Business Case Study: Target SQL

Priyannka Kollekar

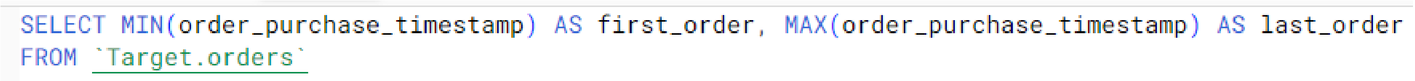
1.1 Data Exploration

After uploading the CSV datasets into Google Cloud, the customer dataset has been initially analyzed.

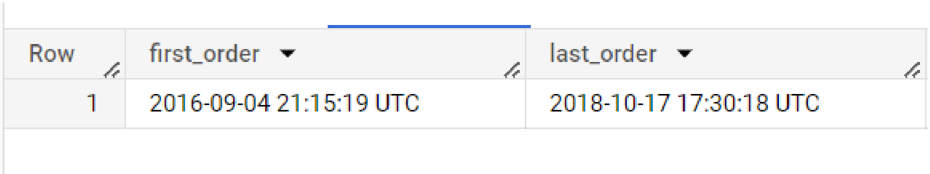
- Customers Table: There are around 99,441 rows representing each customer who has ordered. The table contains alphanumeric values, integers for pincode and string data.
- Geolocations Table: It contains arount 19,015 unique zipcodes, the data types are floating point for latitude and longitude. The name of city and state is also provided as string.
- Order Reviews: Around 99,224 reviews are there in the table whose primary key is review_id. Foreign key is order_id. Each review has a review score from 1–5. And review date having datetime data type.
- Products: There are around 32,951 products, the fields describe the dimensions of the product in integers.
- Order_items: This table shows for each order how many items have been ordered with the corresponding seller_id, order_id, product_id as the foreign keys.
- Payments: In this table we have for each order, the payment type(string), price(float), and number of installments(int), with around 103886 rows.
- Sellers: We have the information about the sellers. Seller_id(string), zip code(int), city and state(string).
- Orders: We have 99,441 orders with information regarding orders like - order_status(int), customer_id(string), date and time of purchase and delivery.

1.2 Time Period

From orders table we see that the time period of the date is from 2016–09–04 21:15:19 UTC to 2018–10–17 17:30:18 UTC. Around 2 years.



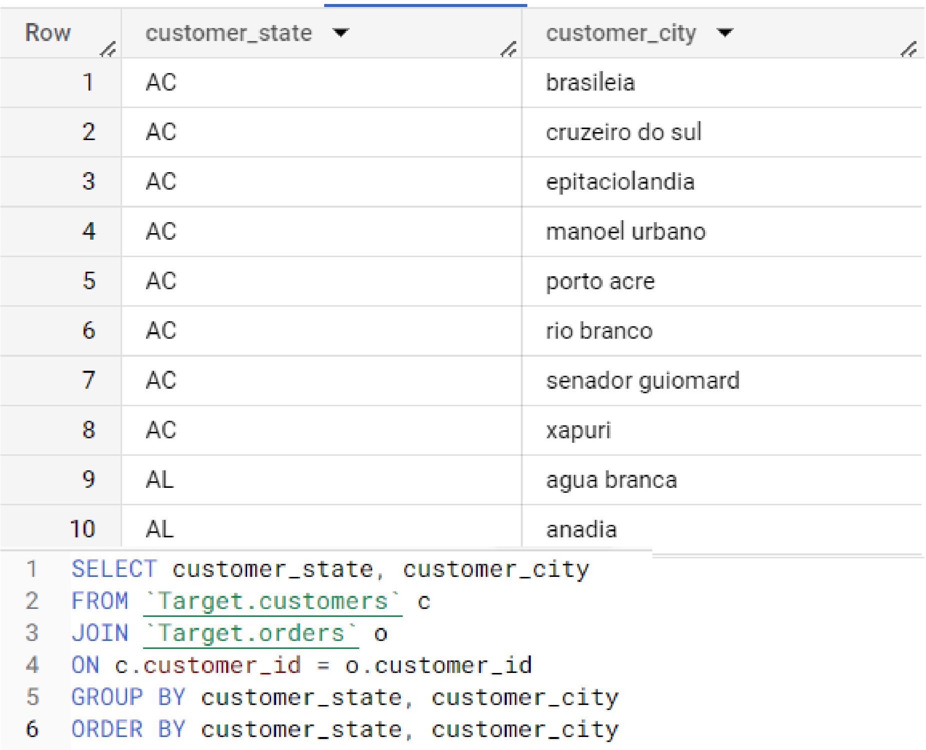
```
SELECT MIN(order_purchase_timestamp) AS first_order, MAX(order_purchase_timestamp) AS last_order
FROM `Target.orders`
```



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

1.3 States and cities of customers

There are around 4310 cities from where the customers ordered.

