Target - Business Case Study

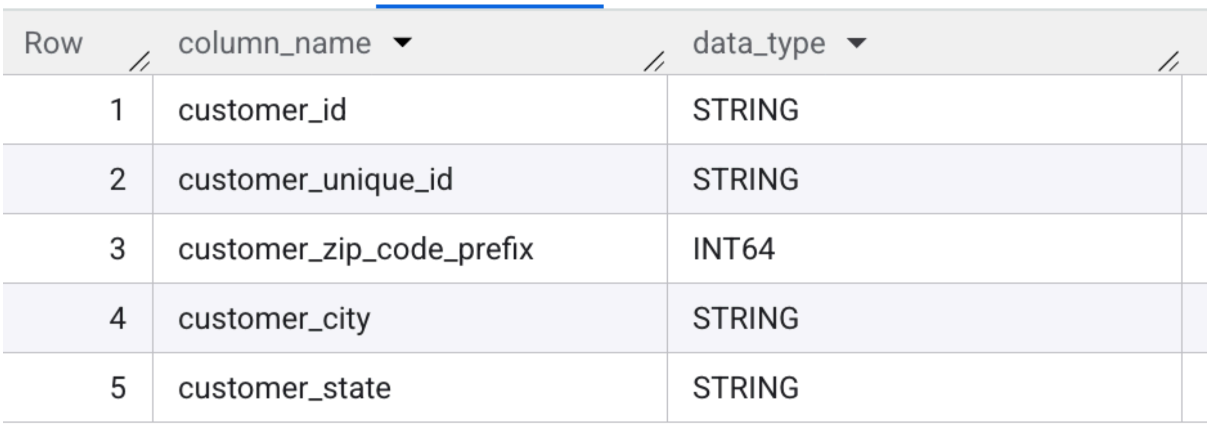
1. What does 'good' look like?

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

Query:

```
Select column_name, data_type from
`Business_case_Study.INFORMATION_SCHEMA.COLUMNS`
where table_name = ‘Customers’
```

Output:



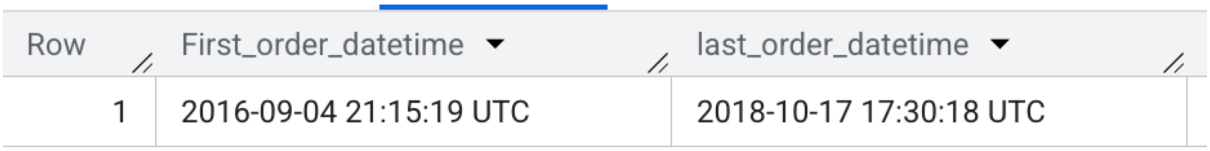
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

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

Query:

```
Select min(order_purchase_timestamp) First_order_datetime,
 max(order_purchase_timestamp) last_order_datetime
from `Business_case_Study.Orders`
```

Output:



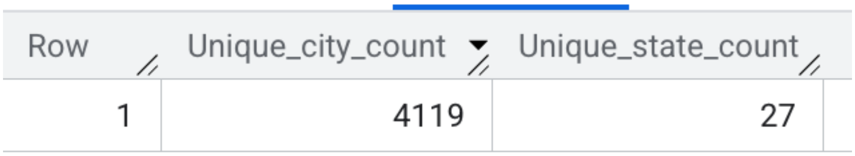
Row	First_order_datetime	last_order_datetime
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Target - Business Case Study

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

Query:

```
Select count(distinct customer_city) as Unique_city_count,
count(distinct customer_state) as Unique_state_count
from `Business_case_Study.Orders` as O
left join `Business_case_Study.Customers` as CC
on O.customer_id = CC.customer_id
```



Row	Unique_city_count	Unique_state_count
1	4119	27

2. In-depth Exploration:

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

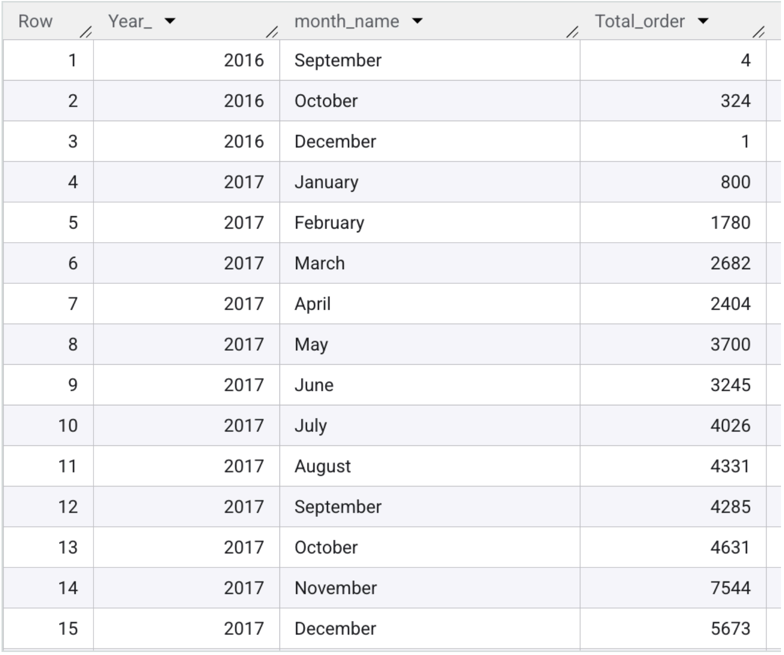
Query:

```
Select extract(year from order_purchase_timestamp) Year_,
format_datetime("%B",order_purchase_timestamp ) as
month_name ,
count(*) as Total_order
from `Business_case_Study.Orders`
group by year_, month_name, extract(month from
order_purchase_timestamp)
order by year_, extract(month from order_purchase_timestamp)


Or

Select format_date("%Y-%B",order_purchase_timestamp ) as
Year_month,
count(*) as Total_order
from `Business_case_Study.Orders`
```

