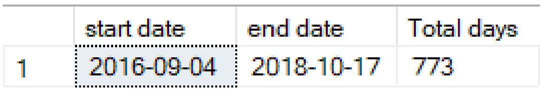
Business Case: Target SQL

Q1.2 Time period for which the data is given

```
Select
        min(cast(order_purchase_timestamp as date)) [start date],
        max(cast(order_purchase_timestamp as date)) [end date],
        datediff(day,min(order_purchase_timestamp),
        max(order_purchase_timestamp)) [Total days]
From
         [Target SQL Project].dbo.[orders (1)]
```

Query Result:



	start date	end date	Total days
1	2016-09-04	2018-10-17	773

Analysis :

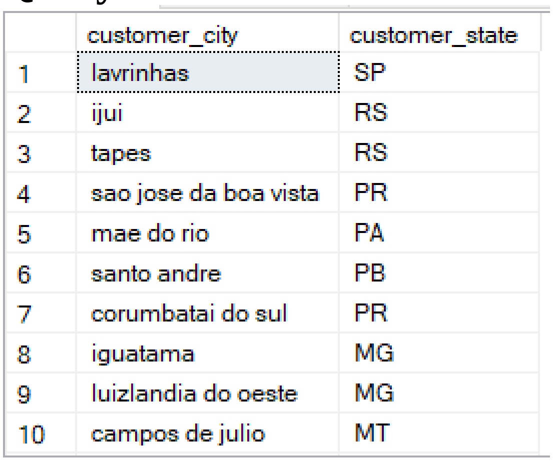
The given dataset includes e-commerce orders data from September 4, 2016 to September 17, 2018, which spans a total of 773 days. This can be useful for understanding the trends and patterns in the data over time.

Q1.3 Cities and States of customers ordered during the given period

```
Select
        distinct c.customer_city,
        c.customer_state
From
         [Target SQL Project].dbo.[customers (1)] as c
Join
```

```
         [Target SQL Project].dbo.[orders (1)] as o
On
        o.customer_id = c.customer_id
```

Query Result:



	customer_city	customer_state
1	lavrinhas	SP
2	ijui	RS
3	tapes	RS
4	sao jose da boa vista	PR
5	mae do rio	PA
6	santo andre	PB
7	corumbatai do sul	PR
8	iguatama	MG
9	luizlandia do oeste	MG
10	campos de julio	MT

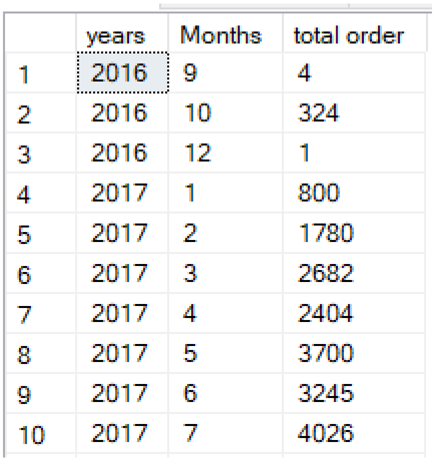
Analysis:

These are the top 10 area where people ordered the most during given time period.

Question 2.1 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?

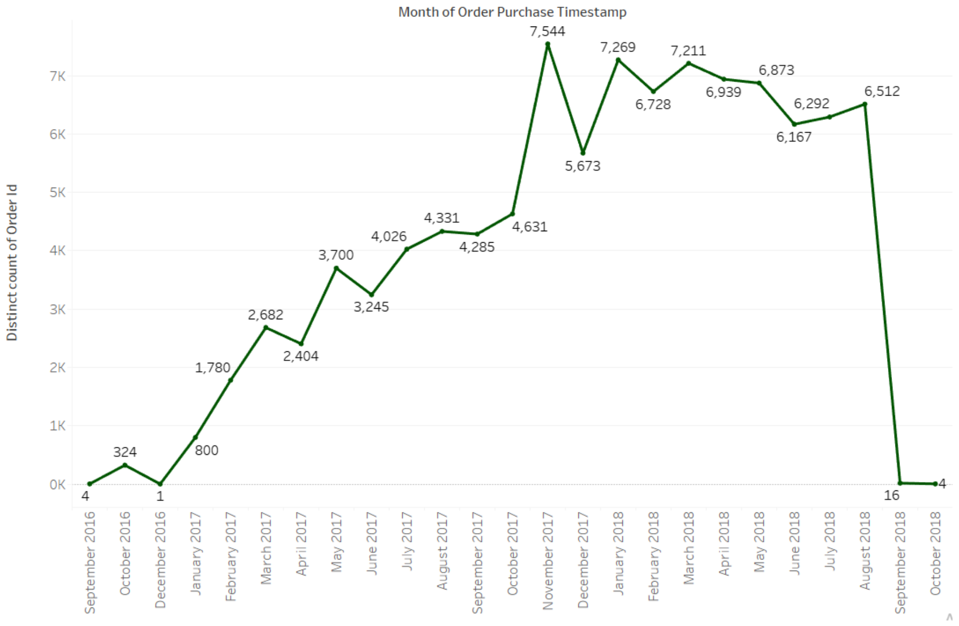
```
Select
        years,
        Months,
        count(*) [total order]
From
        (select
        YEAR(order_purchase_timestamp) as years,
        MONTH(order_purchase_timestamp) as Months
        from [Target SQL Project].dbo.[orders (1)]) as tb1
Group by
        years, Months
Order by
        years, Months
```

Query Result:



	years	Months	total order
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

Visualization on the base of result:



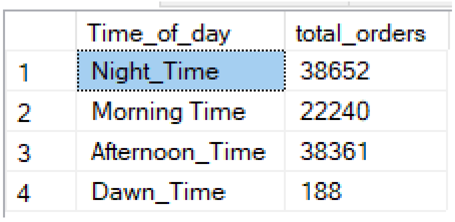
Insights: The data provided indicate a positive yearly sales trend from 2016 to 2018 then we see a sudden dip down from September 2018 onwards apart of that there is a dip from September to December (am fail to understand how this sudden dip happen here, may the data is missing or something went wrong) and a rise from march to august, it suggests that the peak season for sales is during these months.

