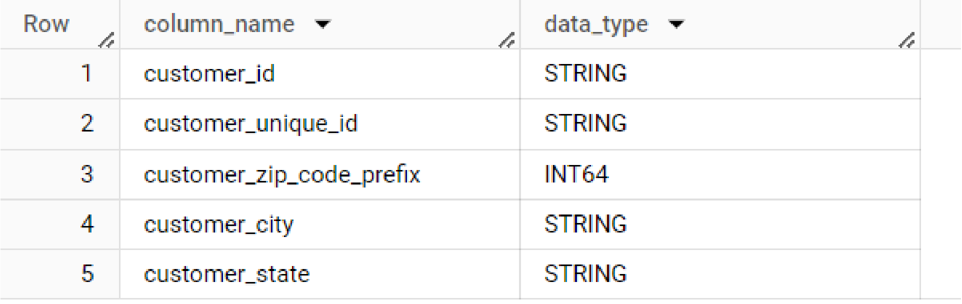
Problem Statement:

1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
    1. Data type of all columns in the "customers" table.

Ans.
```
SELECT column_name, data_type

FROM target-scaler-400905.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

Insights: - customer_id:

Data Type: STRING

Insight: The "customer_id" is represented as a unique identifier for each customer. Being stored as a string allows for flexibility in storing alphanumeric characters and ensures uniqueness for identification purposes.

customer_unique_id:

Data Type: STRING

Insight: The "customer_unique_id" is stored as a string, implying it's a unique identifier associated with each customer. Utilizing a unique identifier is crucial for distinguishing individual customers within the dataset.

customer_zip_code_prefix:

Data Type: INT64

Insight: The "customer_zip_code_prefix" is stored as an integer, indicating the prefix of the customer's zip code. Storing it as an integer allows for efficient numerical operations and comparisons.

customer_city:

Data Type: STRING

Insight: The "customer_city" is represented as a string, containing the name of the city from which the order was made. Storing it as a string enables flexibility in accommodating city names of varying lengths.

customer_state:

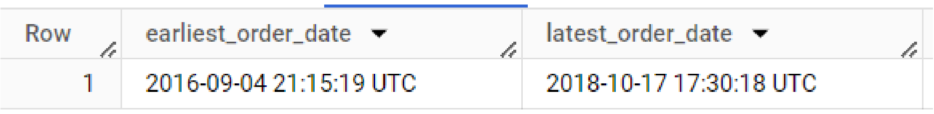
Data Type: STRING

Insight: The "customer_state" is stored as a string, representing the state code from where the order was made (e.g., "são paulo - SP"). Storing it as a string allows for flexibility in capturing state codes or names.

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

Ans.
```
SELECT MIN(order_purchase_timestamp) AS earliest_order_date,

       MAX(order_purchase_timestamp) AS latest_order_date
FROM target-scaler-400905.target.orders;
```



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

Earliest Order Date:

Value: 2016-09-04 21:15:19 UTC

Insight: The earliest order in the dataset was placed on September 4, 2016, at 21:15:19 UTC. This indicates that the dataset spans from this date onwards, capturing orders made after this point in time.

Latest Order Date:

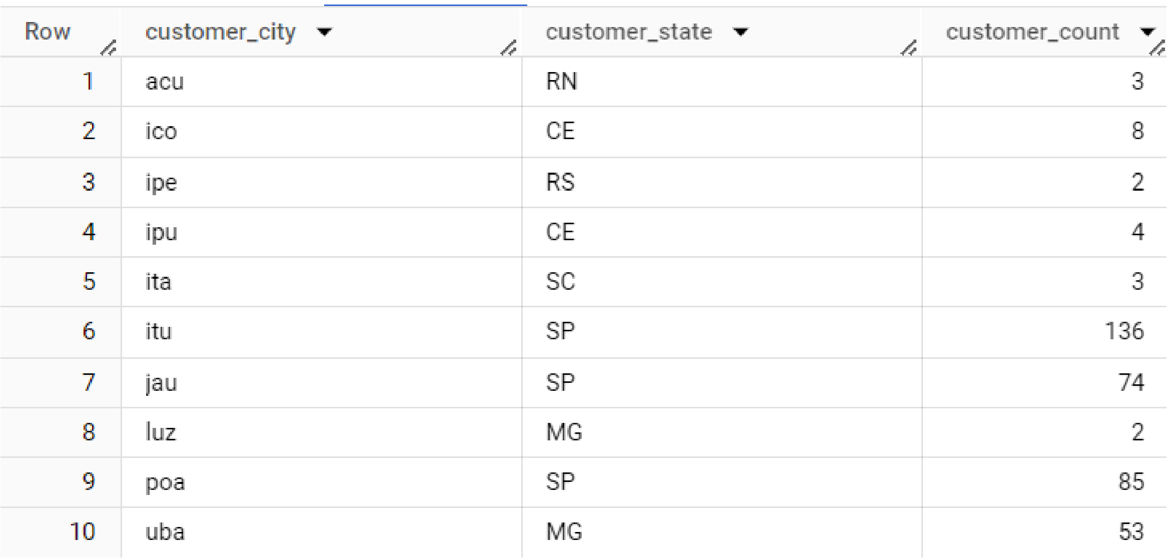
Value: 2018-10-17 17:30:18 UTC

Insight: The latest order in the dataset was placed on October 17, 2018, at 17:30:18 UTC. This represents the endpoint of the dataset, indicating that the dataset encompasses orders placed up to this date and time.

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

Ans.
```
SELECT MIN(order_purchase_timestamp) AS earliest_order_date,