Target - Business Case Study

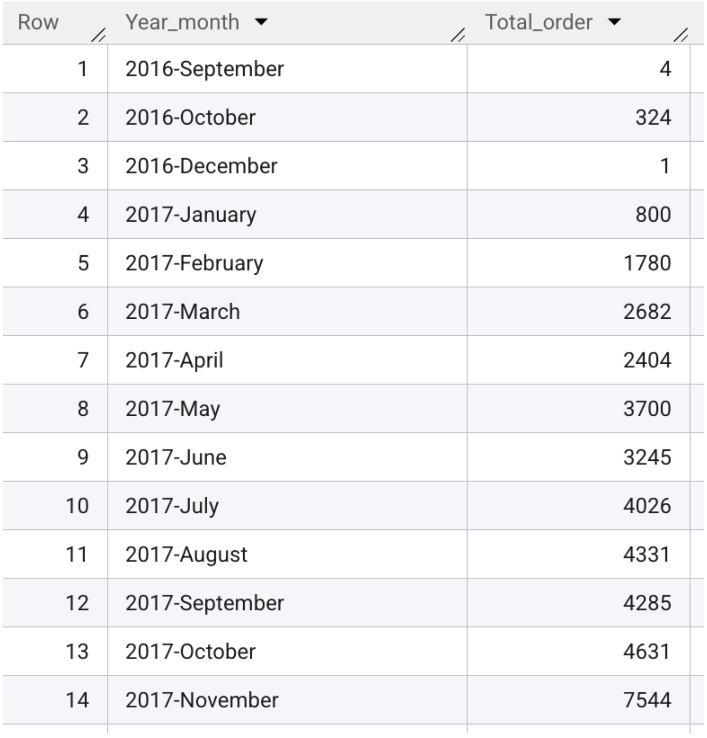
```
GROUP BY year_month, extract(month from
order_purchase_timestamp),extract(year from
order_purchase_timestamp)
order by extract(year from order_purchase_timestamp),
extract(month from order_purchase_timestamp)
```

Output:

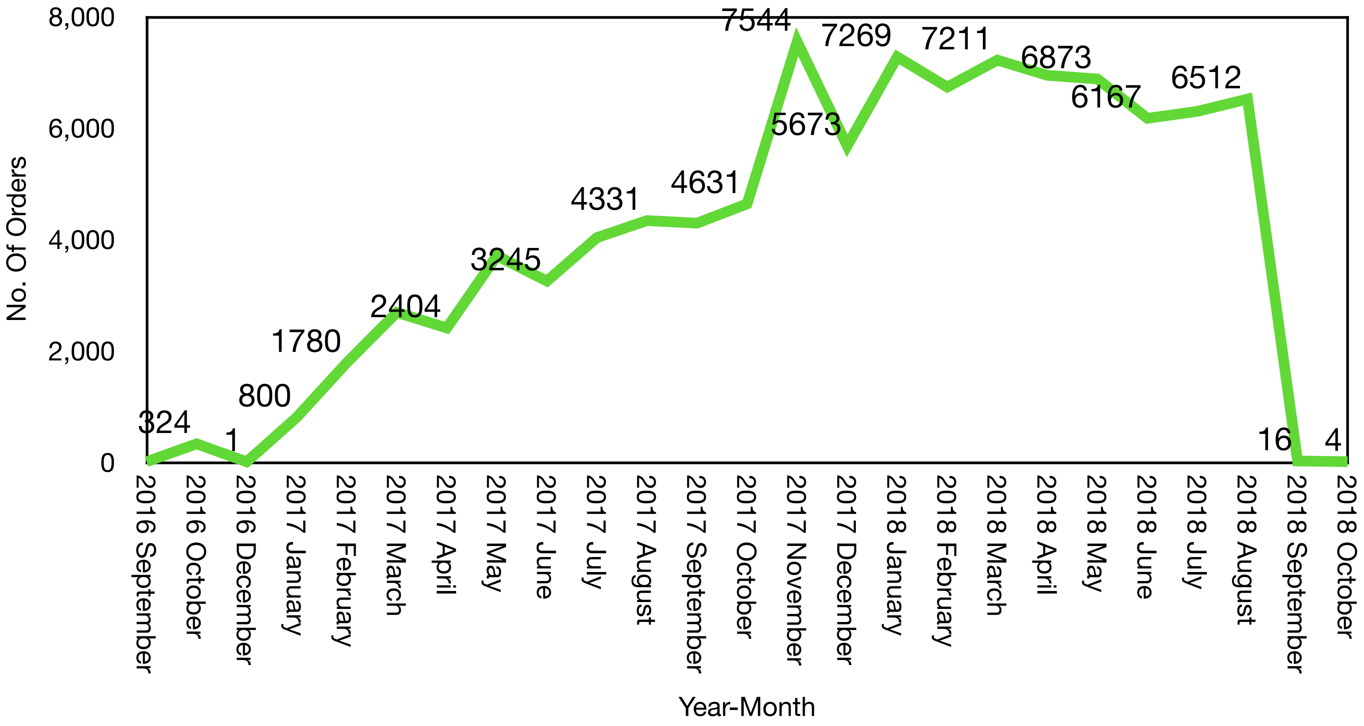


Row	Year_	month_name	Total_order
1	2016	September	4
2	2016	October	324
3	2016	December	1
4	2017	January	800
5	2017	February	1780
6	2017	March	2682
7	2017	April	2404
8	2017	May	3700
9	2017	June	3245
10	2017	July	4026
11	2017	August	4331
12	2017	September	4285
13	2017	October	4631
14	2017	November	7544
15	2017	December	5673

Or



Row	Year_month	Total_order
1	2016-September	4
2	2016-October	324
3	2016-December	1
4	2017-January	800
5	2017-February	1780
6	2017-March	2682
7	2017-April	2404
8	2017-May	3700
9	2017-June	3245
10	2017-July	4026
11	2017-August	4331
12	2017-September	4285
13	2017-October	4631
14	2017-November	7544

