Business Case: Target SQL

Q1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

a) Data type of all columns in the "customers" table.

Answer:

```
SELECT
    column_name,
    data_type
FROM
`directed-racer-396805.target.INFORMATION_SCHEMA.COLUMNS`
WHERE
table_name = 'customers';
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

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: The 'customers' table comprises of five columns , each with specific data type:
i. **'customer_id'**: A string type, which identifies each customer who has made the purchase.
ii. **'customer_unique_id'**: Also a string type, which uniquely identifies each customer.
iii. **'customer_zip_code_prefix':** An integer type, representing zip code of customer's location.
iv. **'customer_city' :** A string type, denoting the city from where the order is made.
v. **'customer_state':** Another string type, indicating the state code from where order is made (Eg. são paulo - SP).
This structured description provides a clear and concise overview of the customer's table's columns and their corresponding data types.

b) Get the time range between which the orders were placed.

Answer:

```
SELECT
```

```
  min(order_purchase_timestamp) as min_time,
  max(order_purchase_timestamp) as max_time
FROM `target.orders`;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

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

Insights: The available data for "target" in Brazil spans a period of precisely 2 years 43 days, commencing with the first recorded order on 04th September 2016 at 21:15:19 UTC and concluding with the most recent order for which detailed information is available, placed on 17th October 2018 at 17:30:18 UTC. This dataset encapsulates the business transactions and customer interactions with "target" over the specified time frame, offering a comprehensive view of their activities during this period.

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

Answer:
```
SELECT count(distinct c1.customer_id) as no_of_city,
       Count(distinct c1.customer_state) as no_of_state
FROM `target.orders` o1
       inner join `target.customers` c1
        on o1.customer_id = c1.customer_id
```

JOB INFORMATION | RESULTS | JSON

Row	no_of_city	no_of_state
1	99441	27

Insights: Within the time frame spanning from September 2016 at 21:15:19 UTC to 17th October 2018 at 17:30:18 UTC, a total of 99,441 unique cities and 27 distinct states have been observed among customers who have placed orders. This comprehensive dataset provides insights into the geographical diversity of customer locations and their distribution across various regions, enhancing our understanding of the market reach and customer base during the specific period.

Q2. In-depth Exploration:

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

Answer:
```
SELECT
       EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
```

```
    COUNT(order_id) AS order_count
FROM
     `target.orders`
GROUP BY
    order_year
ORDER BY 2 desc;
```

JOB INFORMATION | RESULTS | JSON

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

Insights: The data reveals a remarkable and continuous upward trajectory in the volume of orders placed over the course of several years i.e 2016-2018. In the initial year, 329 orders were recorded, demonstrating a substantial surge to 45,101 orders in the subsequent year, marking an impressive growth rate of 13,622.25%. This dramatic increase was followed by another notable surge in the following year, 2018, where the order count reached 54,011, reflecting a growth rate of 19.75% compared to the previous year, 2017.

Taken together, these figures underscore the substantial and sustained growth in order placements, resulting in an impressive overall growth percentage of 16,308.21% across the entire period under consideration. This growth trend not only reflects market dynamics but also underscores the increasing popularity and engagement of customers during this time frame.

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

Answer:

```
SELECT
      EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
      EXTRACT(Month FROM order_purchase_timestamp) AS order_month,
      COUNT(order_id) AS order_count
FROM
     `target.orders`
GROUP BY
      order_year,order_month
ORDER BY
      1,2;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION I

Row	order_year	order_month	order_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800

JOB INFORMATION | RESULTS | JSON | EXECUTION

Row	order_year	order_month	order_count
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: Yes we can see seasonality trend in the given dataset. They are as follows:

i. **Monthly Seasonality:** The data shows a clear monthly seasonality in the number of orders, particularly in 2017. Order counts start low in January and gradually increase each month until they peak in October (but in 2017 the peak month is November). We notice that December month is lowest for 2016 and 2017. This pattern suggests a strong seasonal trend, possibly related to holidays, promotions, or other factors specific to each month.

ii. **Growth Trend:** The year 2016 had a relatively low number of orders, with September and October being the first months with significant order activity. However, in 2017, there was a substantial increase in the number of orders, indicating overall business growth. In 2018 also we see significant growth but the max growth rate was in 2017.

iii. **Peak Month:** November 2017 stands out as the peak month for order counts with 7,544 orders. October 2016 stands out as the peak month with 324 orders and very good growth indicator as this was in initial months. January 2018 stands out as the peak month for order counts with 7,269 orders. This could be due to various factors, such as year-end shopping, holiday sales, or other promotional events.

iv. **Continuous Growth:** The order counts continued to grow throughout 2017 and 2018, demonstrating sustained customer engagement and an expanding customer base. This is a positive sign for the business.

v. **Possible Seasonal Promotions:** Given the pronounced monthly seasonality, it might be beneficial for the business to analyze the factors contributing to these peaks. Consider exploring whether specific marketing campaigns, holidays, or seasonal trends played a role in driving customer orders during certain months.

vi. **Forecasting and Inventory Management**: Understanding monthly seasonality can help in demand forecasting and inventory management. Businesses can prepare for increased demand during peak months and optimize their operations accordingly.

Overall, the data suggests a strong seasonal pattern in the number of orders, with significant growth in the business during the observed period. Further analysis and understanding of the underlying factors driving these trends can help businesses make informed decisions and capitalize on seasonal opportunities.

c) 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

Answer:

