Business Case: Target SQL

Target Corporation is an American retail corporation that operates a chain of discount department stores and hypermarkets, headquartered in Minneapolis, Minnesota. It is the seventh-largest retailer in the United States.Target is notable for its focus on upscale, trend-forward merchandise at lower costs.Its stores typically sell general merchandise—including clothing, household goods, electronics, toys, and more—as well as groceries.Its main competitors include Walmart and Amazon.

PROBLEM STATEMENT

This business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

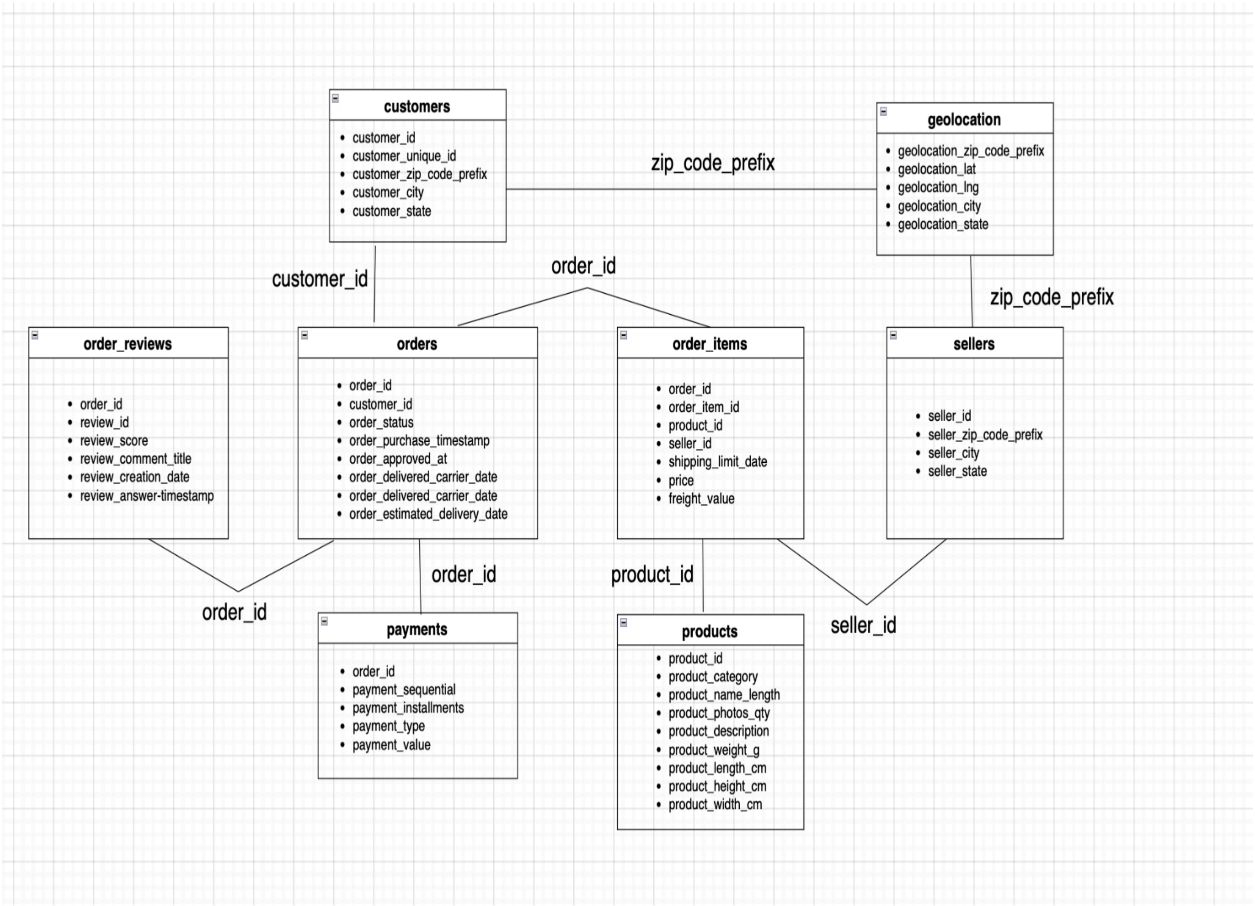
GOAL

Analyzing the given dataset to extract valuable insights:

- Evaluate Operational Efficiency
- Understand Customer Behavior
- Optimize Product Performance
- Enhance Customer Satisfaction
- Drive Data-Driven Decision Making

Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

ER-Diagram



The schema represents Target Brazil's e-commerce operations, detailing the order lifecycle, customer information, product details, payment data, and logistics.

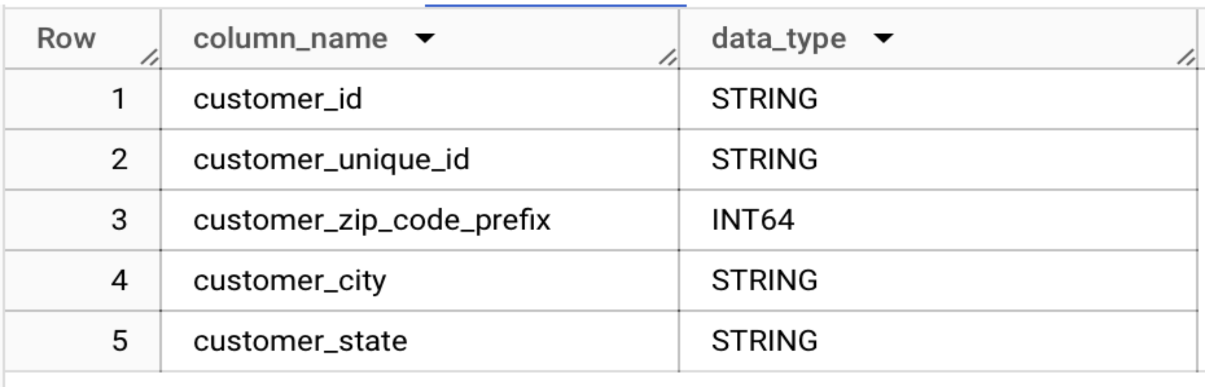
Entity Relationships:

- **Customers ↔ Orders**: Linked via `customer_id` (Customer purchases)
- **Orders ↔ Order Items**: Linked via `order_id` (Each order can have multiple items)
- **Orders ↔ Order Reviews**: Linked via `order_id` (Customer feedback for each order)

- **Orders ↔ Payments**: Linked via `order_id` (Multiple payment installments for a single order)
- **Order Items ↔ Products**: Linked via `product_id` (Product characteristics in each order)
- **Order Items ↔ Sellers**: Linked via `seller_id` (Identify the seller for each product)
- **Customers ↔ Geolocation**: Linked via `zip_code_prefix` (Customer location mapping)
- **Sellers ↔ Geolocation**: Linked via `zip_code_prefix` (Seller location mapping)

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

```
select column_name, data_type
from `sql-scaler-450218.Target.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'customers'
```



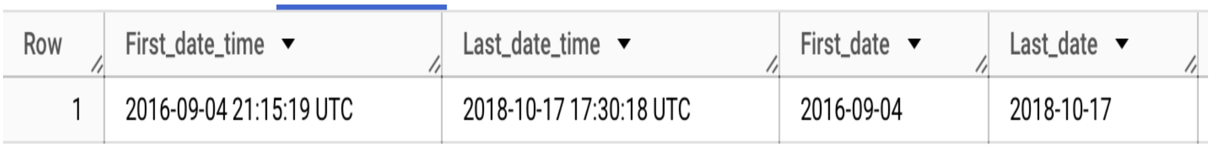
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

ANALYSIS:

The customers table contains customer-related information

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

```
select min(timestamp(order_purchase_timestamp)) as
First_date_time,
            max(timestamp(order_purchase_timestamp)) as
Last_date_time,
           extract(date from
min(timestamp(order_purchase_timestamp))) as First_date,
            extract(date from
max(timestamp(order_purchase_timestamp))) as Last_date
from  `Target.orders`
```



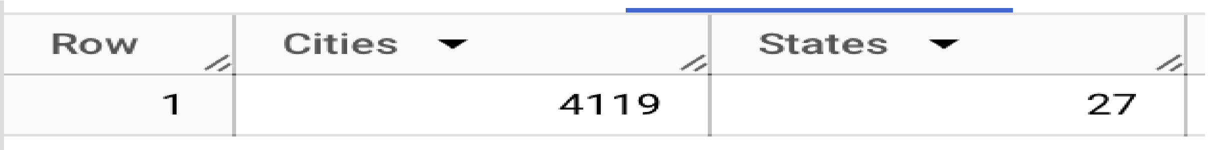
Row	First_date_time	Last_date_time	First_date	Last_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	2016-09-04	2018-10-17

ANALYSIS:

- The first order was placed on 4th September 2016.
- The last order was placed on 17th October 2018.

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

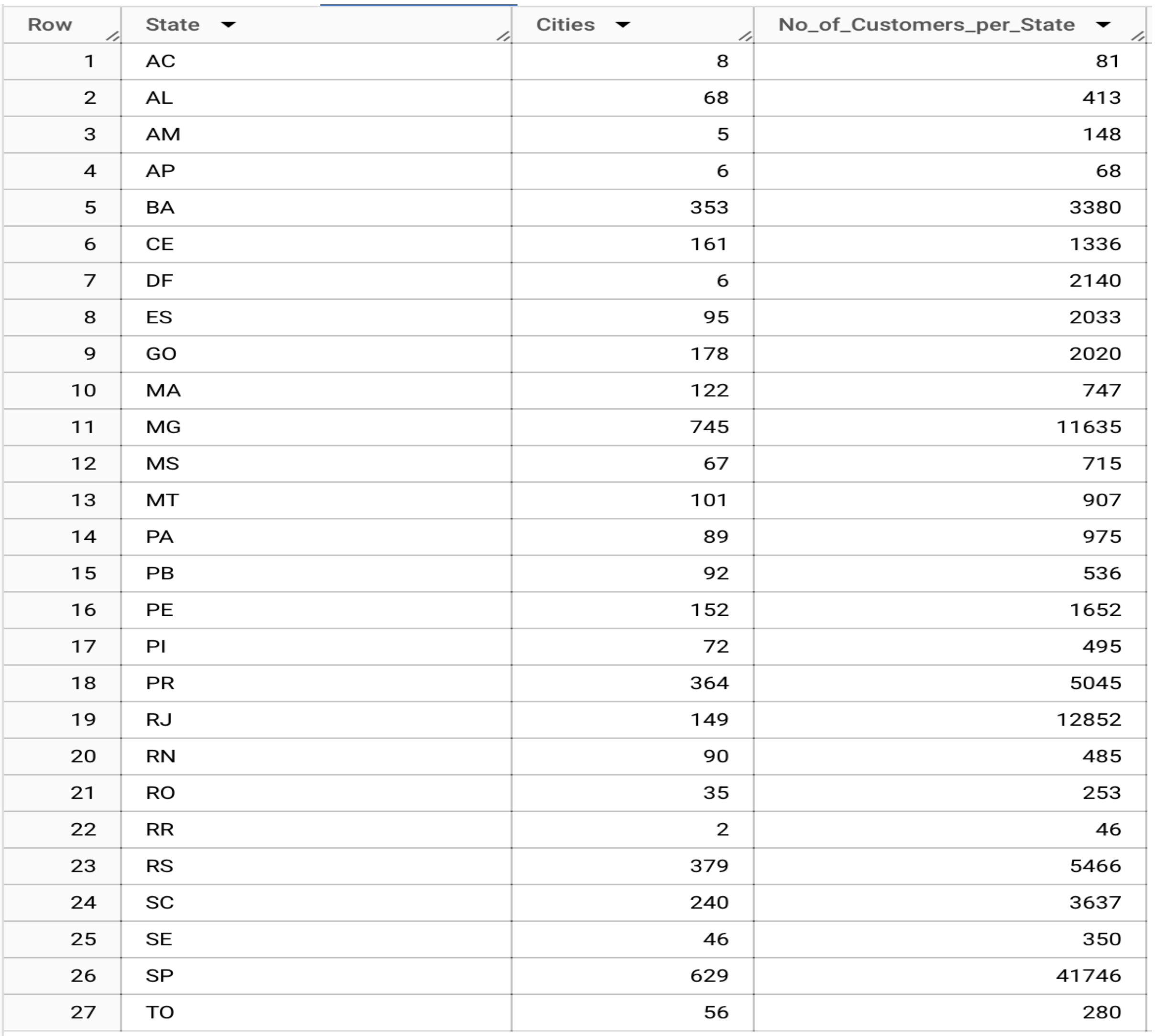
```
select count(distinct customer_city) as Cities,
            count(distinct customer_state) as States
from `Target.customers`
```



