1. **Initial exploratory analysis:**

We’ll first try to explore the data, understand it and answer some simple questions.

**1.a. Data type of all columns in the “customers” table.**

SELECT

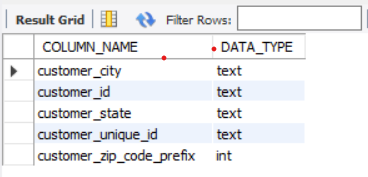
COLUMN\_NAME,

DATA\_TYPE

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_NAME = 'customers'



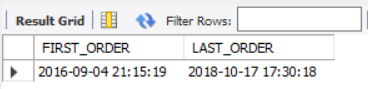
**1.b. Get the time range between which the orders were placed.**

SELECT

MIN(order\_purchase\_timestamp) as FIRST\_ORDER , MAX(order\_purchase\_timestamp) as LAST\_ORDER

FROM

target.orders



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

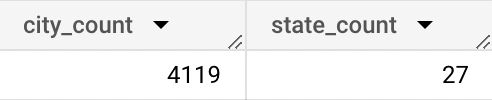
SELECT

COUNT(DISTINCT customer\_city) AS total\_cities,

COUNT(DISTINCT customer\_state) AS total\_states

FROM

target.customers



1. **In-depth Exploration:**

We’ll try to understand the trend in the data and see how things have changed for the data that we have over the course of time.

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

SELECT

YEAR(order\_purchase\_timestamp) as Year, MONTH(order\_purchase\_timestamp) as Month,

COUNT(DISTINCT order\_id) as total\_orders

FROM

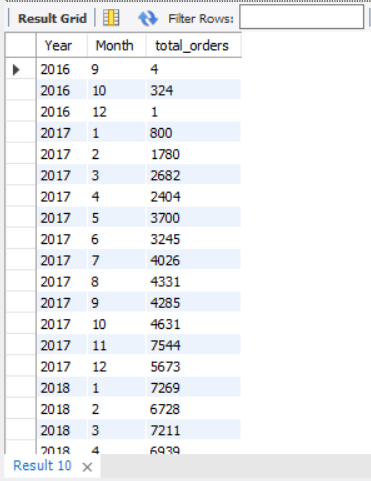
target.orders

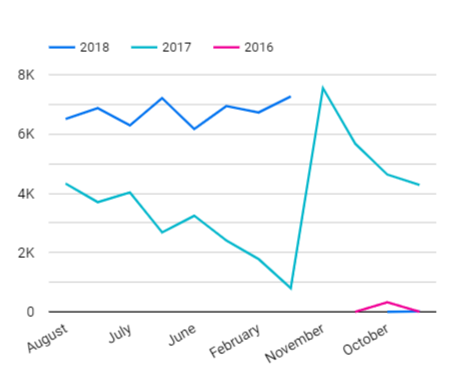
GROUP BY

YEAR(order\_purchase\_timestamp), MONTH(order\_purchase\_timestamp)

ORDER BY

YEAR(order\_purchase\_timestamp) , MONTH(order\_purchase\_timestamp)





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

SELECT

YEAR(order\_purchase\_timestamp) as Year, MONTH(order\_purchase\_timestamp) as Month,

COUNT(DISTINCT order\_id) as total\_orders

FROM

target.orders

GROUP BY

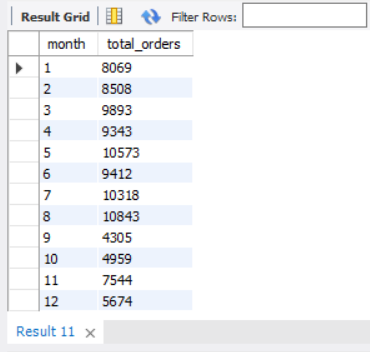
YEAR(order\_purchase\_timestamp),

MONTH(order\_purchase\_timestamp)

ORDER BY

YEAR(order\_purchase\_timestamp) ,

MONTH(order\_purchase\_timestamp)



**2.c. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)**

SELECT

CASE WHEN HOUR(order\_purchase\_timestamp) BETWEEN 0 AND 6 THEN 'dawn'

