Target Business Case Study

Problem Statement

Analyzing Target’s extensive dataset using SQL enables the extraction of valuable insights into Target's operations in Brazil. The data reveals key information about order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction.

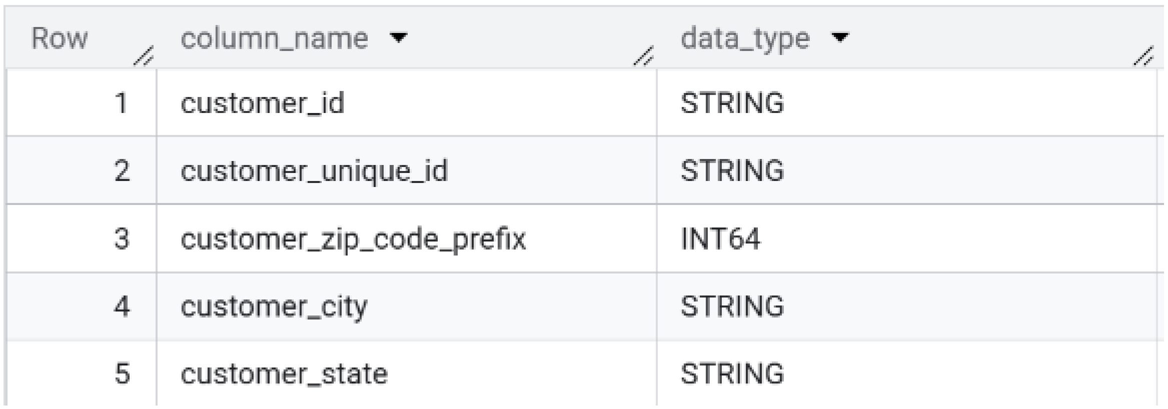
1. Exploratory Analysis

1.1 Data type of all columns in the customers table

Query

```
SELECT
  column_name,
  data_type
FROM
  aerobic-smithy-461311-s2.Target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name = 'customerst'
```

Result



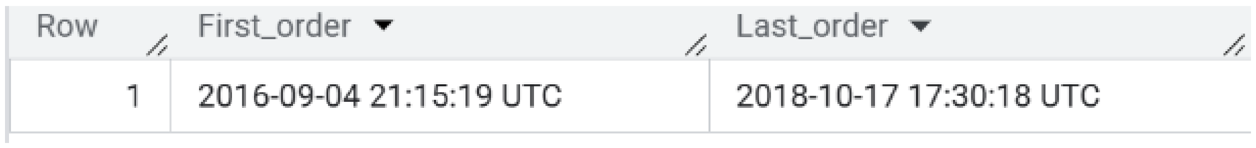
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

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

Query

```
SELECT
   MIN(order_purchase_timestamp) as First_order,
   MAX(order_purchase_timestamp) as Last_order
FROM
   Target.orders;
```

Result



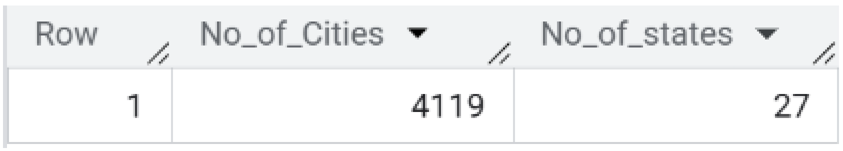
Row	First_order	Last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

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

Query

```
SELECT
  COUNT(DISTINCT customer_city) as customer_city,
  COUNT(DISTINCT customer_state) as customer_state
FROM
  Target.customerst as c
  INNER JOIN Target.orders as o
  ON c.customer_id = o.customer_id;
```

Result



Row	No_of_Cities	No_of_states
1	4119	27

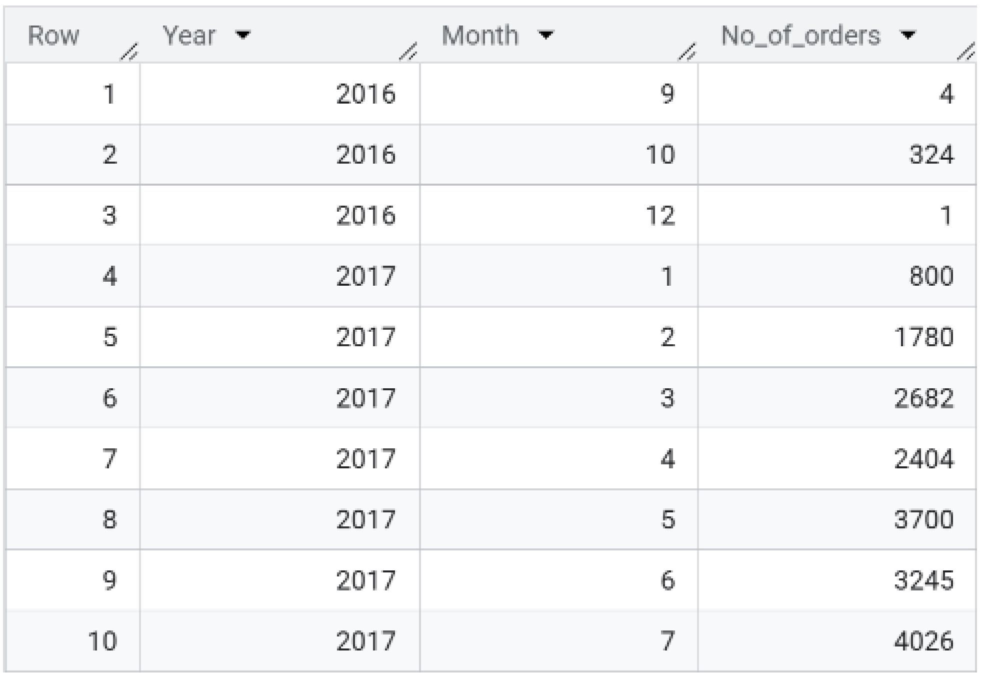
2. In-depth Exploration:

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

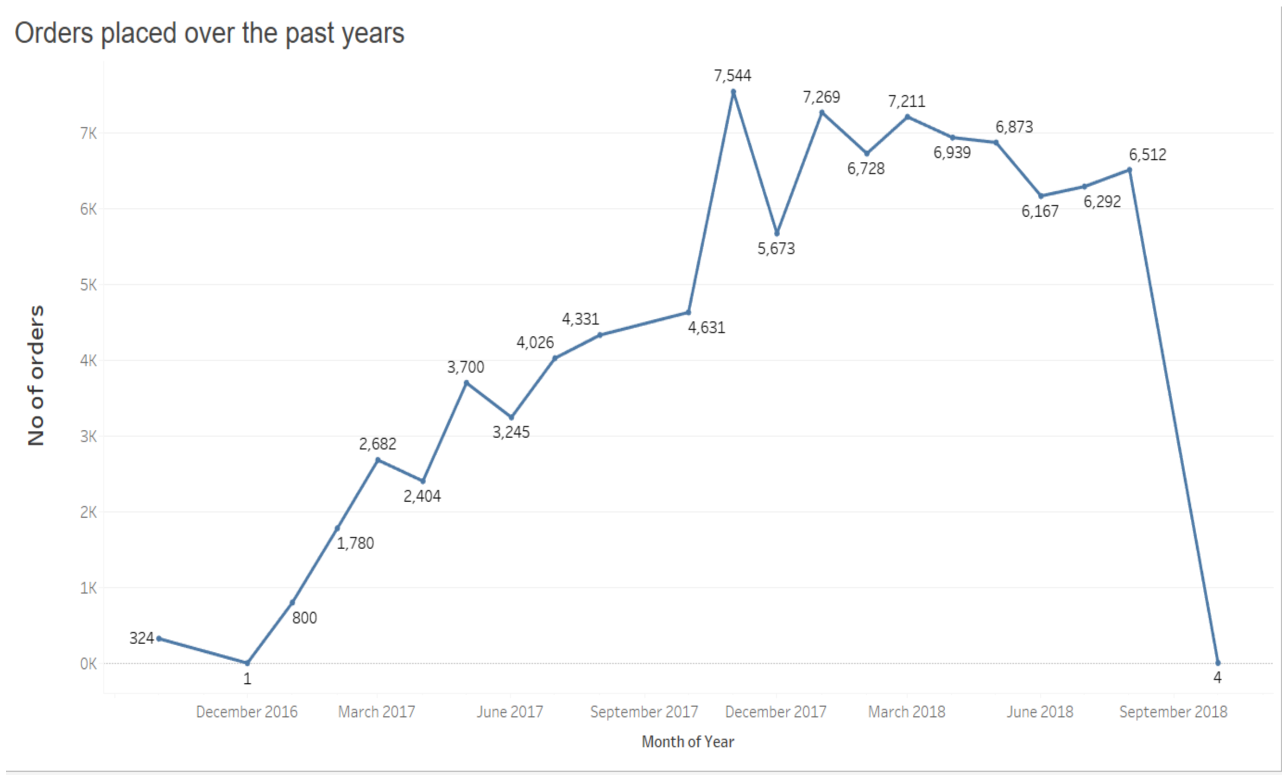
Query

```
SELECT
  EXTRACT(year from order_purchase_timestamp) AS Year,
  EXTRACT(month from order_purchase_timestamp) AS Month,
  COUNT(*) AS No_of_orders
FROM
  Target.orders
GROUP BY
  Year,
  Month
ORDER BY
  Year,
  Month,
  No_of_orders DESC
```