```
SELECT
      CASE
          WHEN order_hour BETWEEN 0 AND 6 THEN 'Dawn'
          WHEN order_hour BETWEEN 7 AND 12 THEN 'Morning'
          WHEN order_hour BETWEEN 13 AND 18 THEN 'Afternoon'
          WHEN order_hour BETWEEN 19 AND 23 THEN 'Night'
      END AS time_of_day,
      count(*) as order_count
 FROM (
      SELECT EXTRACT(HOUR FROM order_purchase_timestamp) AS order_hour
      FROM `target.orders`
      ) A1
group by 1
order by 2 desc;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTIO

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

Insights: Afternoon is the time at which the Brazilian customers mostly place their orders.

i. **Afternoon is the Most Active Time:** The "Afternoon" timeframe, which spans from 13:00 to 18:00, sees the highest order activity among customers i.e 38135 orders were placed at this time. This suggests that a significant portion of orders occurs during these hours.

ii. **Night is the Second Busiest Period**: The "Night" hours, ranging from 19:00 to 23:00, also show substantial order activity i.e 28331 orders. While it's slightly lower than the afternoon, it still represents a significant portion of orders.

iii. **Morning Orders:** The "Morning" period, covering hours from 07:00 to 12:00, demonstrates healthy order activity, although it's slightly lower than both the afternoon and night i.e 27733 orders. Many customers prefer to place orders during this time.

iv. **Dawn Has Fewer Orders:** The "Dawn" hours, which include the early morning hours from 00:00 to 06:00, have the lowest order activity compared to other timeframes i.e 5242. The order count during this time is notably lower.

These insights can be valuable for businesses to optimize their operations, such as adjusting staffing levels or running promotions during peak hours, and to better understand customer behavior patterns regarding order placements throughout the day.

Q3. Evolution of E-commerce orders in the Brazil region:

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

Answer:

```
WITH MonthlyOrders AS (

        SELECT
```

```
        EXTRACT(YEAR FROM o1.order_purchase_timestamp) AS order_year,
        EXTRACT(MONTH FROM o1.order_purchase_timestamp) AS order_month,
        c1.customer_state AS state,
        COUNT(o1.order_id) AS order_count
    FROM
        `target.orders` o1
        INNER JOIN
        `target.customers` c1
        ON
        o1.customer_id = c1.customer_id
    GROUP BY
        order_year,
        order_month,
        state
    ORDER BY
        order_year,
        order_month
)

    SELECT
        order_year,
        order_month,
        state,
        order_count,
        order_count - LAG(order_count) OVER (PARTITION BY state ORDER BY order_year,
order_month) AS month_on_month_change
    FROM
        MonthlyOrders
    ORDER BY
        order_year,
        order_month;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION GRA

Row	order_year	order_month	state	order_count	month_on_month_ch
1	2016	9	SP	2	null
2	2016	9	RS	1	null
3	2016	9	RR	1	null
4	2016	10	SP	113	111

order_year

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION G

Row	order_year	order_month	state	order_count	month_on_month_ch
5	2016	10	RS	24	23
6	2016	10	PE	7	null
7	2016	10	SC	11	null
8	2016	10	MT	3	null

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART PREVIEW | EXECUTION

Row	order_year	order_month	state	order_count	month_on_month_ch
9	2016	10	MA	4	null
10	2016	10	PB	1	null
11	2016	10	RR	1	0
12	2016	10	CE	8	null

Insights: We can see the variation in month-to-month change every year. It's increasing every year that indicates the positive growth in the sales.

i. **Null Values in "month_on_month_change":** The presence of null values in the "month_on_month_change" column for this month indicates that there is no data available for the previous month (September 2016) to calculate a month-on-month change. This is expected because there is no preceding month within the same year for October.

ii. State-Level Analysis: The data allows you to analyze the number of orders for different states. We can identify which states had higher or lower order counts during this specific month.

These insights provide a view of order trends from 2016-2018, with variations in month-on-month changes across different states, including both positive and negative changes. The data reflects the dynamic nature of order activity during this period.

b) How are the customers distributed across all the states?

Answer:

```
SELECT g1.geolocation_state as `state`,
       IFNULL(count(c1.customer_id),0) as cust_count
FROM `target.geolocation`g1
     left join `target.customers` c1
     on g1.geolocation_zip_code_prefix = c1.customer_zip_code_prefix
group by 1
order by 2 desc;
```

JOB INFORMATION | RESULTS | JSON | EXECU

Row	state	cust_count
1	SP	5620430
2	RJ	3015690
3	MG	2878728
4	RS	805370

JOB INFORMATION | RESULTS | JSON | EXECU

Row	state	cust_count
5	PR	626021
6	SC	538638
7	BA	365875
8	ES	316654

JOB INFORMATION | RESULTS | JSON | EXEC

Row	state	cust_count
9	GO	133146
10	MT	122395
11	PE	114588
12	DF	93309

Insights: We can see the spread of customers over 27 distinct states with SP state having the highest number of customers i.e 5620430 whereas RR state reports the lowest number of customers i.e 2087.

i. **Regional Customer Distribution:** The data reveals the distribution of customers across several states in the dataset. Each state has a varying number of customers, indicating different levels of customer engagement with the business in different regions.

ii. **Population Centers:** States with larger customer counts, such as SP, RJ and MG, may be considered population centers or regions with higher customer demand for the products or services offered.

iii. **Market Opportunities:** States with lower customer counts, such as AM, AP and RR, could represent potential market opportunities for the business. There may be room for growth and expansion in these regions.

iv. **Customer Retention:** For states with a significant customer base, customer retention and loyalty programs can be valuable to maintain and grow the customer relationships in those regions.

Further analysis of customer data from these states, such as age, gender, and purchase behavior, could provide insights into the demographics and preferences of customers in different regions.

Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

a) 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.

Answer:

```
WITH Final AS (

          SELECT
                FORMAT_DATE('%Y', o1.order_purchase_timestamp) AS order_year,
                FORMAT_DATE('%m', o1.order_purchase_timestamp) AS order_month,
                ROUND(SUM(p1.payment_value),2) AS total_cost
          FROM
               `target_casestudy.orders` o1
                INNER JOIN
               `target_casestudy.payments` p1
          ON
                o1.order_id = p1.order_id
          WHERE
               EXTRACT(YEAR FROM o1.order_purchase_timestamp) IN (2017, 2018)
               AND EXTRACT(MONTH FROM o1.order_purchase_timestamp) BETWEEN 1 AND 8
         GROUP BY
               order_year, order_month
          )

        SELECT
               order_year,
               order_month,
               total_cost,
        ROUND(
             100 * (total_cost - LAG(total_cost, 8) OVER (ORDER BY order_year, order_month))
/ LAG(total_cost, 8) OVER (ORDER BY order_year, order_month),2) AS percentage_increase
        FROM
            Final
        WHERE
            order_year IN ('2017', '2018')
        ORDER BY
            order_year, order_month;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXEC

Row	order_year	order_month	total_cost	percentage_increase
1	2017	01	138488.04	null
2	2017	02	291908.01	null
3	2017	03	449863.6	null
4	2017	04	417788.03	null

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXE

Row	order_year	order_month	total_cost	percentage_increase
5	2017	05	592918.82	null
6	2017	06	511276.38	null
7	2017	07	592382.92	null
8	2017	08	674396.32	null

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXE

Row	order_year	order_month	total_cost	percentage_increase
9	2018	01	1115004.18	705.13
10	2018	02	992463.34	239.99
11	2018	03	1159652.12	157.78
12	2018	04	1160785.48	177.84

