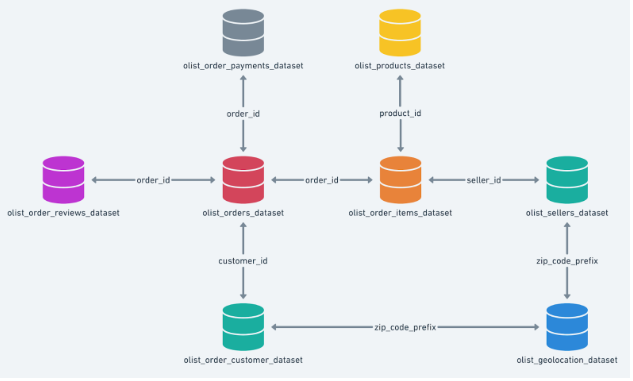
**Business case : Target SQL**

**Dataset Schema :**

****

**The data is available in 8 csv files:**

1. customers.csv
2. sellers.csv
3. order\_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

**1.1 Data type of all columns in the "customers" table.**

**Approach :**

Here we are displaying all the columns that are present in the customers table.

**SELECT**

**column\_name,**

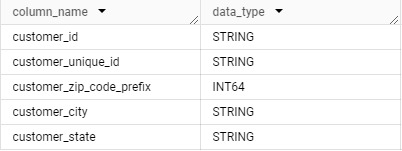
**data\_type**

**FROM**

**`target\_ecomm.INFORMATION\_SCHEMA.COLUMNS`**

**WHERE**

**TABLE\_NAME = "customers"**

****

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

**Approach :**

If we want to get the time range in which the orders were placed we need to get the first order date and last order date from the orders table.For getting the first and last order we will be using Min and Max aggregate functions.

**SELECT**

**MIN(order\_purchase\_timestamp) AS `First\_order`,**

**MAX(order\_purchase\_timestamp) AS `Last\_order`**

**FROM**

**`target\_ecomm.orders`**

****

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

**period.**

**Approach :**

If we want to get the cities and states of customers who ordered during the given time period some times customer may signup and may not order for this case we want Join the customers table with orders table using customer\_id.Apply the filter for given time period , Get the customer state and customer city from the table and create a grouped table on the above two columns customer\_state and customer \_city. Use the aggregate COUNT function to count the number of orders in each city and state.

**SELECT**

**COUNT(DISTINCT c.customer\_city) AS `Customer\_city\_count`,**

**COUNT(DISTINCT c.customer\_state) AS `Customer\_state\_count`**

**FROM**

**target\_ecomm.customers c INNER JOIN target\_ecomm.orders o**

**ON c.customer\_id = o.customer\_id**

**WHERE**

**order\_purchase\_timestamp BETWEEN "2016-09-04" AND "2018-10-17";**



**Insights :**

São Paulo has the most orders by a significant margin.There are over 15,000 orders from

São Paulo, while the next highest city (Rio de Janeiro) has only about 6,800 orders. This

suggests that São Paulo is a major market for your business in Brazil. Several other cities

have notable number of orders.

**Recommendations :**

Consider dedicating a larger portion of your marketing budget to cities like São Paulo, Rio

de Janeiro, Belo Horizonte, Brasília, Curitiba, and Campinas. These areas have a higher

customer base and potentially more opportunity for growth.

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

**Approach :**

We want to see the No.of orders growth over the past years, Extract year from the

order\_purchase\_timestamp, Group by year , Count the number of orders in that year using

COUNT aggregate function , Applying line chart on the result.

**SELECT**

**EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `Year`,**

**COUNT(\*) AS `No\_of\_orders\_placed\_in\_the\_Year`**

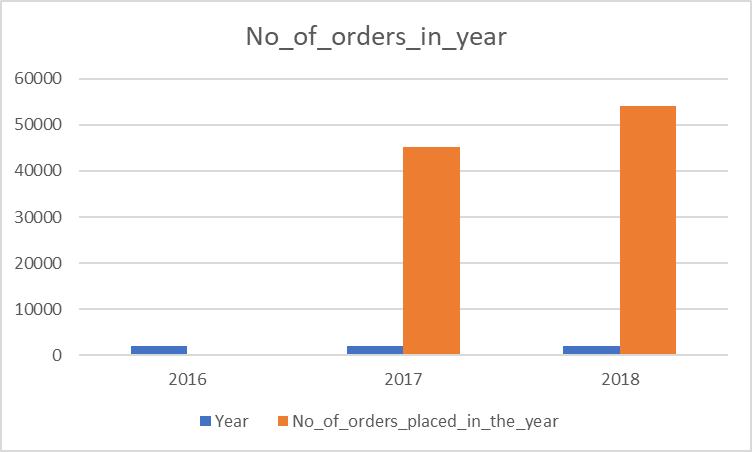
**FROM**

**target\_ecomm.orders**

**GROUP BY 1**

**ORDER BY year;**

****

****

**Insights :**

In 2016 we are having only 3 months of data and in 2017 and 2018 we are having all month's data when we see orders growth significantly increasing from year to year.