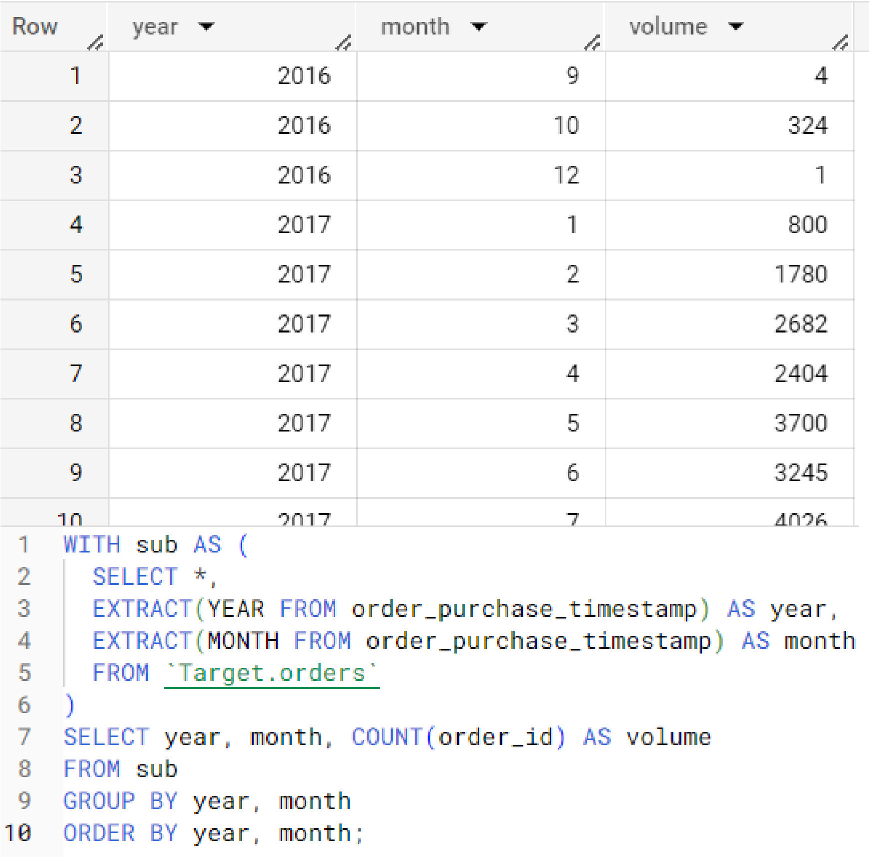


Row	customer_state	customer_city
1	AC	brasileia
2	AC	cruzeiro do sul
3	AC	epitaciolandia
4	AC	manoel urbano
5	AC	porto acre
6	AC	rio branco
7	AC	senador guiomard
8	AC	xapuri
9	AL	agua branca
10	AL	anadia

```sql
SELECT customer_state, customer_city
FROM `Target.customers` c
JOIN `Target.orders` o
ON c.customer_id = o.customer_id
GROUP BY customer_state, customer_city
ORDER BY customer_state, customer_city
```

2.1 Trend in e-commerce

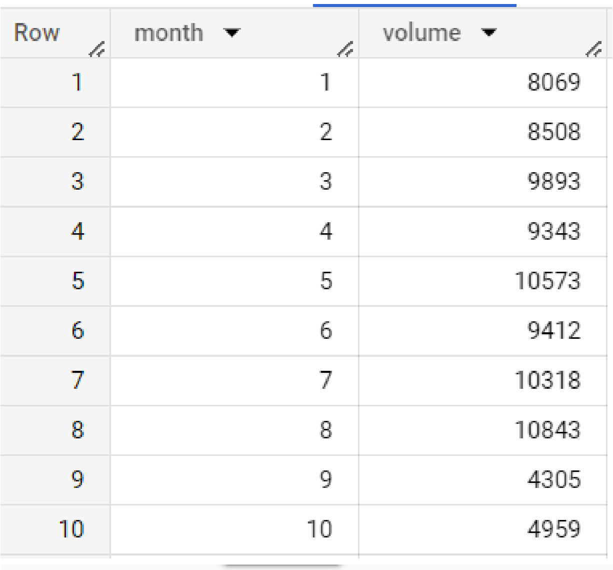
We see that initially the volume of orders (no. of orders) was less in 2016. However, from 2017 the number of orders has increased rapidly. In Nov od 2017, sales peak and eventually they come back down in 2018.



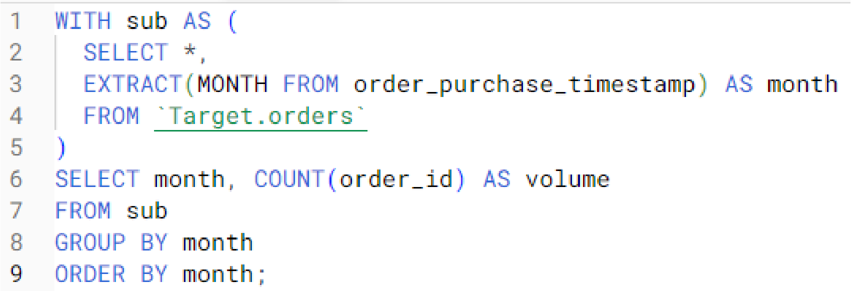
Row	year	month	volume
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

```sql
WITH sub AS (
  SELECT *,
  EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month
  FROM `Target.orders`
)
SELECT year, month, COUNT(order_id) AS volume
FROM sub
GROUP BY year, month
ORDER BY year, month;
```

Further, we see that sales peak in the mid-year period during the months of May, July and August.



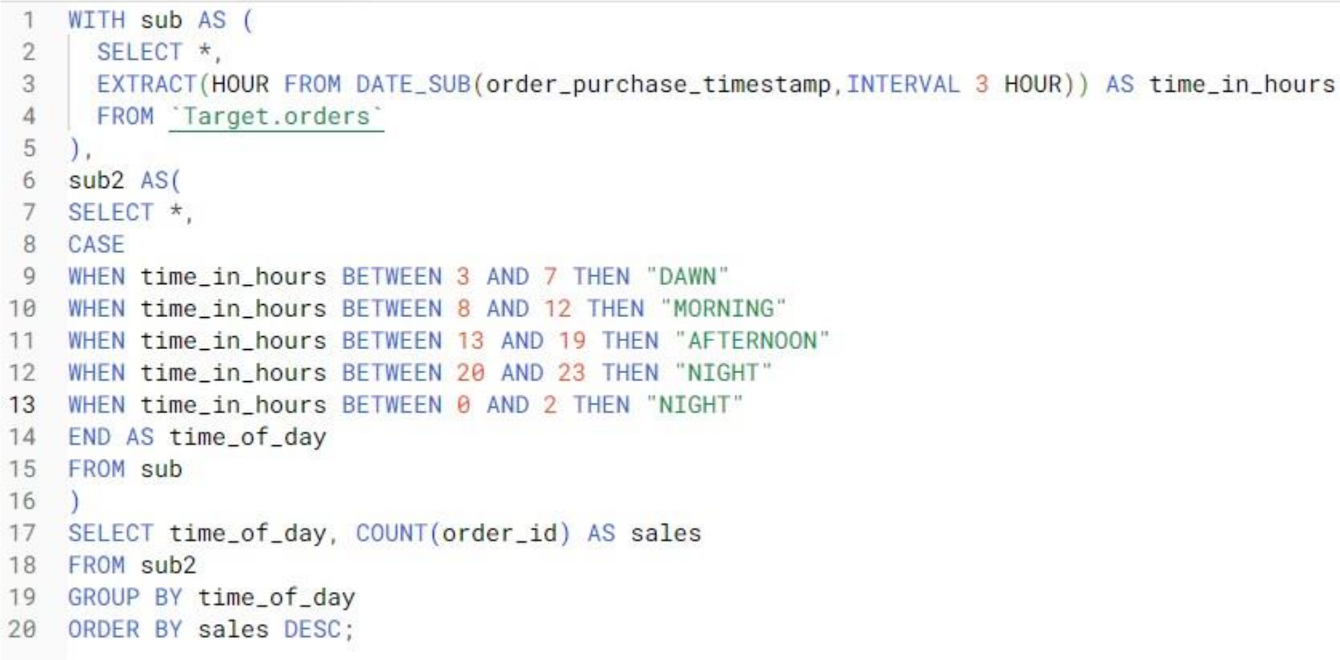
Row	month	volume
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959