Insights:

i. Fluctuating Order Costs: The total cost of orders fluctuates from month to month between January and August in 2017 and 2018. There is no consistent upward or downward trend, indicating that order costs vary throughout the period.

ii. Significant Increase in February: In January 2018, there was a substantial increase in the total cost of orders compared to January 2017. The total cost in January 2018 was 705.13% higher than in February 2017. It is the highest increase amongst all other in cost as compared to 2017(it is from Jan-Aug only). This significant increase may warrant further investigation to understand the factors contributing to it.

iii. Varied Monthly Trends: While there are fluctuations in order costs, some months, such as May and June 2018, show lower total costs with August 2018 as the lowest compared to their corresponding months in 2017. In contrast, months like March and April 2018 experienced higher costs.

iv. Seasonal Variation: The variations in order costs could be influenced by seasonal factors, marketing campaigns, or changes in consumer behavior. Further analysis is needed to identify the specific reasons behind these fluctuations.

v. Cost Control: Understanding the monthly changes in order costs allows businesses to better manage their expenses and make data-driven decisions. They can adjust pricing, marketing strategies, or supply chain operations based on these insights.

vi. Planning for Future Growth: Monitoring the percentage increase in order costs helps businesses plan for future growth and allocate resources effectively. Sudden spikes in costs may require adjustments in budgeting and operations.

In conclusion, analyzing the percentage increase in order costs on a monthly basis provides valuable insights into the dynamics of a business's operations. It helps identify trends, anomalies, and areas for improvement, ultimately contributing to informed decision-making.

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

Answer:

```
SELECT s1.seller_state,
       round(sum(od1.price),2) as `sum_of_price`,
       round( avg(od1.price),2) as `avg_of_price`
FROM `target.sellers` s1
      inner join `target.order_items` od1
      on s1.seller_id = od1.seller_id
      group by 1
      order by sum_of_price desc;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	seller_state	sum_of_price	avg_of_price
1	SP	8753396.21	108.95
2	PR	1261887.21	145.53
3	MG	1011564.74	114.6
4	RJ	843984.22	175.17

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	seller_state	sum_of_price	avg_of_price
5	SC	632426.07	155.2
6	RS	378559.54	172.15
7	BA	285561.56	444.11
8	DF	97749.48	108.73

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	seller_state	sum_of_price	avg_of_price
9	PE	91493.85	204.23
10	GO	66399.21	127.69
11	ES	47689.61	128.2
12	MA	36408.95	89.9

Insights: We can see the various results for 23 distinct states with SP at the top based on the sum of price and AC at the bottom based on sum of price.

i. **Highest Total Sales:** The state SP has the highest total sales among sellers, with a total sales value of approximately 8,551.69 units of currency. This suggests that there is significant sales activity from sellers located in this state.

ii. Highest Average Price per Order: PB has the highest average order item price among sellers, with an average price of approximately 449.87 units of currency. This indicates that sellers in PBtend to sell higher-priced products on average.

iii. **Varied Average Prices:** The average order item prices across states vary. For example, sellers in AM have an average price of approximately 392.33 units of currency, while sellers in PA have an average price of approximately 154.75 units. This suggests that the product offerings and pricing strategies of sellers vary by state.

iv. Lower Total Sales: AC has the lowest total sales value among sellers, with a total of 267.0 units of currency. This may indicate that there is relatively lower sales activity from sellers located in AC.

v.Uniform Average Price: Interestingly, despite lower total sales in AC, the average order item price is equal to the total sales value (267.0 units). This suggests that there may be a specific product or niche market in AC with a consistent order item price.

vi. Opportunities for Growth: States with lower total sales, AC and AP, may represent potential opportunities for sellers to expand their reach and increase sales in these regions.

vii. Market Segmentation: Variations in average order item prices indicate potential market segmentation, where sellers may tailor their products and pricing strategies to meet the preferences and purchasing power of customers in different states.

These insights provide a glimpse into the sales performance and pricing dynamics of sellers across different states, offering valuable information for business decision-making and market analysis.

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

Answer:

```
SELECT s1.seller_state,
       round(sum(od1.freight_value),2) as `sum_of_freight`,
       round( avg(od1.freight_value),2) as `avg_of_freight`
FROM `target.sellers` s1
       inner join `target.order_items` od1
```

```
        on s1.seller_id = od1.seller_id
group by 1
order by sum_of_freight desc;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	seller_state	sum_of_freight	avg_of_freight
1	SP	1482487.67	18.45
2	MG	212595.06	24.08
3	PR	197013.52	22.72
4	SC	106547.06	26.15