WHEN HOUR(order\_purchase\_timestamp) BETWEEN 7 AND 12 THEN 'morning'

WHEN HOUR(order\_purchase\_timestamp) BETWEEN 13 AND 18 THEN 'afternoon'

WHEN HOUR(order\_purchase\_timestamp) BETWEEN 19 AND 23 THEN 'night'

END AS time\_of\_day,

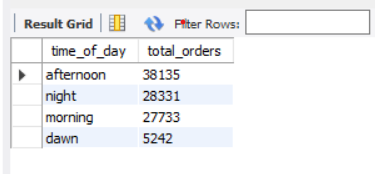
COUNT(\*) AS total\_orders

FROM

target.orders

GROUP BY 1

ORDER BY 2 DESC



Brazilian's customers tend to buy more in the afternoons.

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

Now we’ll try to understand data based on state or city level and see what variations are present and how the people in various states order and receive deliveries.

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

SELECT

MONTH(o.order\_purchase\_timestamp) as mnth, c.customer\_state,

COUNT(DISTINCT o.order\_id) as total\_orders

FROM

target.customers as c

INNER JOIN

target.orders as o

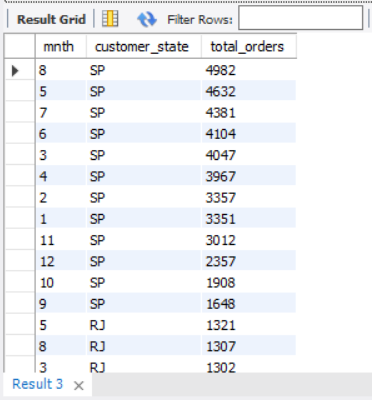
ON c.customer\_id = o.customer\_id

GROUP BY

MONTH(o.order\_purchase\_timestamp), c.customer\_state

ORDER BY

total\_orders DESC



**3.b. How are the customers distributed across all the states?**

SELECT

customer\_state,

COUNT(DISTINCT customer\_unique\_id) as total\_customers

FROM

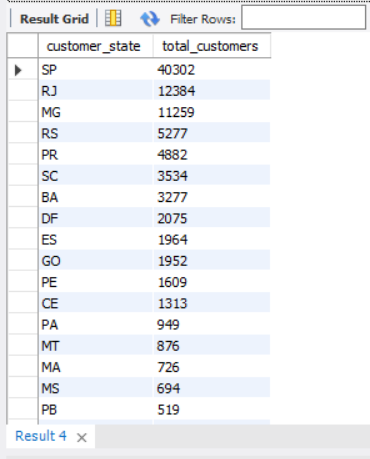
target.customers

GROUP BY

customer\_state

ORDER BY

total\_customers DESC



1. **Impact on Economy:**

Until now, we just answered questions on the E-commerce scenario considering the number of orders received. We could see the volumetry by a month, day of week, time of the day and even the geolocation states.

Now, we will **Analyze the money movement by e-commerce by looking at order prices, freight and others.**

**4.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.**

WITH CTE AS(

SELECT

YEAR(o.order\_purchase\_timestamp) as year, ROUND(SUM(p.payment\_value),2) as cost

FROM

target.orders as o

INNER JOIN

target.payments as p

ON o.order\_id = p.order\_id

WHERE

YEAR(o.order\_purchase\_timestamp) BETWEEN 2017 AND 2018

AND

MONTH(o.order\_purchase\_timestamp) BETWEEN 1 AND 8

GROUP BY

YEAR(o.order\_purchase\_timestamp)

ORDER BY

YEAR(o.order\_purchase\_timestamp)

)

SELECT

year , cost,

LAG(cost,1) OVER(ORDER BY year asc) as next\_year\_cost,

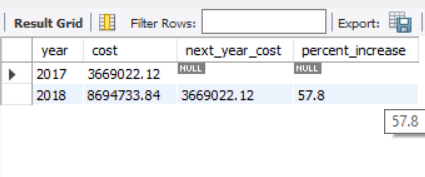
ROUND((100.0\*(cost - LAG(cost,1) OVER(ORDER BY year asc))/ cost),2) as percent\_increase