Result



Row	Year	Month	No_of_orders
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



Insights

- The number of orders steadily increased from September 2016 through November 2017, reaching a peak of 7,544 orders in that month. Post this peak, the order volume remained fairly stable, generally staying between 6,000 to 7,000 orders each month.
- However, there was a sudden drop in September and October 2018, where the number of orders fell noticeably.

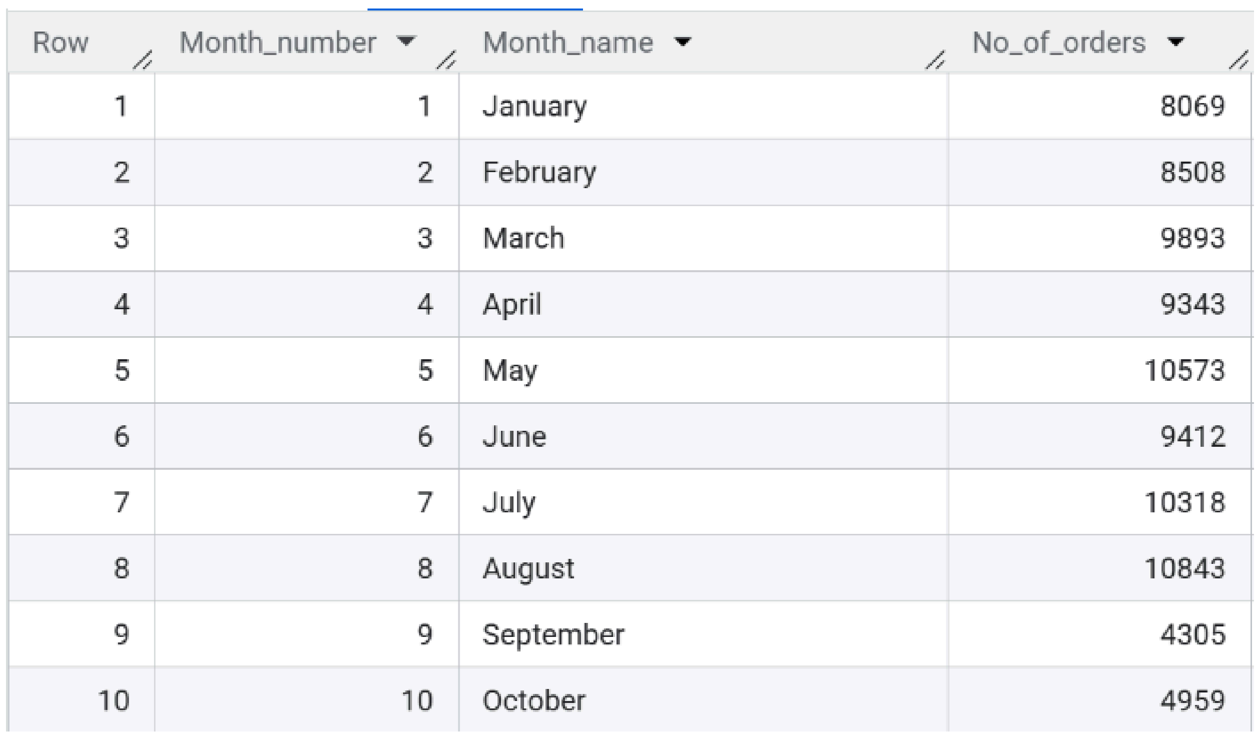
2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Query

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS Month_number,
  FORMAT_DATE("%B",order_purchase_timestamp) AS Month_name,
  COUNT(*) AS No_of_orders
```

```
FROM
  Target.orders
GROUP BY
  Month_number,
  Month_name
ORDER BY
  Month_number
```

Result



Row	Month_number	Month_name	No_of_orders
1	1	January	8069
2	2	February	8508
3	3	March	9893
4	4	April	9343
5	5	May	10573
6	6	June	9412
7	7	July	10318
8	8	August	10843
9	9	September	4305
10	10	October	4959

Insights

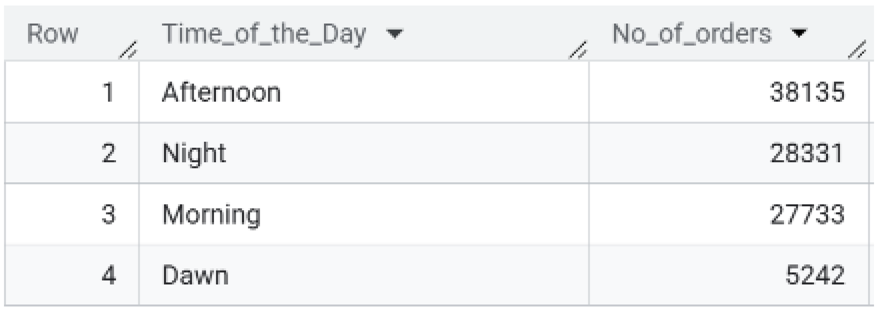
- The number of orders is noticeably higher during May to August, which mostly falls in Brazil’s winter season. Hence the middle part of the year experiences more orders.
- On the other hand, from September to December, fewer orders are placed compared to other months.This might be due to holidays or other seasonal reasons affecting customer purchases

2.3 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

Query