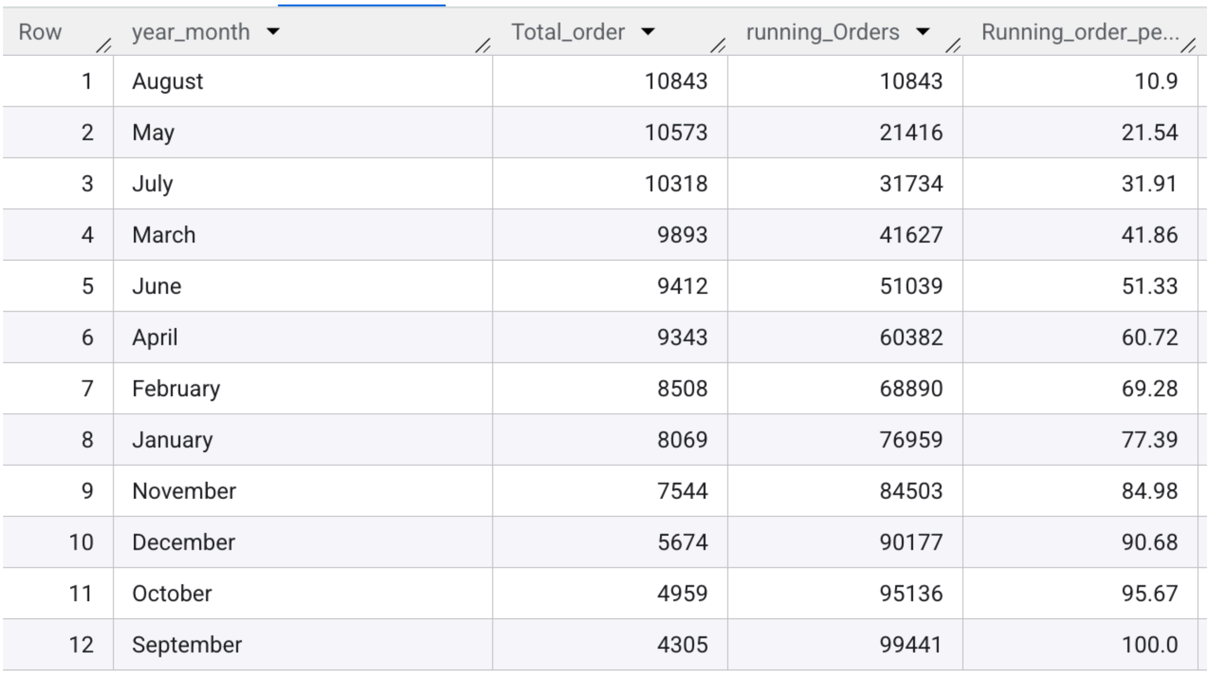


2.Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

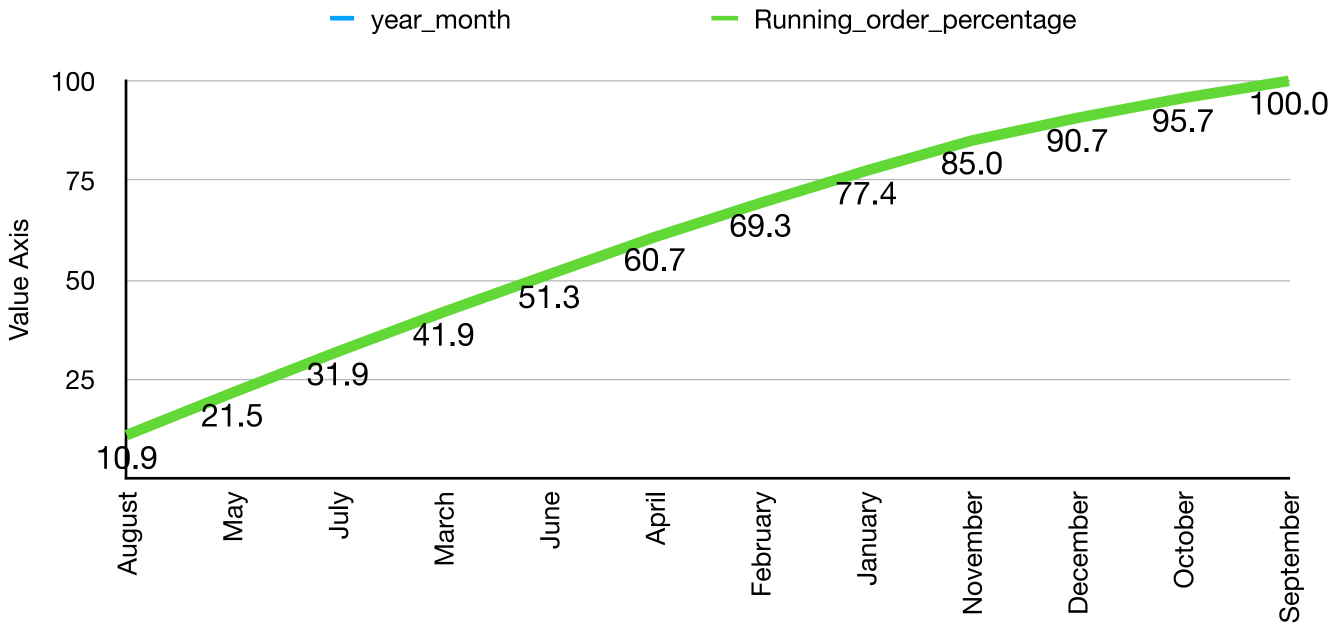
Query:

```
With Monthly_orders as (
Select format_date("%B",order_purchase_timestamp ) as
Year_month,
count(*) as Total_order
from `Business_case_Study.Orders`
GROUP BY year_month
order by total_order desc)

Select year_month, monthly_orders.Total_order,
Sum(monthly_orders.Total_order)
over(order by monthly_orders.Total_order desc) Running_Orders,
Round(Sum(monthly_orders.Total_order)
over(order by monthly_orders.Total_order desc)/(select
sum(monthly_orders.Total_order) from monthly_orders)*100,2)
Running_order_percentage,
from monthly_orders
order by monthly_orders.Total_order desc
```

Output:



Row	year_month	Total_order	running_Orders	Running_order_pe...
1	August	10843	10843	10.9
2	May	10573	21416	21.54
3	July	10318	31734	31.91
4	March	9893	41627	41.86
5	June	9412	51039	51.33
6	April	9343	60382	60.72
7	February	8508	68890	69.28
8	January	8069	76959	77.39
9	November	7544	84503	84.98
10	December	5674	90177	90.68
11	October	4959	95136	95.67
12	September	4305	99441	100.0



Target - Business Case Study

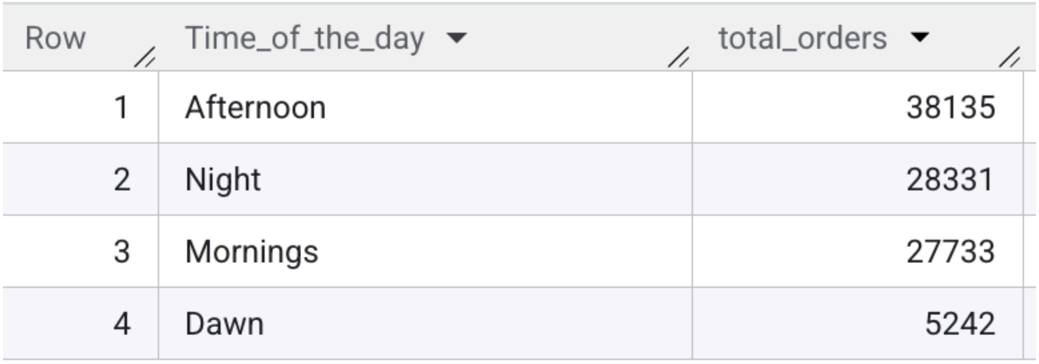
Note: Clearly, 80% of the Orders fall in the month of January-August bracket.
Later months September,October, November and December have the lowest impact as compared to other months