<u>Recommendation: </u>To increase sales, focus on promoting products during the September to December period and continue to capitalize on the high sales trends from March to August.

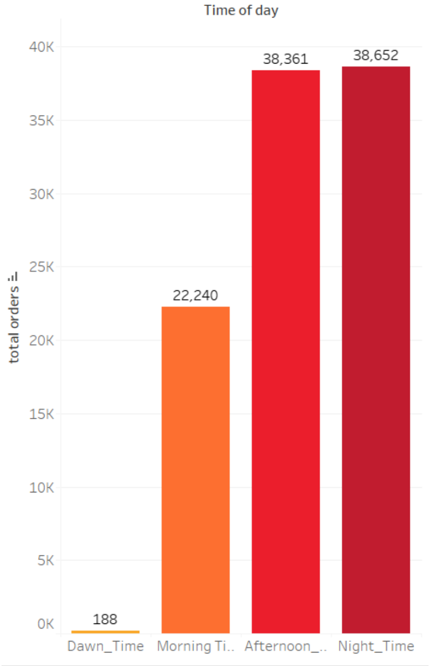
Question 2.2 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

```
select
        Time_of_day,
        COUNT(*) as total_orders
From
         (select order_purchase_timestamp,
Case
         when DATEPART(HOUR, order_purchase_timestamp)= 5 then 'Dawn_Time'
        when DATEPART(HOUR, order_purchase_timestamp) between 6 and 11 then 'Morning Time'
        when DATEPART(HOUR, order_purchase_timestamp) between 12 and 17 then 'Afternoon_Time'
        when DATEPART(HOUR, order_purchase_timestamp) between 18 and 23 then 'Night_Time'
        else 'Night_Time' end as Time_of_day
From
         [Target SQL Project].dbo. [orders (1)]) as table_1
group by
         Time_of_day
```

Query Result:



	Time_of_day	total_orders
1	Night_Time	38652
2	Morning Time	22240
3	Afternoon_Time	38361
4	Dawn_Time	188



Insights: As, result indicate more sales are made in the afternoon and night, dawn has the least sales.
Recommendation: by offering discounts for dawn timings to increase sales.

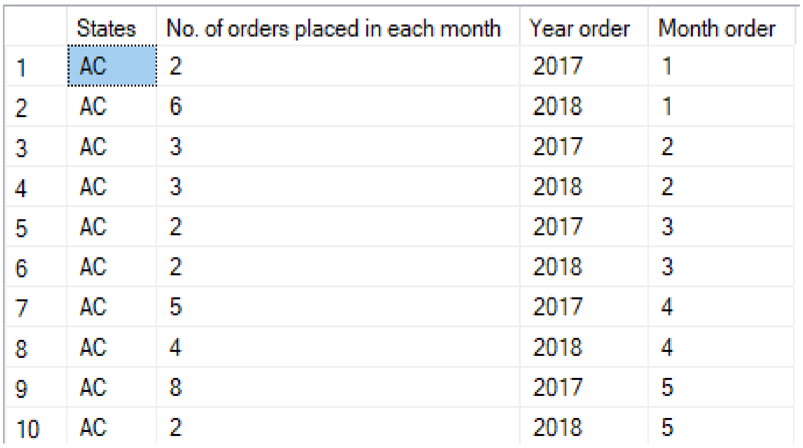
Question 3. Evolution of E-commerce orders in the Brazil region:

3.1 Get month on month orders by states

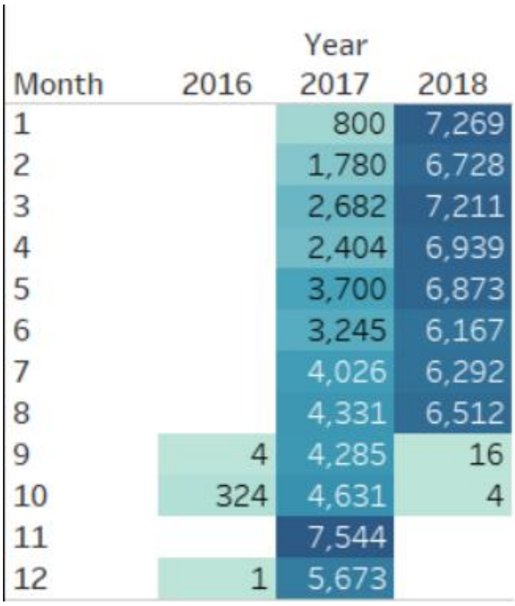
```
Select
       c.customer_state as States,
       count(*) as [No. of orders placed in each month],
       datepart(year, o.order_purchase_timestamp) as [Year order],
       datepart(Month, o.order_purchase_timestamp) as [Month order]
From
```

```
         [Target SQL Project].dbo.[orders (1)] as o
Join
        [Target SQL Project].dbo.[customers (1)] as c
On
         c.customer_id = o.customer_id
Group by
        datepart(year, o.order_purchase_timestamp),
        datepart(Month, o.order_purchase_timestamp),
        c.customer_state
Order by
        States, [Month order], [Year order], [No. of orders placed in each month]
```