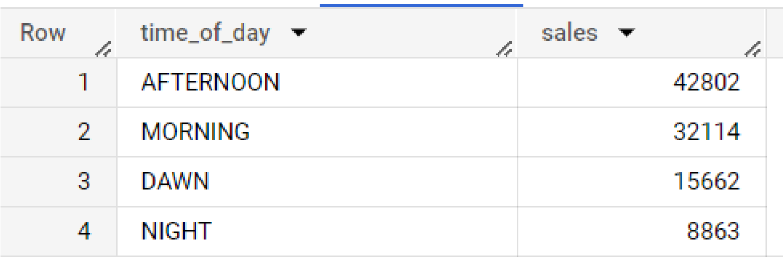
```
WITH sub AS (
  SELECT *,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month
  FROM `Target.orders`
)
SELECT month, COUNT(order_id) AS volume
FROM sub
GROUP BY month
ORDER BY month;
```

2.2 Time at which Brazilian customers tend to buy

Extracting time from order_purchase_timestamp and using Case and when statements, we can classify the time into Dawn, morning, afternoon, night. By grouping by the time_of_day, we get the count of orders which is saved as sales. From this we observe that in afternoon Brazilian customers tend to buy more.

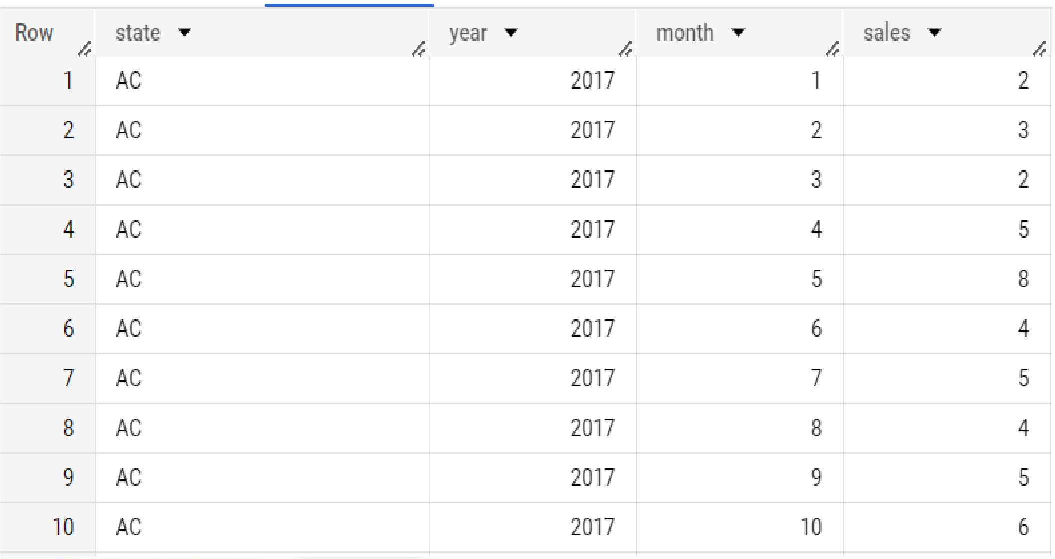


```
WITH sub AS (
  SELECT *,
  EXTRACT(HOUR FROM DATE_SUB(order_purchase_timestamp,INTERVAL 3 HOUR)) AS time_in_hours
  FROM `Target.orders`
),
sub2 AS(
SELECT *,
CASE
WHEN time_in_hours BETWEEN 3 AND 7 THEN "DAWN"
WHEN time_in_hours BETWEEN 8 AND 12 THEN "MORNING"
WHEN time_in_hours BETWEEN 13 AND 19 THEN "AFTERNOON"
WHEN time_in_hours BETWEEN 20 AND 23 THEN "NIGHT"
WHEN time_in_hours BETWEEN 0 AND 2 THEN "NIGHT"
END AS time_of_day
FROM sub
)
SELECT time_of_day, COUNT(order_id) AS sales
FROM sub2
GROUP BY time_of_day
ORDER BY sales DESC;
```

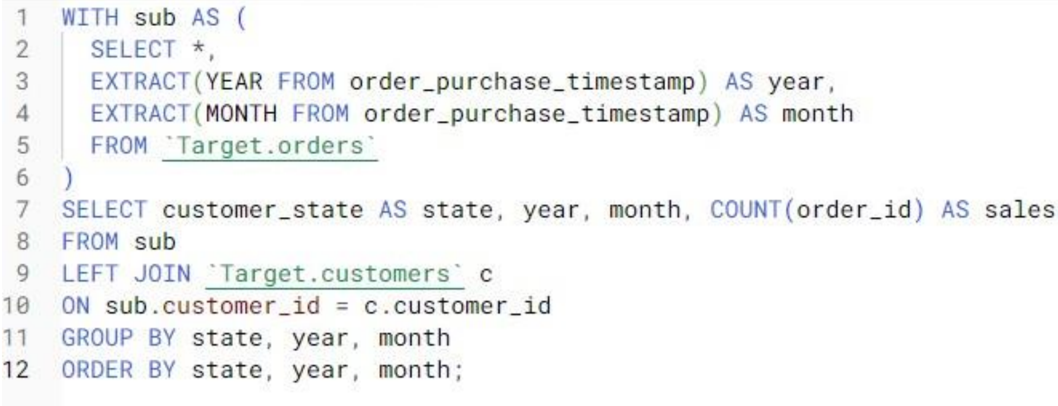


Row	time_of_day	sales
1	AFTERNOON	42802
2	MORNING	32114
3	DAWN	15662
4	NIGHT	8863