FROM

CTE

ORDER BY

Year



**4.b. Year & Month wise Analysis on Price per order , Freight per order**

* + **price\_per\_order = sum(price)/count(order\_id)**
  + **freight\_per\_order= sum(freight\_value)/count(order\_id)**
  + **Group the data on yearly and monthly level4**

SELECT YEAR(o.order\_purchase\_timestamp) as year,

MONTH (o.order\_purchase\_timestamp) as mth ,

SUM(oi.price)/COUNT(DISTINCT oi.order\_id) as price\_per\_order,

SUM(oi.freight\_value)/COUNT(DISTINCT oi.order\_id) as freight\_per\_order

FROM

target.order\_items as oi

INNER JOIN

target.orders as o

ON

oi.order\_id = o.order\_id

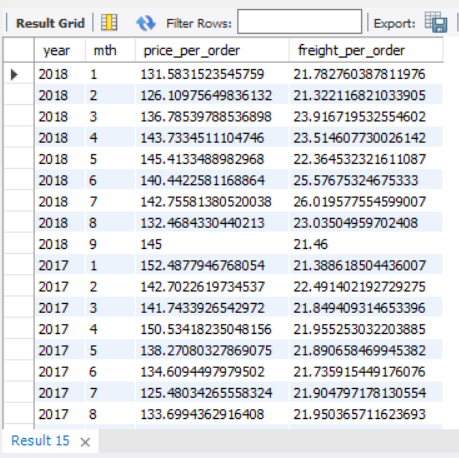
GROUP BY

YEAR(o.order\_purchase\_timestamp)

, MONTH (o.order\_purchase\_timestamp)

ORDER BY

year DESC



**4.c. Calculating the Total & Average value of order price for each state.**

WITH CTE AS(

SELECT

c.customer\_state,

ROUND(SUM(oi.price),2) as total\_price,

COUNT(DISTINCT o.order\_id) as num\_orders

FROM

target.order\_items as oi

INNER JOIN

target.orders as o

ON

oi.order\_id = o.order\_id

INNER JOIN

target.customers as c

ON

o.customer\_id = c.customer\_id

GROUP BY

c.customer\_state

)

SELECT

customer\_state as state,

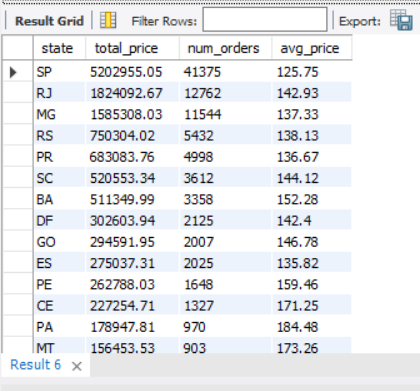
total\_price, num\_orders,

ROUND(total\_price/num\_orders,2) as avg\_price

FROM

CTE

ORDER BY total\_price DESC



It's very interesting to see how some states have a high total amount sold and a low price per order. If we look at SP (São Paulo) for example, it's possible to see that it is the state with most valuable state for e-commerce (5202955 sold) but it is also where customers pay less per order (125.75 per order)

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

WITH CTE AS(

SELECT

c.customer\_state,

ROUND(SUM(oi.freight\_value),2) as total\_freight,

COUNT(DISTINCT o.order\_id) as num\_orders

FROM

target.order\_items as oi

INNER JOIN

target.orders as o

ON

oi.order\_id = o.order\_id

INNER JOIN

target.customers as c

ON

o.customer\_id = c.customer\_id

GROUP BY

c.customer\_state

)

