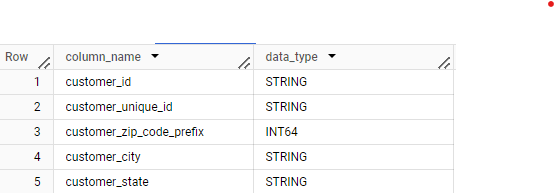
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**
   1. Data type of all columns in the “customers” table

ANS 1A:

select column\_name, data\_type

from scaler-dsml-target-sql1.Target\_SQL.INFORMATION\_SCHEMA.COLUMNS

where table\_name = 'customers'



INSIGHTS: By the code, we can see most of the data is in String type.

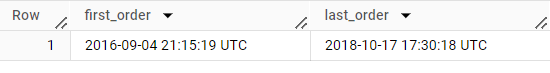
* 1. Get the time range between which the orders were placed.

ANS 1B:

select min(order\_purchase\_timestamp) as first\_order,

       max(order\_purchase\_timestamp) as last\_order

from Target\_SQL.orders



INSIGHTS: First order in our dataset was in 2016-09-04 and last order was in 2018-10-17

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

ANS 1C:

select count(distinct customer\_state) states, count(distinct customer\_city) as cities

from `Target\_SQL.orders` o

inner join `Target\_SQL.customers` c

on o.customer\_id = c.customer\_id

where o.order\_purchase\_timestamp between '2016-01-01 00:00:00' and '2018-12-12 00:00:00'

****

INSIGHT: Total States are 27 and Cities are 4119 in the given dataset.

**II. In-depth Exploration:**

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

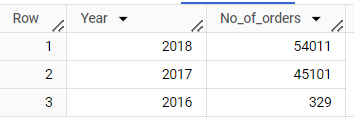
ANS 2A:

select extract(year from order\_purchase\_timestamp) as Year, count(order\_id) as No\_of\_orders

from `Target\_SQL.orders`

group by 1

order by Year desc



INSIGHTS: The growing trend that has emerged over the past years. Our solution aims to meet the demands of this trend and provide customers with a unique and valuable offering.

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

ANS 2B:

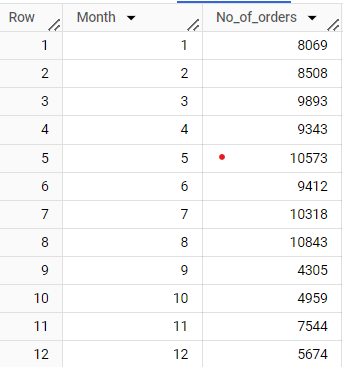
select extract(month from order\_purchase\_timestamp) as Month,

       count(order\_id) as No\_of\_orders

from `Target\_SQL.orders`

group by 1

order by 1



INSIGHTS: Yes, we can see the monthly seasonality in orders placed. By leveraging this valuable insight, businesscan strategically plan inventory, allocate resources, and implement targeted marketing campaigns to maximize profitability.

* 1. 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 2C:

select count(order\_id) as no\_of\_orders,

       case when (a.time between '00:00:00' and '06:59:59') then 'Dawn'

       when (a.time between '07:00:00' and '12:59:59') then 'Morning'

       when (a.time between '13:00:00' and '18:59:59') then 'Afternoon'

       when (a.time between '19:00:00' and '23:59:59') then 'Night' end as Time\_of\_the\_day

from (select \*,O.customer\_id, extract(time from order\_purchase\_timestamp) as time

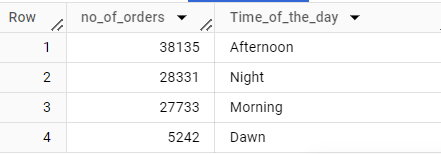
from `Target\_SQL.customers` C

inner join `Target\_SQL.orders` O

on C.customer\_id = O.customer\_id) a

group by 2

order by 1 desc



INSIGHTS: Brazilian consumers, with the majority placing their orders in the afternoon. With this valuable insight, our platform enables businesses to maximize their sales potential by strategically targeting this specific time frame.