3.1 Month on month orders by states



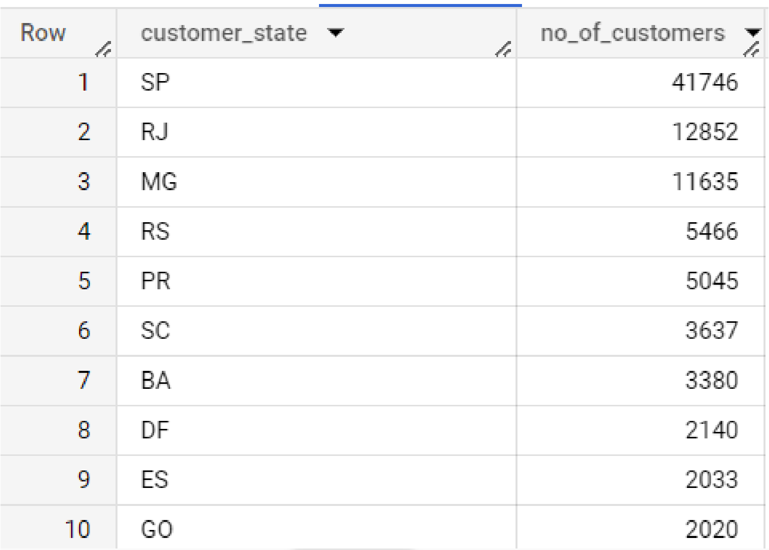
Row	state	year	month	sales
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6



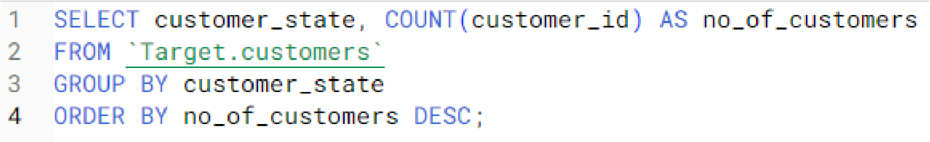
```sql
WITH sub AS (
  SELECT *,
  EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month
  FROM `Target.orders`
)
SELECT customer_state AS state, year, month, COUNT(order_id) AS sales
FROM sub
LEFT JOIN `Target.customers` c
ON sub.customer_id = c.customer_id
GROUP BY state, year, month
ORDER BY state, year, month;
```

3.2 Distribution of customers across States

Distribution of customers across states is found out by grouping states column in customers table and counting the customer id.



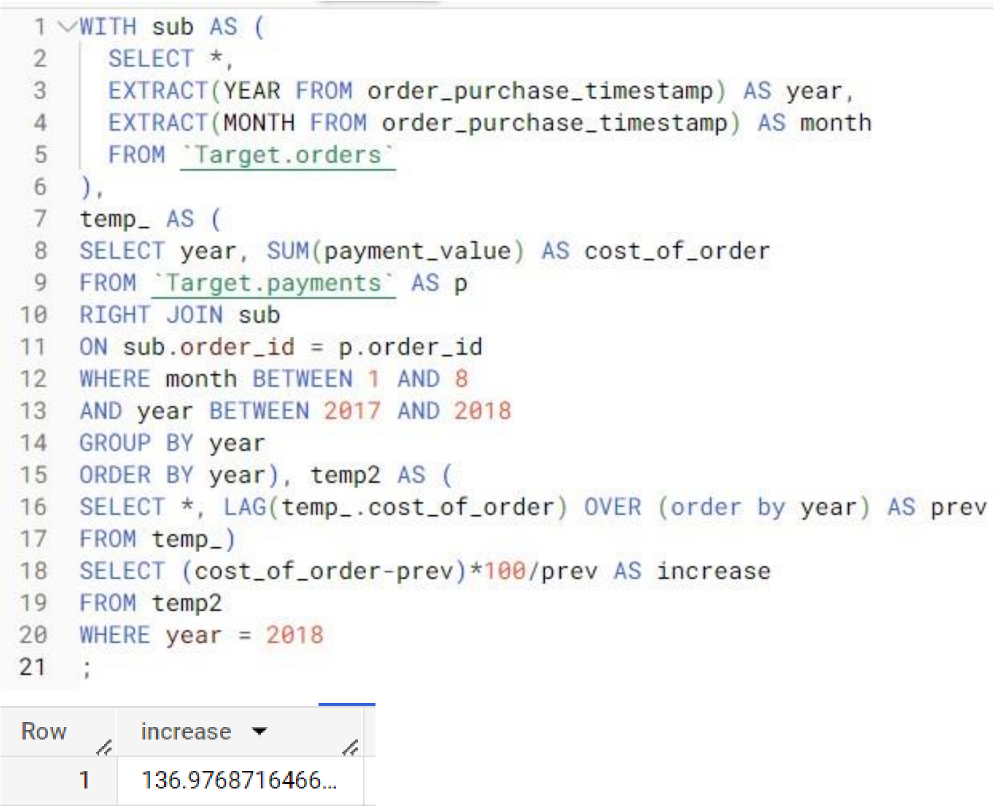
Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020



```sql
SELECT customer_state, COUNT(customer_id) AS no_of_customers
FROM `Target.customers`
GROUP BY customer_state
ORDER BY no_of_customers DESC;
```

4.1 Percentage increase in cost of orders

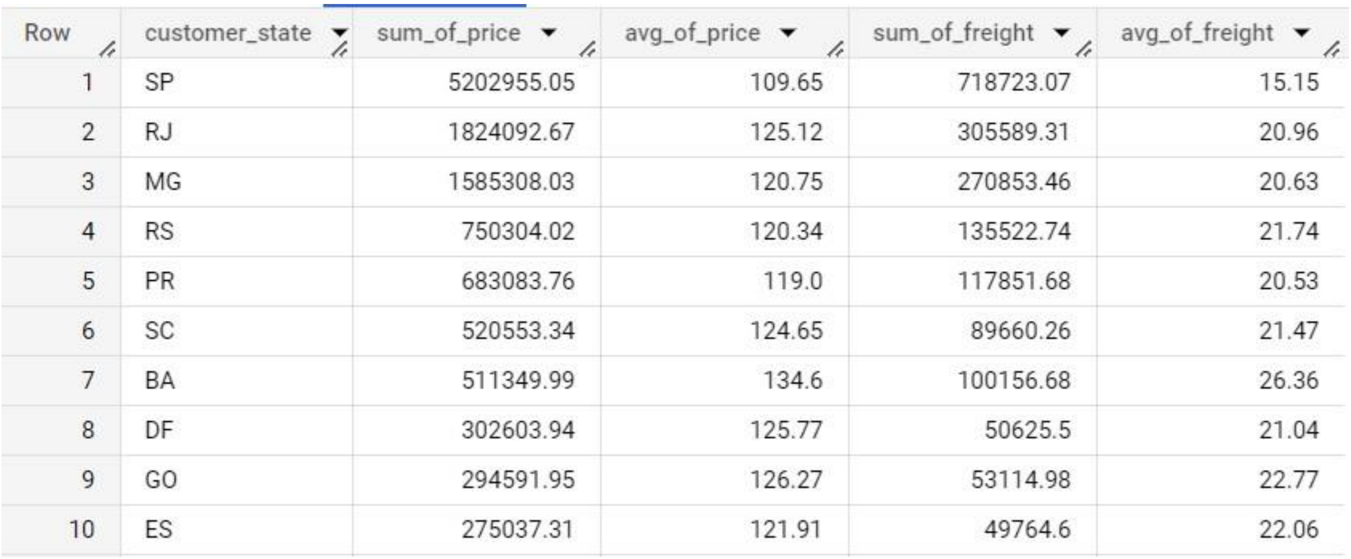
The percentage increase in cost of orders is first calculated by join payments table with orders. We will only take orders which are made from Jan to Aug (1–8). Then finally we group by year 2017 and 2018, where we aggregate the total payments. Using lag function, we can take the difference between the costs in 2017 and 2018 and finally we can take percentage. We get the answer as 136.97% from 2017 to 2018.