       MAX(order_purchase_timestamp) AS latest_order_date
FROM target-scaler-400905.target.orders;
```



Row	customer_city	customer_state	customer_count
1	acu	RN	3
2	ico	CE	8
3	ipe	RS	2
4	ipu	CE	4
5	ita	SC	3
6	itu	SP	136
7	jau	SP	74
8	luz	MG	2
9	poa	SP	85
10	uba	MG	53

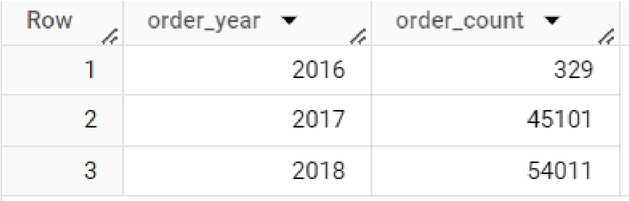
Insight: The dataset offers a comprehensive view of Target's operations in Brazil, covering customer details, order timeframe, and geographical customer distribution. Understanding customer demographics (e.g., unique identifiers, city, state) and order timelines (ranging from September 4, 2016, to October 17, 2018) is essential for analyzing customer behavior, optimizing delivery processes, and tailoring marketing strategies. Additionally, the geographical distribution of customers across various cities and states provides valuable insights into the market reach and potential areas for targeted business growth and expansion.

2. **In-depth Exploration:**

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

Ans.

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       COUNT(*) AS order_count
FROM target-scaler-400905.target.orders
GROUP BY order_year
ORDER BY order_year;
```



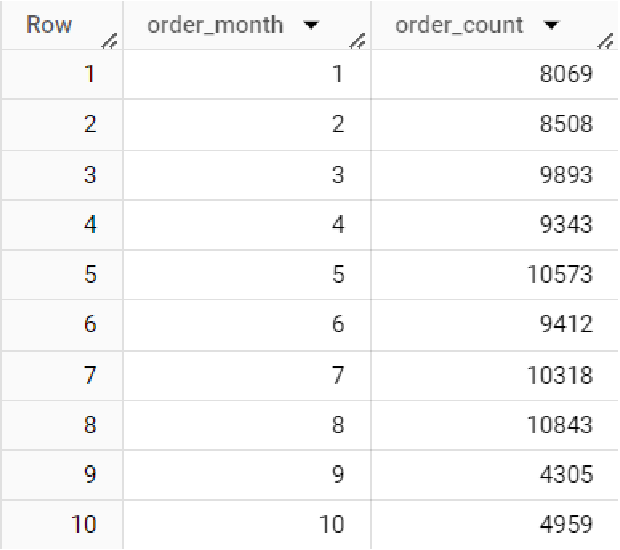
Row	order_year	order_count
1	2016	329
2	2017	45101
3	2018	54011

Insight: The dataset illustrates a clear trend in the number of orders placed over the years in Brazil. The data shows a notable increase in order count from 2016 to 2017, representing a substantial growth in orders. Furthermore, the trend continues in 2018, with a further increase in order count compared to the previous year. This suggests a positive growth trajectory in e-commerce activities during this period. Analyzing this trend can provide valuable insights for business strategies, marketing efforts, and resource allocation to accommodate the increasing demand for online shopping.