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
0-6 hours: Dawn
7-12 hours: Mornings
13-18 hours: Afternoon
19-23 hours: Night

Query:

```
Select case
when extract(hour from order_purchase_timestamp)>=0 and
extract(hour from order_purchase_timestamp)<=6 then "Dawn"
when extract(hour from order_purchase_timestamp)>6 and
extract(hour from order_purchase_timestamp)<=12 then
"Mornings"
when extract(hour from order_purchase_timestamp)>12 and
extract(hour from order_purchase_timestamp)<=18 then
"Afternoon"
when extract(hour from order_purchase_timestamp)>18 and
extract(hour from order_purchase_timestamp)<=23 then "Night"
end as Time_of_the_day,
count(order_purchase_timestamp) total_orders
 from `Business_case_Study.Orders`
 group by Time_of_the_day
 order by total_orders desc
```

Output:



Row	Time_of_the_day	total_orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Target - Business Case Study

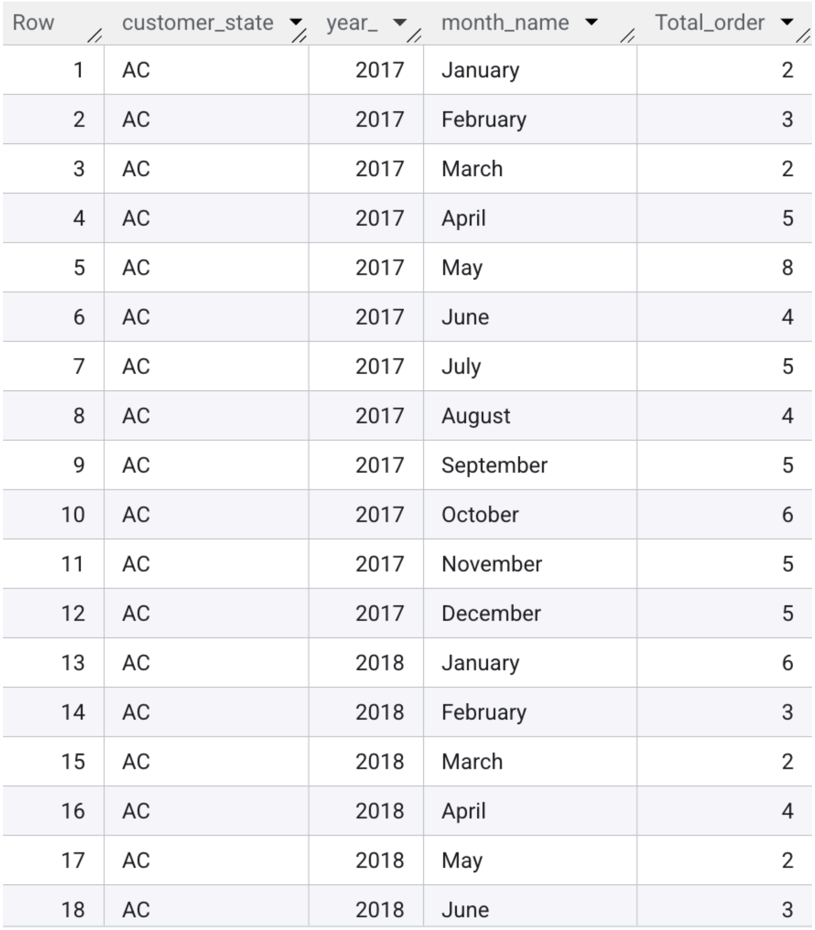
3. Evolution of E-commerce orders in the Brazil region:

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

Query:

```
Select customer_state,
extract(Year from order_purchase_timestamp) year_,
format_datetime("%B", order_purchase_timestamp) as month_name,
count(order_id) as Total_order
 from `Business_case_Study.Orders` as o
left join `Business_case_Study.Customers` as cc
on o.customer_id = cc.customer_id
group by year_,customer_state,month_name,extract(Month from
order_purchase_timestamp)
order by customer_state,year_,extract(Month from
order_purchase_timestamp)
```

Output:



Row	customer_state	year_	month_name	Total_order
1	AC	2017	January	2
2	AC	2017	February	3
3	AC	2017	March	2
4	AC	2017	April	5
5	AC	2017	May	8
6	AC	2017	June	4
7	AC	2017	July	5
8	AC	2017	August	4
9	AC	2017	September	5
10	AC	2017	October	6
11	AC	2017	November	5
12	AC	2017	December	5
13	AC	2018	January	6
14	AC	2018	February	3
15	AC	2018	March	2
16	AC	2018	April	4
17	AC	2018	May	2
18	AC	2018	June	3

Target - Business Case Study

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

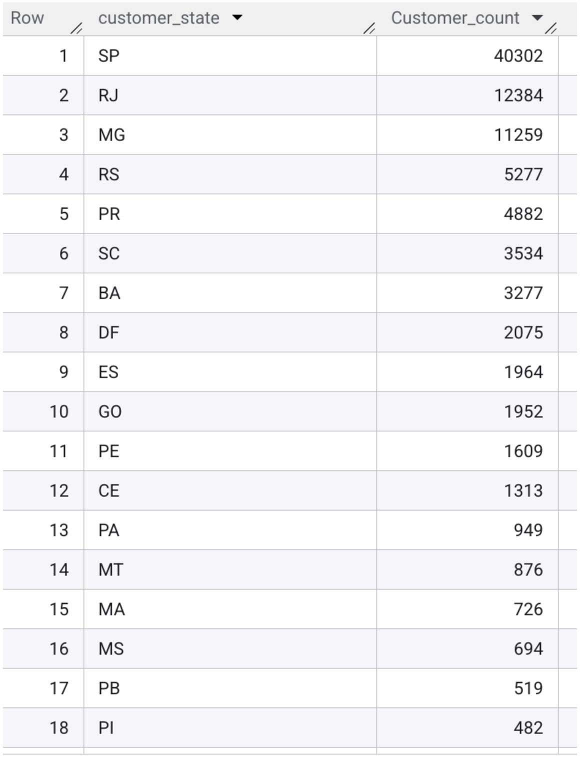
Query:

```
Select customer_state ,count(distinct customer_unique_id)
Customer_count
from `Business_case_Study.Customers`
group by customer_state
order by customer_count desc
```