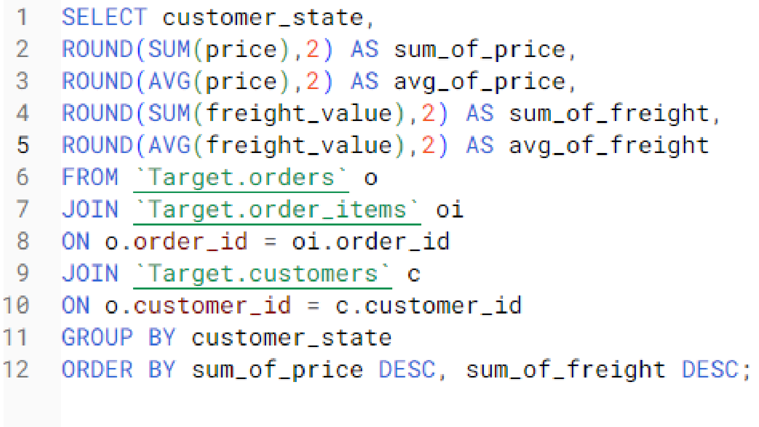
```sql
WITH sub AS (
  SELECT *,
  EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month
  FROM `Target.orders`
),
temp_ AS (
SELECT year, SUM(payment_value) AS cost_of_order
FROM `Target.payments` AS p
RIGHT JOIN sub
ON sub.order_id = p.order_id
WHERE month BETWEEN 1 AND 8
AND year BETWEEN 2017 AND 2018
GROUP BY year
ORDER BY year), temp2 AS (
SELECT *, LAG(temp_.cost_of_order) OVER (order by year) AS prev
FROM temp_)
SELECT (cost_of_order-prev)*100/prev AS increase
FROM temp2
WHERE year = 2018
;
```

Row	increase
1	136.9768716466...

4.2 Mean and sum of freight value and price.



Row	customer_state	sum_of_price	avg_of_price	sum_of_freight	avg_of_freight
1	SP	5202955.05	109.65	718723.07	15.15
2	RJ	1824092.67	125.12	305589.31	20.96
3	MG	1585308.03	120.75	270853.46	20.63
4	RS	750304.02	120.34	135522.74	21.74
5	PR	683083.76	119.0	117851.68	20.53
6	SC	520553.34	124.65	89660.26	21.47
7	BA	511349.99	134.6	100156.68	26.36
8	DF	302603.94	125.77	50625.5	21.04
9	GO	294591.95	126.27	53114.98	22.77
10	ES	275037.31	121.91	49764.6	22.06

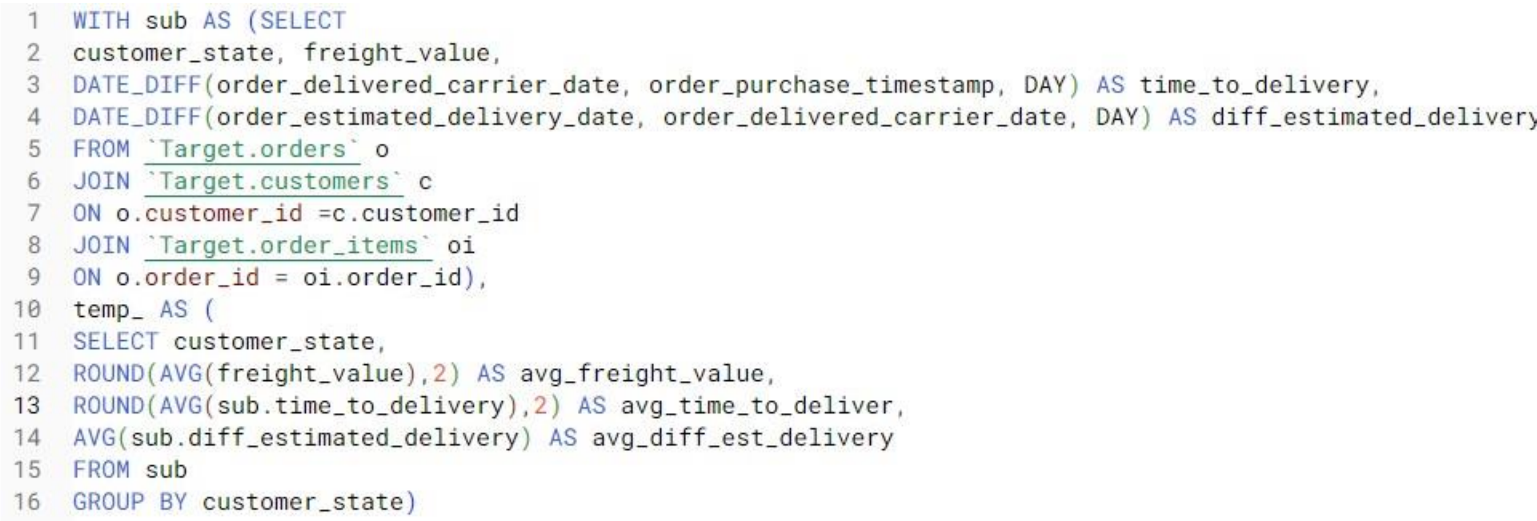


```
SELECT customer_state,
ROUND(SUM(price),2) AS sum_of_price,
ROUND(AVG(price),2) AS avg_of_price,
ROUND(SUM(freight_value),2) AS sum_of_freight,
ROUND(AVG(freight_value),2) AS avg_of_freight
FROM `Target.orders` o
JOIN `Target.order_items` oi
ON o.order_id = oi.order_id
JOIN `Target.customers` c
ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY sum_of_price DESC, sum_of_freight DESC;
```

We see that Sao Paolo state has the highest amount of total price and total freight value. We also notice an important information here, for SP state, the average price and average freight value is the least. Target could decrease the freight value in all the regions. This will increase sales as it would bring down the overall cost to customer.