Row	Cities	States
1	4119	27

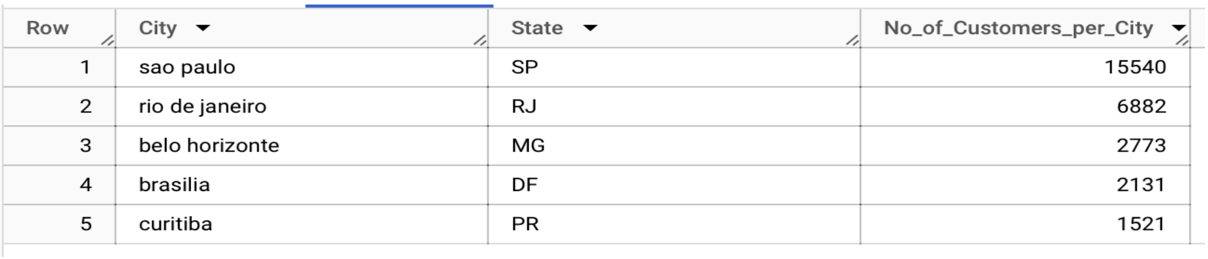
No of Customers per city per state

```
select customer_state as State,
           count(distinct customer_city) as Cities,
           count(distinct customer_id) as No_of_Customers_per_State,
 from `Target.customers`
 group by 1
```



Row	State	Cities	No_of_Customers_per_State
1	AC	8	81
2	AL	68	413
3	AM	5	148
4	AP	6	68
5	BA	353	3380
6	CE	161	1336
7	DF	6	2140
8	ES	95	2033
9	GO	178	2020
10	MA	122	747
11	MG	745	11635
12	MS	67	715
13	MT	101	907
14	PA	89	975
15	PB	92	536
16	PE	152	1652
17	PI	72	495
18	PR	364	5045
19	RJ	149	12852
20	RN	90	485
21	RO	35	253
22	RR	2	46
23	RS	379	5466
24	SC	240	3637
25	SE	46	350
26	SP	629	41746
27	TO	56	280

```
select c.customer_city as City,
      c.customer_state as State,
      count(distinct o.order_id) as No_of_Customers_per_City
from `Target.customers` c
join `Target.orders` o on o.customer_id = c.customer_id
group by 1,2
order by 3 desc
limit 5
```



Row	City	State	No_of_Customers_per_City
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521

ANALYSIS:

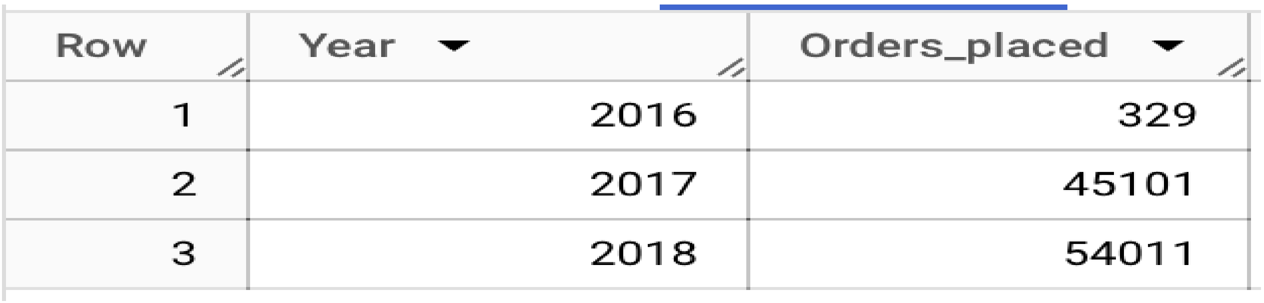
- The query indicates customers are broadly from **4119 cities belonging to 27 states**.
- Maximum Customers are from ‘**SP**’ state, followed by ‘**RJ**’ and ‘**MG**’.
- Same ranking is followed by the states as well -
  - **Sao-paulo from** ‘**SP**’ state tops the chart.
  - **Rio De Janeiro from** ‘**RJ**’ comes in second.
  - **Belo Horizonte from** ‘**MG**’ ranks 3rd.

IN DEPTH EXPLORATION

----

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

```
select extract(year from order_purchase_timestamp) as Year,
       count(*) as Orders_placed
from `Target.orders`
group by 1
order by 1
```



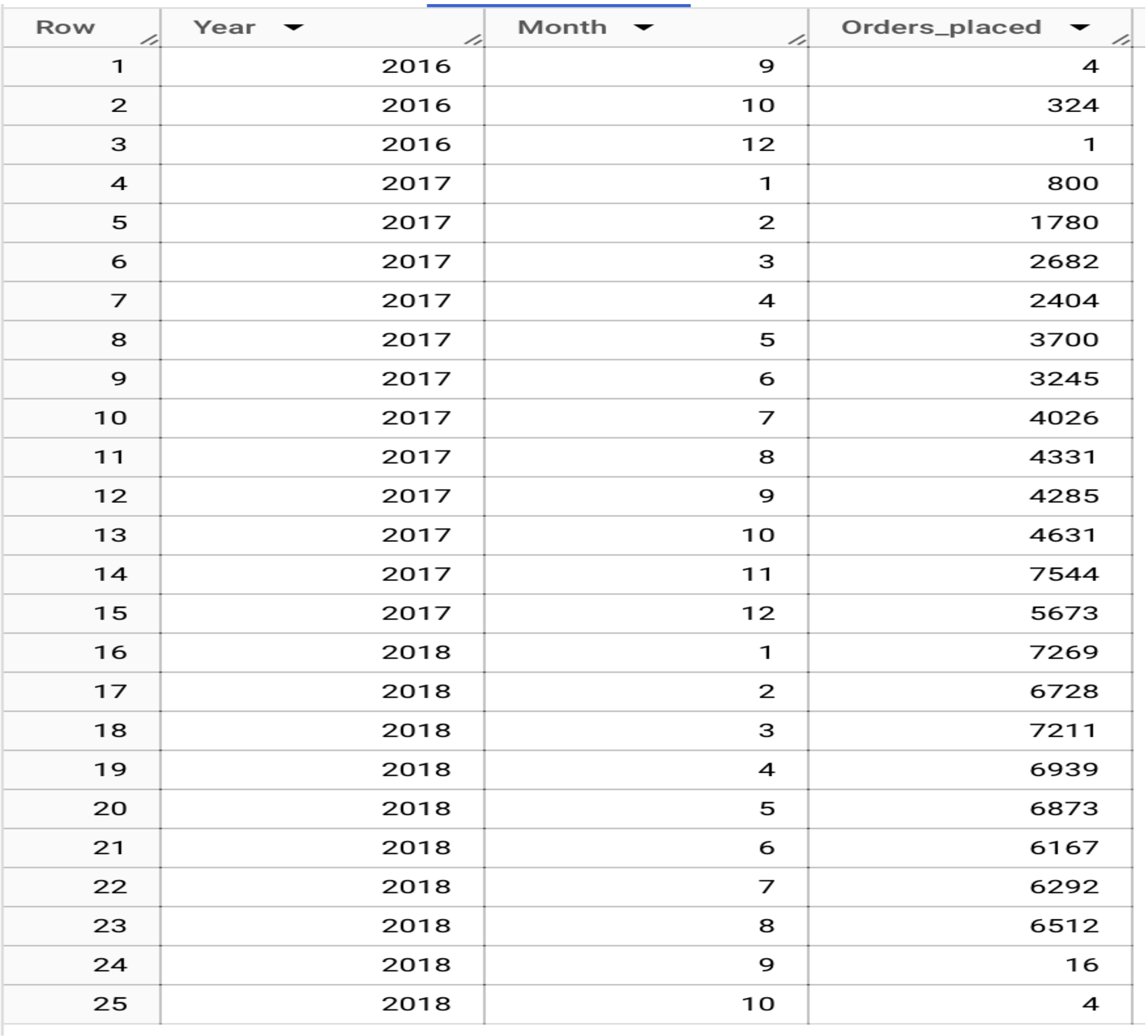
Row	Year	Orders_placed
1	2016	329
2	2017	45101
3	2018	54011

ANALYSIS:

The data depicts an **increasing upward trend** in the number of orders being placed every year.