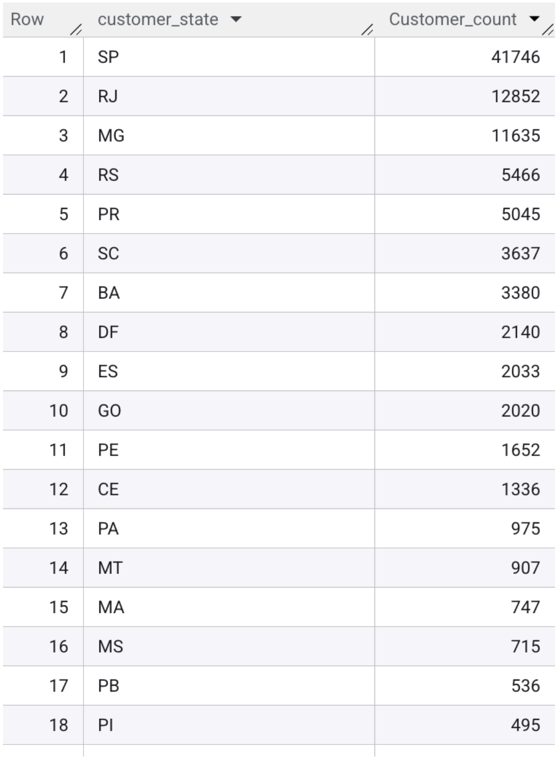
OR

```
Select customer_state ,count(distinct customer_id)
Customer_count
from `Business_case_Study.Customers`
group by customer_state
order by customer_count desc
```

Output:



Row	customer_state	Customer_count
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
11	PE	1609
12	CE	1313
13	PA	949
14	MT	876
15	MA	726
16	MS	694
17	PB	519
18	PI	482



Row	customer_state	Customer_count
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

Target - Business Case Study

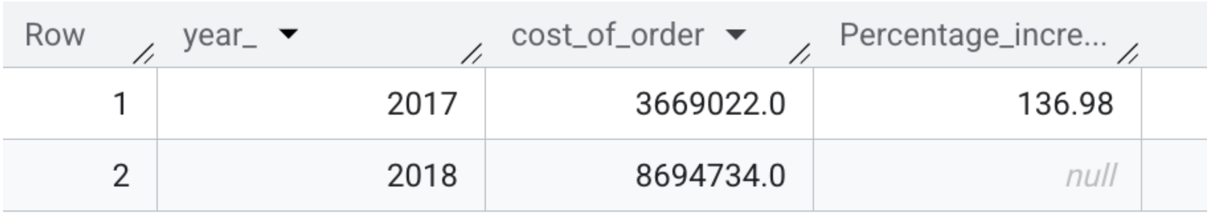
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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.

Query:

```
Select year_,cost_of_order,
Round(((lead(cost_of_order,1) over(order by cost_of_order)-
cost_of_order)/cost_of_order)*100,2)Percentage_increase
from (
Select extract(year from order_purchase_timestamp) year_,
Round(sum(pp.payment_value),0) as cost_of_order
from `Business_case_Study.Orders` as OO
left join `Business_case_Study.Payments` as pp
on OO.order_id = pp.order_id
where extract(month from order_purchase_timestamp) in
(1,2,3,4,5,6,7,8)
group by year_
) as z
order by z.year_
```

Output:



Row	year_	cost_of_order	Percentage_incre...
1	2017	3669022.0	136.98
2	2018	8694734.0	null

Insight:

137% increase in the cost of order from 2017 to 2018.

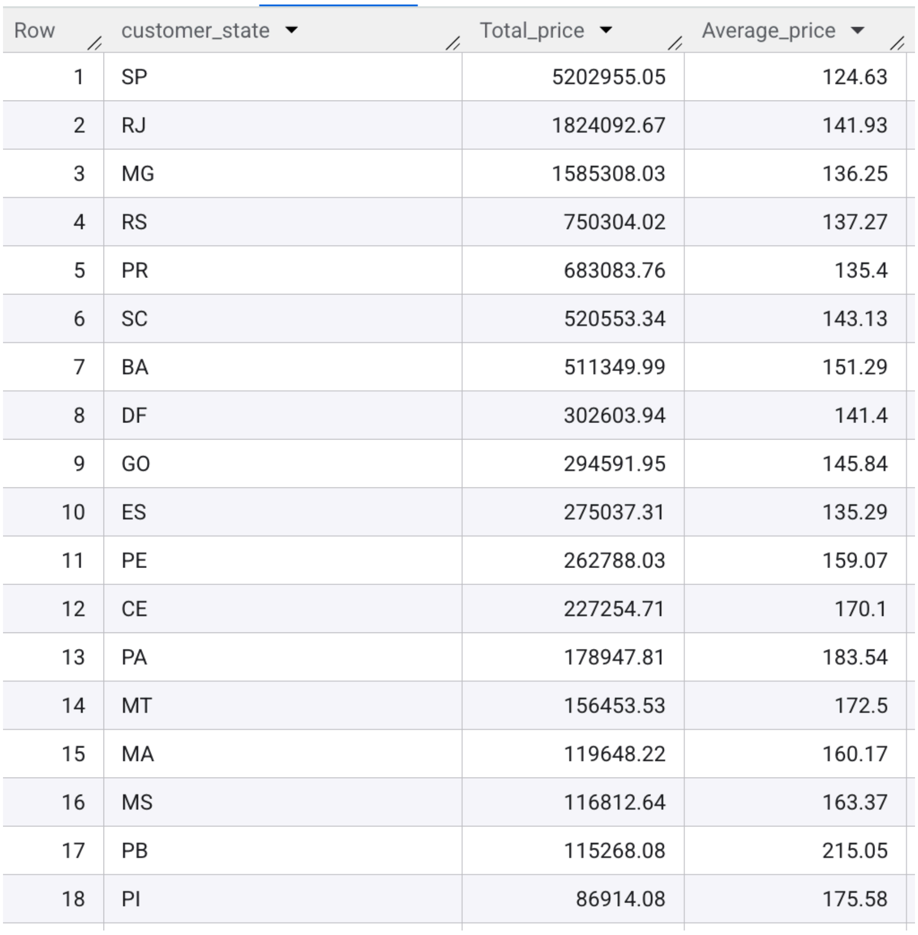
Target - Business Case Study

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

Query:

```
Select  cc.customer_state , Round(sum(price),2) Total_price ,
Round(sum(price)/count(distinct oo.order_id),2)as
Average_price
 from `Business_case_Study.Orders` as oo
left join `Business_case_Study.Customers` as cc


on oo.customer_id = cc.customer_id
left join `Business_case_Study.Order_items` as oi
on oo.order_id = oi.order_id
group by cc.customer_state
order by Total_price desc
```