**Recommendations :**

The orders keep on increasing year over year.so , better to keep the same strategies and try to implement some kind of discounts to attract more new customers.

**Another way to see the above growth is in terms of month in the every year**

**SELECT**

**EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `year`,**

**EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `month`,**

**COUNT(\*) AS `No\_of\_orders`**

**FROM**

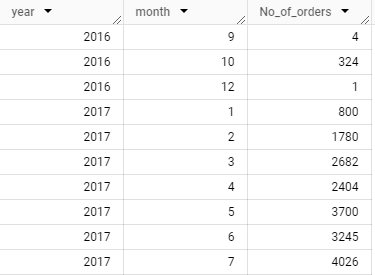
**target\_ecomm.orders**

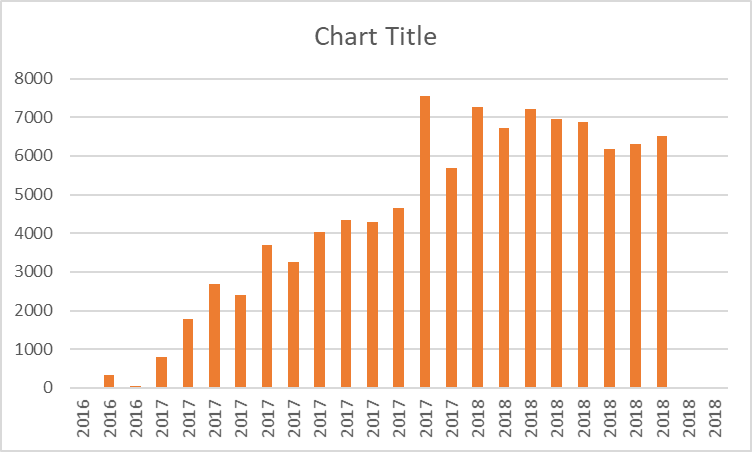
**GROUP BY**

**1, 2**

**ORDER BY**

**year , month;**

****

****

In this method we are analyzing the number of orders in the month over the past years. Here we can see that we are having only 3 months data in the year 2016 .

**Insights :**

Here we are having inconsistent dates we can not compare.And there appears to be a seasonal trend in the data with the number of orders consistently increasing from month 0 to month 7 then decreasing from month 8 to month 12. This could be due to a number of factors such as inconsistent data and the product being more popular in certain times of the year or other may be some offers or discounts may be provided in those months. However, the number of orders appears to be higher in the year 2018 than in the year 2017 .

**Recommendations :**

If the seasonal trend is real, the company could try to take advantage of it by increasing the marketing efforts during the months when sales are lower. For example, they could offer discounts or promotions during those months. If the increase in sales is due to better marketing, the company could continue to invest in marketing.

**2.2 Can we see some kind of monthly seasonality in terms of the no. of orders**

**being placed?**

**Approach :**

If we want monthly seasonality in terms of no\_of\_orders placed in that Month Irrespective

of the year we just want to see the no of orders in the Month and their growth. Extract

month from the order\_purchase\_timestamp Group by month, Count the no of orders using

COUNT aggregate function

**SELECT**

**EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,**

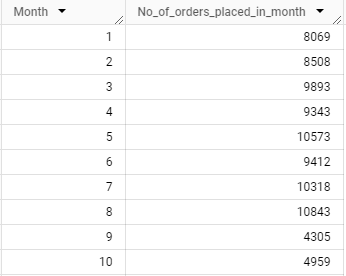
**COUNT(\*) AS `No\_of\_orders\_placed\_in\_month`**