Query Result:



	States	No. of orders placed in each month	Year order	Month order
1	AC	2	2017	1
2	AC	6	2018	1
3	AC	3	2017	2
4	AC	3	2018	2
5	AC	2	2017	3
6	AC	2	2018	3
7	AC	5	2017	4
8	AC	4	2018	4
9	AC	8	2017	5
10	AC	2	2018	5

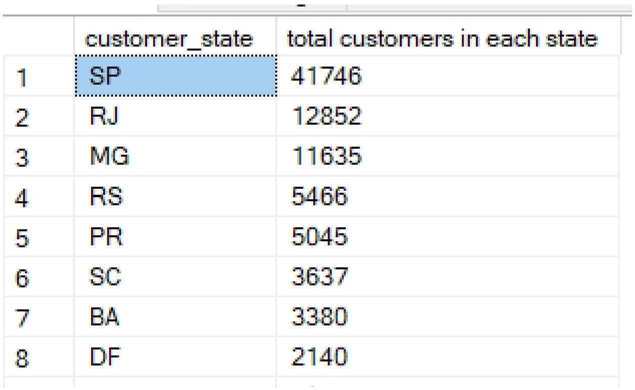


		Year	
Month	2016	2017	2018
1		800	7,269
2		1,780	6,728
3		2,682	7,211
4		2,404	6,939
5		3,700	6,873
6		3,245	6,167
7		4,026	6,292
8		4,331	6,512
9	4	4,285	16
10	324	4,631	4
11		7,544	
12	1	5,673	

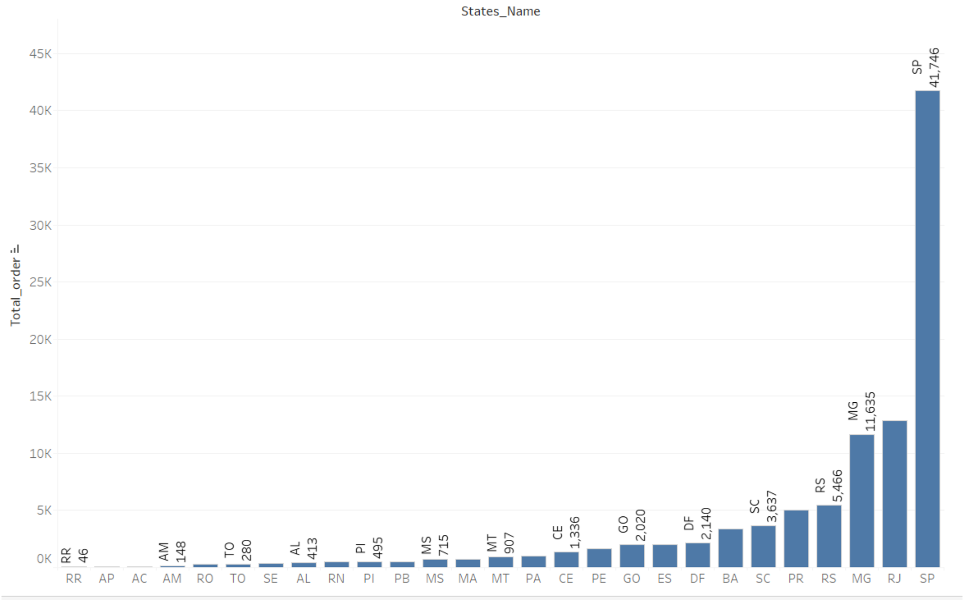
3.2 Distribution of customers across the states in Brazil

```
Select
        customer_state,
        COUNT(*) as [total customers in each state]
from
        [Target SQL Project].dbo.[customers (1)]
Group by
        customer_state
Order by
[total customers in each state] desc
```

Query Result:



	customer_state	total customers in each state
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140



Insights: Here,We can see the Evolution of E-commerce orders in the Brazil region Sao Paulo has consistently been the region with highest number of orders and Customers. There is an overall increasing trend in the number of orders across all regions from 2016 to 2018 with some fluctuations in certain months.

Recommendation: Based on this, can be made focus on expanding operations in these high-performing states to increase overall sale.

Company can also focus in low ordered numbers to increase sales and improve customer satisfaction.

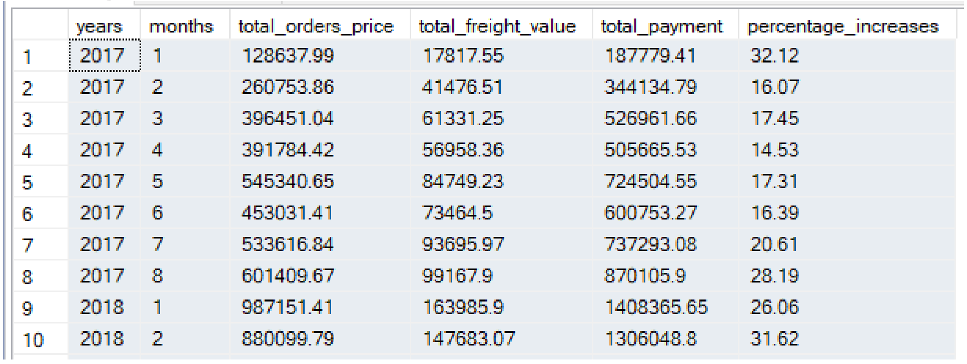
Question 4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