SELECT

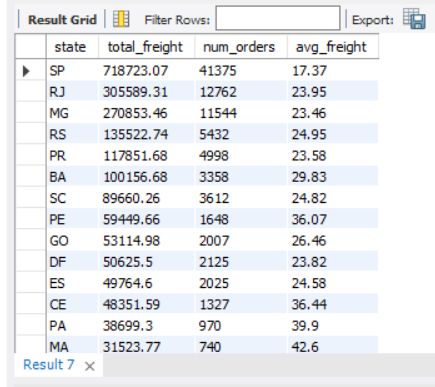
customer\_state as state,

total\_freight, num\_orders, ROUND(total\_freight/num\_orders,2) as avg\_freight

FROM

CTE

ORDER BY total\_freight DESC



1. **Analysis based on sales, freight and delivery time.**

**5.a. No. of days taken to deliver each order as delivery time.**

**Also, calculating the difference (in days) between the estimated & actual delivery date of an order.**

**We’ll be Using these two formulas to gain the Inferences.**

* **time\_to\_deliver** = order\_delivered\_customer\_date - order\_purchase\_timestamp
* **diff\_estimated\_delivery** = order\_delivered\_customer\_date - order\_estimated\_delivery\_date

SELECT

order\_id,

TIMESTAMPDIFF(DAY,order\_purchase\_timestamp,order\_delivered\_customer\_date) as delivery\_days\_taken,

TIMESTAMPDIFF(DAY,order\_delivered\_customer\_date,order\_estimated\_delivery\_date) as differnce\_estimated\_del

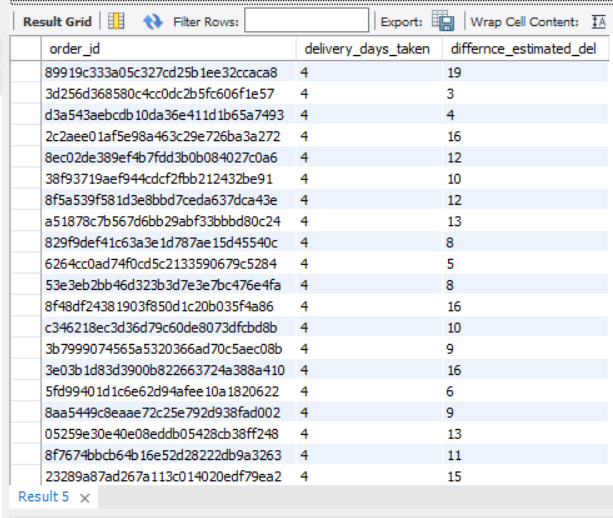
FROM

target.orders

WHERE

order\_status = 'delivered'

ORDER BY delivery\_days\_taken

****

**5.b. Top 5 states with the lowest average delivery time.**

SELECT

c.customer\_state,

ROUND(SUM(ABS(DATEDIFF(o.order\_purchase\_timestamp,o.order\_delivered\_customer\_date)))/COUNT(DISTINCT o.order\_id),2) AS AVG\_DEL

FROM

target.customers as c

INNER JOIN

target.orders as o

ON

c.customer\_id = o.customer\_id

WHERE

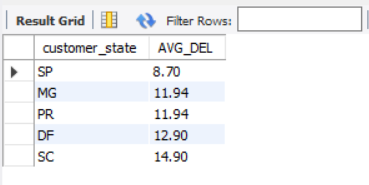
o.order\_status = 'delivered'

GROUP BY

c.customer\_state

ORDER BY AVG\_DEL

LIMIT 5



**5.c. Top 5 states with the highest average freight value.**

SELECT

c.customer\_state,

ROUND(AVG(oi.freight\_value),2) as avg\_freight\_val

FROM

target.order\_items as oi

INNER JOIN

target.orders as o

ON

oi.order\_id = o.order\_id

INNER JOIN

target.customers as c

ON

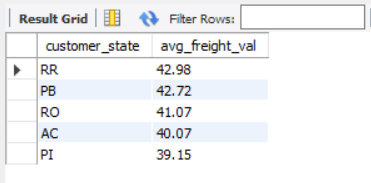
o.customer\_id = c.customer\_id

GROUP BY

c.customer\_state

ORDER BY avg\_freight\_val DESC

LIMIT 5



**5.d. Top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**

SELECT

c.customer\_state,

ROUND(SUM(ABS(DATEDIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp)))/COUNT(DISTINCT o.order\_id)) as avg\_del\_time,