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	seller_state	sum_of_freight	avg_of_freight
5	RJ	93829.9	19.47
6	RS	57243.09	26.03
7	BA	19700.68	30.64
8	DF	18494.06	20.57

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	seller_state	sum_of_freight	avg_of_freight
9	GO	12565.5	24.16
10	PE	12392.46	27.66
11	ES	12171.13	32.72
12	MA	12141.29	29.98

Insights:

i. Highest Total Freight Costs: SP has the highest total freight costs among sellers, with a total freight cost of approximately 1482487.67 units of currency. This indicates that there is significant transportation activity associated with sellers located in SP.

ii. Highest Average Freight Cost per Order: RO has the highest average freight cost per order among sellers, with an average cost of approximately 50.91 units of currency. This suggests that sellers in RO tend to have relatively higher shipping costs per order.

iii. Varied Average Freight Costs: The average freight costs per order vary across states. For instance, sellers in AC have an average freight cost of approximately 32.84 units of currency, while sellers in PA have an average cost of approximately 19.39 units. These variations may reflect differences in shipping distances, courier services, or shipping policies.

iv. Lower Total Freight Costs: AC has the lowest total freight costs among sellers, with a total freight cost of 32.84 units of currency. This suggests that there is relatively lower transportation activity associated with sellers located in AC.

v. Uniform Average Freight Cost: Similar to the total freight cost, AC also has a uniform average freight cost of approximately 32.84 units of currency. This suggests consistent freight costs per order in AC.

vi. Logistics Considerations: The variation in freight costs across states highlights the importance of logistics and transportation in e-commerce operations. Sellers may need to consider transportation costs when setting product prices and expanding into different regions.

vii. Cost Efficiency: Sellers in states with higher average freight costs may need to focus on cost-efficient shipping solutions to maintain competitive pricing and customer satisfaction.

viii. Market Analysis: Understanding the relationship between freight costs and order value can provide insights into customer behavior and market dynamics. High shipping costs relative to product prices may impact purchasing decisions.

These insights provide valuable information for sellers to manage transportation costs, optimize logistics, and make informed decisions regarding pricing and market expansion.

Q5. Analysis based on sales, freight and delivery time.

a) 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.

Answer:

```
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`;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_deliver	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_deliver	diff_estimated_delive
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_deliver	diff_estimated_delive
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5
11	66057d37308e787052a32828...	38	-6
12	19135c945c554eebfd7576c73...	36	-2

Insights:

i. Delivery Time Analysis: Delivery times vary significantly, with some orders delivered earlier than the estimated date and others taking longer.

ii. Estimated vs. Actual Delivery:

a. Estimated Delivery Accuracy: Many orders were delivered close to the estimated delivery date, as indicated by values around 0 in diff_estimated_delivery. This suggests that the estimated delivery dates are reasonably accurate for these orders.

b.Early Deliveries: Some orders had negative values in time_to_deliver, indicating that they were delivered earlier than expected. This could be due to efficient logistics or shorter delivery distances.

c.Late Deliveries: On the other hand, positive values in time_to_deliver suggest that certain orders took longer to deliver than initially estimated. Late deliveries might result from logistical challenges or unexpected delays.

iii. Order Fulfillment Optimization:

a. Efficiency Improvement: Analyzing orders with positive values in time_to_deliver can help identify areas where delivery processes can be optimized to reduce delays and improve efficiency.

b. Customer Satisfaction: Monitoring delivery times and ensuring orders are delivered close to or earlier than estimated can contribute to higher customer satisfaction.

iv. Tracking and Reporting: These metrics can be used for tracking delivery performance over time, identifying trends, and generating reports for stakeholders. For instance, businesses can track whether improvements in logistics or processes lead to reduced delivery times.

v. Data-Driven Decisions: By having insights into delivery times and estimated vs. actual delivery performance, businesses can make data-driven decisions to enhance their supply chain operations, manage customer expectations better, and optimize resource allocation.

vi. Customer Experience: Understanding delivery times allows businesses to communicate effectively with customers regarding expected delivery dates, helping manage customer expectations and enhance their overall experience.

In conclusion, the analysis of delivery times and the comparison between estimated and actual delivery dates provide valuable insights for businesses to improve their delivery processes, optimize logistics, and enhance customer satisfaction.

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

Answer:

```
WITH Final AS (
            SELECT
                 s1.seller_state AS state,
                 ROUND(AVG(od1.freight_value),2) AS avg_freight
            FROM
                 `target.sellers` s1
                  INNER JOIN
                 `target.order_items` od1
                  ON
                  s1.seller_id = od1.seller_id
            GROUP BY
                  state
      )
SELECT A1.state,
       A1.avg_freight
FROM
```

```
(SELECT
       state,
       avg_freight,
       DENSE_RANK() OVER (ORDER BY avg_freight DESC) AS high_freight_rank,
       DENSE_RANK() OVER (ORDER BY avg_freight ASC) AS low_freight_rank
 FROM
       Final)A1
 WHERE
       A1.high_freight_rank <= 5 OR A1.low_freight_rank <= 5
 ORDER BY
      avg_freight DESC;
```

JOB INFORMATION | RESULTS | JSON | EXECUTI

Row	state	avg_freight
1	RO	50.91
2	CE	46.38
3	PB	39.19
4	PI	36.94
5	AC	32.84

JOB INFORMATION | RESULTS | JSON | EXECU

Row	state	avg_freight
6	PR	22.72
7	DF	20.57
8	RJ	19.47
9	PA	19.39
10	SP	18.45

Insights: The top five states with highest average freight value are RO, CE, PB, PI, AC whereas the top five states with lowest average freight value are PR, DF, RJ, PA and SP. The top most value of average freight is 50.91 and the lowest value of average freight is 18.45.

These insights can help businesses make informed decisions about logistics, pricing, and distribution strategies in different regions of Brazil. Understanding the factors influencing freight costs is crucial for optimizing supply chain operations and ensuring competitive pricing for customers.

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

Answer:

```
WITH Final AS (

            SELECT
```

```
                c1.customer_state AS state,
                ROUND(AVG(DATE_DIFF(o1.order_delivered_customer_date,
o1.order_purchase_timestamp, DAY)),2) AS avg_time_to_deliver
            FROM
              `target.customers` c1
               INNER JOIN
              `target.orders` o1
               ON
              c1.customer_id = o1.customer_id
            GROUP BY
              state
)
SELECT A1.state,
      A1.avg_time_to_deliver
FROM
     (SELECT
            state,
            avg_time_to_deliver,
      DENSE_RANK() OVER (ORDER BY avg_time_to_deliver DESC) AS high_rank,
      DENSE_RANK() OVER (ORDER BY avg_time_to_deliver ASC) AS low_rank
     FROM
           Final)A1
     WHERE
         A1.high_rank <= 5 OR A1.low_rank <= 5
     ORDER BY
         avg_time_to_deliver DESC;
```

JOB INFORMATION | RESULTS | JSON | EXECU

Row	state	avg_time_to_deliver
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

JOB INFORMATION | RESULTS | JSON | EXECU

Row	state	avg_time_to_deliver
6	SC	14.48
7	DF	12.51
8	MG	11.54
9	PR	11.53
10	SP	8.3

Insights: The top five states with highest average delivery time are RR, AP, AM, AL, PA whereas the top five states with lowest average delivery time are SC, DF, MG, PR and SP. The top most value of average freight is 28.98 and the lowest value of average freight is 8.3.

These insights can assist businesses in optimizing their logistics and delivery strategies in different regions of Brazil, taking into account the varying delivery times and potential factors influencing them.

d) 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.

Answer:

```
WITH Final AS (

          SELECT
               c1.customer_state AS state,
               ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date
, DAY)),2) AS avg_est_delivery
          FROM
              `target.customers` c1
               INNER JOIN
              `target.orders` o1
               ON
               c1.customer_id = o1.customer_id
          GROUP BY
               state
            )
       SELECT A1.state,
              A1.avg_est_delivery
       FROM
          (SELECT
                 state,
                 avg_est_delivery,
                 DENSE_RANK() OVER (ORDER BY avg_est_delivery ASC) AS low_rank,
           FROM
                Final)A1
           WHERE
                A1.low_rank <= 5
           ORDER BY
                avg_est_delivery DESC;
```

JOB INFORMATION | RESULTS | JSON | EXECU

Row	state	avg_est_delivery
1	BA	9.93
2	ES	9.62
3	SE	9.17
4	MA	8.77
5	AL	7.95

Insights: AL completes the top 5 with an average delivery time 7.95 days ahead of the estimated date. It is the state with fastest delivery. Customers in AL experience faster delivery times, potentially due to streamlined logistics.

These insights can help businesses and logistics providers identify regions in Brazil where orders are consistently delivered ahead of schedule, providing a positive customer experience and potentially leading to higher customer satisfaction.

Q6. Analysis based on the payments:

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

Answer:

```
WITH Final AS (

        SELECT
              EXTRACT(YEAR FROM o1.order_purchase_timestamp) AS order_year,
              EXTRACT(MONTH FROM o1.order_purchase_timestamp) AS order_month,
              p1.payment_type,
              COUNT(*) AS no_of_orders
        FROM
             `target.orders` o1
              INNER JOIN
             `target.payments` p1
              ON
              o1.order_id = p1.order_id
        GROUP BY
              order_year,
              order_month,
              p1.payment_type
            )

        SELECT
              order_year,
              order_month,
              payment_type,
              no_of_orders,
```

```
            LAG(no_of_orders, 1) OVER (PARTITION BY payment_type ORDER BY order_year,
order_month) AS prev_month_orders
        FROM
            Final
        ORDER BY
            order_year,
            order_month,
            payment_type;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXECUTION

Row	order_year	order_month	payment_type	no_of_orders	prev_month_orders
1	2016	9	credit_card	3	null
2	2016	10	UPI	63	null
3	2016	10	credit_card	254	3
4	2016	10	debit_card	2	null

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXECUTION

Row	order_year	order_month	payment_type	no_of_orders	prev_month_orders
5	2016	10	voucher	23	null
6	2016	12	credit_card	1	254
7	2017	1	UPI	197	63
8	2017	1	credit_card	583	1

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW | EXECUTION

Row	order_year	order_month	payment_type	no_of_orders	prev_month_orders
9	2017	1	debit_card	9	2
10	2017	1	voucher	61	23
11	2017	2	UPI	398	197
12	2017	2	credit_card	1356	583

Results per page: 50

Insights:

i. Payment Types: The data includes several payment types such as "voucher," "UPI," "credit_card," and "debit_card."

ii. Month-to-Month Trends: The "no_of_orders" column shows the number of orders placed for each payment type in a given month. The "prev_month_orders" column shows the number of orders for the same payment type in the previous month.

iii. Month-over-Month Changes: We can observe the changes in the number of orders for each payment type from one month to the next. For example, in rows 19 and 20, there is a decrease in the number of "UPI" orders from April (496 orders) to May (772 orders).

iv. Different Payment Types: The data allows you to compare how different payment types perform over time. We can see how the number of orders for each payment type fluctuates month by month.

v. Potential Analysis: With this data, we can perform further analysis to identify trends, patterns, or anomalies in the ordering behavior based on payment types. For instance, we can calculate growth rates or identify which payment type is the most popular over time. We see that highest number of transactions are done with credit cards which is 5000 plus in a month whereas the lowest number of transactions are done with debit cards.

Overall, this data can be valuable for understanding the dynamics of customer payment preferences and how they change month by month. Further analysis and visualization can provide deeper insights into customer behavior and payment trends.

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

Answer:

```
SELECT
     payment_installments,
     COUNT(DISTINCT order_id) AS no_of_orders
FROM
    `target.payments`
WHERE
    payment_sequential = payment_installments
GROUP BY
    payment_installments
ORDER BY
    payment_installments;
```

Row	payment_installment	no_of_orders
1	1	48236
2	2	53
3	3	1

Insights:

i. Single Payment Installment (1): There are 48,236 orders where customers chose to pay for their purchases in a single installment. This is the most common payment option, indicating that a significant portion of customers prefers to pay for their orders all at once.

ii. Two Payment Installments (2): There are 53 orders where customers opted to pay in two installments. While this option is less common compared to a single installment, it

suggests that some customers may choose to spread their payments over two installments for larger orders.

iii. Three Payment Installments (3): There is only one order where the payment was divided into three installments. This is the least common option among the analyzed orders, indicating that the majority of customers prefer one or two installments for payment.

These insights provide an understanding of the distribution of payment installments chosen by customers when placing orders. Most customers prefer a single payment, while only a few make all the payments of installments. Only 53 paid the 2 installments and have opted for 2 installments, with three installments being the least common paid. So we should avoid giving more than 2 installments options as people generally don’t pay the installments. People have taken till 24 installments but nobody has paid them.