5 Analysis on sales, freight, and delivery time.

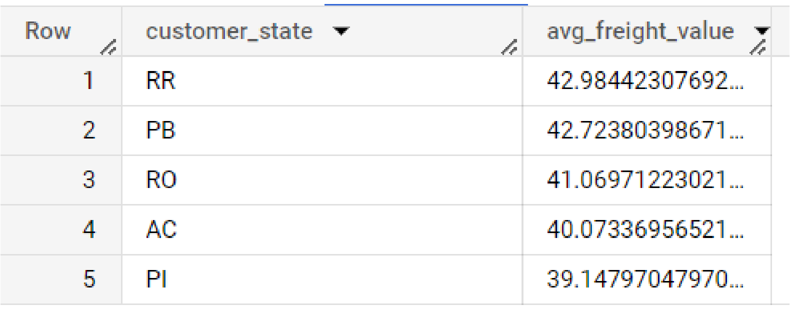
I have made the table which contains all the necessary data to solve this problem.



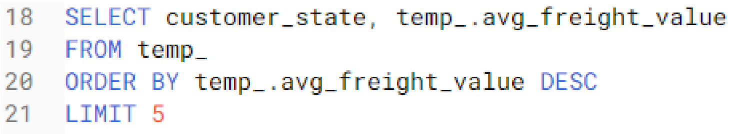
```
WITH sub AS (SELECT
customer_state, freight_value,
DATE_DIFF(order_delivered_carrier_date, order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(order_estimated_delivery_date, order_delivered_carrier_date, DAY) AS diff_estimated_delivery
FROM `Target.orders` o
JOIN `Target.customers` c
ON o.customer_id =c.customer_id
JOIN `Target.order_items` oi
ON o.order_id = oi.order_id),
temp_ AS (
SELECT customer_state,
ROUND(AVG(freight_value),2) AS avg_freight_value,
ROUND(AVG(sub.time_to_delivery),2) AS avg_time_to_deliver,
AVG(sub.diff_estimated_delivery) AS avg_diff_est_delivery
FROM sub
GROUP BY customer_state)
```

Using this table we can further solve the remaining part of the problem.

5.5 States with the highest average freight cost.



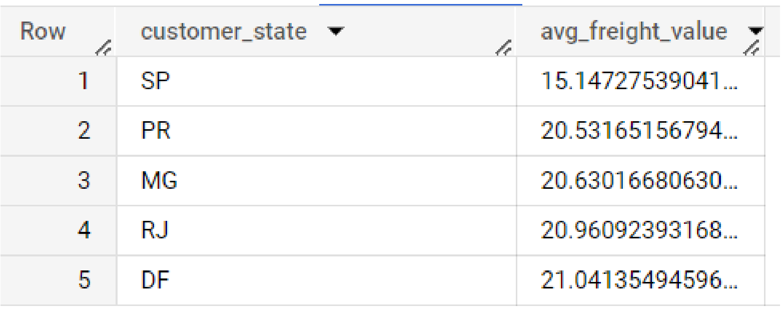
Row	customer_state	avg_freight_value
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...



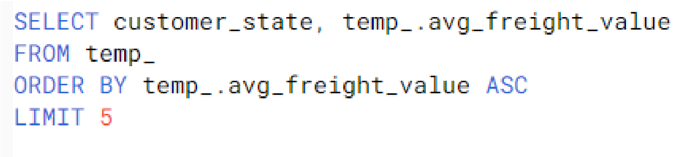
```
SELECT customer_state, temp_.avg_freight_value
FROM temp_
ORDER BY temp_.avg_freight_value DESC
LIMIT 5
```

Target should improve inventory management and logistics in these states so as to bring down the average freight value.

5.5 States with the lowest average freight cost.

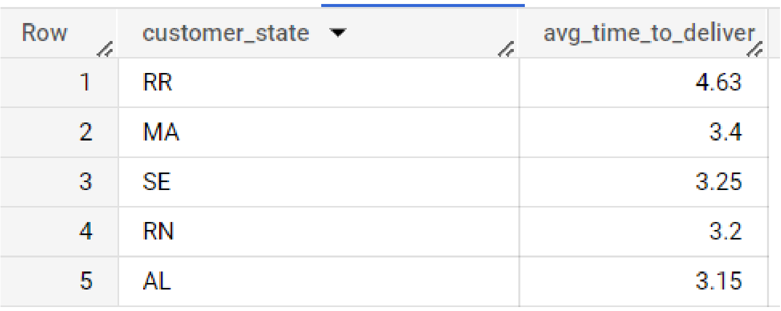


Row	customer_state	avg_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

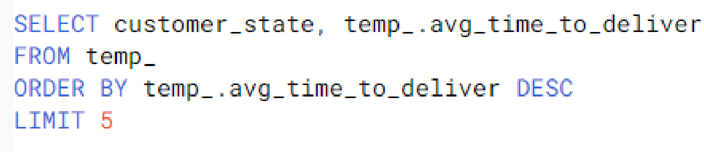


```
SELECT customer_state, temp_.avg_freight_value
FROM temp_
ORDER BY temp_.avg_freight_value ASC
LIMIT 5
```

5.6 States with the highest average time to delivery.



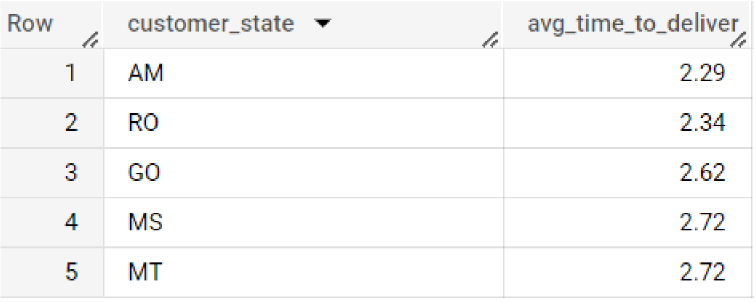
Row	customer_state	avg_time_to_deliver
1	RR	4.63
2	MA	3.4
3	SE	3.25
4	RN	3.2
5	AL	3.15



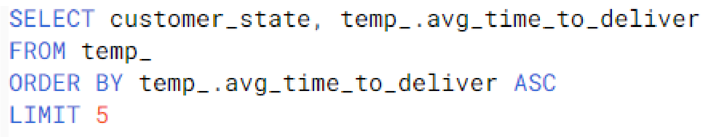
```
SELECT customer_state, temp_.avg_time_to_deliver
FROM temp_
ORDER BY temp_.avg_time_to_deliver DESC
LIMIT 5
```

Target should improve inventory management and logistics in these states to bring down the average time of delivery. We notice that states that have high average freight cost also have high average time of delivery.