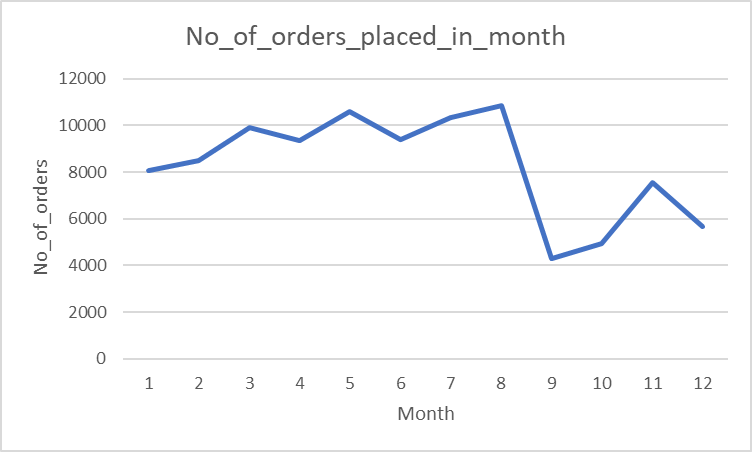
**FROM**

**target\_ecomm.orders**

**GROUP BY 1**

**ORDER BY Month;**

****

****

**Insights :**

The data that we had provided is not equal for all the months in one year. Some years are missing and same with the other year as well .There appears to be an upward trend in the number of orders placed throughout the years. From month 1 to month 8 we can see the sales are increasing , but there is a slight dip in the number of orders in the months 9 and 10. It's possible there is a seasonal trend at play, with sales significantly lower in month 9 and 10 compared to the rest of the months.

**Recommendations :**

Look into what might have caused the significant drop in sales in the months 9 and 10 , were there any marketing promotions or changes in product availability during that time? Consider Promotions During Lower Sales Months: If there is a seasonal trend, then consider running promotions or discounts during months 9 and 10 to boost sales during that time.

**2.3 During what time of the day, do the Brazilian customers mostly place their**

**orders? (Dawn, Morning, Afternoon or Night).**

**Approach :**

Brazilian customers places their orders in different timings they are 0 - 6 - Dawn , 7 - 12 - Morning , 13 - 18 - Afternoon , 19 - 23 - Night Using Case When statements we will be giving the name like above one And we will be grouping them to get the count of orders in that time.

**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 "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 `time\_of\_ordering`,**

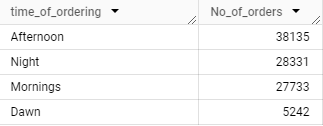
**COUNT(order\_id) AS `No\_of\_orders`**

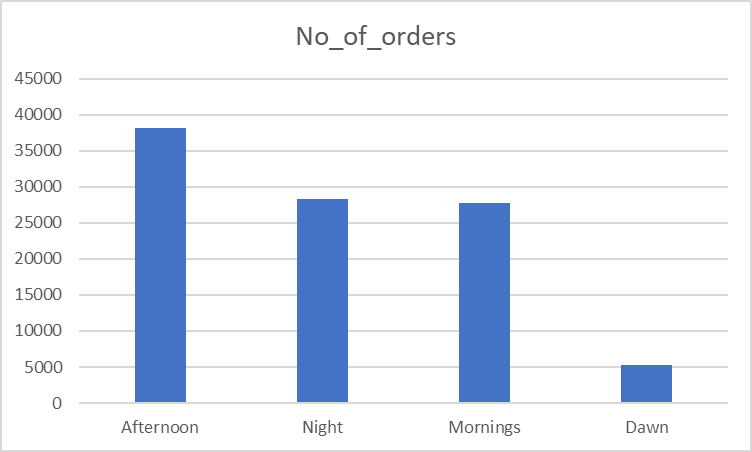
**FROM**

**target\_ecomm.orders**

**GROUP BY 1**

**ORDER BY No\_of\_orders DESC;**

****

****

**Insights :**

The highest number of orders were placed in the afternoon at 38,135. This suggests that a

significant portion of Brazilian customers prefer to shop online in the afternoon, possibly

during lunch breaks or after work. Night is the second most popular time for orders with

28,331 orders. This indicates that many Brazilian customers also shop online in the evening,

which could be due to having more free time to browse and shop after work or other

obligations. Then followed by morning and dawn.

**Recommendations :**

Focus marketing and advertising efforts on reaching Braziliancustomers during the

afternoon and evening since these are the times when they are most likely to be shopping

online. Explore offering special deals or discounts during afternoons and evenings to

incentivize purchases during these peak periods.

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

**Approach :**

We want to perform month on month analysis on no of orders placed in each state.Before

starting we need to join the customers table with orders table using customer\_id column and

then Extract month from order\_purchase\_timestamp column form orders table. Get

customer\_state column from the customers table and Group by both the extracted month

column and customer state column Use COUNT aggregate function to count the no of

orders placed in each state.