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

```
select extract(year from order_purchase_timestamp) as Year,
       extract(month from order_purchase_timestamp) as Month,
       count(*) as Orders_placed
from `Target.orders`
group by 1,2
order by 1,2
```

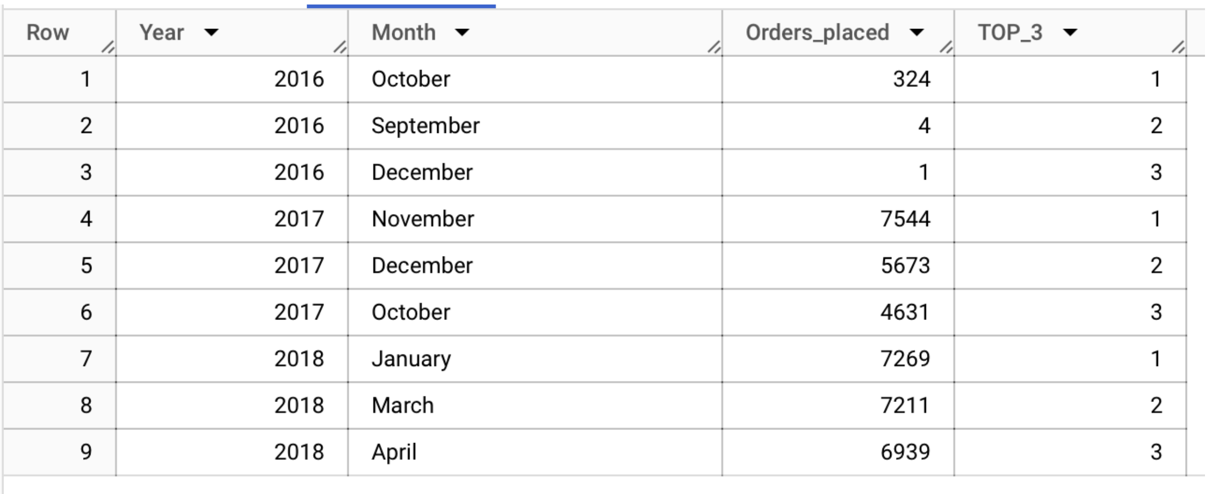


Row	Year	Month	Orders_placed
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728
18	2018	3	7211
19	2018	4	6939
20	2018	5	6873
21	2018	6	6167
22	2018	7	6292
23	2018	8	6512
24	2018	9	16
25	2018	10	4

Top 3 months for orders_placed Year Wise

```
with data_distribution as
(
select extract(year from order_purchase_timestamp) as Year,
     FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) as
Month,
     count(*) as Orders_placed
from `Target.orders`
group by 1,2
order by 1,2 desc
)
```

```
select *
from
(
select *,
      dense_rank()over(partition by d.Year order by Orders_placed
desc) as TOP_3
from data_distribution d
) e
where TOP_5 <=3
order by 1
```



Row	Year	Month	Orders_placed	TOP_3
1	2016	October	324	1
2	2016	September	4	2
3	2016	December	1	3
4	2017	November	7544	1
5	2017	December	5673	2
6	2017	October	4631	3
7	2018	January	7269	1
8	2018	March	7211	2
9	2018	April	6939	3

ANALYSIS:

- **Orders increased** significantly from 2016 to 2018.
- In **2016**, there were very **few orders**, **indicating the initial phase of operations**.
- **2017** showed **substantial growth**, with a peak in November (7,544 orders), likely due to promotional events like Black Friday.
- **2018 continued strong growth** with the highest orders in January (7,269 orders).

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

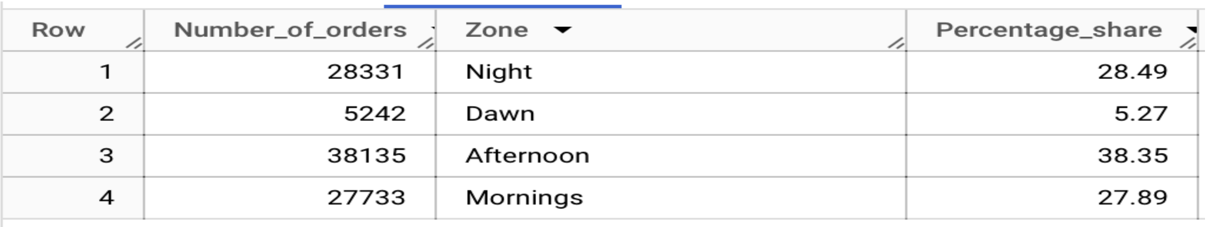
   0-6 hrs : Dawn

   7-12 hrs : Mornings

   13-18 hrs : Afternoon

   19-23 hrs : Night

```
with zone_cte as
(
select count(*) as Number_of_orders,
      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 'Mornings'
           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 Zone
from `Target.orders`
group by 2
order by 1
)
select *,
      (z.Number_of_orders/(select count(*) from
`Target.orders`))*100 as Percentage_share
from zone_cte z
```



Row	Number_of_orders	Zone	Percentage_share
1	28331	Night	28.49
2	5242	Dawn	5.27
3	38135	Afternoon	38.35
4	27733	Mornings	27.89

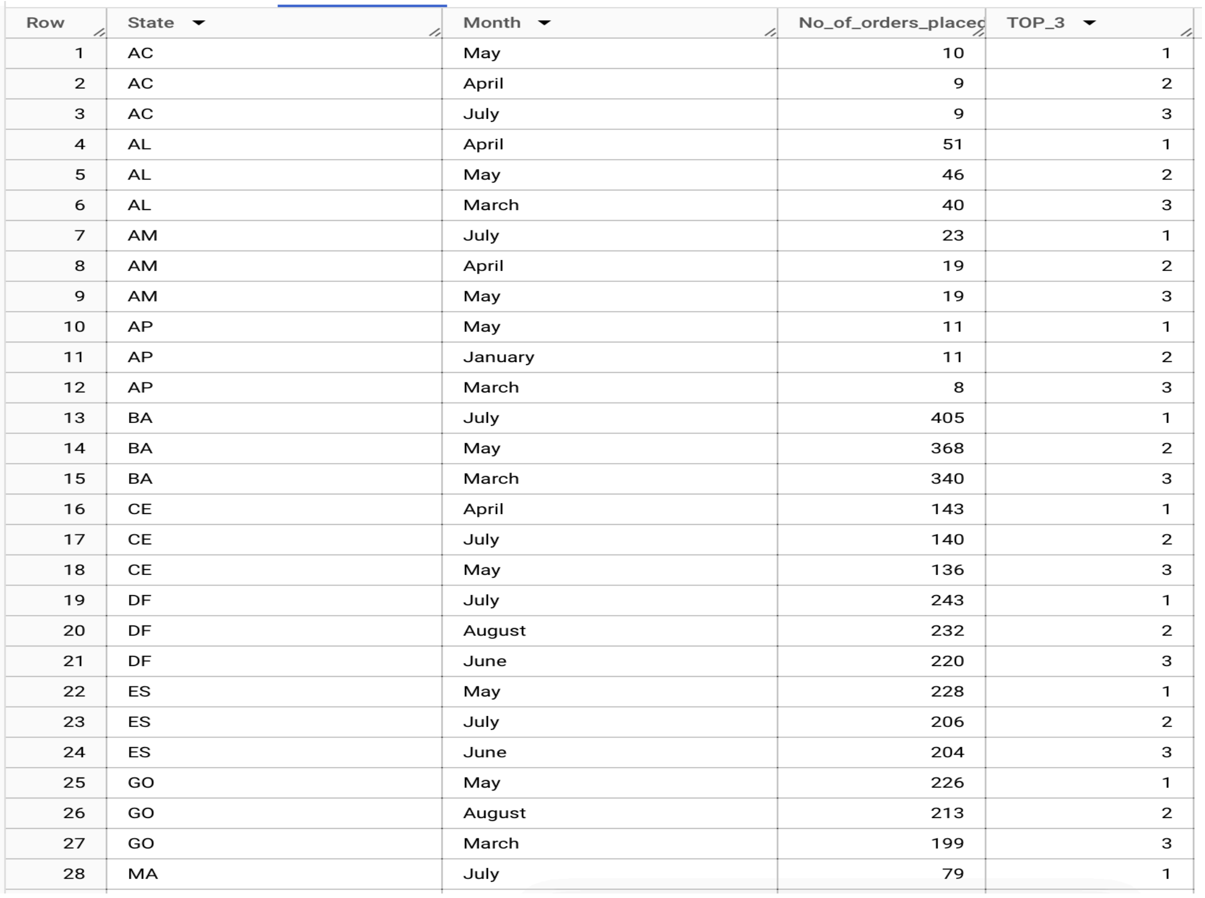
ANALYSIS:

- **Afternoon Dominates**: **38% of orders** occur in the afternoon, making it the most active period.This suggests a preference for midday shopping, possibly due to breaks or post-lunch browsing.
- **Morning Surge**: **28% of orders** are placed in the morning, indicating strong early-day demand.This indicates customers placing orders before starting their day.
- **Night Activity**: **28% of orders** happen at night, showing that customers actively shop after typical work hours.This could reflect a second wave of online shopping post-dinner.
- **Low Dawn Orders**: Only **5% of orders** occur during dawn, suggesting minimal customer activity in the early morning hours.

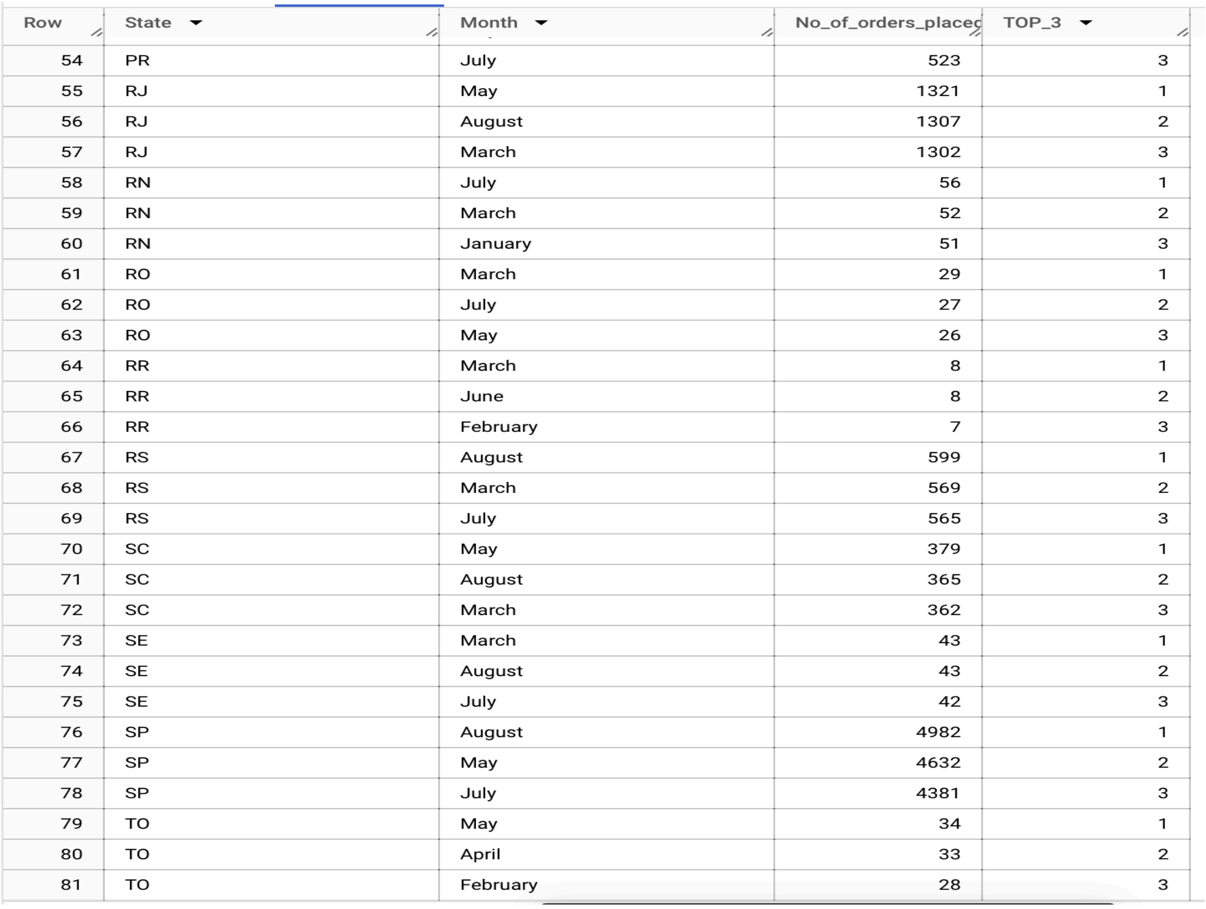
Evolution of E-commerce orders in the Brazil region.