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

Ans.
```
SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,

       COUNT(*) AS order_count
FROM target-scaler-400905.target.orders
GROUP BY order_month
ORDER BY order_month;
```

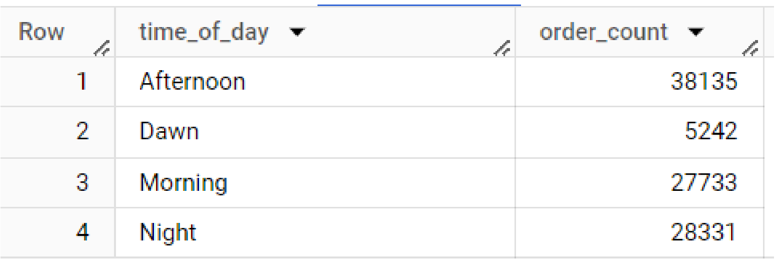


Row	order_month	order_count
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959

Insight: Analyzing the month-wise distribution of order counts reveals interesting patterns. Months from January to August consistently show a steady increase in order counts, reaching their peak in August. However, there is a significant drop in order counts in September and October. This could be attributed to seasonal factors, marketing strategies, or external influences impacting consumer behavior during this period. Understanding these trends is crucial for planning marketing campaigns, inventory management, and offering promotions to maintain or boost order numbers during lower-performing months.

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

Ans.
```
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_day,
       COUNT(*) AS order_count
FROM target-scaler-400905.target.orders
GROUP BY time_of_day
ORDER BY time_of_day;
```

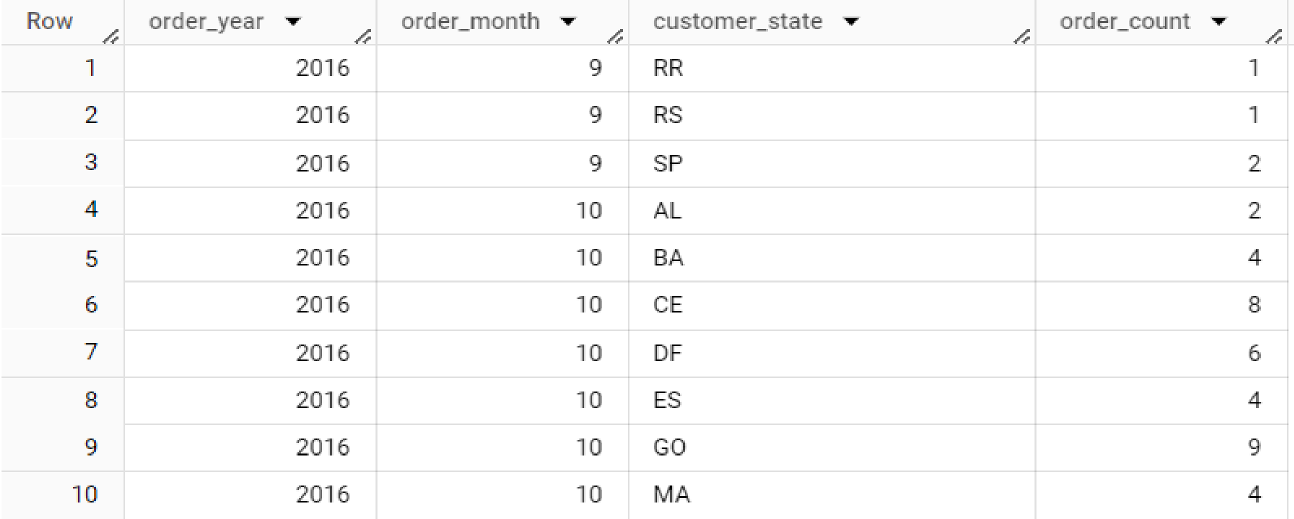


Row	time_of_day	order_count
1	Afternoon	38135
2	Dawn	5242
3	Morning	27733
4	Night	28331

Insight: The dataset provides insights into the preferred time of day for placing orders by Brazilian customers. The data categorizes orders into four time slots: Afternoon, Dawn, Morning, and Night. The highest order count is observed during the Afternoon, indicating that this time period is the most popular for placing orders. Conversely, Dawn has the lowest order count, suggesting it is the least favored time for making purchases. Understanding these time preferences is valuable for businesses to optimize marketing strategies, delivery schedules, and customer support, aligning with the peak order times to enhance customer satisfaction and operational efficiency.