**SELECT**

**EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,**

**customer\_state,**

**COUNT(\*) AS `orders\_placed`**

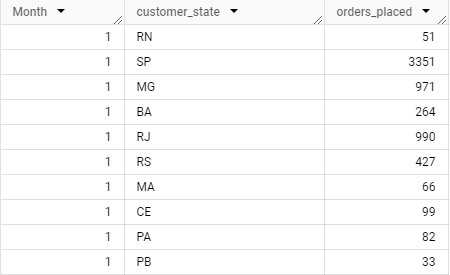
**FROM**

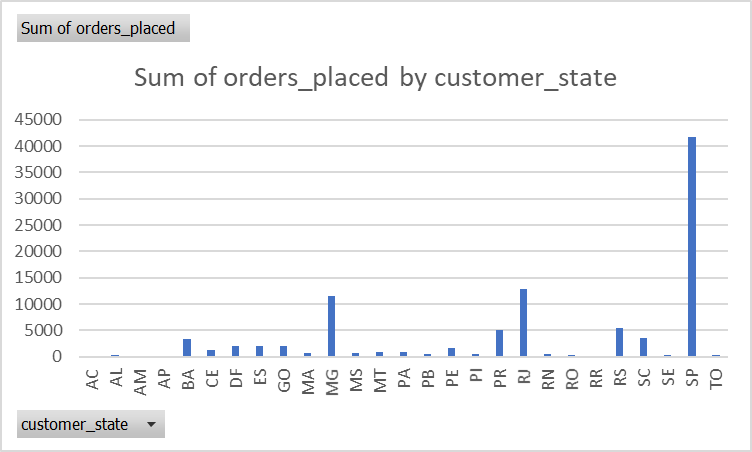
**target\_ecomm.customers c INNER JOIN target\_ecomm.orders o**

**ON c.customer\_id = o.customer\_id**

**GROUP BY 1,2**

**ORDER BY 1**

****

****

**Insights :**

The above chart provides the insights of month on month number of orders placed in various states. SP , BA, MS stand out with the highest number of orders placed across multiple months.These states to be significant market. Some states such as RN and ES show relatively consistent numbers of orders.States like AM and SE display fluctuating order numbers , indicating variability in demand or external factors affecting order volumes. Several states like PB (Paraíba) and PI (Piauí) have consistently low order numbers.

**Recommendations :**

Focus on high activity states like SP, BA, MS should be primary targets for marketing.Strategies could include targeted promotions.For states with consistent but lower order volumes like RN and ES,consider strategies to incrementally increase orders. This could include partnership with local businesses, enhancing distribution channels, or improving product availability.

**3.2 How are the customers distributed across all the states?**

**Approach :**

Here in this case we want to get the number of customers distributed across the different states. Get the customer\_state column from the customers tableGroup by customer\_state to get the distinct states where customers are present. Use the COUNT aggregate function to get the count of Unique customers across different states.

**SELECT**

**customer\_state,**

**COUNT(DISTINCT customer\_id) AS `No\_of\_customers`**

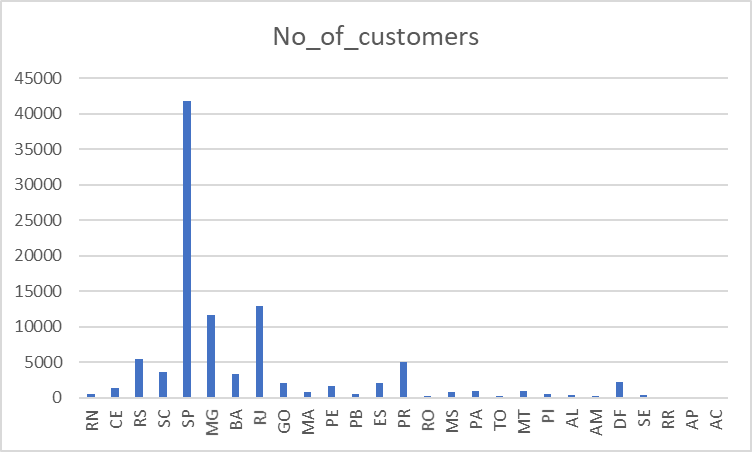
**FROM**

**target\_ecomm.customers**

**GROUP BY**

**customer\_state**

****

****

**Insights :**

SP has an overwhelmingly higher number of customers compared to other states. Indicating

It is a key market with significant customer base. We can say this is a dominated state. MG

and RJ states also have substantial customer bases, though much smaller than SP.