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

```
With state_vs_month_cte as
(
select c.customer_state as State,
format_datetime("%B",o.order_purchase_timestamp) as Month,
count(o.customer_id) as No_of_orders_placed
from `Target.orders` o
join `Target.customers` c
on o.customer_id = c.customer_id
group by 1,2,extract(month from o.order_purchase_timestamp)
order by 1,extract(month from o.order_purchase_timestamp)
)
select * from
(select *,row_number()over(partition by State order by
No_of_orders_placed desc) as TOP_3
from state_vs_month_cte s) e
where TOP_3 <=3
order by 1
```



Row	State	Month	No_of_orders_placeq	TOP_3
1	AC	May	10	1
2	AC	April	9	2
3	AC	July	9	3
4	AL	April	51	1
5	AL	May	46	2
6	AL	March	40	3
7	AM	July	23	1
8	AM	April	19	2
9	AM	May	19	3
10	AP	May	11	1
11	AP	January	11	2
12	AP	March	8	3
13	BA	July	405	1
14	BA	May	368	2
15	BA	March	340	3
16	CE	April	143	1
17	CE	July	140	2
18	CE	May	136	3
19	DF	July	243	1
20	DF	August	232	2
21	DF	June	220	3
22	ES	May	228	1
23	ES	July	206	2
24	ES	June	204	3
25	GO	May	226	1
26	GO	August	213	2
27	GO	March	199	3
28	MA	July	79	1



Row	State	Month	No_of_orders_placeq	TOP_3
54	PR	July	523	3
55	RJ	May	1321	1
56	RJ	August	1307	2
57	RJ	March	1302	3
58	RN	July	56	1
59	RN	March	52	2
60	RN	January	51	3
61	RO	March	29	1
62	RO	July	27	2
63	RO	May	26	3
64	RR	March	8	1
65	RR	June	8	2
66	RR	February	7	3
67	RS	August	599	1
68	RS	March	569	2
69	RS	July	565	3
70	SC	May	379	1
71	SC	August	365	2
72	SC	March	362	3
73	SE	March	43	1
74	SE	August	43	2
75	SE	July	42	3
76	SP	August	4982	1
77	SP	May	4632	2
78	SP	July	4381	3
79	TO	May	34	1
80	TO	April	33	2
81	TO	February	28	3

ANALYSIS:

The highest performing states are:

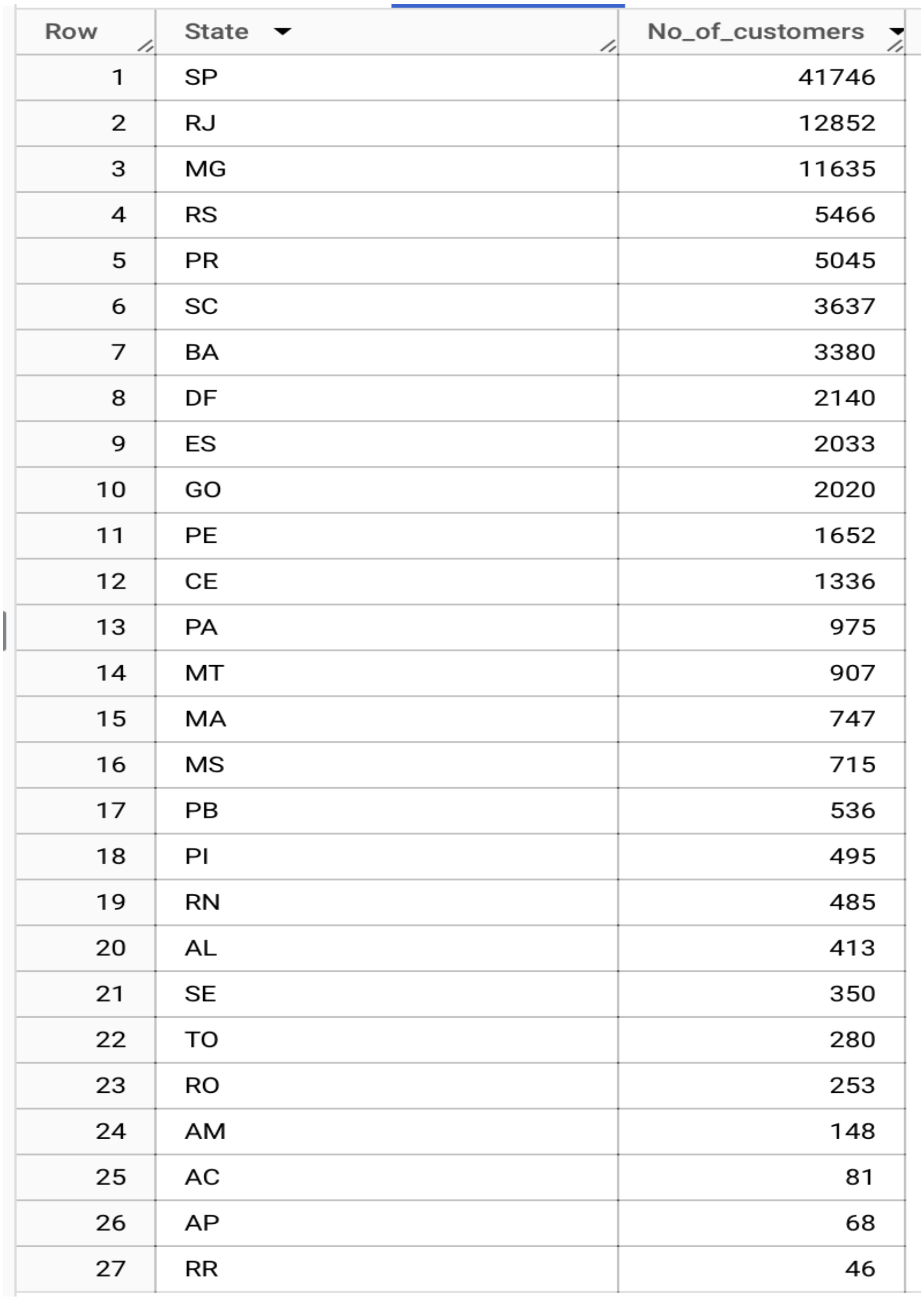
- SP :
  - August: 4,982 orders (Rank 1)
  - May: 4,632 orders (Rank 2)
  - July: 4,381 orders (Rank 3)
- BA :
  - July: 405 orders (Rank 1)
  - May: 368 orders (Rank 2)
  - March: 340 orders (Rank 3)
- RS :
  - August: 599 orders (Rank 1)
  - March: 569 orders (Rank 2)
  - July: 565 orders (Rank 3)

The lowest performing states are:

- RR :
  - March: 8 orders (Rank 1)
  - June: 8 orders (Rank 2)
  - February: 7 orders (Rank 3)
- RO :
  - March: 29 orders (Rank 1)
  - July: 27 orders (Rank 2)
  - May: 26 orders (Rank 3)
- TO :
  - May: 34 orders (Rank 1)
  - April: 33 orders (Rank 2)
  - February: 28 orders (Rank 3)

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

```
select customer_state as State, count(customer_id) as
No_of_customers
from `Target.customers`
group by customer_state
order by 2 desc
```



Row	State	No_of_customers
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

ANALYSIS:

Top 5 States by Number of Customers

1. **SP** – 41,746 customers
2. **RJ** – 12,852 customers
3. **MG** – 11,635 customers
4. **RS** – 5,466 customers
5. **PR** – 5,045 customers

- SP has a significant lead with over 41k customers, accounting for a substantial portion of the total customer base.
- RJ and MG are secondary hubs, but their customer volume is significantly lower compared to SP.
- RS and PR also exhibit strong customer bases, suggesting these regions are key for business growth.

Bottom 5 States by Number of Customers

1. **RR** - 46 customers
2. **AP** - 68 customers
3. **AC** - 81 customers
4. **AM** - 148 customers
5. **RO** - 253 customers

- **Northern** states like **RR, AP, and AC** have **very low customer engagement**, highlighting a potential **service gap**.
- These states combined account for **less than 1%** of the total customer base.

—---

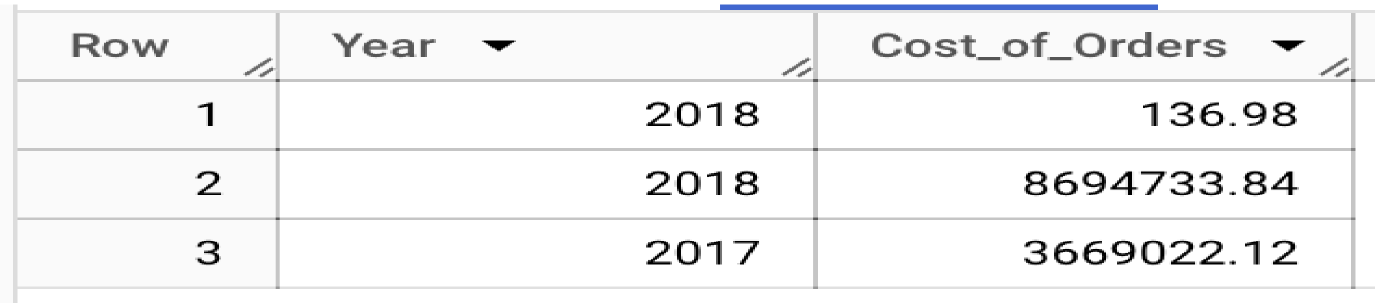
Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

—---

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

```
with orders_total as
(
select extract(year from o.order_purchase_timestamp) as Year,
       round(sum(p.payment_value),2) as Cost_of_Orders,
from `Target.payments` p
join `Target.orders` o on o.order_id = p.order_id
where extract(year from o.order_purchase_timestamp) in
(2017,2018)
and extract(month from o.order_purchase_timestamp) in
(1,2,3,4,5,6,7,8)
group by 1
order by 1
),
first_year as
(
select ord.Year,
         ord.Cost_of_Orders
from orders_total ord
where ord.Year = 2017
),
second_year as
(
select ord.Year,
        ord.Cost_of_Orders
from orders_total ord
where ord.Year = 2018)
```

```
select * from first_year
UNION ALL
Select s.year,
round(((s.Cost_of_Orders-f.Cost_of_Orders)*100)/f.Cost_of_Orde
rs),2)
from
first_year f
join second_year s
on f.year<s.Year
UNION ALL
select * from second_year
```



Row	Year	Cost_of_Orders
1	2018	136.98
2	2018	8694733.84
3	2017	3669022.12

ANALYSIS:

- The data indicates an **increase of 137%** in the yearly cost of orders in 2018 as compared to 2017.