3. **Evolution of E-commerce orders in the Brazil region:**
    1. Get the month on month no. of orders placed in each state.

    Ans.
```
SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
    c.customer_state,
    COUNT(*) AS order_count
FROM
    target-scaler-400905.target.orders AS o
JOIN
    target-scaler-400905.target.customers AS c
ON
    o.customer_id = c.customer_id
GROUP BY
    order_year, order_month, customer_state
ORDER BY
    order_year, order_month, customer_state;
```



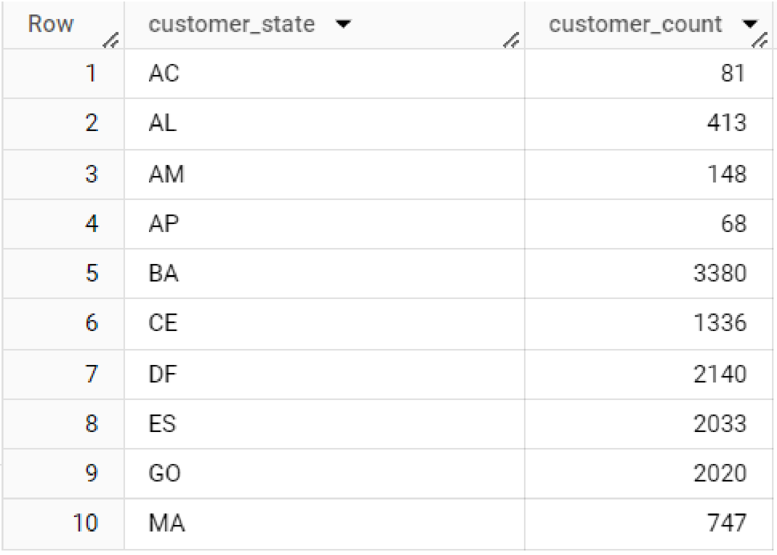
Row	order_year	order_month	customer_state	order_count
1	2016	9	RR	1
2	2016	9	RS	1
3	2016	9	SP	2
4	2016	10	AL	2
5	2016	10	BA	4
6	2016	10	CE	8
7	2016	10	DF	6
8	2016	10	ES	4
9	2016	10	GO	9
10	2016	10	MA	4

Insight: The dataset provides a detailed view of the month-wise distribution of orders across different states in Brazil for the year 2016. It's evident that orders are distributed across various states, with differing order counts for each state-month combination. For instance, in October 2016, the state of Goiás (GO) had the highest order count (9), indicating a relatively high level of e-commerce activity. On the other hand, states like Roraima (RR) and Rio Grande do Sul (RS) had the lowest order counts in the given time frame.

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

Ans. SELECT

```
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
    c.customer_state,
    COUNT(*) AS order_count
FROM
    target-scaler-400905.target.orders AS o
JOIN
    target-scaler-400905.target.customers AS c
ON
    o.customer_id = c.customer_id
GROUP BY
    order_year, order_month, customer_state
ORDER BY
    order_year, order_month, customer_state;
```



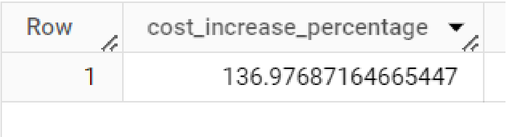
Row	customer_state	customer_count
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

Insight: The dataset provides a breakdown of customer counts for various states in Brazil. The counts range from 68 customers in Amapá (AP) to 3380 customers in Bahia (BA). São Paulo (SP), represented by "DF," has a substantial customer count of 2140. Understanding customer distribution across states is crucial for businesses to tailor their marketing strategies, improve service quality, and efficiently allocate resources. States like Bahia (BA) and São Paulo (DF) show a notable customer presence, making them potentially significant markets for e-commerce businesses.

4. **Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.**
    1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
       You can use the "payment_value" column in the payments table to get the cost of orders.