```
SELECT
  CASE
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN 'Morning'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN  'Night'
  END as Time_of_the_Day,
  COUNT(order_id) as No_of_orders
FROM
  Target.customerst as c
  INNER JOIN Target.orders as o
  ON c.customer_id = o.customer_id
GROUP  BY
  Time_of_the_Day
ORDER BY No_of_orders DESC;
```

Result



Row	Time_of_the_Day	No_of_orders
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights

Brazilian customers mostly place their orders during the afternoon, followed by night time. The number of orders placed at dawn is very low.

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

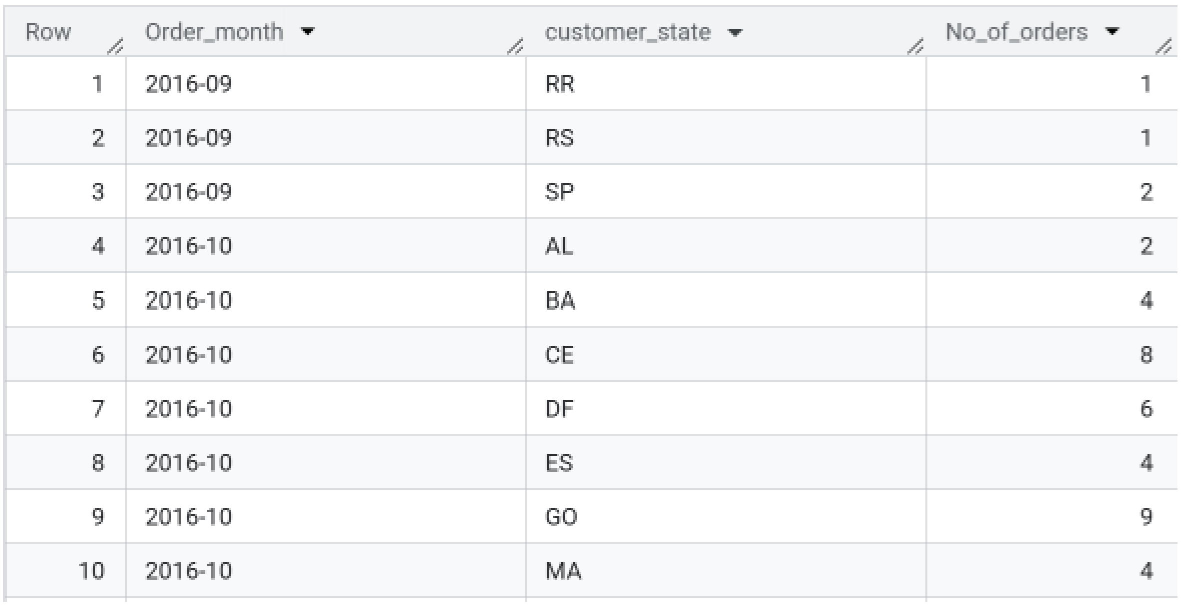
3.1 Get the month on month no. of orders placed in each state.

Query

```
 SELECT
   FORMAT_DATE("%Y-%m",order_purchase_timestamp) AS Order_month,
   customer_state,
   COUNT(order_id) AS No_of_orders

FROM Target.customerst AS c
JOIN Target.orders AS o
   ON c.customer_id = o.customer_id
GROUP BY Order_month,customer_state
ORDER BY
   Order_month,
   customer_state
```

Result



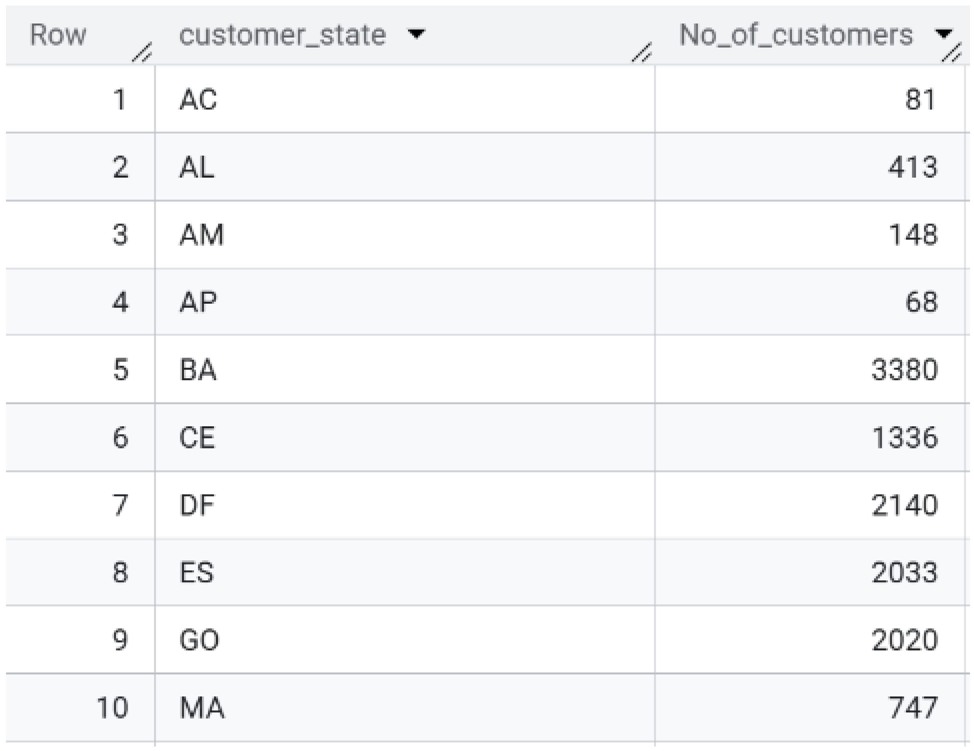
Row	Order_month	customer_state	No_of_orders
1	2016-09	RR	1
2	2016-09	RS	1
3	2016-09	SP	2
4	2016-10	AL	2
5	2016-10	BA	4
6	2016-10	CE	8
7	2016-10	DF	6
8	2016-10	ES	4
9	2016-10	GO	9
10	2016-10	MA	4

3.2 How are the customers distributed across all the states?

Query

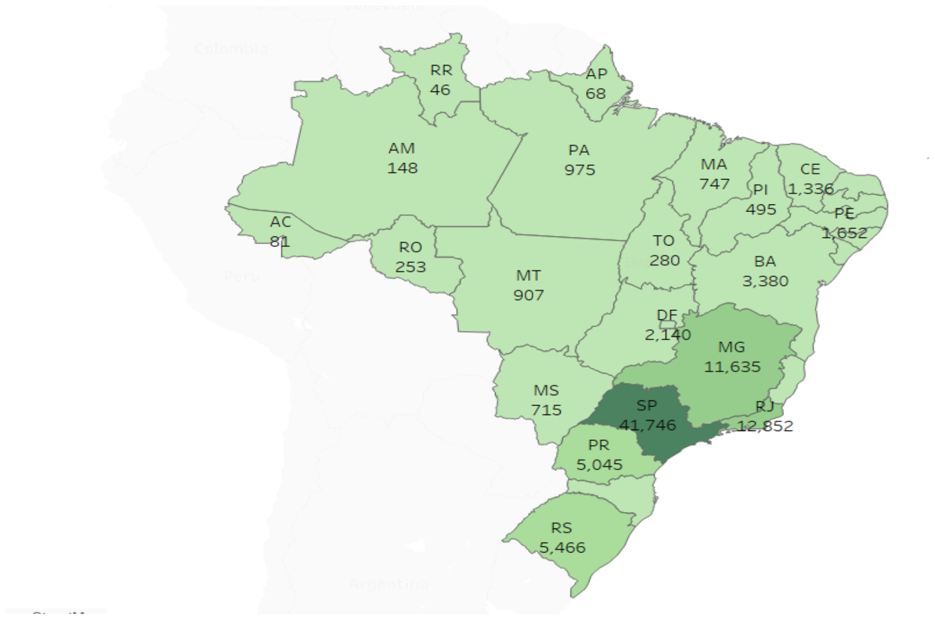
```
 SELECT
  customer_state,
  COUNT(customer_id) AS No_of_customers
FROM
  `Target.customerst`
GROUP BY
  customer_state
ORDER BY
  customer_state;
```

Result



Row	customer_state	No_of_customers
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

Customer Distribution Across States



Insights

- Among all the Brazilian states, São Paulo **(SP)** has the highest number of customers, with **41,746**. On the other hand, **Amapá (AP)** has the lowest, with just **81 customers**.
- This clearly indicates that **most of the orders are placed from the SP state**, which aligns with its large customer base.

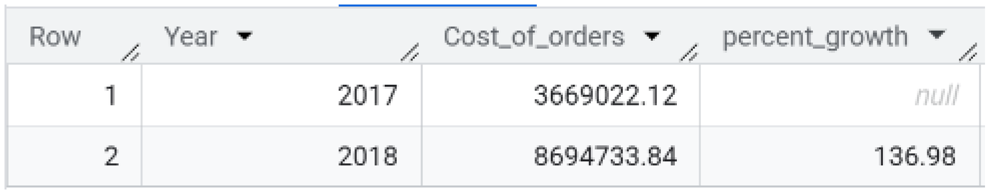
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

Query 1