4.1 Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table

```
SELECT
	YEAR(o.order_purchase_timestamp) as years,
	MONTH(o.order_purchase_timestamp) as months,
	round(SUM(ot.price),2)  as total_orders_price,
	round(SUM(ot.freight_value),2) as total_freight_value,
	round(SUM(p.payment_value),2) total_payment,
	round((SUM(p.payment_value) - SUM(ot.price) - SUM(ot.freight_value) )
	/
	SUM(ot.price) * 100, 2) as percentage_increases
FROM
	 [Target SQL Project].dbo.[orders (1)] as o
Join
	 [Target SQL Project].dbo.payments as p
on
	p.order_id = o.order_id
Join
	 [Target SQL Project].dbo.order_items as ot
on
	ot.order_id = p.order_id
WHERE
	YEAR(o.order_purchase_timestamp) between 2017 and 2018 and
```

```
    MONTH(o.order_purchase_timestamp) between 1 and 8
GROUP BY
    YEAR(o.order_purchase_timestamp),
    MONTH(o.order_purchase_timestamp)
ORDER BY
    YEAR(o.order_purchase_timestamp),
    MONTH(o.order_purchase_timestamp)
```

Query Result:



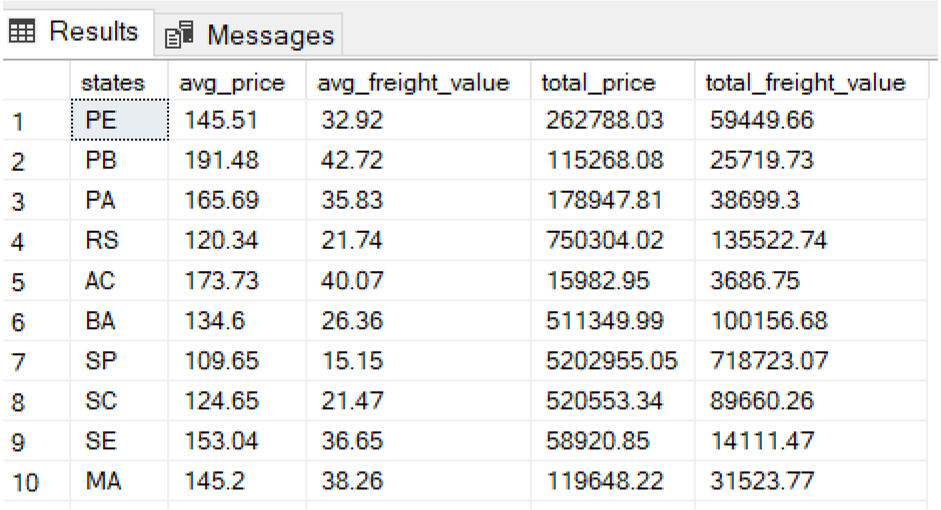
	years	months	total_orders_price	total_freight_value	total_payment	percentage_increases
1	2017	1	128637.99	17817.55	187779.41	32.12
2	2017	2	260753.86	41476.51	344134.79	16.07
3	2017	3	396451.04	61331.25	526961.66	17.45
4	2017	4	391784.42	56958.36	505665.53	14.53
5	2017	5	545340.65	84749.23	724504.55	17.31
6	2017	6	453031.41	73464.5	600753.27	16.39
7	2017	7	533616.84	93695.97	737293.08	20.61
8	2017	8	601409.67	99167.9	870105.9	28.19
9	2018	1	987151.41	163985.9	1408365.65	26.06
10	2018	2	880099.79	147683.07	1306048.8	31.62

4.2 Mean & Sum of price and freight value by customer state

```
SELECT
    c.customer_state as states,
    round(AVG(ot.price),2) as avg_price,
    round(AVG(ot.freight_value),2) as avg_freight_value,
    round(SUM(ot.price),2) as total_price,
    round(SUM(ot.freight_value),2) as total_freight_value
FROM
     [Target SQL Project].dbo.order_items as ot
join
    [Target SQL Project].dbo.[orders (1)] as o
on
    o.order_id = ot.order_id
join
    [Target SQL Project].dbo.[customers (1)] as c
on
    c.customer_id = o.customer_id
GROUP BY
```

c.customer_state

Query Result:



	states	avg_price	avg_freight_value	total_price	total_freight_value
1	PE	145.51	32.92	262788.03	59449.66
2	PB	191.48	42.72	115268.08	25719.73
3	PA	165.69	35.83	178947.81	38699.3
4	RS	120.34	21.74	750304.02	135522.74
5	AC	173.73	40.07	15982.95	3686.75
6	BA	134.6	26.36	511349.99	100156.68
7	SP	109.65	15.15	5202955.05	718723.07
8	SC	124.65	21.47	520553.34	89660.26
9	SE	153.04	36.65	58920.85	14111.47
10	MA	145.2	38.26	119648.22	31523.77

Analysis:

The analysis shows a significant increase in the the cost of orders in 2018 compared to 2017 that's indicating a positive impact on the ecommerce industry in terms of economic growth.