Ans.
```
WITH order_costs AS (

    SELECT p.order_id,
           EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
           EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
           p.payment_value
    FROM target-scaler-400905.target.payments p
    JOIN target-scaler-400905.target.orders o ON p.order_id = o.order_id
    WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018)
      AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
)
SELECT
    (SUM(CASE WHEN order_year = 2018 THEN payment_value END) -
    SUM(CASE WHEN order_year = 2017 THEN payment_value END)) /
    SUM(CASE WHEN order_year = 2017 THEN payment_value END) * 100 AS
cost_increase_percentage
FROM order_costs;
```



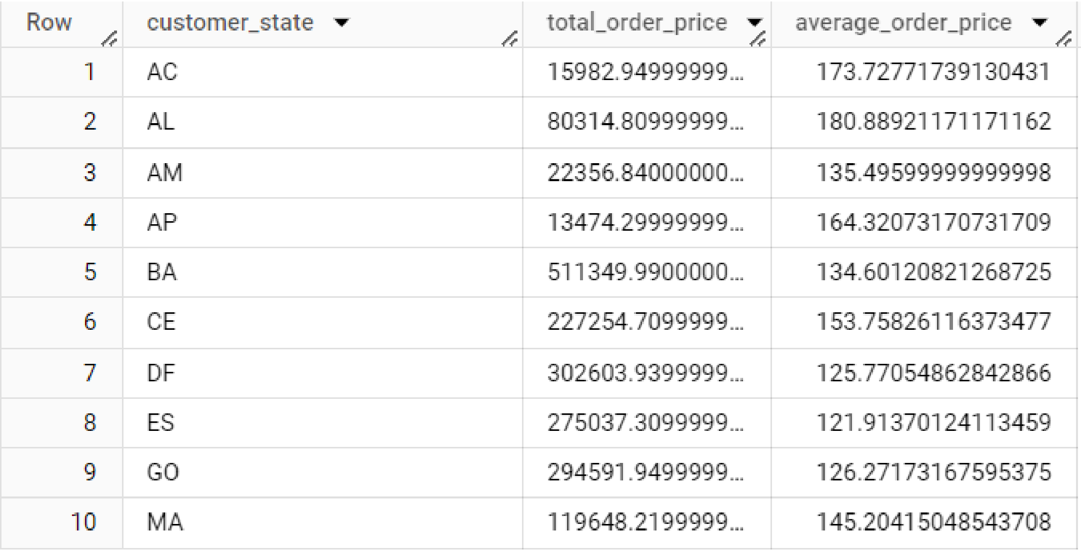
Row	cost_increase_percentage
1	136.97687164665447

Insight: The dataset presents a significant percentage increase in the cost of orders from the year 2017 to 2018. The calculated cost increase percentage is approximately 136.98%. This increase suggests a substantial rise in order costs during this period. Understanding these cost fluctuations is vital for businesses to assess profitability, pricing strategies, and cost management. It's important to delve deeper into the factors influencing this increase to make informed decisions and ensure sustainable business growth.

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

Ans.
```
SELECT c.customer_state,

       SUM(oi.price) AS total_order_price,
       AVG(oi.price) AS average_order_price
FROM target-scaler-400905.target.order_items oi
JOIN target-scaler-400905.target.orders o ON oi.order_id = o.order_id
JOIN target-scaler-400905.target.customers c ON o.customer_id =
c.customer_id
GROUP BY c.customer_state
ORDER BY c.customer_state;
```