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

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

ANS 3A:

select customer\_state, extract(month from order\_purchase\_timestamp) as Month, count(order\_id) as no\_of\_orders

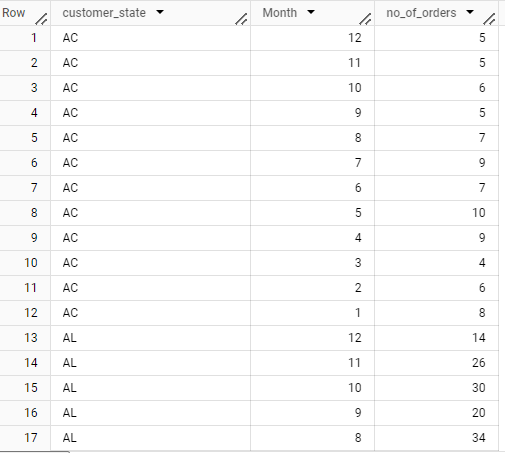
from `Target\_SQL.customers` C

inner join Target\_SQL.orders O

on C.customer\_id = O.customer\_id

group by 1,2

order by 1,2 desc



INSIGHT: This provides real-time insights into the fluctuating monthly order volumes placed by Brazilian customers across different states.

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

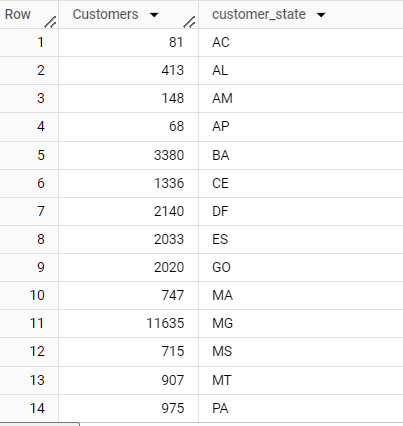
ANS 3B:

select count(customer\_id) as Customers, customer\_state

from Target\_SQL.customers

group by 2

order by customer\_state asc



INISIGHT: Got 27ROWS in the output for each state. By the innovative solution for businesses with customers distributed across various states.

1. **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 4A:

with final as

(select extract(year from order\_purchase\_timestamp) as Year,

        round(sum(payment\_value)) as cost\_of\_orders

from `Target\_SQL.orders` o

inner join `Target\_SQL.payments` p

on o.order\_id = p.order\_id

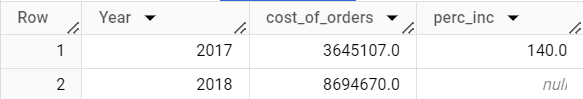
where order\_purchase\_timestamp between '2017-01-01' and '2017-08-31' or order\_purchase\_timestamp between '2018-01-01' and '2018-08-31'

group by 1)

select Year, cost\_of\_orders, 100\*round((lead(cost\_of\_orders) over (order by Year)-cost\_of\_orders)/cost\_of\_orders,1) as perc\_inc

from final

order by 1



INSIGHTS: Percentage increase in cost of orders in the year 2017 is 140 but there is no percentage increase in the year 2018.There might be fluctuations in the orders placed.

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

ANS 4B:

select round(sum(price)) as Total\_order\_price, round(avg(price)) as Avg\_order\_price, customer\_state

from `Target\_SQL.order\_items` Oi

inner join `Target\_SQL.orders` O

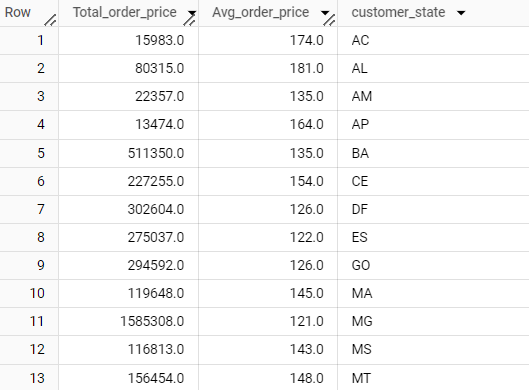
on Oi.order\_id = O.order\_id

inner join `Target\_SQL.customers` C

on O.customer\_id = C.customer\_id

group by 3

order by 3 asc



INSIGHT: Got 27ROWS in the output. These are the total and average order prices for each state.