**Recommendations :**

Given SP the most dominant customer base , continue to invest heavily in this state, and

introduce premium offerings to capitalize on this large market. With significant customer

numbers in MG and RJ, focus on expanding market share. Tailor marketing campaigns to

regional preferences.

**4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include**

**months between Jan to Aug only).**

**Approach :**

We want to calculate the % increase in the cost of orders from 2017 to 2018 and the months

between Jan to Aug only Cost of orders is known as payment\_value which is present in the

payments table. Join the orders table with payments table to extract year and month from

the column order\_purchase\_timestamp and join the table using order\_id. Filter out the

orders that we placed between the months January and August in the years of 2017 and

2018. Group by year to get the 2 years and use an aggregated SUM function to calculate the

total payment value of that year. Make the above query as CTE to calculate the percentage

increase in cost of Orders. We can use LAG / LEAD functions to calculate the percentage

increase. Create a new column named prev\_year\_payment\_value using LAG function.

Formula : percentage\_increase = (total\_value - prev\_total\_value) / prev\_total\_value × 100

**WITH `result` AS (**

**SELECT**

**EXTRACT(YEAR FROM order\_purchase\_timestamp) AS `year`,**

**SUM(payment\_value) AS `total\_payment\_value`**

**FROM**

**target\_ecomm.orders o**

**INNER JOIN target\_ecomm.payments p ON o.order\_id = p.order\_id**

**WHERE**

**EXTRACT(YEAR FROM order\_purchase\_timestamp) IN (2017, 2018)**

**AND EXTRACT(MONTH FROM order\_purchase\_timestamp) BETWEEN 1 AND 8**

**GROUP BY**

**EXTRACT(YEAR FROM order\_purchase\_timestamp)**

**ORDER BY year ASC**

**)**

**SELECT**

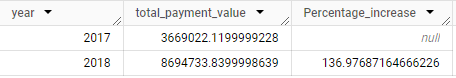
**\*,**

**((total\_payment\_value - LAG(total\_payment\_value, 1) OVER (ORDER BY year ASC)) / LAG(total\_payment\_value, 1) OVER (ORDER BY year ASC) \* 100) AS `Percentage\_increase`**

**FROM**

**result**

**ORDER BY year ASC**

****

**Insights :**

We have taken only 2017 and 2018 data from the month 1 to 8.The total payment value

increased from 3,669,022.12 in 2017 to 8,694,733.84 in 2018, marking a substantial growth

of approximately 137%.

**Recommendations :**

Invest in marketing and customer acquisition strategies to sustain and capitalize on the

significant growth trend.

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

**Approach :**

We want to calculate the Total and Average value of order price for each state. Price column

is available in the order\_items table. We want to get the state as well right so we want to

join three tables: customers , orders and order\_items. Fetch the customer\_state column

from customer table and group by state Use the aggregate SUM function to get the total

order value and aggregate AVG function to get the average of total order value across the

different states.

**SELECT**

**customer\_state,**

**SUM(i.price) AS `Total\_order\_value`,**

**AVG(i.price) AS `Average\_order\_value`**

**FROM**

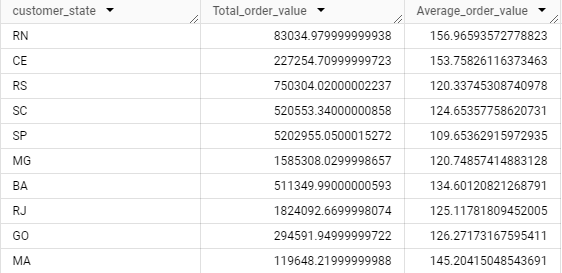
**target\_ecomm.customers c INNER JOIN target\_ecomm.orders o**

**ON c.customer\_id = o.customer\_id**

**INNER JOIN target\_ecomm.order\_items i**

**ON o.order\_id = i.order\_id**

**GROUP BY 1**

****

**Insights :**

The data reveals trends in customer order value across various states.State SP has the highest total order value. This could be due to factors like large population , higher average income or wider availability of premium products and state SP having the most dominant customer base. State PB with the highest average order value,this suggests customers in PB tend to spend more per order compared to other states This might be due to various reasons one of them is a smaller number of total orders(which can inflate the average).

**Recommendations :**

Targeted marketing in the higher order value states ex: states like SP with high total order value represents a profitable market.Consider allocating more marketing budget to these states To improve sales in the lower average order value , consider promotional offers or discounts. These can attract new customers and exciting ones to increase their spending.