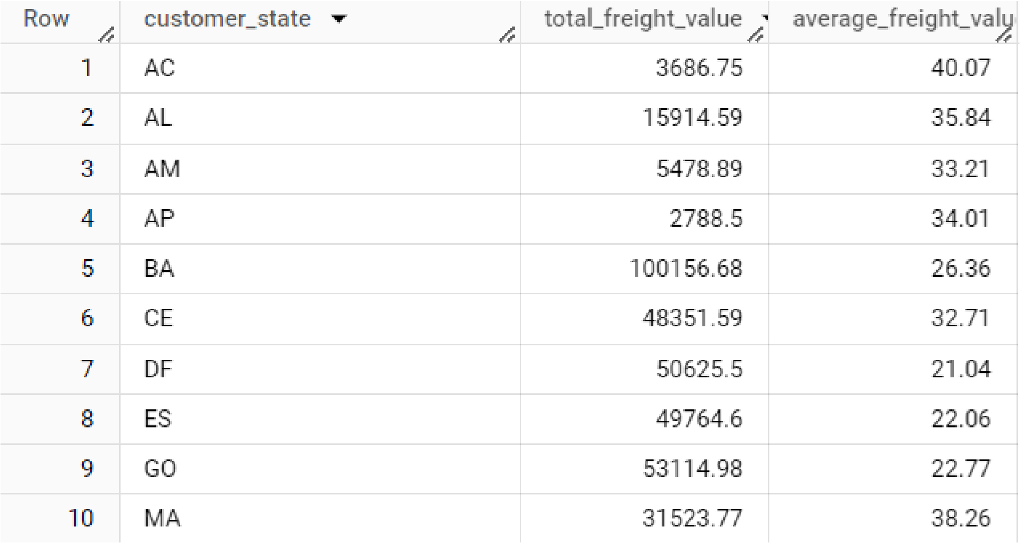
Row	customer_state	total_order_price	average_order_price
1	AC	15982.94999999...	173.72771739130431
2	AL	80314.80999999...	180.88921171171162
3	AM	22356.84000000...	135.49599999999998
4	AP	13474.29999999...	164.32073170731709
5	BA	511349.9900000...	134.60120821268725
6	CE	227254.7099999...	153.75826116373477
7	DF	302603.9399999...	125.77054862842866
8	ES	275037.3099999...	121.91370124113459
9	GO	294591.9499999...	126.27173167595375
10	MA	119648.2199999...	145.20415048543708

Insight: The dataset provides a breakdown of the total order price and the average order price for various states in Brazil. Each state has its respective total order price and average order price, showcasing the financial aspects of orders in those states. For example, in Bahia (BA), the total order price is approximately 511,349.99, with an average order price of around 134.60. Understanding the financial aspects of orders in different states is essential for businesses to optimize pricing strategies, identify lucrative markets, and tailor marketing efforts accordingly.

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

Ans.
```
SELECT

    c.customer_state,
    ROUND(SUM(oi.freight_value), 2) AS total_freight_value,
    ROUND(AVG(oi.freight_value), 2) AS average_freight_value
FROM
    target-scaler-400905.target.order_items oi
JOIN
    target-scaler-400905.target.orders o ON oi.order_id = o.order_id
JOIN
    target-scaler-400905.target.customers c ON o.customer_id =
c.customer_id
GROUP BY
    c.customer_state
ORDER BY
    c.customer_state;
```



Row	customer_state	total_freight_value	average_freight_valu
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

Insight: The dataset provides insights into the total and average freight values for various states in Brazil. Each state has its respective total freight value and average freight value, showcasing the financial aspects related to shipping costs for orders in those states. For example, in Bahia (BA), the total freight value is approximately 100,156.68, with an average freight value of around 26.36. Understanding the freight costs across different states is crucial for businesses to optimize logistics, pricing strategies, and operational efficiency.

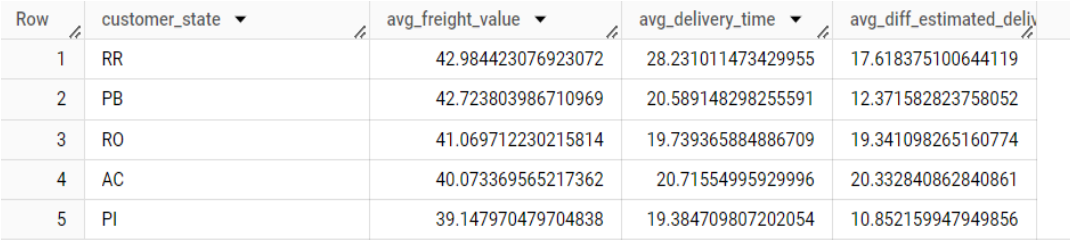
5. **Analysis based on sales, freight and delivery time.**
    1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
       Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
       Do this in a single query.

       You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:
        - **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
        - **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

```
      WITH delivery_analysis AS (

    SELECT
        customers.customer_state,
        orders.order_delivered_customer_date,
        orders.order_purchase_timestamp,
        orders.order_estimated_delivery_date,
        order_items.freight_value
    FROM
        target-scaler-400905.target.orders
    JOIN
        target-scaler-400905.target.order_items
    ON
        orders.order_id = order_items.order_id
    JOIN
        target-scaler-400905.target.customers
    ON
        orders.customer_id = customers.customer_id
)

SELECT
    customer_state,
    AVG(freight_value) AS avg_freight_value,
    AVG(TIMESTAMP_DIFF(order_delivered_customer_date,
order_purchase_timestamp, SECOND) / (24 * 3600)) AS avg_delivery_time,
    AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, SECOND) / (24 * 3600)) AS
avg_diff_estimated_delivery
FROM
    delivery_analysis
GROUP BY
    customer_state
ORDER BY
    avg_freight_value DESC
LIMIT 5;
```