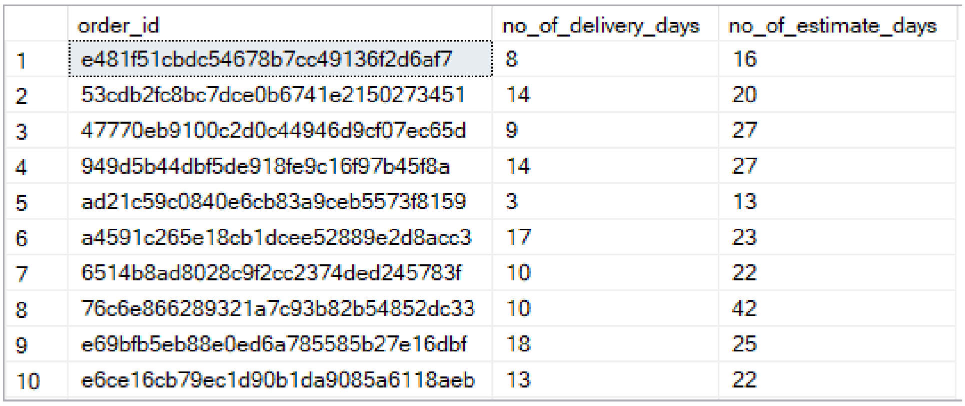
Recommendation: Based on the result of the analysis, a recommendation could be to assess the reasons for the increase in cost and see if there are any areas for optimization or cost reduction to improve profitability.

5. Analysis on sales, freight and delivery time:

5.1 Calculate days between purchasing, delivering and estimated delivery

```
select
        order_id,
        DATEDIFF(day,order_purchase_timestamp,
        order_delivered_customer_date)
        as no_of_delivery_days,
        DATEDIFF(day,order_purchase_timestamp,
        order_estimated_delivery_date)
        as no_of_estimate_days
from
        [Target SQL Project].dbo.[orders (1)]
where
        order_purchase_timestamp is not null and order_delivered_customer_date
        is not null and order_estimated_delivery_date is not null
```

Query Result:

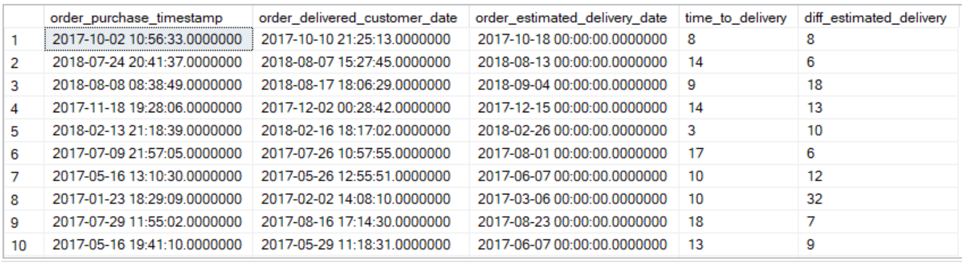


	order_id	no_of_delivery_days	no_of_estimate_days
1	e481f51cbdc54678b7cc49136f2d6af7	8	16
2	53cdb2fc8bc7dce0b6741e2150273451	14	20
3	47770eb9100c2d0c44946d9cf07ec65d	9	27
4	949d5b44dbf5de918fe9c16f97b45f8a	14	27
5	ad21c59c0840e6cb83a9ceb5573f8159	3	13
6	a4591c265e18cb1dcee52889e2d8acc3	17	23
7	6514b8ad8028c9f2cc2374ded245783f	10	22
8	76c6e866289321a7c93b82b54852dc33	10	42
9	e69bfb5eb88e0ed6a785585b27e16dbf	18	25
10	e6ce16cb79ec1d90b1da9085a6118aeb	13	22

5.2 Find time_to_delivery & diff_estimated_delivery.

```
SELECT
         order_purchase_timestamp,
         order_delivered_customer_date,
         order_estimated_delivery_date,
        DATEDIFF(day,order_purchase_timestamp,order_delivered_customer_dat
        e)
         AS time_to_delivery,
        DATEDIFF(day,order_delivered_customer_date,order_estimated_delivery
        _date)
        AS diff_estimated_delivery
FROM
        [Target SQL Project].dbo.[orders (1)]
where
        order_purchase_timestamp is not null
        and order_delivered_customer_date is not null
        and order_estimated_delivery_date is not null
```

Query Result:



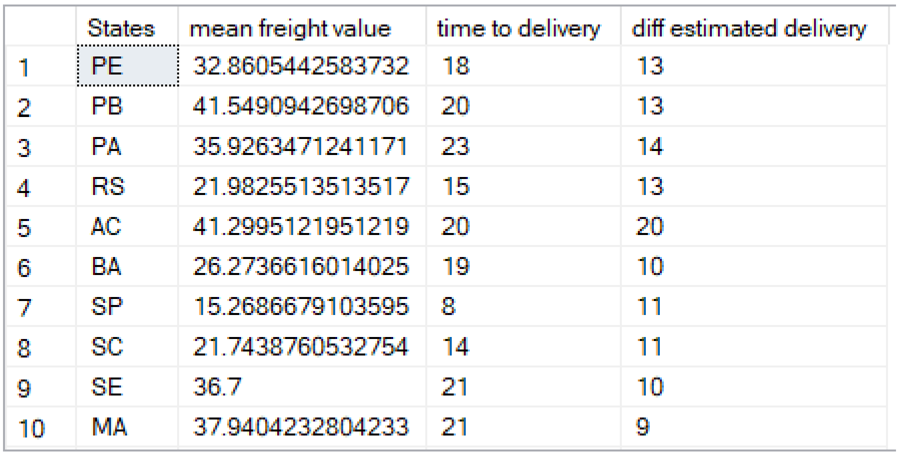
	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	diff_estimated_delivery
1	2017-10-02 10:56:33.0000000	2017-10-10 21:25:13.0000000	2017-10-18 00:00:00.0000000	8	8
2	2018-07-24 20:41:37.0000000	2018-08-07 15:27:45.0000000	2018-08-13 00:00:00.0000000	14	6
3	2018-08-08 08:38:49.0000000	2018-08-17 18:06:29.0000000	2018-09-04 00:00:00.0000000	9	18
4	2017-11-18 19:28:06.0000000	2017-12-02 00:28:42.0000000	2017-12-15 00:00:00.0000000	14	13
5	2018-02-13 21:18:39.0000000	2018-02-16 18:17:02.0000000	2018-02-26 00:00:00.0000000	3	10
6	2017-07-09 21:57:05.0000000	2017-07-26 10:57:55.0000000	2017-08-01 00:00:00.0000000	17	6
7	2017-05-16 13:10:30.0000000	2017-05-26 12:55:51.0000000	2017-06-07 00:00:00.0000000	10	12
8	2017-01-23 18:29:09.0000000	2017-02-02 14:08:10.0000000	2017-03-06 00:00:00.0000000	10	32
9	2017-07-29 11:55:02.0000000	2017-08-16 17:14:30.0000000	2017-08-23 00:00:00.0000000	18	7
10	2017-05-16 19:41:10.0000000	2017-05-29 11:18:31.0000000	2017-06-07 00:00:00.0000000	13	9