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

ANS 4C:

select round(sum(freight\_value)) as Total\_order\_freight\_value, round(avg(freight\_value)) as Avg\_order\_freight\_value, customer\_state

from `Target\_SQL.order\_items` Oi

inner join `Target\_SQL.orders` O

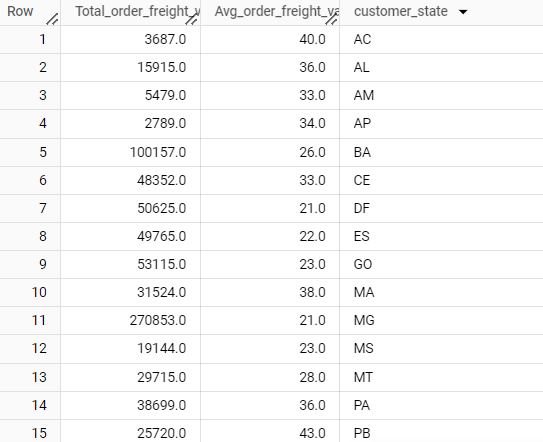
on Oi.order\_id = O.order\_id

inner join `Target\_SQL.customers` C

on O.customer\_id = C.customer\_id

group by 3

order by 3 asc



INSIGHTS: Got 27ROWS in the output. These are the total and average freight values for each state.

1. **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

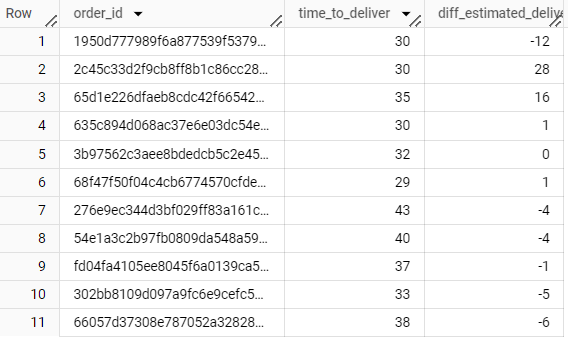
ANS 5A:

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\_estimated\_delivery

from Target\_SQL.orders



INSIGHTS: Got many rows in the output. We can see days taken to deliver the order and difference in estimated for each order varies. If the order delivered before the estimated date, customers prefer to order regularly.

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

ANS 5B:

(select customer\_state,

        round(avg(freight\_value),2) as avg\_freight\_value,

        'low' as level\_of\_state

from Target\_SQL.order\_items Oi

inner join `Target\_SQL.orders` O

on Oi.order\_id = O.order\_id

inner join Target\_SQL.customers C

on O.customer\_id = C.customer\_id

group by 1

order by avg\_freight\_value asc

limit 5)

union distinct

(select customer\_state,

        round(avg(freight\_value),2) as avg\_freight\_value,

        'high' as level\_of\_state

from Target\_SQL.order\_items Oi

inner join `Target\_SQL.orders` O

on Oi.order\_id = O.order\_id

inner join Target\_SQL.customers C

on O.customer\_id = C.customer\_id

group by 1

order by avg\_freight\_value desc

limit 5)



INSIGHTS: Lowest and highest average freight value of states are listed with values.

By this we get to know which states has high and low freight values.