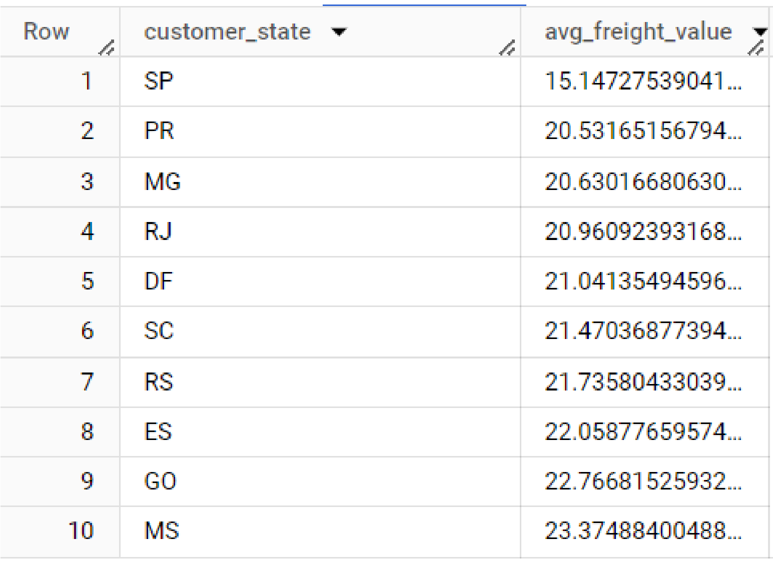


Row	customer_state	avg_freight_value	avg_delivery_time	avg_diff_estimated_deli
1	RR	42.984423076923072	28.231011473429955	17.618375100644119
2	PB	42.723803986710969	20.589148298255591	12.371582823758052
3	RO	41.069712230215814	19.739365884886709	19.341098265160774
4	AC	40.073369565217362	20.71554995929996	20.332840862840861
5	PI	39.147970479704838	19.384709807202054	10.852159947949856

Insight: The dataset analysis for the top 5 states with the highest average freight values shows that states like Roraima (RR), Paraíba (PB), and Rondônia (RO) have notably high average freight costs. Additionally, these states also exhibit relatively higher average delivery times and differences between estimated and actual delivery dates. On the other hand, the states of Acre (AC) and Piauí (PI) have lower average freight costs, indicating potentially efficient logistics. Understanding these variations in freight costs and delivery times across different states is crucial for optimizing shipping strategies and enhancing customer satisfaction.

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

```
Ans. SELECT
  customer_state,
  AVG(freight_value) AS avg_freight_value,
  AVG(TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, SECOND) / (24 * 3600)) AS avg_delivery_time,
  AVG(TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, SECOND) / (24 * 3600)) AS avg_diff_estimated_delivery
FROM
  delivery_analysis
GROUP BY
  customer_state
ORDER BY
  avg_freight_value DESC
LIMIT 5;
```