```
WITH Total AS
(
SELECT
  Extract(YEAR FROM order_purchase_timestamp) AS Year,
  ROUND(SUM(payment_value),2) AS Cost_of_orders,
FROM
  Target.orders AS o
  INNER JOIN `Target.payments` AS p
  ON o.order_id = p.order_id
WHERE EXTRACT(YEAR from order_purchase_timestamp) BETWEEN 2017 and 2018
  AND EXTRACT(MONTh from order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY
  Year)
SELECT
  *,
  ROUND(
  ((Cost_of_orders - LAG(Cost_of_orders) OVER(ORDER BY Year))/ LAG(Cost_of_orders)
OVER(ORDER BY Year)) * 100,2)
  AS percent_growth
FROM Total
ORDER BY
  Year
```

Result 1



Row	Year	Cost_of_orders	percent_growth
1	2017	3669022.12	null
2	2018	8694733.84	136.98

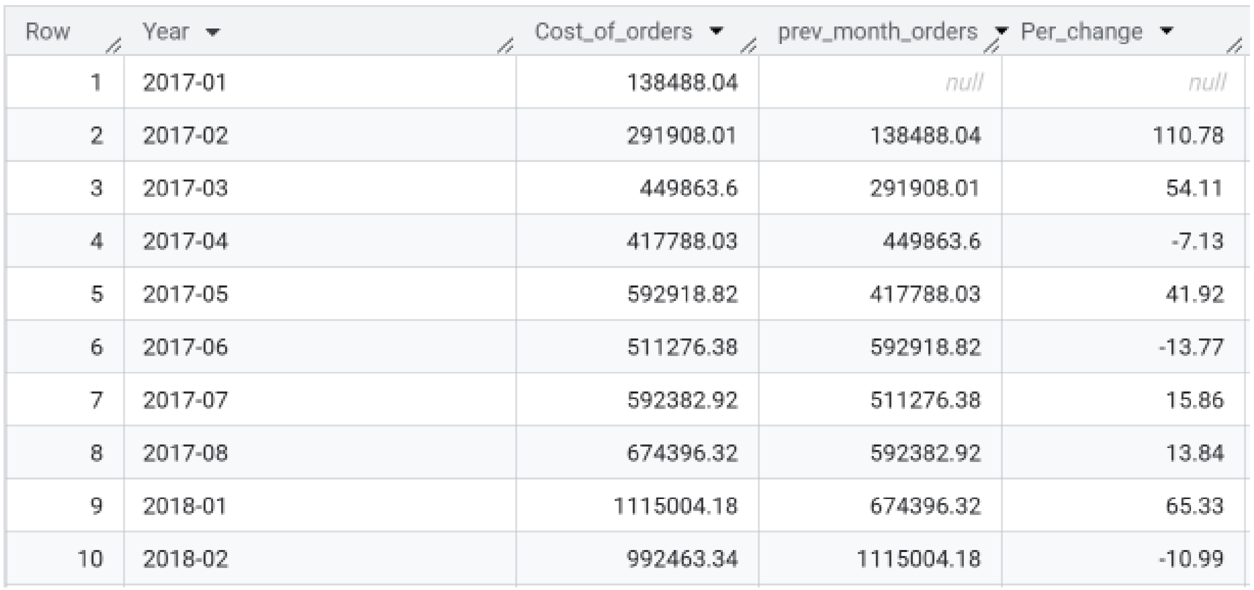
Query 2

```
SELECT
  *,
  LAG(Cost_of_orders) OVER(ORDER BY Year) AS prev_month_orders,
  ROUND(((Cost_of_orders - LAG(Cost_of_orders) OVER(ORDER BY
Year))/LAG(Cost_of_orders) OVER(ORDER BY Year))*100,2) AS Per_change
FROM
(
SELECT
  FORMAT_DATE("%Y-%m",order_purchase_timestamp) AS Year,
  ROUND(SUM(payment_value),2) AS Cost_of_orders
FROM
```

```
  Target.orders AS o
  INNER JOIN `Target.payments` AS p
  ON o.order_id = p.order_id
WHERE EXTRACT(YEAR from order_purchase_timestamp) BETWEEN 2017 and 2018
  AND EXTRACT(MONTh from order_purchase_timestamp) BETWEEN 1 AND 8

GROUP BY
  Year
  )
ORDER BY
  Year,
  Cost_of_orders DESC
```

Result 2



Row	Year	Cost_of_orders	prev_month_orders	Per_change
1	2017-01	138488.04	null	null
2	2017-02	291908.01	138488.04	110.78
3	2017-03	449863.6	291908.01	54.11
4	2017-04	417788.03	449863.6	-7.13
5	2017-05	592918.82	417788.03	41.92
6	2017-06	511276.38	592918.82	-13.77
7	2017-07	592382.92	511276.38	15.86
8	2017-08	674396.32	592382.92	13.84
9	2018-01	1115004.18	674396.32	65.33
10	2018-02	992463.34	1115004.18	-10.99

Insights

- There is a noticeable growth in the cost of orders compared to the previous year.
- This suggests that changes in the economy may have influenced people's buying behavior, leading them to spend more or choose higher-value products.

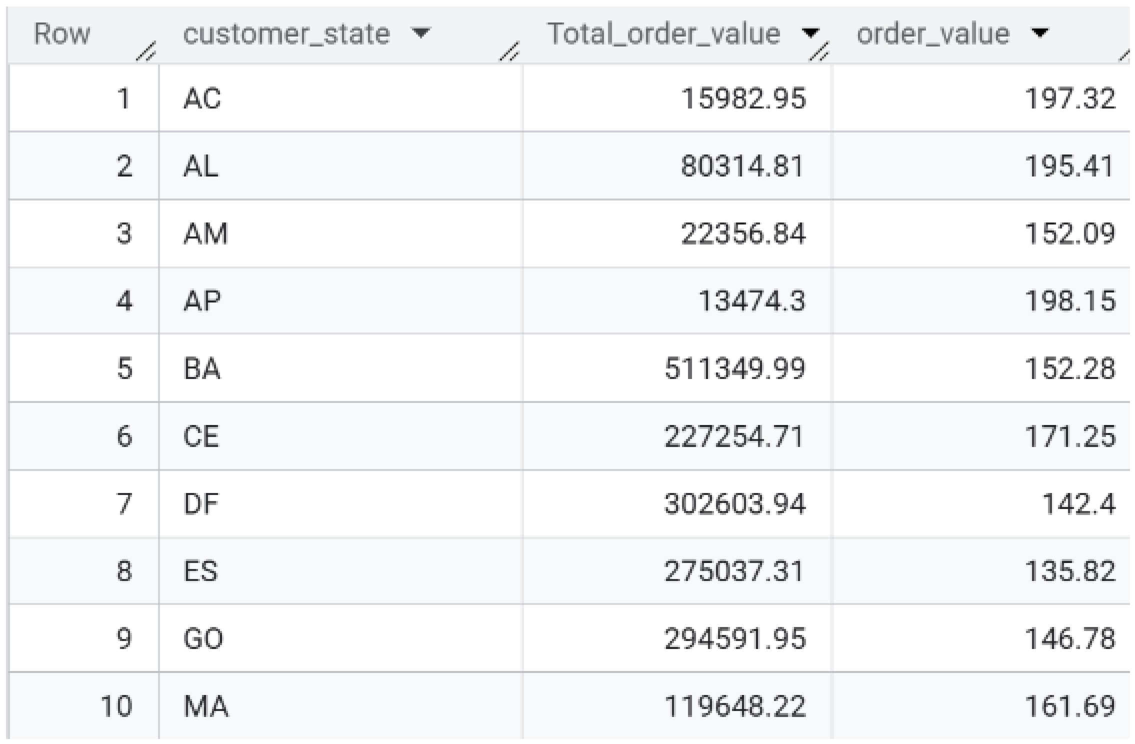
4.2 Calculate the Total & Average value of order price for each state.

Query

```
SELECT
  customer_state,
  ROUND(SUM(price),2) AS Total_order_value,
  ROUND(SUM(price)/COUNT(DISTINCT i.order_id),2) AS order_value