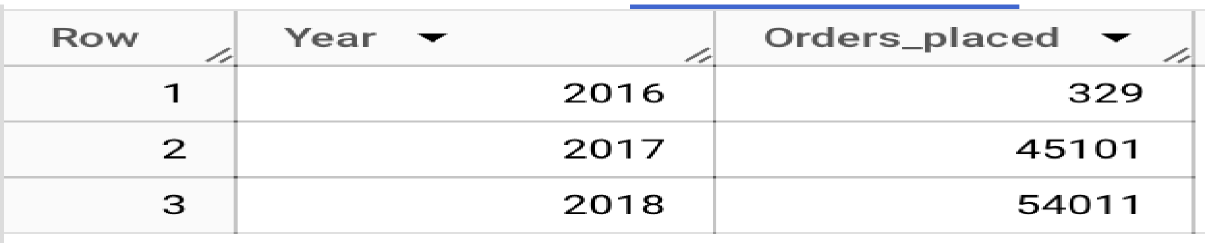
Cost_of_orders in 2017 = 3669022

Cost_of_orders in 2018 = 8694734

Increase in cost_of_orders = 8694734 - 3669022 = 5025712

% change = (5025712 / 3669022) * 100 = 136.9%

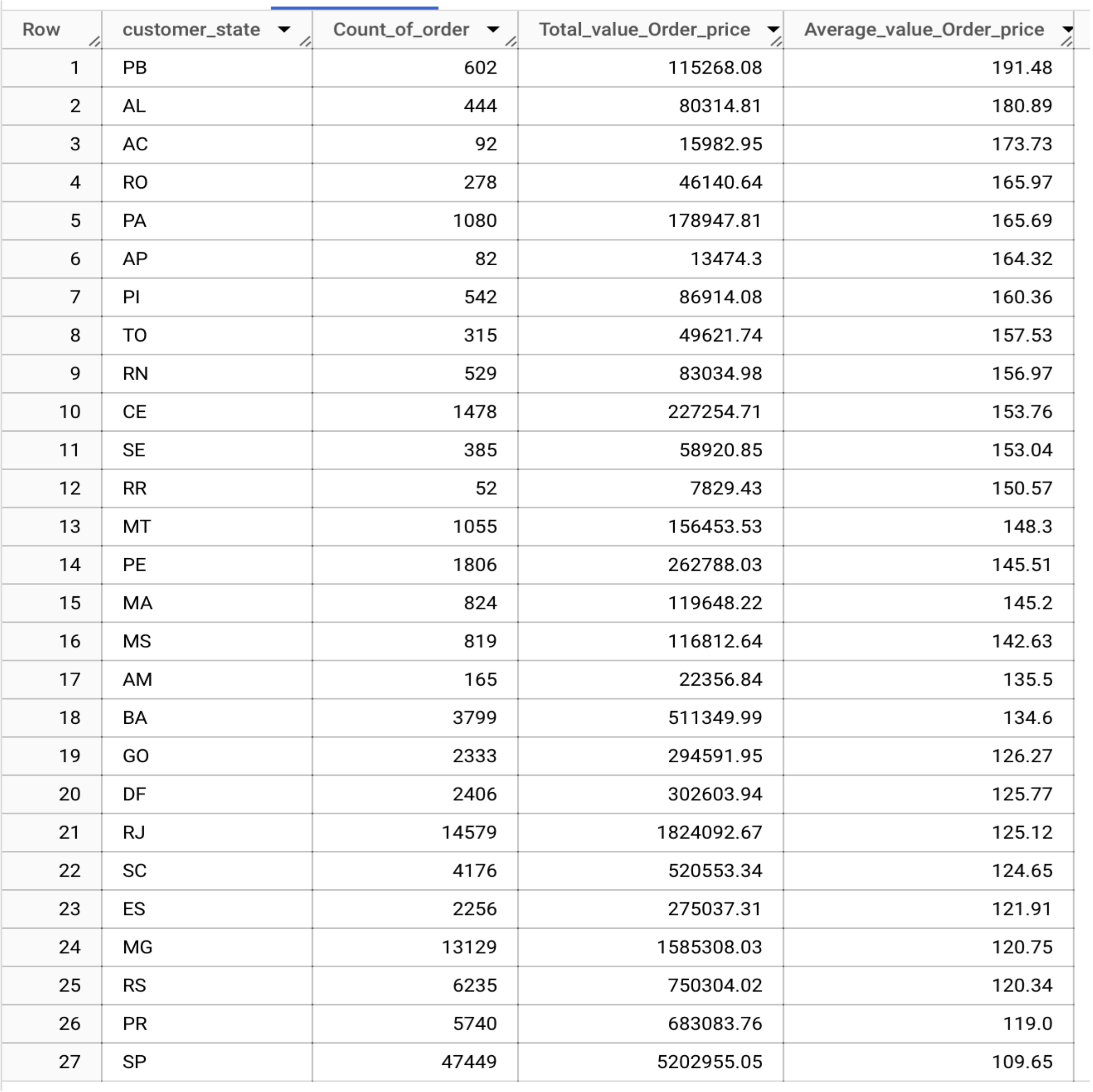
- This positive change owes to the increase in number of orders during 2018 as compared to 2017 as calculated earlier.



Row	Year	Orders_placed
1	2016	329
2	2017	45101
3	2018	54011

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

```
select c.customer_state,
      count(c.customer_state) as Count_of_order,
      round(sum(oi.price),2) as Total_value_Order_price,
      round(avg(oi.price),2) as Average_value_Order_price
from `Target.order_items` oi
join `Target.orders` o on o.order_id=oi.order_id
join `Target.customers` c on c.customer_id=o.customer_id
group by 1
order by 4 desc
```



Row	customer_state	Count_of_order	Total_value_Order_price	Average_value_Order_price
1	PB	602	115268.08	191.48
2	AL	444	80314.81	180.89
3	AC	92	15982.95	173.73
4	RO	278	46140.64	165.97
5	PA	1080	178947.81	165.69
6	AP	82	13474.3	164.32
7	PI	542	86914.08	160.36
8	TO	315	49621.74	157.53
9	RN	529	83034.98	156.97
10	CE	1478	227254.71	153.76
11	SE	385	58920.85	153.04
12	RR	52	7829.43	150.57
13	MT	1055	156453.53	148.3
14	PE	1806	262788.03	145.51
15	MA	824	119648.22	145.2
16	MS	819	116812.64	142.63
17	AM	165	22356.84	135.5
18	BA	3799	511349.99	134.6
19	GO	2333	294591.95	126.27
20	DF	2406	302603.94	125.77
21	RJ	14579	1824092.67	125.12
22	SC	4176	520553.34	124.65
23	ES	2256	275037.31	121.91
24	MG	13129	1585308.03	120.75
25	RS	6235	750304.02	120.34
26	PR	5740	683083.76	119.0
27	SP	47449	5202955.05	109.65

ANALYSIS:

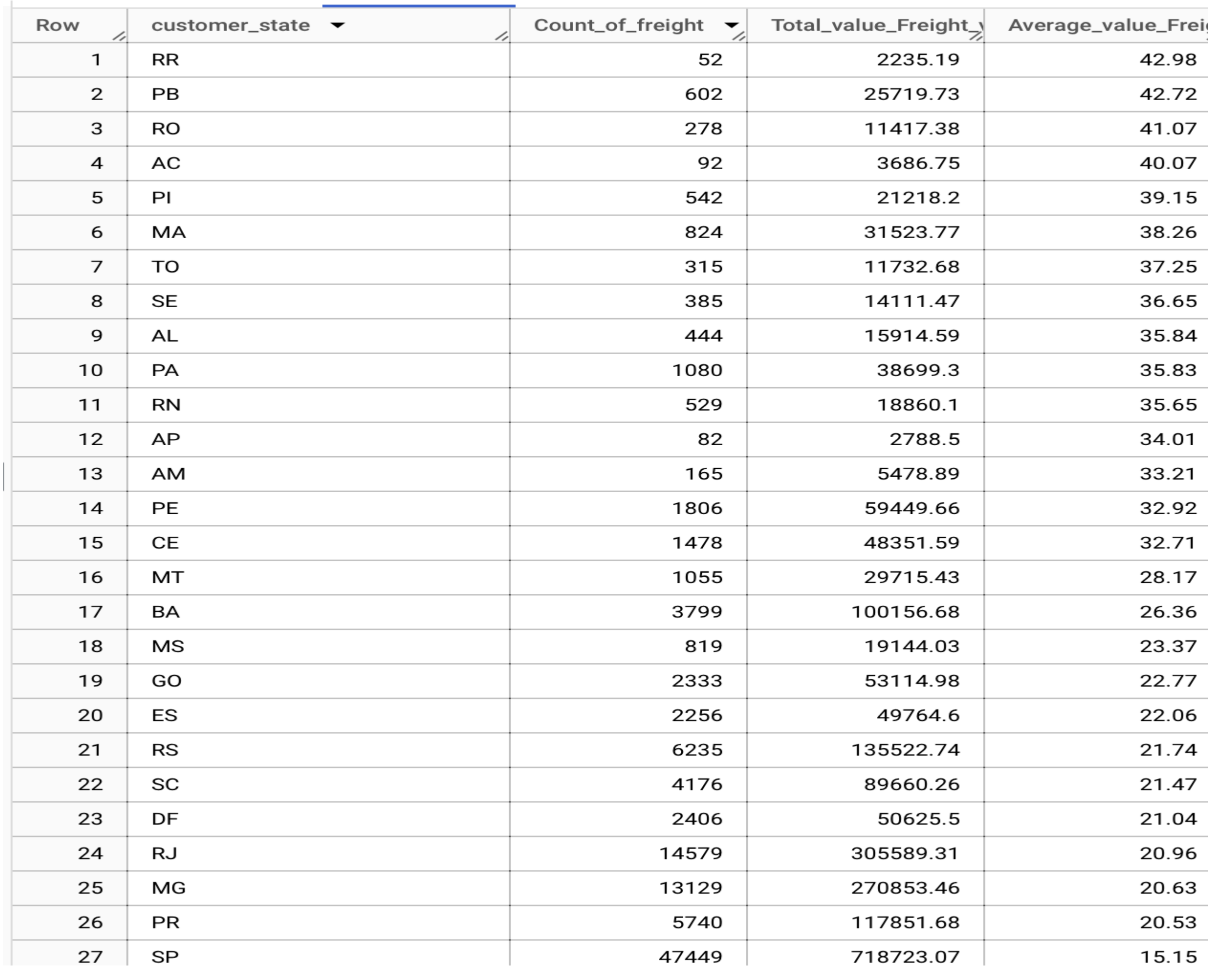
Average Order Value (AOV) – High vs. Low:

- **Highest AOV:**
  - PB: **$191.48**.
  - AL: **$180.89**.
- **Lowest AOV:**
  - **SP**: **$109.65** – Despite being the largest by volume, SP has the lowest average order value.

- PB, AL have higher AOV but fewer orders overall.
- SP, RJ dominate in volume but have lower AOV.