Output:



Row	customer_state	Total_price	Average_price
1	SP	5202955.05	124.63
2	RJ	1824092.67	141.93
3	MG	1585308.03	136.25
4	RS	750304.02	137.27
5	PR	683083.76	135.4
6	SC	520553.34	143.13
7	BA	511349.99	151.29
8	DF	302603.94	141.4
9	GO	294591.95	145.84
10	ES	275037.31	135.29
11	PE	262788.03	159.07
12	CE	227254.71	170.1
13	PA	178947.81	183.54
14	MT	156453.53	172.5
15	MA	119648.22	160.17
16	MS	116812.64	163.37
17	PB	115268.08	215.05
18	PI	86914.08	175.58

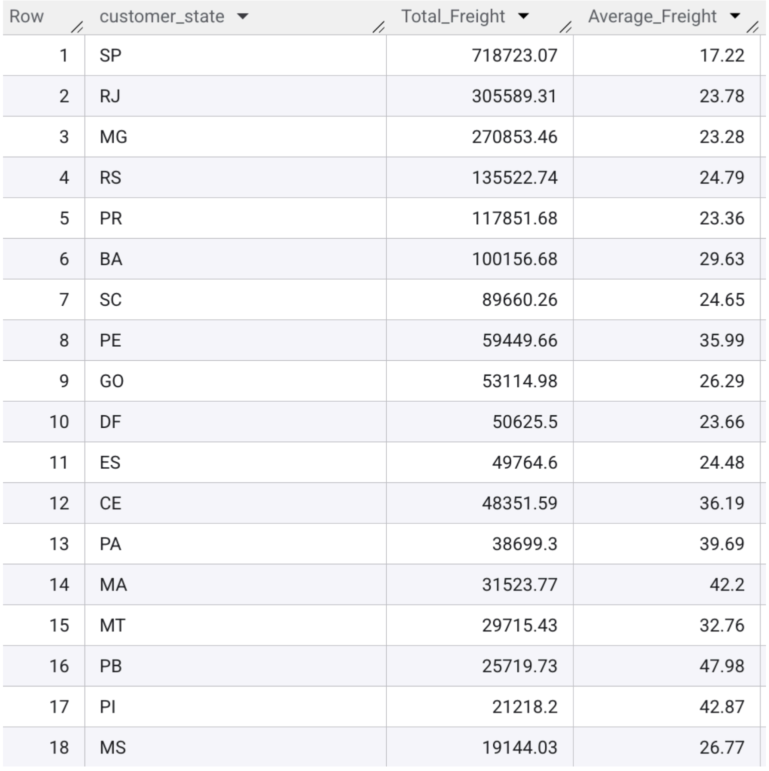
Target - Business Case Study

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

Query:

```
Select  cc.customer_state , Round(sum(freight_value),2)
Total_Freight ,
Round(sum(freight_value)/count(distinct oo.order_id),2)as
Average_Freight

from `Business_case_Study.Orders` as oo
left join `Business_case_Study.Customers` as cc
on oo.customer_id = cc.customer_id
left join `Business_case_Study.Order_items` as oi
on oo.order_id = oi.order_id
group by cc.customer_state
order by Total_Freight desc
```

Output:



Row	customer_state	Total_Freight	Average_Freight
1	SP	718723.07	17.22
2	RJ	305589.31	23.78
3	MG	270853.46	23.28
4	RS	135522.74	24.79
5	PR	117851.68	23.36
6	BA	100156.68	29.63
7	SC	89660.26	24.65
8	PE	59449.66	35.99
9	GO	53114.98	26.29
10	DF	50625.5	23.66
11	ES	49764.6	24.48
12	CE	48351.59	36.19
13	PA	38699.3	39.69
14	MA	31523.77	42.2
15	MT	29715.43	32.76
16	PB	25719.73	47.98
17	PI	21218.2	42.87
18	MS	19144.03	26.77

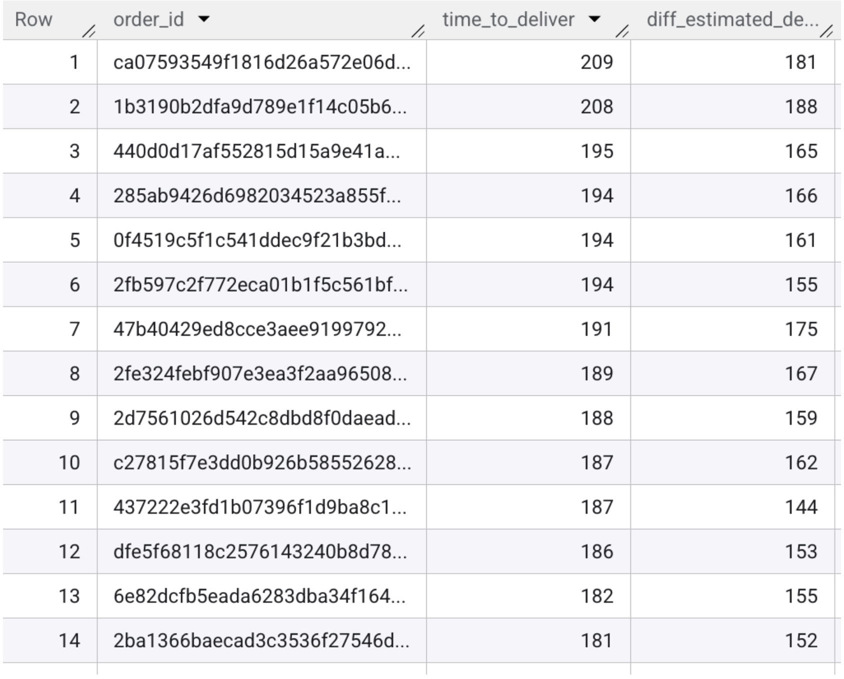
5. Analysis based on sales, freight and delivery time

Target - Business Case Study

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_delivered_customer_date - order_estimated_delivery_date