FROM
 `Target.customerst` AS  c
  INNER JOIN Target.orders AS  o
  ON c.customer_id = o.customer_id
  INNER JOIN `Target.order_items` AS  i
  ON o.order_id = i.order_id
GROUP BY
  customer_state
ORDER BY
  customer_state
```

Result



Row	customer_state	Total_order_value	order_value
1	AC	15982.95	197.32
2	AL	80314.81	195.41
3	AM	22356.84	152.09
4	AP	13474.3	198.15
5	BA	511349.99	152.28
6	CE	227254.71	171.25
7	DF	302603.94	142.4
8	ES	275037.31	135.82
9	GO	294591.95	146.78
10	MA	119648.22	161.69

Insights

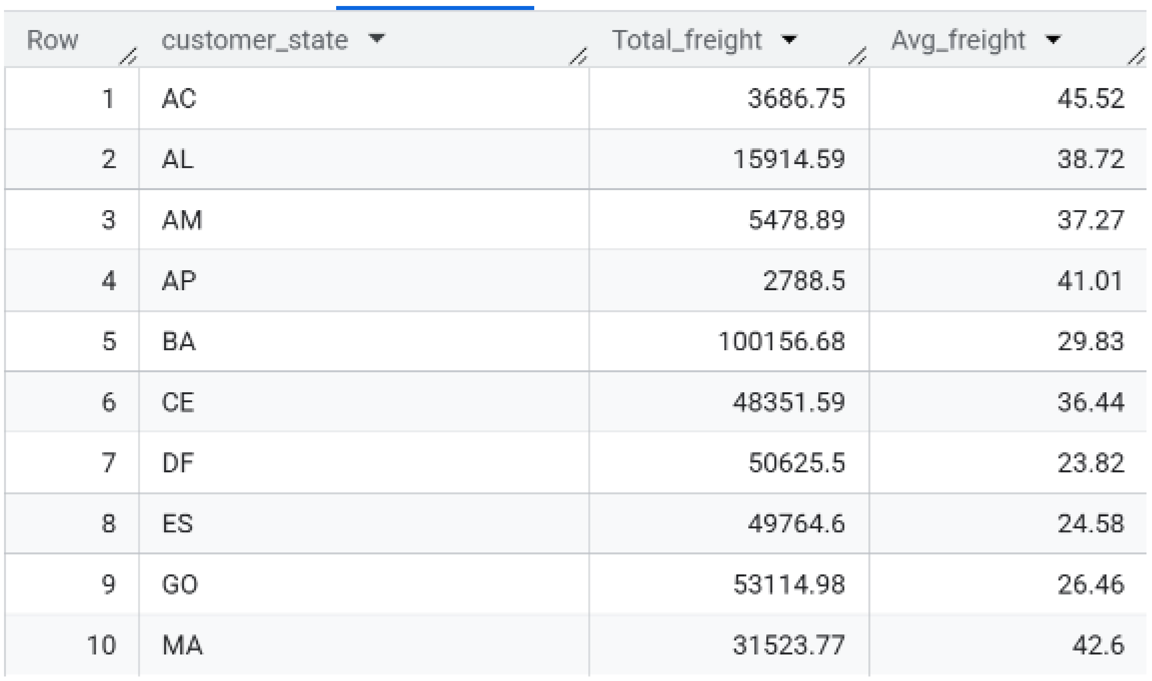
- The highest average order value was recorded in the PB state, at 216.67. This suggests that customers in **PB** may place fewer but higher-value orders, possibly due to bulk purchases or a preference for premium products.
- In contrast, **São Paulo (SP)** reported the lowest average order value, at 125.75. This may indicate that customers in SP tend to make more frequent purchases..

4.3 Calculate the Total & Average value of order freight for each state.

Query

```
SELECT
  customer_state,
  ROUND(SUM(freight_value),2) AS Total_freight,
  ROUND(SUM(freight_value)/COUNT(DISTINCT i.order_id),2) AS Avg_freight
FROM
  Target.customerst AS  c
  INNER JOIN Target.orders AS  o
  ON c.customer_id = o.customer_id
  INNER JOIN `Target.order_items` AS  i
  ON o.order_id = i.order_id
GROUP BY
  customer_state
ORDER BY
  Customer_state
```

Result



Row	customer_state	Total_freight	Avg_freight
1	AC	3686.75	45.52
2	AL	15914.59	38.72
3	AM	5478.89	37.27
4	AP	2788.5	41.01
5	BA	100156.68	29.83
6	CE	48351.59	36.44
7	DF	50625.5	23.82
8	ES	49764.6	24.58
9	GO	53114.98	26.46
10	MA	31523.77	42.6

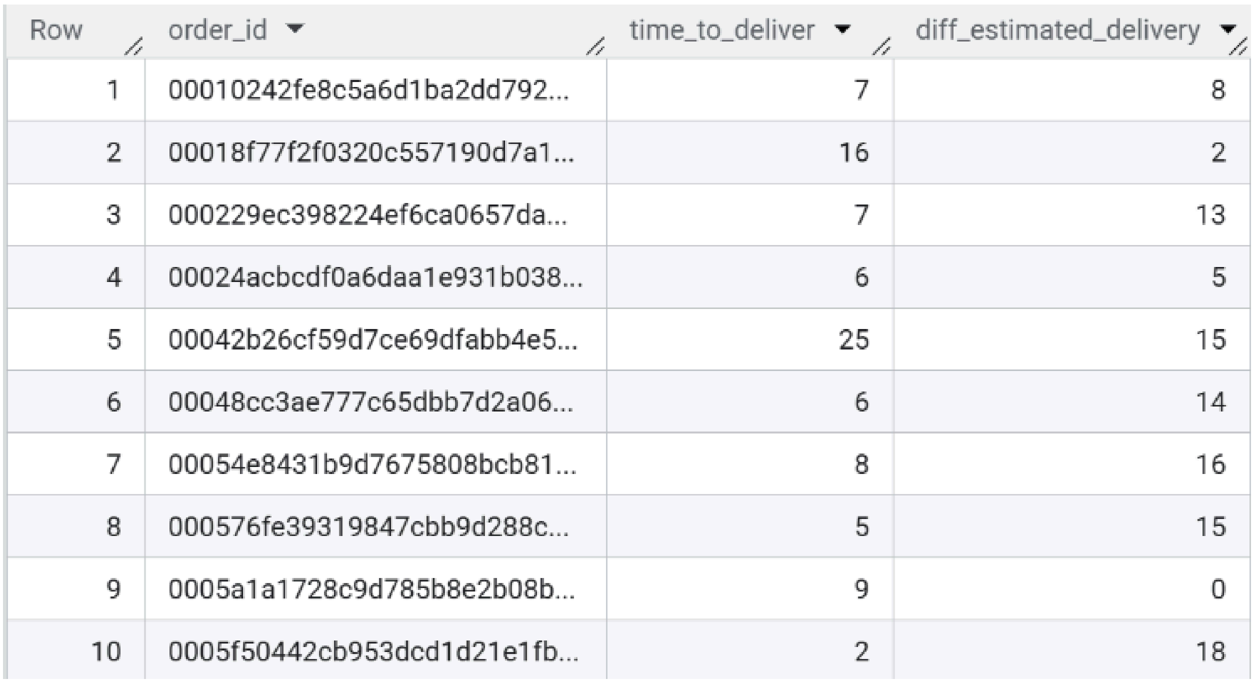
5. Analysis based on sales, freight and delivery time.

5.1 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.

Query