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

```
select c.customer_state,
 count(c.customer_state) as Count_of_freight,
 round(sum(oi.freight_value),2) as Total_value_Freight_value,
 round(avg(oi.freight_value),2) as Average_value_Freight_value
from `Target.order_items` oi
join `Target.orders` o on o.order_id=oi.order_id
join `Target.customers` c on c.customer_id=o.customer_id
group by 1
order by 4 desc
```



Row	customer_state	Count_of_freight	Total_value_Freight_	Average_value_Frei
1	RR	52	2235.19	42.98
2	PB	602	25719.73	42.72
3	RO	278	11417.38	41.07
4	AC	92	3686.75	40.07
5	PI	542	21218.2	39.15
6	MA	824	31523.77	38.26
7	TO	315	11732.68	37.25
8	SE	385	14111.47	36.65
9	AL	444	15914.59	35.84
10	PA	1080	38699.3	35.83
11	RN	529	18860.1	35.65
12	AP	82	2788.5	34.01
13	AM	165	5478.89	33.21
14	PE	1806	59449.66	32.92
15	CE	1478	48351.59	32.71
16	MT	1055	29715.43	28.17
17	BA	3799	100156.68	26.36
18	MS	819	19144.03	23.37
19	GO	2333	53114.98	22.77
20	ES	2256	49764.6	22.06
21	RS	6235	135522.74	21.74
22	SC	4176	89660.26	21.47
23	DF	2406	50625.5	21.04
24	RJ	14579	305589.31	20.96
25	MG	13129	270853.46	20.63
26	PR	5740	117851.68	20.53
27	SP	47449	718723.07	15.15

ANALYSIS:

Average Freight Value (AFV) – High vs. Low:

- **Highest AFV:**
  - **RR : $42.98** – Smallest volume (52 shipments) but the highest freight cost, likely due to geographic remoteness.
- **Lowest AOV:**
  - **SP : $15.15** – Economies of scale from high shipment volumes.

- SP and RJ maintain low average freight values despite their large shipment sizes.
- Northern regions like RR and PB show a significant increase in average freight cost relative to their shipment count.

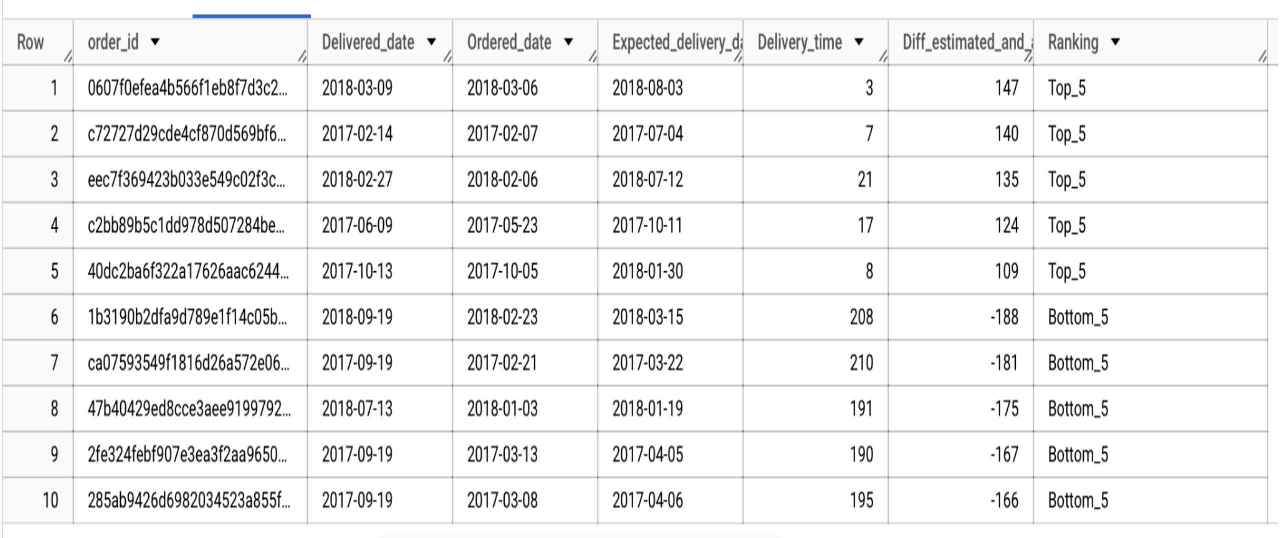
—---

Analysis based on sales, freight and delivery time.

—---

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.

```
with deliverytime_cte as
(select order_id,
      extract(date from order_delivered_customer_date) as Delivered_date,
      extract(date from order_purchase_timestamp) as Ordered_date,
      extract(date from order_estimated_delivery_date) as
Expected_delivery_date,
      date_diff(extract(date from
order_delivered_customer_date),extract(date from
order_purchase_timestamp),day) as Delivery_time,
      date_diff(extract(date from
order_estimated_delivery_date),extract(date from
order_delivered_customer_date),day) as Diff_estimated_and_actual_delivery
from `Target.orders`
where order_delivered_customer_date is not null
order by 6 desc)
(select *,
      'Top_5' as Ranking
from deliverytime_cte
limit 5)
UNION ALL
(select *,
      'Bottom_5' as Ranking
from deliverytime_cte
order by 6
limit 5)
```



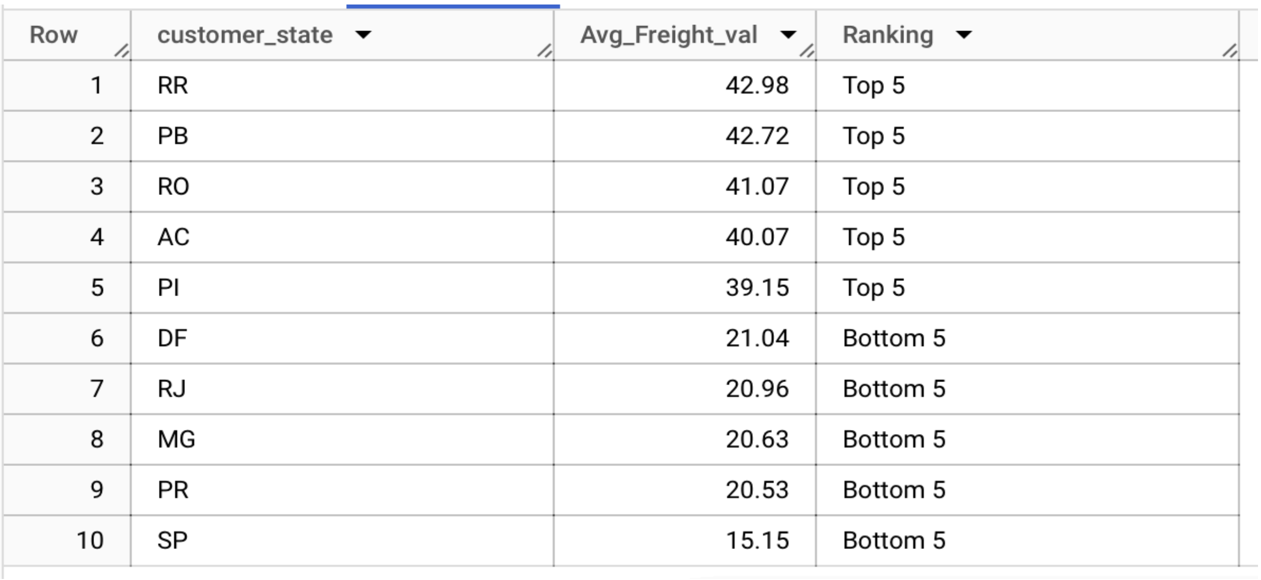
Row	order_id ▾	Delivered_date ▾	Ordered_date ▾	Expected_delivery_d	Delivery_time ▾	Diff_estimated_and_	Ranking ▾
1	0607f0efea4b566f1eb8f7d3c2...	2018-03-09	2018-03-06	2018-08-03	3	147	Top_5
2	c72727d29cde4cf870d569bf6...	2017-02-14	2017-02-07	2017-07-04	7	140	Top_5
3	eec7f369423b033e549c02f3c...	2018-02-27	2018-02-06	2018-07-12	21	135	Top_5
4	c2bb89b5c1dd978d507284be...	2017-06-09	2017-05-23	2017-10-11	17	124	Top_5
5	40dc2ba6f322a17626aac6244...	2017-10-13	2017-10-05	2018-01-30	8	109	Top_5
6	1b3190b2dfa9d789e1f14c05b...	2018-09-19	2018-02-23	2018-03-15	208	-188	Bottom_5
7	ca07593549f1816d26a572e06...	2017-09-19	2017-02-21	2017-03-22	210	-181	Bottom_5
8	47b40429ed8cce3aee9199792...	2018-07-13	2018-01-03	2018-01-19	191	-175	Bottom_5
9	2fe324febf907e3ea3f2aa9650...	2017-09-19	2017-03-13	2017-04-05	190	-167	Bottom_5
10	285ab9426d6982034523a855f...	2017-09-19	2017-03-08	2017-04-06	195	-166	Bottom_5

ANALYSIS:

- **Top 5 Fastest Deliveries:**
  - The **fastest delivery was completed in 3 days** (Order ID: 0607f0...), which was 147 days earlier than the estimated date.
  - All top deliveries were completed at least 109 days ahead of schedule, highlighting efficiency in handling these orders.
  - The average difference for the Top 5 deliveries was approximately +131 days (early).
- **Bottom 5 Slowest Deliveries:**
  - The **slowest delivery took 210 days**, which was 188 days late beyond the estimated delivery date.
  - All bottom deliveries were at least 166 days late, suggesting **major fulfillment delays**.
  - The average difference for the Bottom 5 deliveries was approximately -180 days (late).