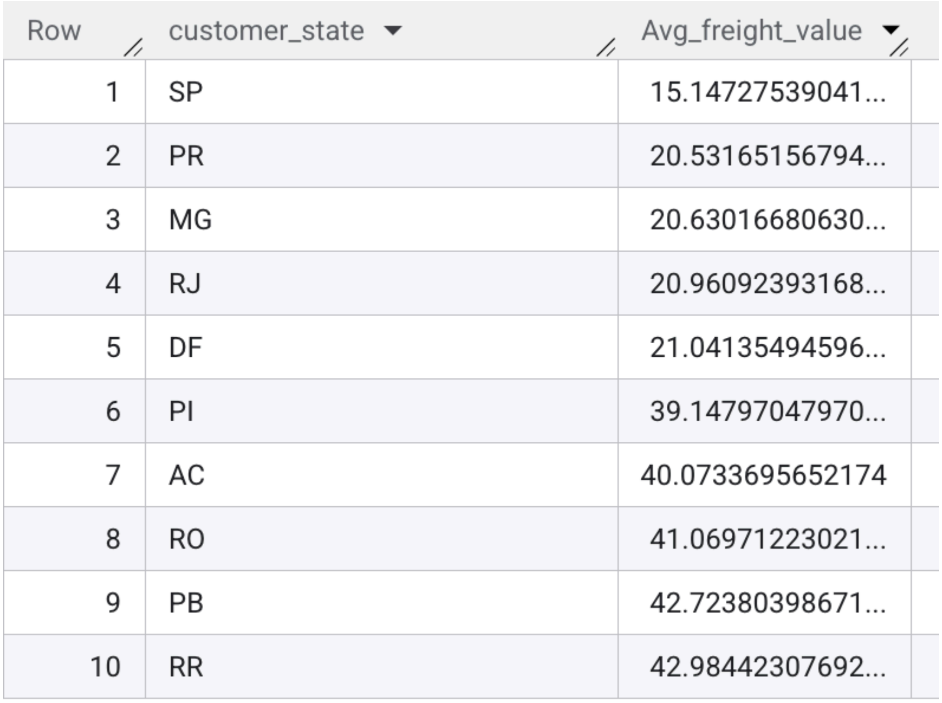
```
Select order_id, date_diff(order_delivered_customer_date,
order_purchase_timestamp, Day)  as time_to_deliver,
date_diff(order_delivered_customer_date,
order_estimated_delivery_date, Day) as diff_estimated_delivery
from `Business_case_Study.Orders`
order by time_to_deliver desc
```



Row	order_id	time_to_deliver	diff_estimated_de...
1	ca07593549f1816d26a572e06d...	209	181
2	1b3190b2dfa9d789e1f14c05b6...	208	188
3	440d0d17af552815d15a9e41a...	195	165
4	285ab9426d6982034523a855f...	194	166
5	0f4519c5f1c541ddec9f21b3bd...	194	161
6	2fb597c2f772eca01b1f5c561bf...	194	155
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324febf907e3ea3f2aa96508...	189	167
9	2d7561026d542c8dbd8f0daead...	188	159
10	c27815f7e3dd0b926b58552628...	187	162
11	437222e3fd1b07396f1d9ba8c1...	187	144
12	dfe5f68118c2576143240b8d78...	186	153
13	6e82dcfb5eada6283dba34f164...	182	155
14	2ba1366baecad3c3536f27546d...	181	152

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

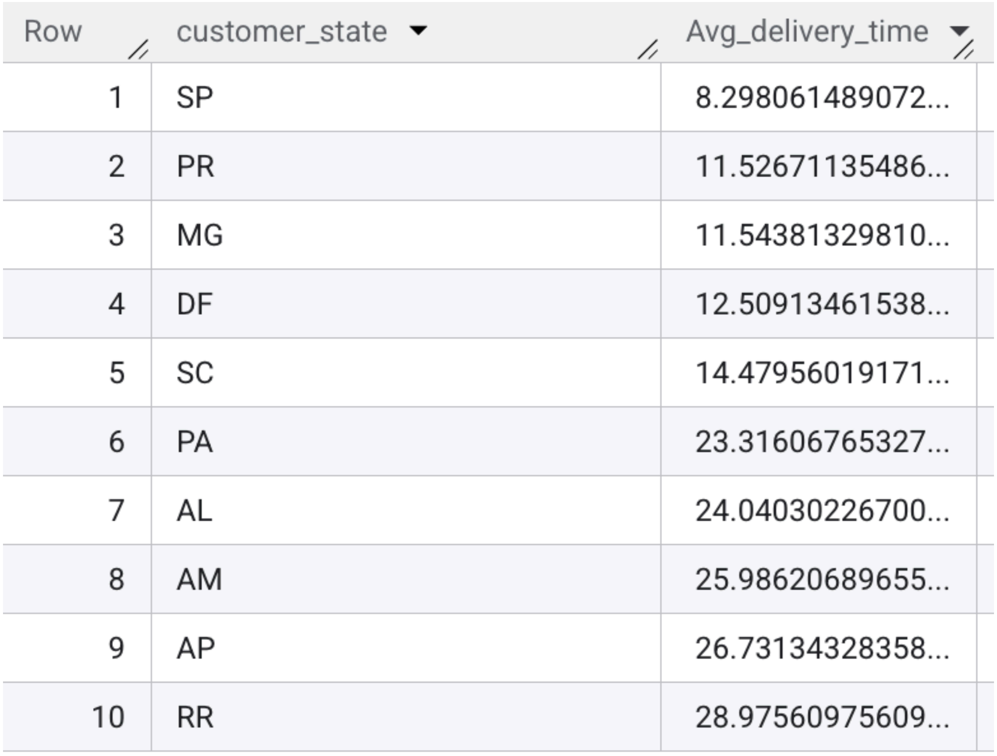
```
with Avg_value as (
Select customer_state, Avg(freight_value) as Avg_freight_value
from `Business_case_Study.Orders` as o
left join `Business_case_Study.Order_items` as oi
on o.order_id = oi.order_id

left join `Business_case_Study.Customers` as cc
on o.customer_id = cc.customer_id
group by customer_state),

Ranked_states as (

select customer_state, Avg_freight_value,
row_number() over(order by Avg_freight_value desc) as
desc_rank,
row_number() over(order by Avg_freight_value asc) as asc_rank
from Avg_value
)

Select customer_state, Avg_freight_value from Ranked_states
where desc_rank <=5 or asc_rank <=5
ORDER BY avg_freight_value asc;
```



Row	customer_state	Avg_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...
6	PI	39.14797047970...
7	AC	40.0733695652174
8	RO	41.06971223021...
9	PB	42.72380398671...
10	RR	42.98442307692...