```
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS
time_to_deliver,
  DATE_DIFF(order_estimated_delivery_date ,order_delivered_customer_date,DAY) AS
diff_estimated_delivery
FROM
  `Target.orders`
WHERE
  order_purchase_timestamp IS NOT NULL
  AND order_delivered_customer_date IS NOT NULL
  AND order_estimated_delivery_date IS NOT NULL
ORDER BY
  Order_id;
```

Result



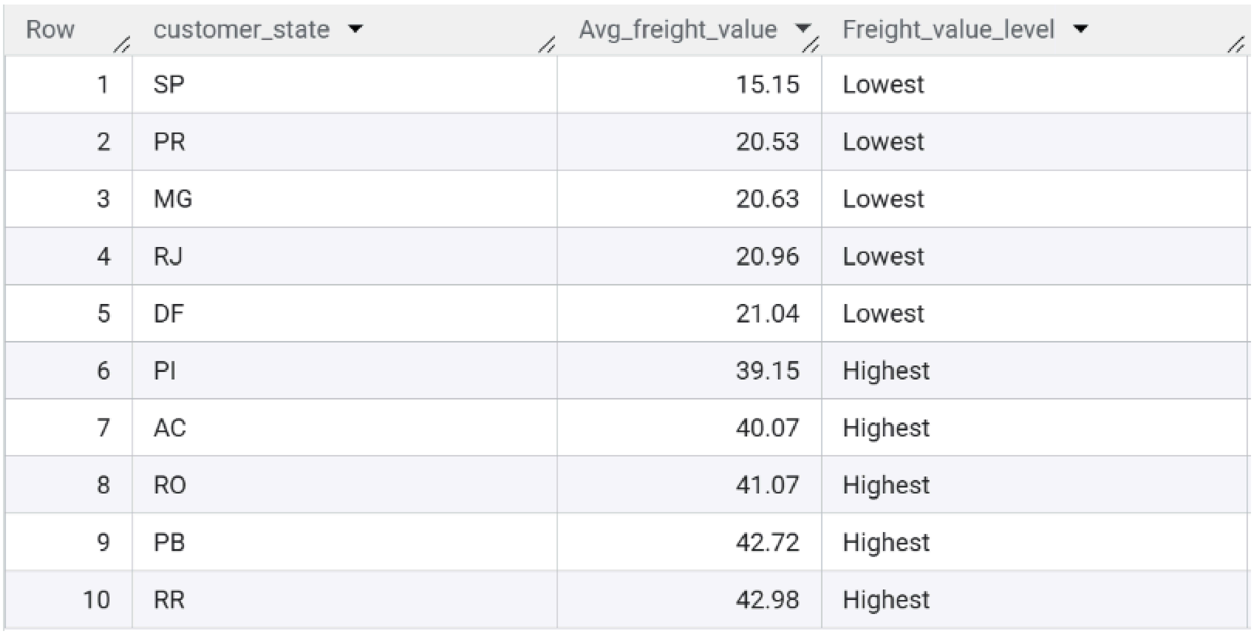
Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b038...	6	5
5	00042b26cf59d7ce69dfabb4e5...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb81...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08b...	9	0
10	0005f50442cb953dcd1d21e1fb...	2	18

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

Query

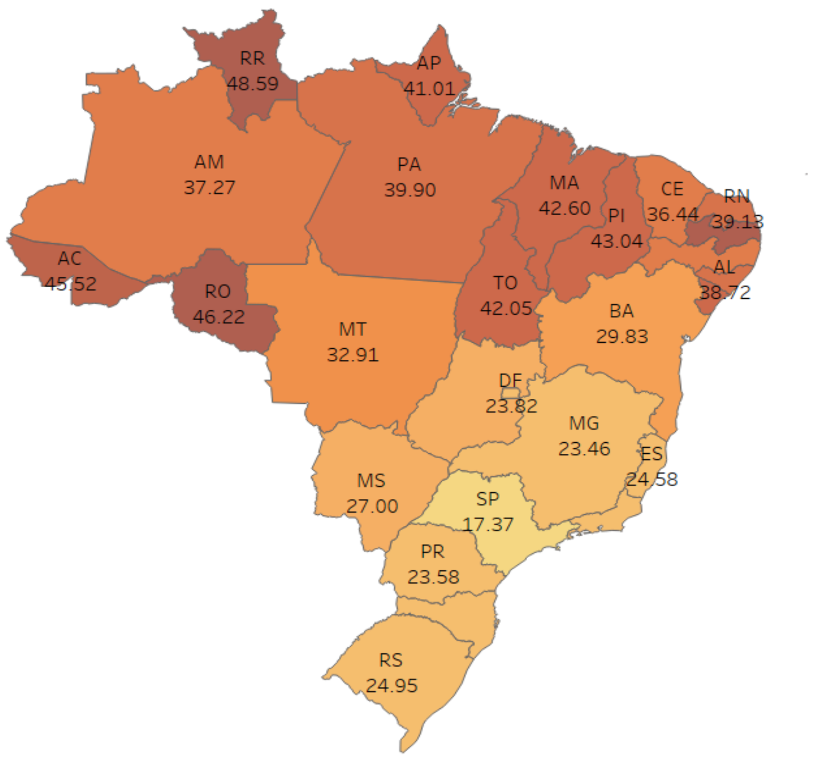
```
SELECT
  *,
  CASE WHEN ROW_NUMBER() OVER(ORDER BY Avg_freight_value) <=5 THEN 'Lowest'
  ELSE 'Highest'
  END AS Freight_value_level
FROM(
(
SELECT
  customer_state,
  ROUND(AVG(freight_value),2) AS Avg_freight_value
FROM
  `Target.customerst` AS c
  INNER JOIN Target.orders AS o
  ON c.customer_id = o.customer_id
  INNER JOIN `Target.order_items` AS i
  ON o.order_id = i.order_id
GROUP BY
  customer_state
ORDER BY
  Avg_freight_value
  LIMIT 5)
UNION ALL
(
SELECT
    customer_state,
  ROUND(AVG(freight_value),2) AS Avg_freight_value
FROM
  `Target.customerst` AS c
  INNER JOIN Target.orders AS o
  ON c.customer_id = o.customer_id
  INNER JOIN `Target.order_items` AS i
  ON o.order_id = i.order_id
GROUP BY
  customer_state
ORDER BY
  Avg_freight_value DESC
LIMIT 5))
```

Result



Row	customer_state	Avg_freight_value	Freight_value_level
1	SP	15.15	Lowest
2	PR	20.53	Lowest
3	MG	20.63	Lowest
4	RJ	20.96	Lowest
5	DF	21.04	Lowest
6	PI	39.15	Highest
7	AC	40.07	Highest
8	RO	41.07	Highest
9	PB	42.72	Highest
10	RR	42.98	Highest

Average Freight Value in each state



Insights

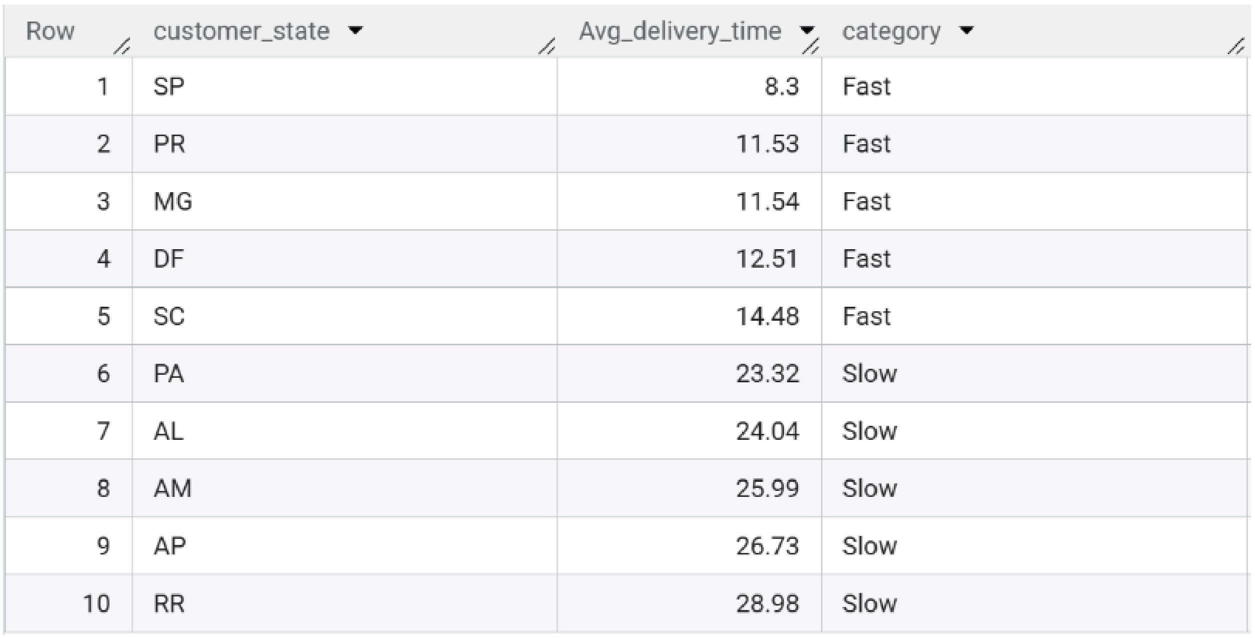
- SP has the lowest shipping cost in Brazil probably because it’s close to warehouses or has better delivery options.
- On the other hand, states in the northern regions have much higher shipping costs, almost three times more than SP. This might be because they are far away and have fewer delivery services.

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

Query