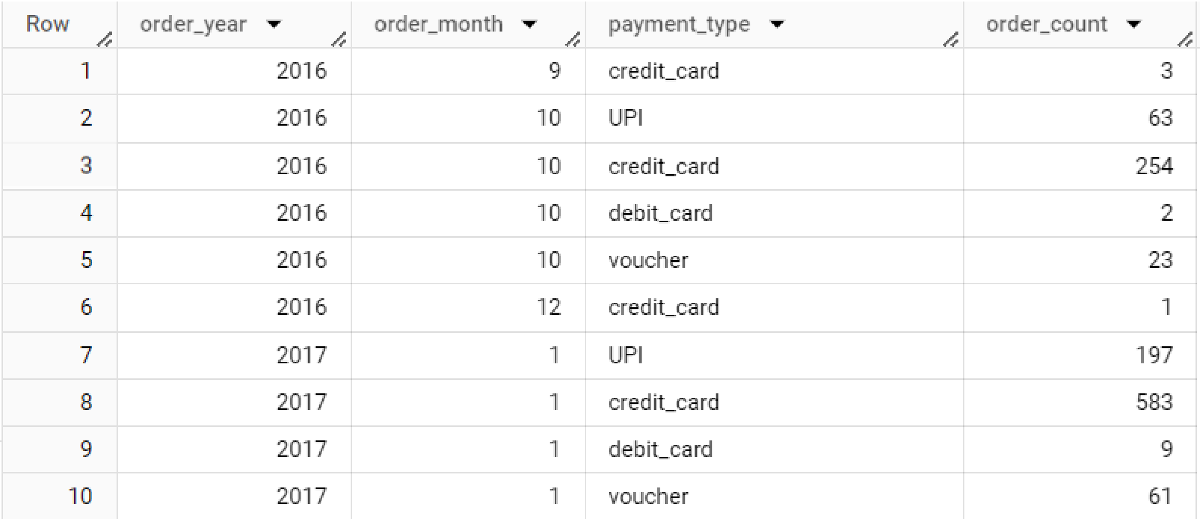
Row	customer_state	avg_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...
6	SC	21.47036877394...
7	RS	21.73580433039...
8	ES	22.05877659574...
9	GO	22.76681525932...
10	MS	23.37488400488...

Insight: Analyzing the average freight values for different states, it is apparent that São Paulo (SP) has the lowest average freight cost among the top 10 states, indicating potentially efficient logistics and shipping processes. Conversely, Mato Grosso do Sul (MS) has the highest average freight cost among the top 10 states. Understanding these variations in freight costs across different states can aid in devising strategies to optimize shipping costs, potentially leading to enhanced customer satisfaction and operational efficiency.

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

6. **Analysis based on the payments:**
   1. Find the month on month no. of orders placed using different payment types.

Ans.
```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
       payment_type,
       COUNT(*) AS order_count
FROM target-scaler-400905.target.payments
JOIN target-scaler-400905.target.orders ON payments.order_id =
orders.order_id
GROUP BY order_year, order_month, payment_type
ORDER BY order_year, order_month, payment_type;
```



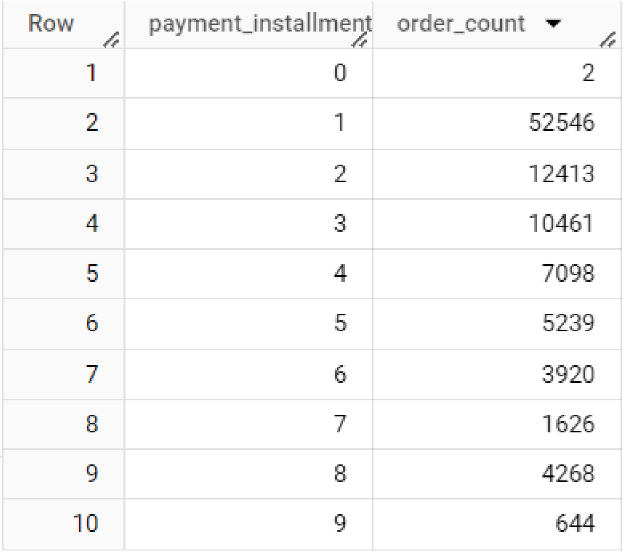
Row	order_year	order_month	payment_type	order_count
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197
8	2017	1	credit_card	583
9	2017	1	debit_card	9
10	2017	1	voucher	61

Insight: The dataset offers insights into the number of orders placed in various months of different years, categorized by payment types. For example, in October 2016, "credit_card" was the dominant payment type with 254 orders, followed by "voucher" (23 orders) and "UPI" (63 orders). This information is valuable for businesses to understand the preferred payment methods in different months and tailor their payment processing systems and promotions accordingly to optimize customer satisfaction and conversion rates.

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

Ans.
```
SELECT payment_installments,

       COUNT(*) AS order_count
FROM target-scaler-400905.target.payments
GROUP BY payment_installments
ORDER BY payment_installments;
```



Row	payment_installment	order_count
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Insight: The dataset showcases the number of orders based on the number of payment installments chosen by customers. It reveals a varied distribution of orders across different installment options. The majority of customers seem to prefer either no installments (0) or a single installment (1), contributing to a significant portion of the order count. However, a notable number of customers also opt for two or three installments, indicating their preference for spreading payments over multiple periods. Understanding customer preferences in payment installments is vital for businesses to design flexible payment options and enhance customer satisfaction.