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

**Approach :**

We want to calculate the total and average freight value for each state freight \_value is nothing but Price rate at which a product is delivered from one point to another.We want to join by three table customers, orders and order\_items. Fetch the customer\_state column from the customers table and group by the customer\_state column. Use the aggregate SUM and AVG functions to calculate total and average freight value.

**SELECT**

**customer\_state,**

**SUM(i.freight\_value) AS `Total\_freight\_value`,**

**AVG(i.freight\_value) AS `Average\_freight\_value`**

**FROM**

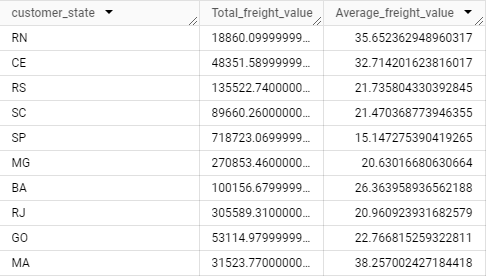
**target\_ecomm.customers c INNER JOIN target\_ecomm.orders o**

**ON c.customer\_id = o.customer\_id**

**INNER JOIN target\_ecomm.order\_items i**

**ON o.order\_id = i.order\_id**

**GROUP BY 1**

****

**Insights :**

State SP has the highest total order freight value , suggesting a large volume of goods are shipped compared to other states. State RR has the highest average order freight value , indicating that orders in this state might contain heavier or bulk items on an average.

**Recommendations :**

For states with high total order freight value : we can negotiate bulk discounts with the carriers Given the large volume of shipments to SP, negotiate discounted rates with carriers based on your consistent business. This can significantly reduce your overall freight costs.

Investigate the possibility of partnering with warehouses or fulfillment centers in SP. Storing inventory closer to customers reduces shipping distances and potentially lowers freight costs.

**5.1 Find the no. of days taken to deliver each order from the order’s purchase**

**date as delivery time. Also, calculate the difference (in days) between the**

**estimated & actual delivery date of an order.**

**Approach :**

Here there are 2 cases where we want to find the delivery time of product delivered to initial order purchase time and another case is we want to find the difference in days between estimated delivery date and product delivered. Time\_to\_deliver\_order = order\_delivered\_customer\_date - order\_purchase\_timestamp , Difference\_estimated\_delivery = order\_delivered\_customer\_date - order\_estimated\_delivery\_date. Use the DATE\_DIFF function to calculate the difference between 2 dates Fetch all those above things from the orders table.

**SELECT**

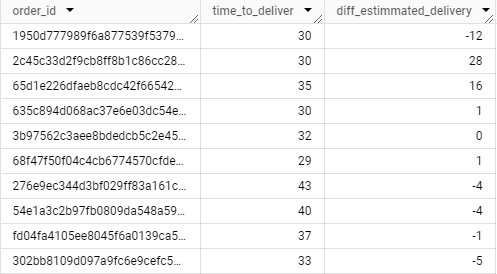
**DISTINCT order\_id,**

**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\_estimmated\_delivery`**

**FROM**

**target\_ecomm.orders;**

****

**Insights :**

The delivery time varies across the orders ranging from 1 day to 200 days. This could be

due to factors like product availability , shipping distance and order fulfillment workload.

We have some differences in between the order estimated delivery date and order delivered

date for some orders, products are delivered earlier than the estimated delivery date and for

some other orders were delayed than the estimated delivery date.

**Recommendations :**

Consider providing more accurate or conservative estimated delivery times to customers.

This can help manage expectations and avoid potential dissatisfaction if deliveries take

longer than anticipated.In cases where delivery delays are unavoidable, proactively

communicate with customers to explain the reasons for the delay and keep them updated on

the revised delivery timeframe. Transparency and clear communication can help maintain

customer trust and satisfaction.

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

**Approach :**

Here we want to calculate the top 5 states with highest and lowest average freight value.We

want to join 3 table customers , orders and order\_items to get all the required data. We want

to fetch the customer\_state column from the customers table and Group by the

customer\_state column to create a grouped table. Get the Average of freight value using

AVG aggregate function. Make the above query as CTE From CTE we will fetch the top 5

and lowest states with average freight value.