```
WITH delivery_time AS
(
  SELECT
  customer_state,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),2)
AS Avg_delivery_time
FROM
 `Target.customerst` AS c
  INNER JOIN `Target.orders` AS o
  ON c.customer_id = o.customer_id
WHERE order_delivered_customer_date IS NOT NULL
GROUP BY
  customer_state),
ranked AS
(
SELECT
  *,
  ROW_NUMBER() OVER (ORDER BY Avg_delivery_time ASC) AS rn_asc,
  ROW_NUMBER() OVER (ORDER BY Avg_delivery_time DESC) AS rn_desc
FROM delivery_time)
SELECT
  customer_state,
  Avg_delivery_time,
  CASE
  WHEN rn_asc <= 5 THEN 'Fast'
  WHEN rn_desc <=5 THEN 'Slow'
  END AS category
FROM ranked
WHERE rn_asc <= 5 OR rn_desc <= 5
ORDER BY
  Avg_delivery_time
```

Result



Row	customer_state	Avg_delivery_time	category
1	SP	8.3	Fast
2	PR	11.53	Fast
3	MG	11.54	Fast
4	DF	12.51	Fast
5	SC	14.48	Fast
6	PA	23.32	Slow
7	AL	24.04	Slow
8	AM	25.99	Slow
9	AP	26.73	Slow
10	RR	28.98	Slow

Insights

- The state code SP has the fastest delivery time among all states in Brazil. This could be because it has local vendors or is located close to Target's distribution center.
- On the other hand, RR has the highest delivery time, which may indicate limited logistics or vendor availability in that region.

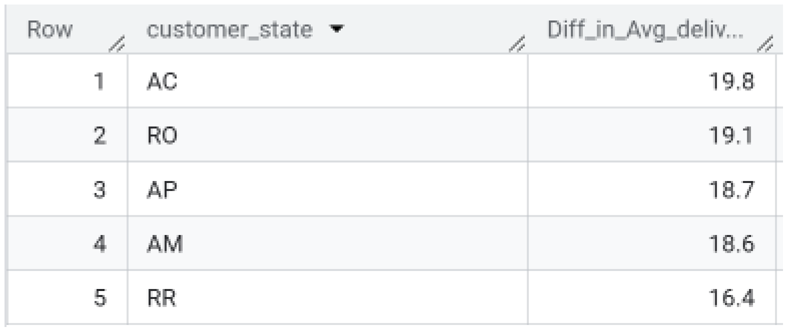
5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery

Query

```
SELECT
  customer_state,
  ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,DAY)),1) AS Diff_in_Avg_delivery
FROM
 `Target.customerst` AS  c
  INNER JOIN `Target.orders` AS  o
  ON c.customer_id = o.customer_id
WHERE order_status = 'delivered'
```

```
GROUP BY
  customer_state
ORDER BY
  Diff_in_Avg_delivery DESC
LIMIT 5;
```

Result



Row	customer_state	Diff_in_Avg_deliv...
1	AC	19.8
2	RO	19.1
3	AP	18.7
4	AM	18.6
5	RR	16.4

Insights

Although AC, AM, and RR are states where delivery times are generally slow based on previous analysis, customers in these regions still receive their products earlier than expected.This shows that the actual delivery performance is better than what was estimated.

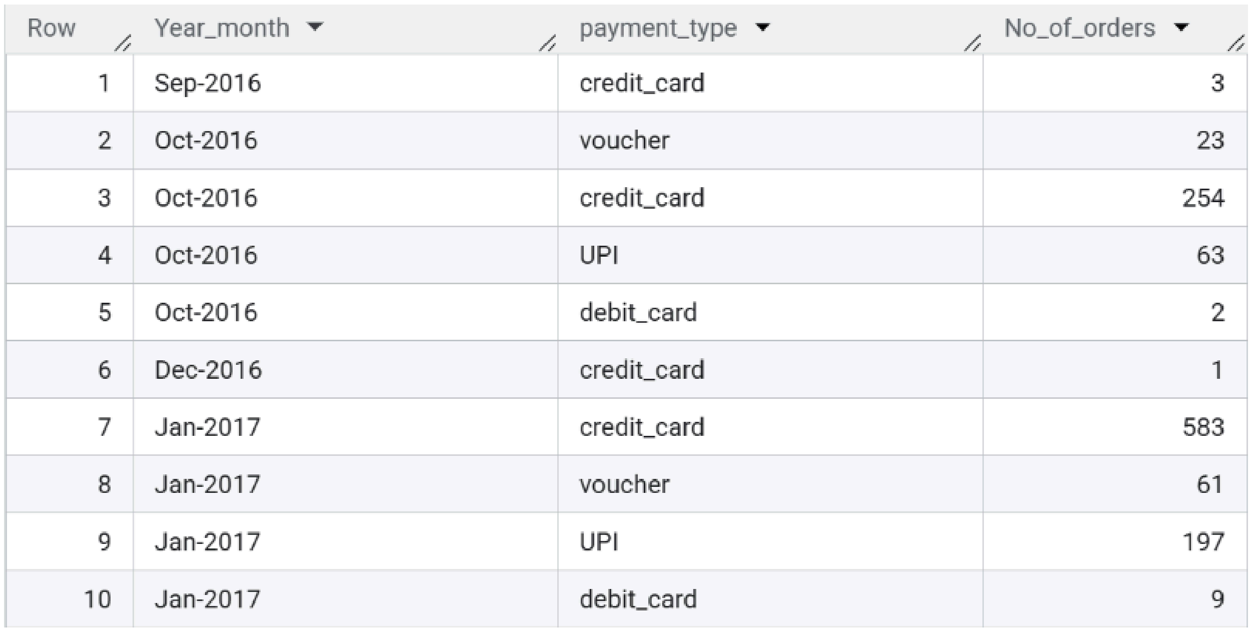
6. Analysis based on the payments

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

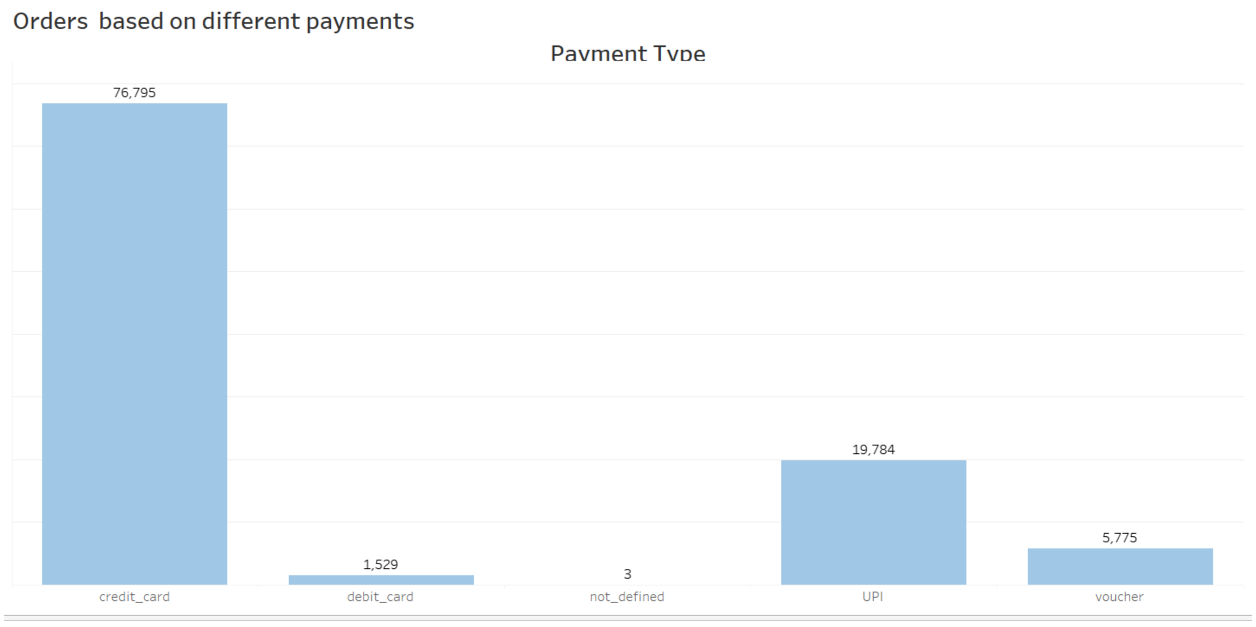
Query