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

```
With Avg_value as (
Select customer_state,
Avg(date_diff(order_delivered_customer_date,
order_purchase_timestamp,Day)) as Avg_delivery_time
from `Business_case_Study.Orders` as o
left join `Business_case_Study.Customers` as cc
on o.customer_id = cc.customer_id
group by customer_state),

ranked_states as (
select customer_state, Avg_delivery_time,
row_number() over(order by Avg_delivery_time desc) as
desc_rank,
row_number() over(order by Avg_delivery_time asc) as asc_rank
from Avg_value
)

Select customer_state, Avg_delivery_time from ranked_states
where desc_rank <=5 or asc_rank <=5
ORDER BY Avg_delivery_time
```



Row	customer_state	Avg_delivery_time
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...
4	DF	12.50913461538...
5	SC	14.47956019171...
6	PA	23.31606765327...
7	AL	24.04030226700...
8	AM	25.98620689655...
9	AP	26.73134328358...
10	RR	28.97560975609...

\

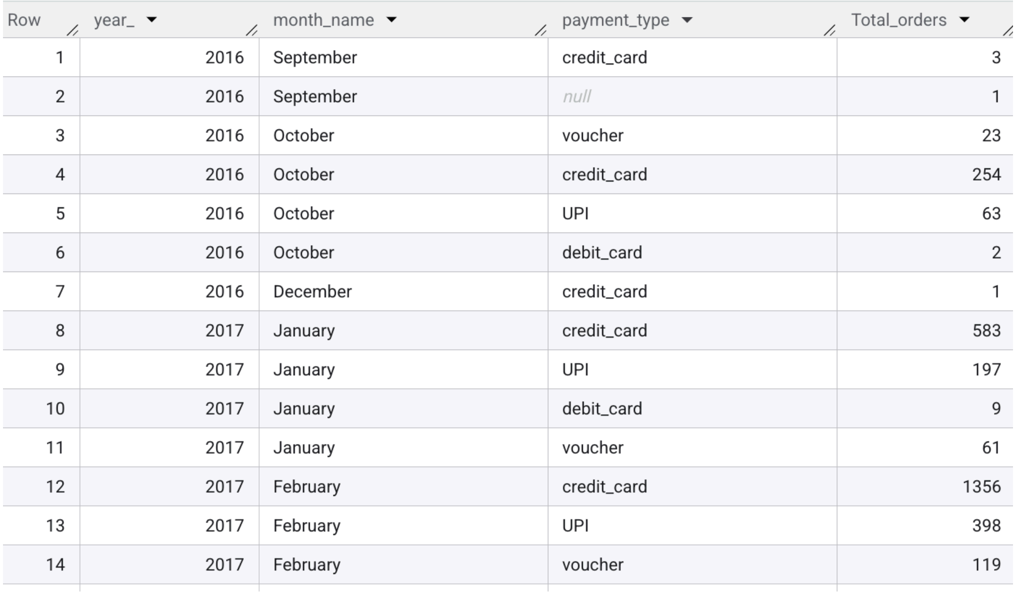
4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery. You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
Select customer_state,
Avg(date_diff(order_delivered_customer_date,order_purchase_tim
estamp, DAY))-
Avg(date_diff(order_delivered_customer_date,order_estimated_de
livery_date, DAY))
from `Business_case_Study.Orders` as oo
left join `Business_case_Study.Customers` as cc
on oo.customer_id = cc.customer_id
where order_delivered_customer_date is not null
group by customer_state
```

6. Analysis based on the payments

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

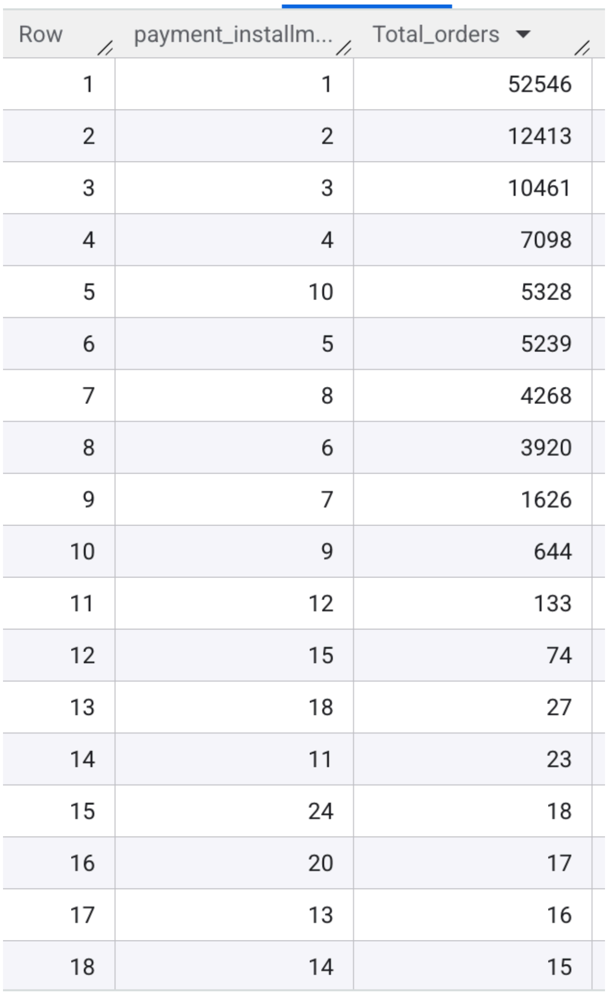
```
Select extract(Year from order_purchase_timestamp) year_,
format_datetime("%B",order_purchase_timestamp ) as month_name,
payment_type, count(*) as Total_orders
from `Business_case_Study.Orders` as OO
left join `Business_case_Study.Payments` as p
on oo.order_id = p.order_id
group by year_, month_name, payment_type, extract(month from
order_purchase_timestamp)
order by year_, extract(month from order_purchase_timestamp)
```



Row	year_	month_name	payment_type	Total_orders
1	2016	September	credit_card	3
2	2016	September	null	1
3	2016	October	voucher	23
4	2016	October	credit_card	254
5	2016	October	UPI	63
6	2016	October	debit_card	2
7	2016	December	credit_card	1
8	2017	January	credit_card	583
9	2017	January	UPI	197
10	2017	January	debit_card	9
11	2017	January	voucher	61
12	2017	February	credit_card	1356
13	2017	February	UPI	398
14	2017	February	voucher	119

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

```
Select payment_installments, Count(*) as Total_orders
from `Business_case_Study.Orders` as OO
left join `Business_case_Study.Payments` as pp
on oo.order_id = pp.order_id
where payment_installments is not null and
payment_installments <> 0
group by payment_installments
order by Total_orders desc
```



Row	payment_installm...	Total_orders
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
18	14	15