**WITH `result` AS (**

**SELECT**

**c.customer\_state,**

**AVG(i.freight\_value) AS `Avg\_freight\_value`**

**FROM**

**target\_ecomm.order\_items i INNER JOIN target\_ecomm.orders o**

**ON i.order\_id = o.order\_id**

**INNER JOIN target\_ecomm.customers c**

**ON o.customer\_id = c.customer\_id**

**GROUP BY**

**c.customer\_state**

**)**

**This query is for getting the top 5 states with highest Average freight value**

**SELECT**

**customer\_state,**

**Avg\_freight\_value**

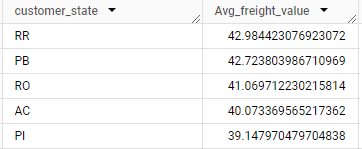
**FROM**

**result**

**ORDER BY**

**Avg\_freight\_value DESC**

**LIMIT 5**

****

**Now this query is for getting the top 5 states with lowest Average freight value**

**SELECT**

**customer\_state,**

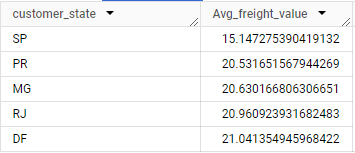
**Avg\_freight\_value**

**FROM**

**result**

**ORDER BY Avg\_freight\_value ASC**

**LIMIT 5**

****

**Insights :**

There are significant differences in average freight value across different states. This could be due to variations in , quantity of goods shipped. States with higher average freight value might have a larger volume of goods shipped on average , States with higher average freight value might have heavier or bulkier items being shipped on average.

**Recommendations :**

Are there specific product categories driving this, or is it due to other factors like customer demographics (e.g., a higher concentration of business customers ordering in bulk)? If feasible, consider collaborating with local fulfillment centers or carriers in states with high average freight value. This could potentially reduce shipping distances and freight costs.

Analyze product types and packaging practices in states with lower average freight value.

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

**Approach :**

We want to find the top 5 highest states with the highest and lowest average delivery time in terms of days. We want to join the customers table with the orders table to fetch the customer\_state and Difference in date. After joining the tables get the customer\_state column and group by using customer\_state to get the grouped table. After this use AVG aggregate function and DATE\_DIFF function to get the difference in days to get the average across different states. Make Order by Desc and limit 5 to get the top 5 highest average delivery time states. Order by Asc will give you the lowest average delivery time states

**This query will give you the top 5 states with highest average delivery time**

**SELECT**

**customer\_state,**

**AVG(DATETIME\_DIFF(order\_delivered\_customer\_date ,**

**order\_purchase\_timestamp , DAY)) AS `Avg\_time\_to\_deliver`**

**FROM**

**target\_ecomm.orders o INNER JOIN target\_ecomm.customers c**

**ON o.customer\_id = c.customer\_id**

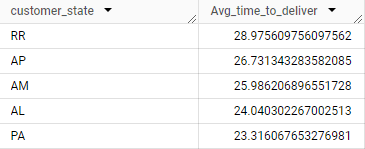
**GROUP BY**

**customer\_state**

**ORDER BY**

**Avg\_time\_to\_deliver DESC**

**LIMIT 5**

****

**This query will give you the top 5 states with lowest average delivery time**

**SELECT**

**customer\_state,**

**AVG(DATETIME\_DIFF(order\_delivered\_customer\_date , order\_purchase\_timestamp , DAY)) AS `Avg\_time\_to\_deliver`**

**FROM**

**target\_ecomm.orders o INNER JOIN target\_ecomm.customers c**

**ON o.customer\_id = c.customer\_id**

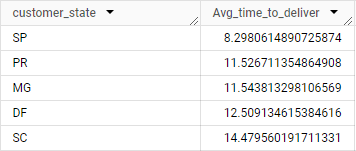
**GROUP BY**

**customer\_state**

**ORDER BY**

**Avg\_time\_to\_deliver ASC**

**LIMIT 5**

****

**Insights :**

We are getting the date of the top 5 states with the highest average delivery time and lowest average delivery time. States with the highest average delivery time could be various reasons the goods they are ordering in bulk quantity or they might be ordering from faraway places for the premium products and there won’t be any premium service delivery partners. States with lowest average delivery time indicates that there might be less products to deliver or there might be some inventory management stores near to their places to store the products which they regularly order and they might have some premium delivery services.

**Recommendations :**

Strategic Warehouse Network Design: Analyze your sales data and delivery time trends to identify potential gaps in your fulfillment network. Consider establishing or expanding

fulfillment centers in states with high average delivery times to shorten shipping distances and improve delivery speed for those regions.

