Over payments and months:

```
WITH monthadd AS(
 SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS month
 FROM orders
)
```

```
SELECT p.payment_type,m.month,COUNT(m.order_id) AS Number_of_orders
FROM monthadd AS m
JOIN payments AS p
ON m.order_id = p.order_id
GROUP BY p.payment_type,m.month
ORDER BY p.payment_type,COUNT(m.order_id);
```

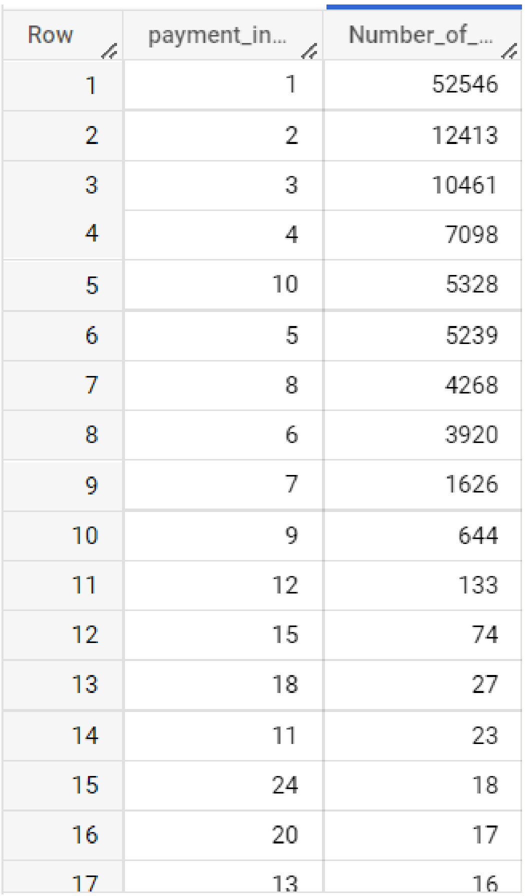


Row	payment_type	month	Number_of_orders
1	UPI	9	903
2	UPI	10	1056
3	UPI	12	1160
4	UPI	11	1509
5	UPI	1	1715
6	UPI	2	1723
7	UPI	4	1783
8	UPI	6	1807
9	UPI	3	1942
10	UPI	5	2035
11	UPI	7	2074
12	UPI	8	2077
13	credit_card	9	3286
14	credit_card	10	3778
15	credit_card	12	4378
16	credit_card	11	5897
17	credit_card	1	6103

2. Distribution of payment installments and count of orders

Query:

```
SELECT payment_installments,COUNT(order_id) AS Number_of_orders
FROM payments
GROUP BY payment_installments
ORDER BY COUNT(order_id) DESC
```



Row	payment_in...	Number_of_...
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644
11	12	133
12	15	74
13	18	27
14	11	23
15	24	18
16	20	17
17	13	16

Distribution – Most of our payment installments are between 1 to 10.

Actionable Insights and Recommendations:

- Comparing 2017- 2018 growth rate is above 100 percent in sales which implies we need to increase the physical workforce and branches in the cities where most sales happened.
- AS 2nd quarter have more sales basically in a year hiring the intern or temporary workers for that month will be good.
- Sao Paulo have the highest number of purchases by far -- Increase the workforce in the stores open new branch in the city.
- SP state have lowest freight value (lowest cost to transport goods) and also have most number of sales which means here we can derive maximum profit should set up more resource towards this state.
- AL,MA,SE are the states where the delivery are done fast (here fast means how much less difference in delivered time and estimated time) , Call a meeting with this state managers and analyse to find out how they deliver fast and try to replicate in other states as well.

- Most of the payments are done using Credit cards, if able to get credit data find out which bank card is used most and try to have a partnership with the bank might help in increase revenue.
- Most used payment installments by users are 1,2,3,4,10. Increasing these installments to be valid for many products as possible will increase the interest in customers.
- Evening and Night times we see increase in purchases, Alerting the employees in the store to have better focus and have best employee at sight in this period of time.