5.6 States with the lowest average time to delivery.

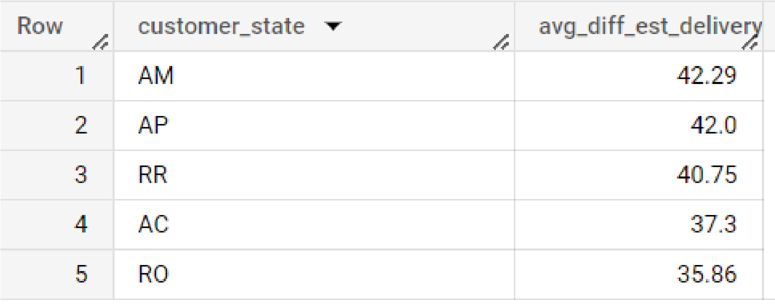


Row	customer_state	avg_time_to_deliver
1	AM	2.29
2	RO	2.34
3	GO	2.62
4	MS	2.72
5	MT	2.72

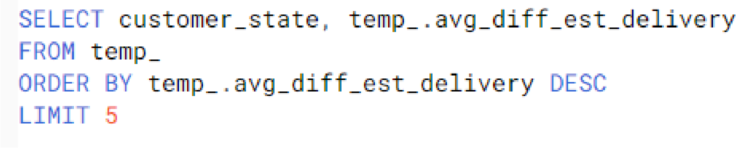


```
SELECT customer_state, temp_.avg_time_to_deliver
FROM temp_
ORDER BY temp_.avg_time_to_deliver ASC
LIMIT 5
```

5.7 States where delivery time is faster than estimated.



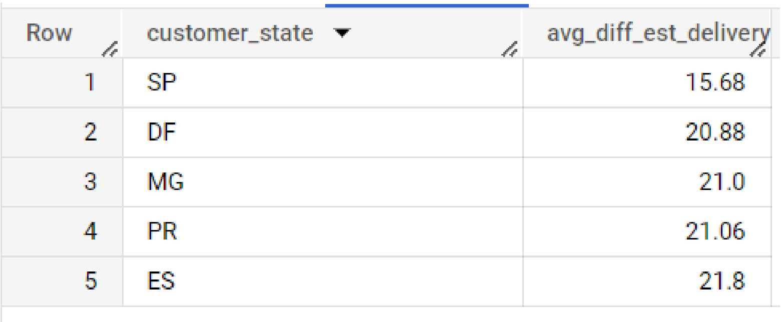
Row	customer_state	avg_diff_est_delivery
1	AM	42.29
2	AP	42.0
3	RR	40.75
4	AC	37.3
5	RO	35.86



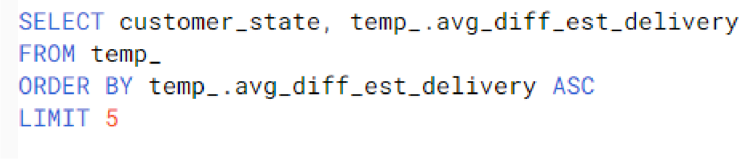
```
SELECT customer_state, temp_.avg_diff_est_delivery
FROM temp_
ORDER BY temp_.avg_diff_est_delivery DESC
LIMIT 5
```

Top 5 states where delivery is very fast compared to the estimated delivery time. Target should improve the delivery estimation algorithm, because the estimation is wrong by around whopping 40 days.

5.7 States where delivery time is slower than estimated



Row	customer_state	avg_diff_est_delivery
1	SP	15.68
2	DF	20.88
3	MG	21.0
4	PR	21.06
5	ES	21.8

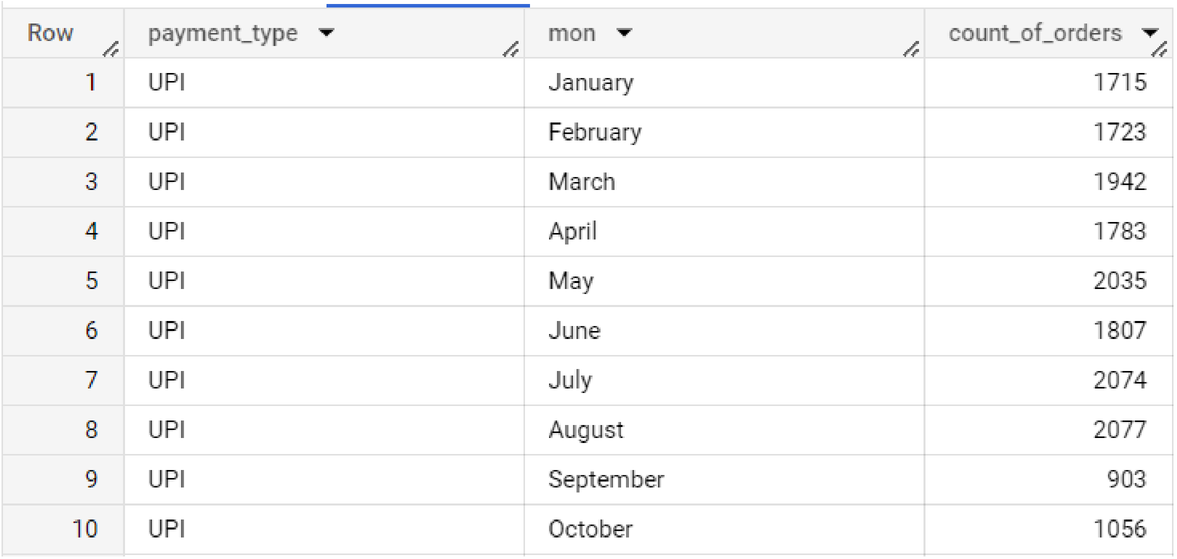


```
SELECT customer_state, temp_.avg_diff_est_delivery
FROM temp_
ORDER BY temp_.avg_diff_est_delivery ASC
LIMIT 5
```