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

ANS 5C:

(select customer\_state, 'high' as level\_of\_state, round(avg(date\_diff(order\_delivered\_customer\_date, order\_purchase\_timestamp, day)),3) as time\_to\_deliver

from Target\_SQL.order\_items Oi

inner join `Target\_SQL.orders` O

on Oi.order\_id = O.order\_id

inner join Target\_SQL.customers C

on O.customer\_id = C.customer\_id

group by 1

order by time\_to\_deliver desc

limit 5)

union all

(select customer\_state, 'low' as level\_of\_state, round(avg(date\_diff(order\_delivered\_customer\_date, order\_purchase\_timestamp, day)),3) as time\_to\_deliver

from Target\_SQL.order\_items Oi

inner join `Target\_SQL.orders` O

on Oi.order\_id = O.order\_id

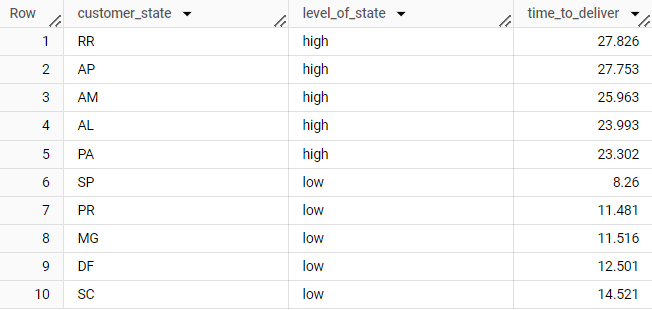
inner join Target\_SQL.customers C

on O.customer\_id = C.customer\_id

group by 1

order by time\_to\_deliver asc

limit 5)



INSIGHTS: In the above code, the time taken to deliver the order is mentioned. Less time to deliver the order should be preferred by the customers.

* 1. 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.

ANS 5D:

select customer\_state,

round(avg(date\_diff(order\_estimated\_delivery\_date, order\_delivered\_customer\_date, day)),2) as Estimated\_date

from `Target\_SQL.orders` O

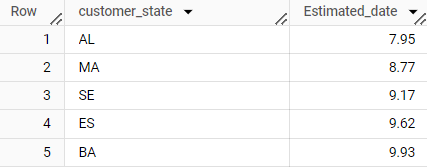
inner join `Target\_SQL.customers` C

on O.customer\_id = C.customer\_id

group by 1

order by Estimated\_date asc

limit 5



INSIGHTS: AL,MA,SE,ES,BA are the states with fast in delivering the order than estimated. It is suggested that other states should follow same to deliver.

1. **Analysis based on the payments:**

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

ANS 6A:

select extract(month from b.order\_purchase\_timestamp) as month\_on\_month, count(b.order\_id) as no\_of\_orders, payment\_type

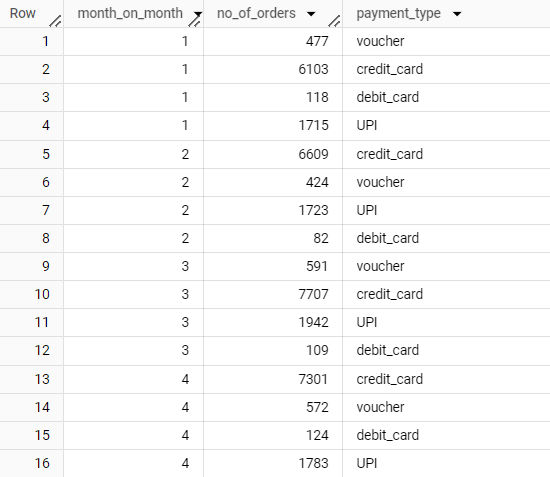
from Target\_SQL.payments a

INNER JOIN `Target\_SQL.orders` b

on a.order\_id = b.order\_id

group by 1,3

order by 1



INSIGHT: Got 50ROWS. Customers has used different payment types for orders.

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

ANS 6B:

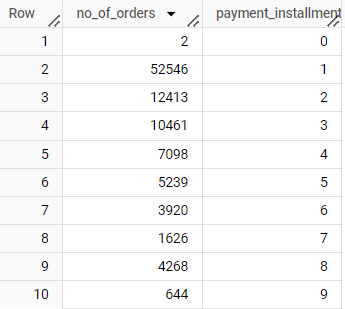
select count(payment\_installments) as no\_of\_orders, payment\_installments

from Target\_SQL.payments a

INNER JOIN `Target\_SQL.orders` b

on a.order\_id = b.order\_id

group by 2



INSIGHTS: By the code we can observe number of orders placed in payment installments.

Customers mostly preferred installments for their payments.