**5.4 Find out the top 5 states where the order delivery is really fast as**

**compared to the estimated date of delivery.**

**Approach :**

We need to find the top 5 states where the order delivery is really fast as compared to the estimated delivery date. We need to join the orders table with customers table using customer\_id Calculate the difference between two dates(order\_delivered\_customer\_date

- order\_estimated\_delivery\_date) using DATE\_DIFF function. Get the customer\_state column and group by using the same column. Then find the average of diff\_estimated\_delivery using aggregate AVG Function. Now do order by diff\_estimated\_delivery in ASC to get the deliveries which are faster than the estimated delivery. To get the top 5 rows use LIMIT 5.

**SELECT**

**customer\_state,**

**AVG(DATE\_DIFF(order\_delivered\_customer\_date , order\_estimated\_delivery\_date, DAY)) AS `diff\_estimated\_delivery`**

**FROM**

**target\_ecomm.orders o INNER JOIN target\_ecomm.customers c**

**ON o.customer\_id = c.customer\_id**

**GROUP BY**

**customer\_state**

**ORDER BY**

**diff\_estimated\_delivery ASC**

**LIMIT 5**

****

**Insights :**

Here we got the top 5 states where the order delivery is really fast as compared to the estimated date of delivery. Negative values indicate that orders are delivered faster than the expected Delivery. ex : if our estimated delivery date is 22-06-2024 but the order delivered on 20-06-2024 it delivered 5 days prior to the estimated delivery date so (-5). So for all the above states deliveries are delivered earlier than the estimated delivery date.

**Recommendations :**

Orders might be delayed in some situations and might deliver earlier than estimated delivery date in some situations this could be due to various reasons not providing order estimated delivery date and try to update the customer incase of delay in order delivery . If the same products are ordered from the same place try to keep inventory stores near to that place where we can deliver the products fast to the customer.

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

**types.**

**Approach :**

Here we want to perform month on month analysis to count no of orders placed using different payment methods.Join the payments table with orders table using order\_id

Get the payment\_type column from the payments table and Extract month from the order\_purchase\_timestamp. Group by both payment\_type column and extracted month.

Count the no of orders using the COUNT aggregate function.

**SELECT**

**payment\_type,**

**EXTRACT(MONTH FROM order\_purchase\_timestamp) AS `Month`,**

**COUNT(\*) AS `No\_of\_orders`**

**FROM**

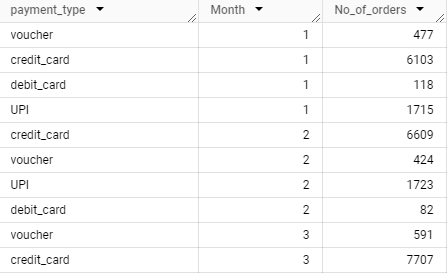
**target\_ecomm.payments P INNER JOIN target\_ecomm.orders o**

**ON P.order\_id = o.order\_id**

**GROUP BY**

**payment\_type , 2**

**ORDER BY Month ASC**

****

**Insights :**

Credit cards consistently have the highest number of orders across all the months and the number of orders keep on increasing month over month.

**Recommendations :**

Since credit cards are seeing substantial month-on-month growth, it would be beneficial to continue promoting this payment method, possibly with additional rewards or cashback offers. Given the stable yet minimal growth, ensuring a seamless and secure UPI transaction experience could help in maintaining or even increasing the usage.

**6.2 Find the no. of orders placed on the basis of the payment installments**

**that have been paid.**

**Approach :**

Here we want to get the number of orders placed on the basis of payment installments that have been paid. Fetch the payment\_instalments column from the payments table Apply filter that if the installment has been paid it it should be greater than or equal to 1.Then group by the payment\_instalments column. Apply COUNT aggregate function to get the no of orders.

**SELECT**

**payment\_installments,**

**COUNT(\*) AS `No\_of\_orders`**

**FROM**

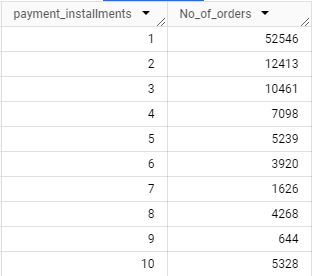
**target\_ecomm.payments**

**WHERE**

**payment\_installments >= 1**

**GROUP BY**

**payment\_installments**

****

**Insights :**

Here we can observe that most of the payments paid in single instalments only.

**Recommendations :**

We recommend giving more offers if they pay bill in single instalments where they get the payment in the single instalment.

In this section, I present a comprehensive analysis based on the data provided. This analysis encompasses a detailed examination of the dataset, highlighting key patterns, trends, and insights.