```
WITH time_period AS
(
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS Year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS Month,
  FORMAT_DATE("%b-%Y", order_purchase_timestamp) AS Year_month,
  payment_type,
  COUNT(o.order_id) AS No_of_orders
FROM
 `Target.orders` AS o
  INNER JOIN `Target.payments` AS p
  ON o.order_id = p.order_id
GROUP BY
  Year,Month,
  Year_month,
  payment_type
ORDER BY
  Year,Month)
SELECT
  Year_month,
  payment_type,
  No_of_orders
FROM time_period
```

Result



Row	Year_month	payment_type	No_of_orders
1	Sep-2016	credit_card	3
2	Oct-2016	voucher	23
3	Oct-2016	credit_card	254
4	Oct-2016	UPI	63
5	Oct-2016	debit_card	2
6	Dec-2016	credit_card	1
7	Jan-2017	credit_card	583
8	Jan-2017	voucher	61
9	Jan-2017	UPI	197
10	Jan-2017	debit_card	9



Insights

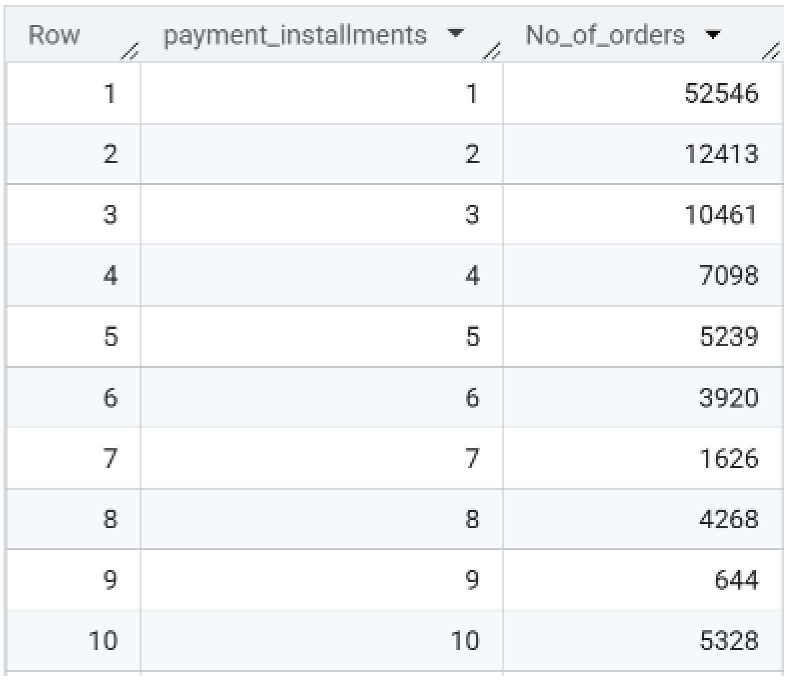
- Customers have consistently preferred using credit cards as their primary mode of payment over the years .
- Debit card usage remains the least preferred, with only 1,529 orders.

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

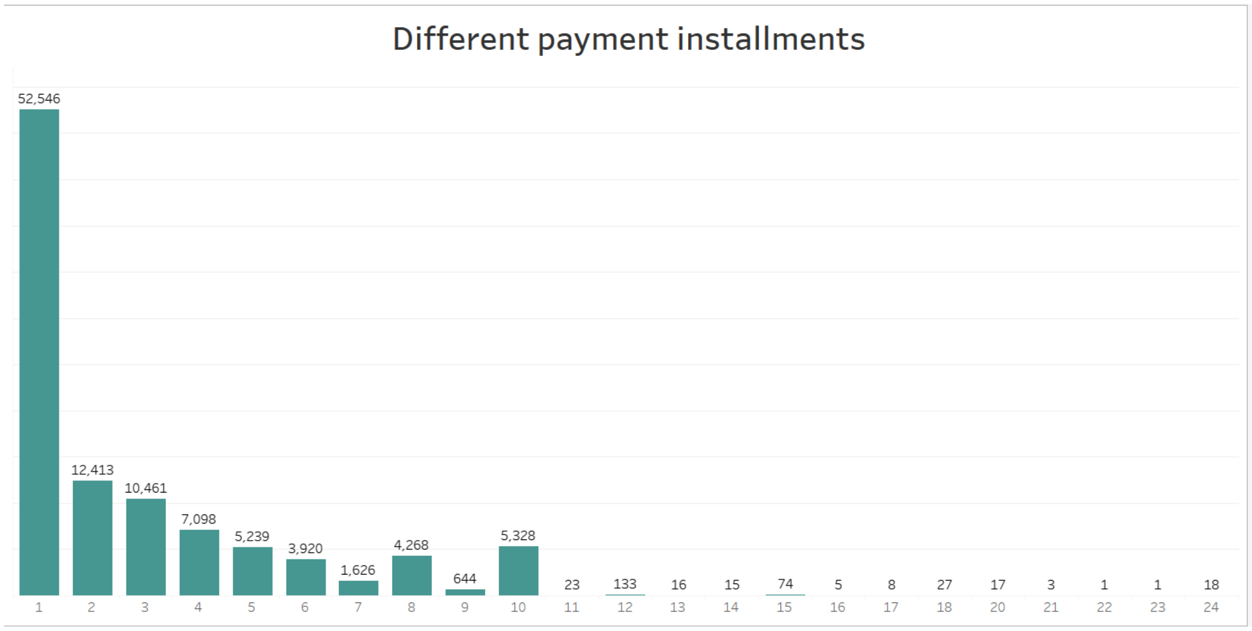
Query

```
SELECT
  payment_installments,
  COUNT(order_id) AS No_of_orders
FROM
  Target.payments
WHERE payment_installments >= 1
GROUP BY
  Payment_installments
```

Result



Row	payment_installments	No_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328



Insights

- Customers mostly choose one-month payment installments. This shows they like to finish their payments quickly instead of spreading them over a long time.
- Long-term installment plans receive very few orders.

Recommandations

- To increase orders during the low-demand months (September to December), Target can introduce special shopping events or offer holiday deals around Christmas and other local holidays.
- Afternoon and Night time experience the highest traffic. Target ensures that their website runs smoothly during these peak hours. Additionally dispatch teams are well-staffed during this period, and supported with automated alerts to enable faster processing and timely deliveries.
- Target can reduce avg freight value and delivery time by partnering with local vendors to enable faster delivery and lower shipping expenses. This approach can help Boost order volume in the Northern regions like Paraiba (PB) by making products more affordable and improving delivery speed.
- Since the mode of payment through credit card is very high Target should retain this strategy to make the credit card payment process as smooth as possible.
- To increase debit card adoption, Target can collaborate with key banking partners to introduce exclusive debit card deals.
- Since most customers choose one month installment or shorter plans, the company should focus on promoting shorter installment options.