5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
create view cust as
Select
        o.customer_id as cu,
        round(avg(ot.freight_value),2) [MFV],
        avg(DATEDIFF(day,o.order_purchase_timestamp,
        o.order_delivered_customer_date)) [TTD],
        avg(DATEDIFF(day,o.order_delivered_customer_date,
        o.order_estimated_delivery_date)) [DED]
From
      [Target SQL Project].dbo.[orders (1)] as o
left join
        [Target SQL Project].dbo.order_items as ot
on ot.order_id = o.order_id
Group by
        o.customer_id,ot.freight_value,o.order_delivered_customer_date,o.order_e
        stimated_delivery_date,o.order_purchase_timestamp
Select
        c.customer_state as States,
        avg(ct.MFV)[mean freight value],
        avg(ct.TTD)[time to delivery],
        avg(ct.DED)[diff estimated delivery]
From
```

```
        [Target SQL Project].dbo.[customers (1)] as c
left join cust  as ct
on ct.cu = c.customer_id
Group by c.customer_state
```

Query Result:



	States	mean freight value	time to delivery	diff estimated delivery
1	PE	32.8605442583732	18	13
2	PB	41.5490942698706	20	13
3	PA	35.9263471241171	23	14
4	RS	21.9825513513517	15	13
5	AC	41.2995121951219	20	20
6	BA	26.2736616014025	19	10
7	SP	15.2686679103595	8	11
8	SC	21.7438760532754	14	11
9	SE	36.7	21	10
10	MA	37.9404232804233	21	9

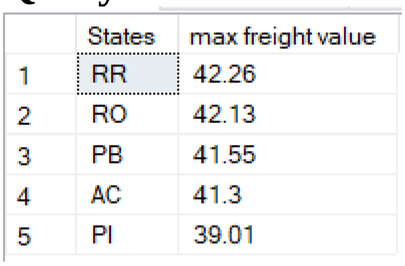
Sort the data to get the following:

5.4 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
--using view created in 5.3


--highest AVERAGE FREIGHT VALUE

select top 5
        c.customer_state as States,
        round(avg(ct.MFV),2)[max freight value]
From
         [Target SQL Project].dbo.[customers (1)] as c
left join
        cust  as ct
On
```

```
        ct.cu = c.customer_id
group by
        c.customer_state
order by
        [max freight value] desc
```

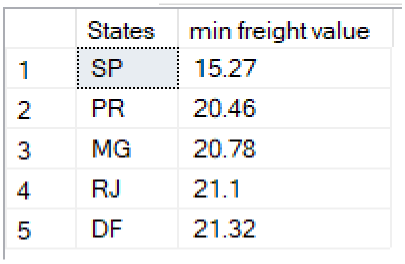
Query Result:



	States	max freight value
1	RR	42.26
2	RO	42.13
3	PB	41.55
4	AC	41.3
5	PI	39.01

```
--LOWEST AVERAGE FREIGHT VALUE

Select top 5
       c.customer_state as States,
       round(avg(ct.MFV),2)[min freight value]
From
        [Target SQL Project].dbo.[customers (1)] as c
left join
        cust  as ct
on
       ct.cu = c.customer_id
Group by
       c.customer_state
Order by
        [min freight value]
```

Query Result:

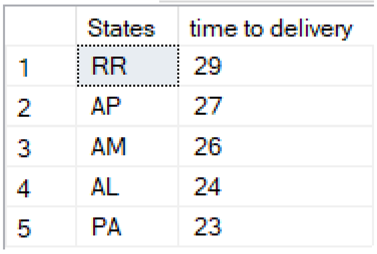


	States	min freight value
1	SP	15.27
2	PR	20.46
3	MG	20.78
4	RJ	21.1
5	DF	21.32

Q5.6 Top 5 states with highest/lowest average time to delivery

```
--using view created in 5.3

--top 5 states with highest average time to delivery

Select top 5
        c.customer_state as States,
        avg(ct.TTD)[time to delivery]
From
        [Target SQL Project].dbo.[customers (1)] as c
Left join
         cust  as ct
On
        ct.cu = c.customer_id
Group by
         c.customer_state
Order by
        [time to delivery] desc
```