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

```
with Top_5_cte as
(
select customer_state,round(average_freight_value,2) as Avg_Freight_val
,'Top 5' as Ranking
from
(select c.customer_state,avg(oi.freight_value) as
average_freight_value,
      dense_rank()over(order by avg(oi.freight_value)desc) as ranking
from `Target.order_items` oi
join `Target.orders` o on o.order_id = oi.order_id
join `Target.customers` c on c.customer_id = o.customer_id
group by c.customer_state
)tb1
where ranking <=5),
Bottom_5_cte as
(
select customer_state,round(average_freight_value,2) as Avg_Freight_val
,'Bottom 5' as Ranking
from
(select c.customer_state,avg(oi.freight_value) as
average_freight_value,
      dense_rank()over(order by avg(oi.freight_value) asc) as ranking
from `Target.order_items` oi
join `Target.orders` o on o.order_id = oi.order_id
join `Target.customers` c on c.customer_id = o.customer_id
group by c.customer_state
)tb2
where ranking <=5
order by 2 asc
)
select * from Top_5_cte
union ALL
select * from Bottom_5_cte
order by 2 desc
```



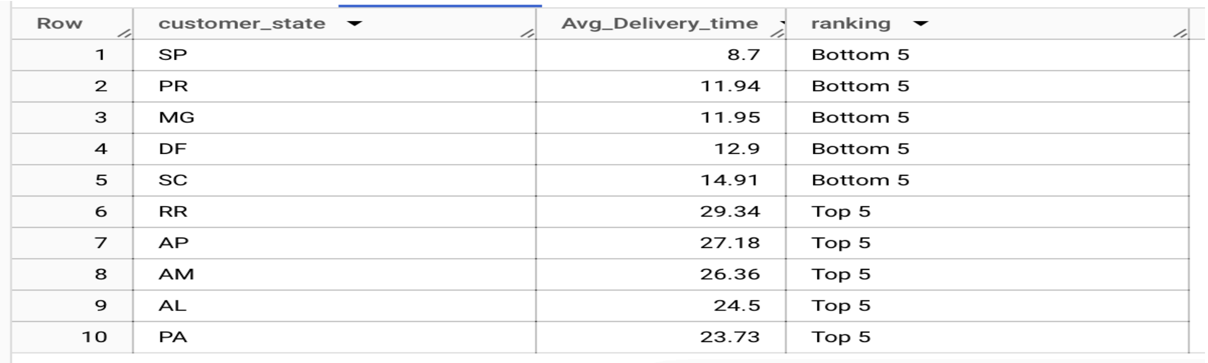
Row	customer_state	Avg_Freight_val	Ranking
1	RR	42.98	Top 5
2	PB	42.72	Top 5
3	RO	41.07	Top 5
4	AC	40.07	Top 5
5	PI	39.15	Top 5
6	DF	21.04	Bottom 5
7	RJ	20.96	Bottom 5
8	MG	20.63	Bottom 5
9	PR	20.53	Bottom 5
10	SP	15.15	Bottom 5

ANALYSIS:

- **Top 5 States (Highest Freight Costs)**:
  - **RR, PB, RO, AC, PI** have the **highest average freight values**, ranging from 39.15 to 42.98.
  - These states **may be remote areas or difficult to access, leading to higher transportation costs**.
- **Bottom 5 States (Lowest Freight Costs):**
  - **DF, RJ, MG, PR, SP** have the **lowest freight values**, ranging from 15.15 to 21.04.
  - These states might be **major urban hubs or well-connected regions, resulting in lower logistics expenses**.

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

```
with delivery_time_calc as
(select o.order_id,
       o.customer_id,
       c.customer_state,
       extract(date from o.order_purchase_timestamp) as Ordered_date,
extract(date from o.order_delivered_customer_date) as Delivered_date,
date_diff(extract(date from o.order_delivered_customer_date),
extract(date from o.order_purchase_timestamp),day) as Delivery_time
from `Target.orders` o
join `Target.customers` c
on o.customer_id = c.customer_id
where extract(date from o.order_delivered_customer_date) is not null
order by 4,5,6
)
(select customer_state,round(avg(Delivery_time),2) as
Avg_Delivery_time,'Top 5' as Ranking
from delivery_time_calc
group by customer_state
order by 2 desc
limit 5)
UNION ALL
(select customer_state,round(avg(Delivery_time),2),'Bottom 5' as Ranking
from delivery_time_calc
group by customer_state
order by 2 asc
limit 5)
```



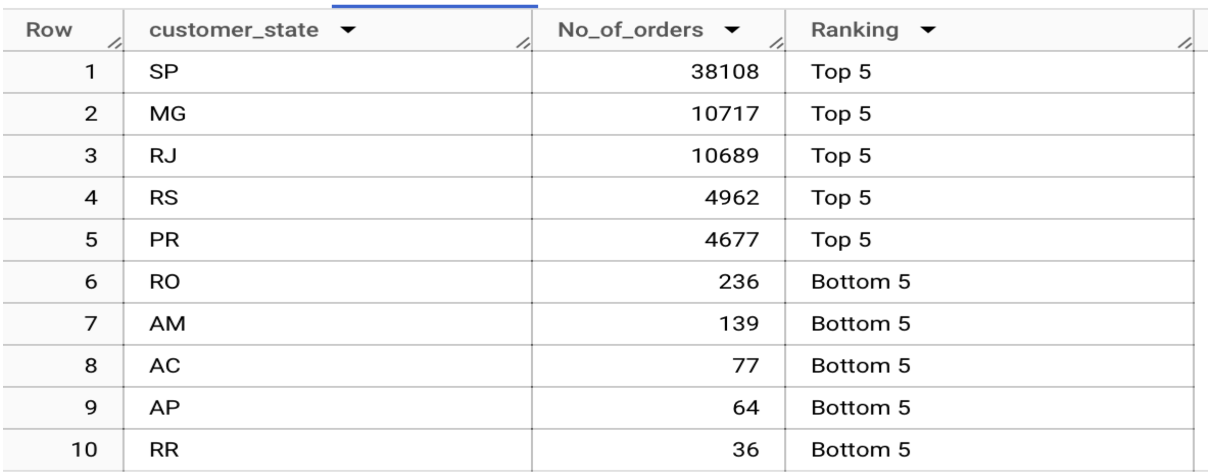
Row	customer_state	Avg_Delivery_time	ranking
1	SP	8.7	Bottom 5
2	PR	11.94	Bottom 5
3	MG	11.95	Bottom 5
4	DF	12.9	Bottom 5
5	SC	14.91	Bottom 5
6	RR	29.34	Top 5
7	AP	27.18	Top 5
8	AM	26.36	Top 5
9	AL	24.5	Top 5
10	PA	23.73	Top 5

ANALYSIS:

- **Top 5 states with Very Low Average Delivery Time**:
    - **SP, PR, MG, DF, SC** have the **lowest average delivery times** ranging from **9 to 15 days**.
    - These states have **better logistics** and **faster delivery networks**, likely due to their **urban locations** or **centralized infrastructure**.
    - **SP** has the quickest average delivery time, reflecting its position as a major hub.
- **Bottom 5 States with Very High Average Delivery Time:**
    - **RR, AP, AM, AL, PA** have the **highest average delivery times**, ranging from **24 to 29 days**.
    - These states are likely **remote or less connected regions**, contributing to longer delivery times.
    - **RR** experiences the slowest deliveries, aligning with its geographical position in northern Brazil and challenging transport routes.

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

```
(select c.customer_state,
      count(*) as No_of_orders,
      'Top 5'
from `Target.orders` o join `Target.customers` c on c.customer_id =
o.customer_id
where extract(date from o.order_delivered_customer_date) < extract(date
from o.order_estimated_delivery_date)
group by 1
order by 2 desc
limit 5)
UNION ALL
(select c.customer_state,
      count(*) as No_of_orders,
      'Bottom 5'
from `Target.orders` o join `Target.customers` c on c.customer_id =
o.customer_id
where extract(date from o.order_delivered_customer_date) < extract(date
from o.order_estimated_delivery_date)
group by 1
order by 2 asc
limit 5)
order by 2 desc
```



Row	customer_state	No_of_orders	Ranking
1	SP	38108	Top 5
2	MG	10717	Top 5
3	RJ	10689	Top 5
4	RS	4962	Top 5
5	PR	4677	Top 5
6	RO	236	Bottom 5
7	AM	139	Bottom 5
8	AC	77	Bottom 5
9	AP	64	Bottom 5
10	RR	36	Bottom 5

ANALYSIS:

- **Top 5 states with Highest orders** :
    - **SP, MG, RJ, RS, PR** have the highest orders placed.
    - These states represent **economic hubs** with **high population density** and **stronger logistical capabilities**, driving **higher order volumes**.
    - **SP** significantly outperforms other states, handling **over 3.5x** more orders than **MG** or **RJ.**
    - **MG and RJ are relatively close in order volume, reflecting a balanced demand across these regions.**

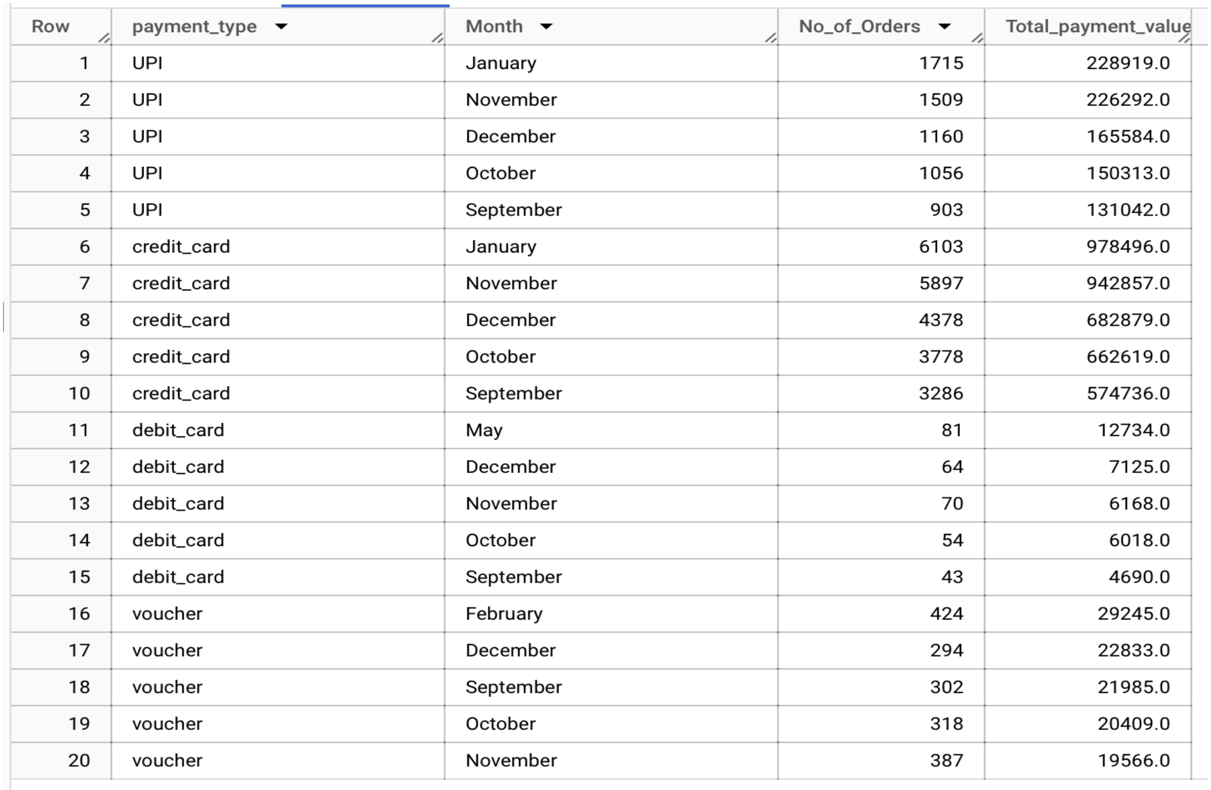
- **Bottom 5 States with Lowest Orders:**
    - **RO, AM, AC, AP, RR** have the **lowest Orders.**
    - These regions exhibit **very low order volumes,** especially **RR with only 36 orders.**
    - **Geographical isolation**, **lower population**, and **logistical challenges** likely contribute to **reduced customer demand**.
    - **AM and RO, while larger in area, still receive fewer orders,** indicating **limited market penetration.**

Analysis based on the payments:

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

```
with payments_cte as
(select p.payment_type,FORMAT_DATETIME("%B",
DATETIME(order_purchase_timestamp)) as Month,count(p.order_id) as
No_of_Orders,round(sum(payment_value)) as Total_payment_value,
      dense_rank()over(partition by p.payment_type order by
count(p.order_id))as rank_of_orders
from `Target.payments` p
join `Target.orders` o on o.order_id = p.order_id
group by 1,2)

select payment_type,Month,No_of_Orders,Total_payment_value
from payments_cte
where rank_of_orders <= 5
order by 1,3 desc
```



Row	payment_type	Month	No_of_Orders	Total_payment_value
1	UPI	January	1715	228919.0
2	UPI	November	1509	226292.0
3	UPI	December	1160	165584.0
4	UPI	October	1056	150313.0
5	UPI	September	903	131042.0
6	credit_card	January	6103	978496.0
7	credit_card	November	5897	942857.0
8	credit_card	December	4378	682879.0
9	credit_card	October	3778	662619.0
10	credit_card	September	3286	574736.0
11	debit_card	May	81	12734.0
12	debit_card	December	64	7125.0
13	debit_card	November	70	6168.0
14	debit_card	October	54	6018.0
15	debit_card	September	43	4690.0
16	voucher	February	424	29245.0
17	voucher	December	294	22833.0
18	voucher	September	302	21985.0
19	voucher	October	318	20409.0
20	voucher	November	387	19566.0

ANALYSIS:

By Total Value

- **Credit Card** is the **most popular** payment method by a **large margin**, contributing to **80%** of the total payment value.
- **UPI** is the **second-highest** but far behind credit cards, accounting for **19%** of the total payment value.
- **Debit Cards** show **low adoption**, contributing the **least** in both order volume and payment value.
- **Vouchers** play a **minor role** in transactions but have **moderate usage**.

Month vs payment Type

UPI:

- **Peak Month:** January - 1715 Orders
- **Trend:** Consistent decline from January to September.
- UPI usage is seasonal, with a spike at the beginning of the year and a gradual decrease over time.

Credit Card:

- **Peak Month**: January - 6103 orders
- **Trend:** Consistently high with November (5897 orders)
- Credit card payments are the dominant choice across all months, with a year-end boost likely due to holiday sales.

Debit Card:

- **Peak Month:** May - 81 orders
- **Trend:** Minimal with no clear pattern.
- Debit card usage is infrequent and shows low customer preference.

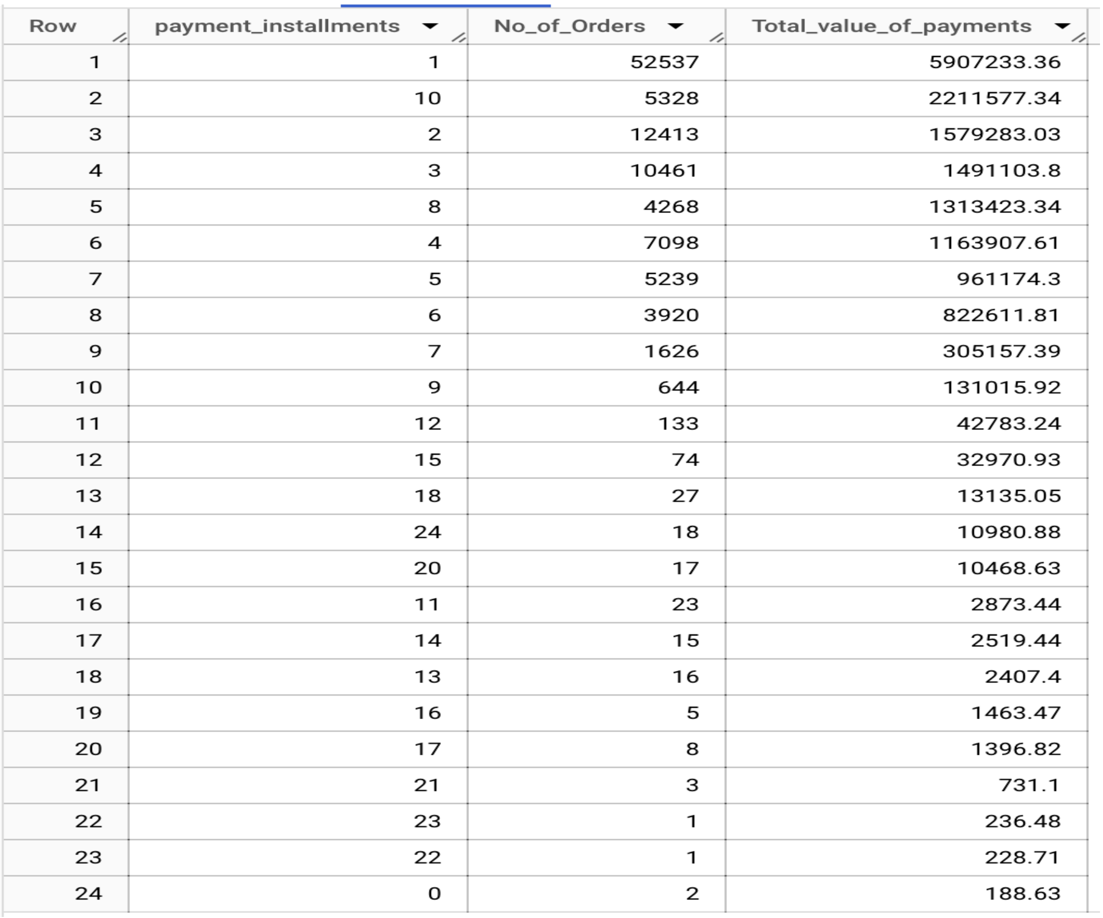
Voucher:

- **Peak Month:** February - 424 orders
- **Trend:** Steady but with a slight decline from February to November.

- Vouchers experience higher activity during the early year promotions but slow down later.

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

```
select
      Payment_installments,
      count(order_id) as No_of_Orders,
      sum(payment_value) as Total_value_of_payments
from `Target.payments`
where payment_value>0
group by 1
order by 3 desc
```



Row	payment_installments	No_of_Orders	Total_value_of_payments
1	1	52537	5907233.36
2	10	5328	2211577.34
3	2	12413	1579283.03
4	3	10461	1491103.8
5	8	4268	1313423.34
6	4	7098	1163907.61
7	5	5239	961174.3
8	6	3920	822611.81
9	7	1626	305157.39
10	9	644	131015.92
11	12	133	42783.24
12	15	74	32970.93
13	18	27	13135.05
14	24	18	10980.88
15	20	17	10468.63
16	11	23	2873.44
17	14	15	2519.44
18	13	16	2407.4
19	16	5	1463.47
20	17	8	1396.82
21	21	3	731.1
22	23	1	236.48
23	22	1	228.71
24	0	2	188.63

ANALYSIS:

- **Single Payment Dominance:**
    - **51.4% of the orders (52,537 out of 102,164)** are made with **1 installment**, contributing to 38.3% of the total payment value.
    - Customers **prefer paying in full rather than in multiple installments.**
- **High-Value Installments:**
    - **10 Installments account for 5,328 orders but contribute 14.4% of the total payment value**, making it the **second-highest payment method.**
    - 2 Installments also show a significant contribution with 12,413 orders.
- **Longer Installments Are Rare:**
    - **Installments beyond 12 are not common**, with orders dropping sharply.
    - **Orders with 22 and 23 installments have only 1 record each**, indicating very low adoption.
- **Mid-Range Preferences (3-10 Installments):**
    - **3-10 installments cover a substantial share of the orders and payments**, accounting for approximately 37% of total orders.
    - Customers may prefer mid-range payment flexibility over extended installments.

OBSERVATIONS

- **State vs Total Orders.**
  - **Top 5 States (SP, MG, RJ, RS, PR) contribute to the bulk of the orders**, with SP (38,108 orders) accounting for over 50% of the total volume.
  - **Bottom 5 States (RO, AM, AC, AP, RR) reflect low consumer activity,** with RR contributing the least (36 orders).
- **State vs Freight Charges**
  - **States with lower order volumes (such as RR, AP, and AC)** are likely to **incur higher freight costs per order due to distance and less frequent deliveries.**
  - **High-order states (SP, MG)** benefit from economies of scale and reduced average shipping costs.
- **State vs Customer Base**
  - **SP is the largest customer base, implying higher urban density and purchasing power.**
  - **States like RR and AP have limited customer penetration**, suggesting room for growth through customer acquisition campaigns.
- **Payment Insights**
  - **Payment Method Preferences**
    - UPI leads both order volume and payment value, particularly in high-order states.
    - Credit Cards are the second most popular method, especially for high-ticket items.
    - Debit Cards and Vouchers account for a small fraction of total transactions.
  - **Payment Installments**
    - Single payments dominate (over 50% of orders).
    - Installments between 3-10 months are popular, while 12+ month plans have minimal adoption.
  - **Seasonal Trends**
    - January, November, and December are the busiest months for both UPI and Credit Card payments.
    - September reflects moderate activity, suggesting potential for a pre-holiday push.

RECOMMENDATIONS

- **High-Performing States (SP, MG, RJ, RS, PR)**
  1. Exclusive Loyalty Programs: Implement tier-based reward programs to encourage frequent purchases.
  2. Personalized Promotions: Offer state-specific deals aligned with their preferred payment methods.
  3. Priority Delivery: Introduce same-day or next-day delivery for these regions to maintain a competitive edge.
- **Low-Performing States (RO, AM, AC, AP, RR)**
  1. Localized Campaigns: Design hyper-local advertising in regional languages to engage underrepresented areas.
  2. Strategic Partnerships: Partner with local delivery services to reduce freight costs and improve delivery speed.
  3. Community Outreach: Use community influencers to build trust and promote the brand in smaller markets.
- **Reduce Delivery Time in Key Regions**
  1. Set up localized warehouses in regions with high order volumes (SP, MG).
  2. Implement AI-based delivery routing to shorten transit times.
  3. Smart Inventory Allocation
- **UPI & Credit Card Optimization**
  1. Exclusive Discounts: Offer UPI-exclusive cashback or credit card EMI discounts to drive usage.
  2. Promote Contactless Payments: Push UPI-first marketing during festive seasons and flash sales.
- **Increase Adoption of Debit Cards and Vouchers**
  1. Limited-Time Offers: Provide exclusive product access or extra discounts for voucher users.
  2. Introduce zero-cost EMIs for debit card transactions to boost large-order adoption.
- **Customer Acquisition & Retention**
  1. Increase Customer Base in Low-Performing States.
  2. Drive Customer Loyalty in High-Performing Regions

CONCLUSION

In conclusion, **Target's** data-driven insights reveal clear opportunities to enhance operational efficiency and customer satisfaction. By **focusing on high-performing states, optimizing logistics in low-demand regions, and leveraging popular payment methods,** Target can drive sustainable growth. Strengthening seasonal marketing strategies and tailoring payment options will further boost customer engagement and sales. **With a strategic, data-backed approach, Target is well-positioned to maximize market share and deliver exceptional value to its customers.**