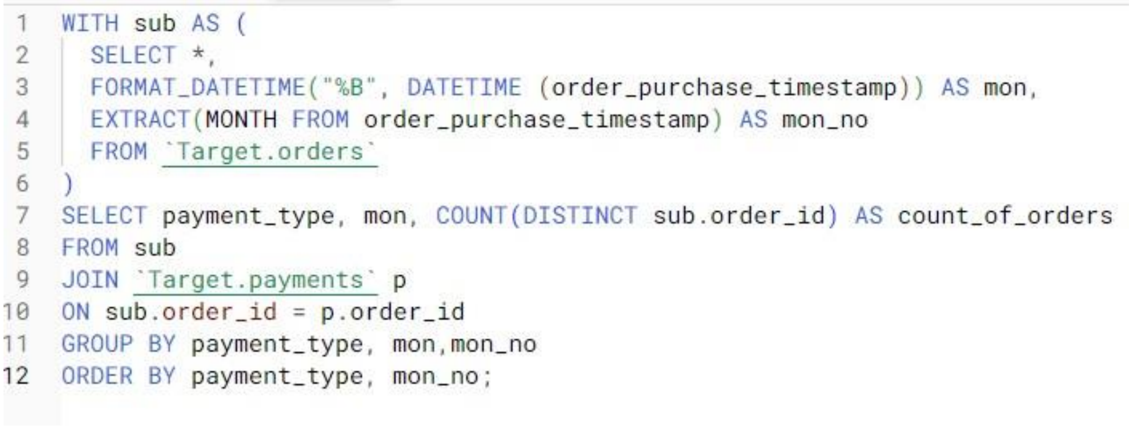
From the date we can draw following insights:

1. Inventory management and logistics should be improved in these states, to decrease the delivery time.
2. Estimation of delivery should be more aligned with the actual delivery time, so that customers don’t receive the orders at unexpected time.

6.1 Payment type analysis - Month over month count of different payment types



Row	payment_type	mon	count_of_orders
1	UPI	January	1715
2	UPI	February	1723
3	UPI	March	1942
4	UPI	April	1783
5	UPI	May	2035
6	UPI	June	1807
7	UPI	July	2074
8	UPI	August	2077
9	UPI	September	903
10	UPI	October	1056

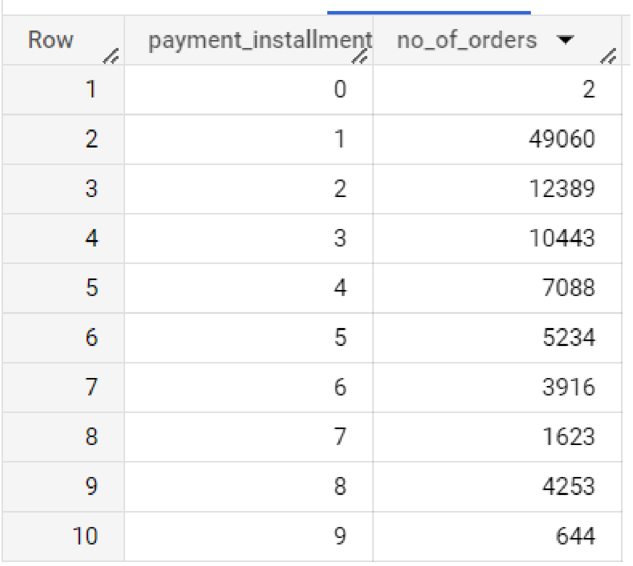


```
WITH sub AS (
  SELECT *,
  FORMAT_DATETIME("%B", DATETIME (order_purchase_timestamp)) AS mon,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS mon_no
  FROM `Target.orders`
)
SELECT payment_type, mon, COUNT(DISTINCT sub.order_id) AS count_of_orders
FROM sub
JOIN `Target.payments` p
ON sub.order_id = p.order_id
GROUP BY payment_type, mon,mon_no
ORDER BY payment_type, mon_no;
```

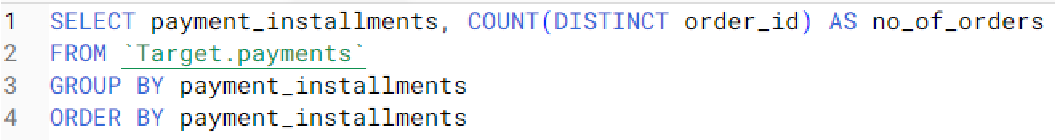
We see that no. of orders steadily increase month over month for all payment types up until august and then it drastically falls. Credit card payments are the highest.

We can recommend Target to give more discounts and benefits to debit card customers, because this payment type is the least used.

6.2 Payment type analysis - Count of orders based on no. of installments



Row	payment_installment	no_of_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623
9	8	4253
10	9	644



```sql
SELECT payment_installments, COUNT(DISTINCT order_id) AS no_of_orders
FROM `Target.payments`
GROUP BY payment_installments
ORDER BY payment_installments
```

We can observe the number of one-time purchases is highest.

Insights from the data:

- Number of orders increased rapidly from 2016 up until 2017. However, it was less in 2018 due to the fact that there was no sufficient data from 2018. But overall the sales trend is upward.
- Further, we see that sales peak in the mid-year period during the months of May, July and August.
- We can see that customers buy more during the afternoons and mornings.
- We can see that most of the customers come from the state of SP, RJ and MG which contribute to more than 60% of total customers.
- There is 137% increase in total cost of orders from 2017 to 2018.
- We see that the cost of freight is around 1/6 of the price.
- The state of SP has the highest number of orders and the lowest average cost of freight.
- We can also see that as the volume of orders decreases from state to state, the freight price increases.
- The state RR has the highest average freight cost most time taken to deliver.
- The state AM is the fastest in terms of delivery time, which is around 2.29 days.
- The number of credit card payments is the highest. Whereas payments made through debit cards are the lowest.
- We can observe the number of one-time purchases is highest. The number of purchases decreases when the number of installments increase.

Recommendation:

- We can recommend Target to give more discounts and benefits to debit card customers, because this payment type is least used.
- Inventory management and logistics should be improved in certain states (mentioned above), to decrease the delivery time.
- Estimation of delivery should be more aligned with the actual delivery time, so that customers don’t receive the orders at unexpected time.
- Improved logistics will also decrease the average freight costs.
- The state SP is very important for Target, since it is where most orders come from. Target should focus on reducing the delivery time in this region.
- The number of orders is less for products where the number of installments is more - which are generally higher value purchases. To increase the affordability of bigger purchases, Target should provide no cost EMI and discounts for purchases with 12 installments or more.
- Overall month on month sales is increasing. Therefore, Target should focus on expanding in the Brazil market so as to not face any shortages or delays.