Query Result:

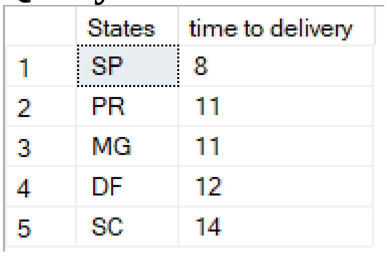


	States	time to delivery
1	RR	29
2	AP	27
3	AM	26
4	AL	24
5	PA	23

```
--top 5 states with lowest average time to delivery

Select top 5
        c.customer_state as States,
        avg(ct.TTD)[time to delivery]
From
        [Target SQL Project].dbo.[customers (1)] as c
Left join
         cust  as ct
On
        ct.cu = c.customer_id
Group by
```

```
        c.customer_state
Order by
        [time to delivery] asc
```

Query Result:

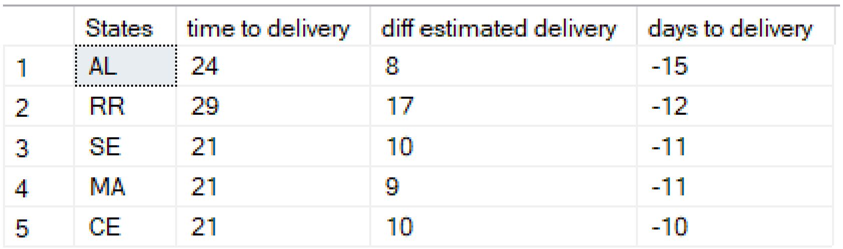


	States	time to delivery
1	SP	8
2	PR	11
3	MG	11
4	DF	12
5	SC	14

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

```
--using view created in 5.3



--top 5 Slow delivery
Select top 5
        c.customer_state as States,
        avg(ct.TTD)[time to delivery],
        avg(ct.DED)[diff estimated delivery],
        avg(ct.DED - ct.TTD)[days to delivery]
From
        [Target SQL Project].dbo.[customers (1)] as c
Left join
         cust  as ct
On
        ct.cu = c.customer_id
Group by
        c.customer_state
Order by
         [days to delivery]
```

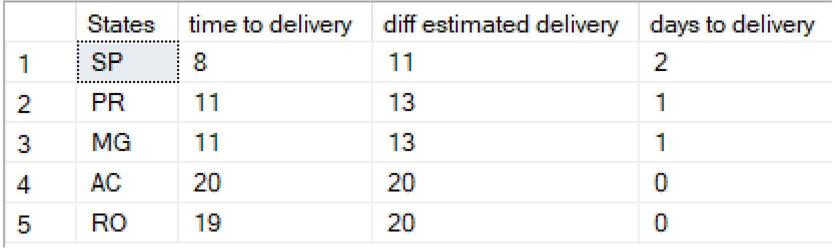
Query Result:



	States	time to delivery	diff estimated delivery	days to delivery
1	AL	24	8	-15
2	RR	29	17	-12
3	SE	21	10	-11
4	MA	21	9	-11
5	CE	21	10	-10

```
--top 5 Fast delivery

select top 5
        c.customer_state as States,
        avg(ct.TTD)[time to delivery],
        avg(ct.DED)[diff estimated delivery],
        avg(ct.DED - ct.TTD)[days to delivery]
from
        [Target SQL Project].dbo.[customers (1)] as c
left join
        cust  as ct
on
        ct.cu = c.customer_id
group by
        c.customer_state
order by
         [days to delivery] desc
```

Query Result:



	States	time to delivery	diff estimated delivery	days to delivery
1	SP	8	11	2
2	PR	11	13	1
3	MG	11	13	1
4	AC	20	20	0
5	RO	19	20	0

Analysis on sales, freight and delivery time:

The analysis on sales, freight and delivery time provides valuable insights into the performance of the business across different states, allow identidication of areas for improvement and optimze operations to enhance customer satisfaction and increase sales.

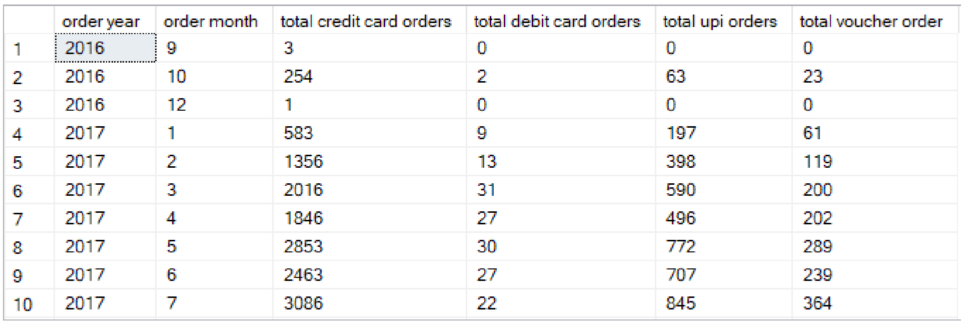
6. Payment type analysis:

6.1 Month over Month count of orders for different payment types

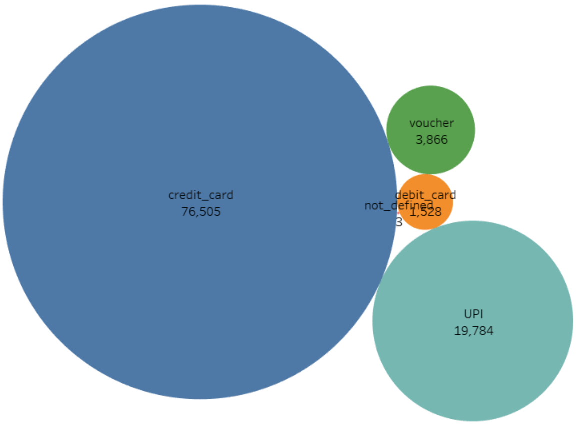
```
SELECT

      YEAR(o.order_purchase_timestamp) [order year],

      MONTH(o.order_purchase_timestamp) [order month],

      COUNT(CASE WHEN p.payment_type = 'credit_card' THEN 1 END) [total credit card orders],

      COUNT(CASE WHEN p.payment_type = 'debit_card' THEN 1 END) [total debit card orders],

      COUNT(CASE WHEN p.payment_type = 'UPI' THEN 1 END)  [total upi orders],

      COUNT(CASE WHEN p.payment_type = 'voucher' THEN 1 END)  [total voucher order]

FROM

    [Target SQL Project].dbo.[orders (1)] AS o

LEFT JOIN

    [Target SQL Project].dbo.payments AS p ON p.order_id = o.order_id

GROUP BY

    YEAR(o.order_purchase_timestamp) ,

    MONTH(o.order_purchase_timestamp)
```

```
Order by
        [order year], [order month]
```

Query Result:



	order year	order month	total credit card orders	total debit card orders	total upi orders	total voucher order
1	2016	9	3	0	0	0
2	2016	10	254	2	63	23
3	2016	12	1	0	0	0
4	2017	1	583	9	197	61
5	2017	2	1356	13	398	119
6	2017	3	2016	31	590	200
7	2017	4	1846	27	496	202
8	2017	5	2853	30	772	289
9	2017	6	2463	27	707	239
10	2017	7	3086	22	845	364



Insights:

Here customers are using various type of payment mode but the highest amount order has been done via credit card. So the highest amount of transaction is done via credit cards.

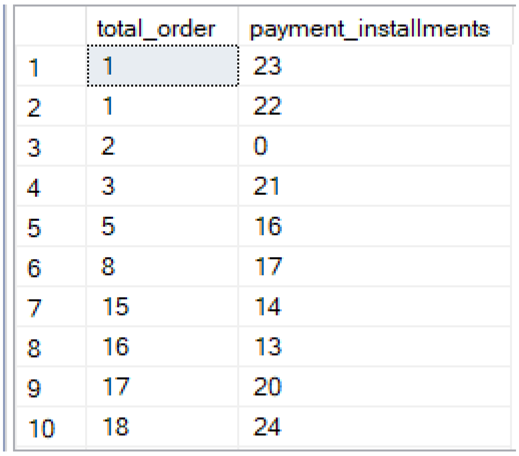
Recommendation:

As the highest amount of transactions are happened through credit cards, we can make a partnership with credit card companies so that customers can get some discounts and it will help in increase sales.

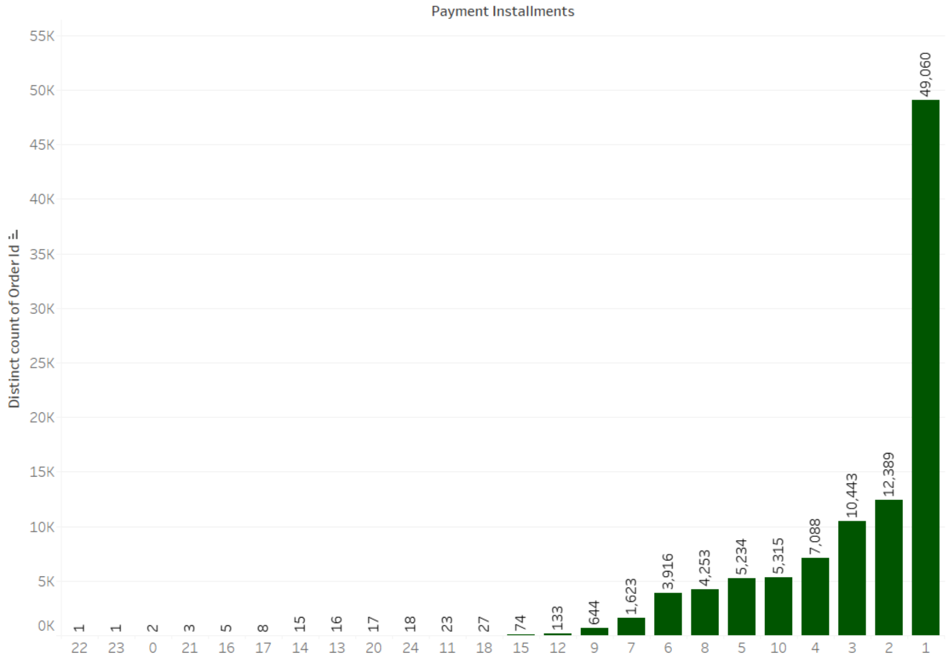
6.2 Count of orders based on the no. of payment installments

```
Select
        COUNT(*) as total_order,
        payment_installments
From
         [Target SQL Project].dbo.payments
Group by
        payment_installments
Order by
         total_order
```

Query Result:



	total_order	payment_installments
1	1	23
2	1	22
3	2	0
4	3	21
5	5	16
6	8	17
7	15	14
8	16	13
9	17	20
10	18	24



Insights: The result showsthat the majority of orders were made with single payment installment, but there is still a significant number of orders made with multiple installments, indicating a potential need for more flexible payment option.