ROUND(SUM(ABS(DATEDIFF(o.order\_estimated\_delivery\_date, o.order\_purchase\_timestamp)))/COUNT(DISTINCT o.order\_id)) as avg\_est\_del\_time

FROM

target.customers as c

INNER JOIN

target.orders as o

ON

c.customer\_id = o.customer\_id

WHERE

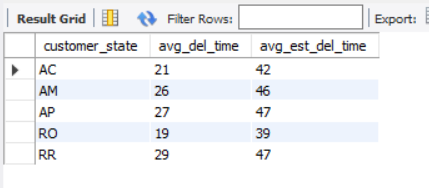
o.order\_status = 'delivered'

GROUP BY

c.customer\_state

ORDER BY (avg\_del\_time - avg\_est\_del\_time)

LIMIT 5



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

**6.a. Month on Month no. of orders placed using different payment types.**

SELECT

p.payment\_type,

MONTH(o.order\_purchase\_timestamp) as mth, YEAR(o.order\_purchase\_timestamp) as yr

,COUNT(DISTINCT o.order\_id) as order\_count

FROM

target.payments as p

INNER JOIN

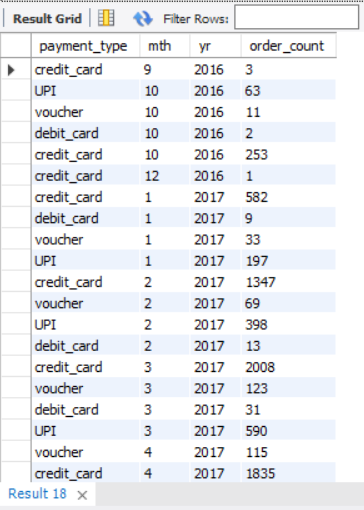
target.orders as o

ON

p.order\_id = o.order\_id

GROUP BY 1,2,3

ORDER BY 3,2



**6.b. No. of orders placed on the basis of the payment installments that have been paid.**

SELECT

payment\_installments as installments ,

COUNT(DISTINCT order\_id) as total\_orders

FROM

target.payments

WHERE

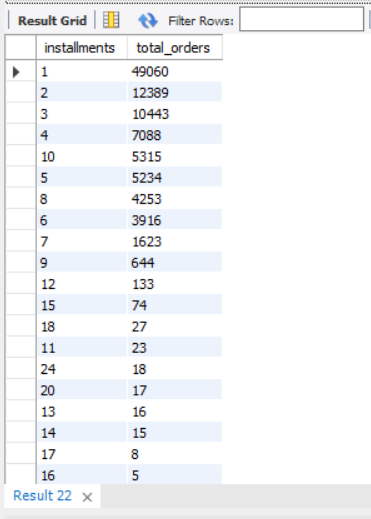
payment\_installments >= 1

GROUP BY

payment\_installments

ORDER BY

total\_orders DESC



**ADDITIONAL QUESTIONS**

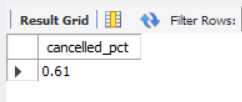
* **What percentage of orders were cancelled or unavailable**

SELECT

ROUND(100.0\*COUNT(DISTINCT CASE WHEN order\_status in ('cancelled','unavailable') THEN order\_id END)/COUNT(DISTINCT order\_id),2) AS cancelled\_pct

FROM

target.orders



* **Top 5 Customers**

SELECT

c.customer\_unique\_id,

COUNT(DISTINCT o.order\_id) as total\_orders

FROM

target.customers as c

INNER JOIN

target.orders as o

ON

c.customer\_id = o.customer\_id

GROUP BY

c.customer\_unique\_id

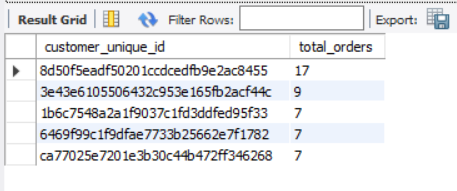
HAVING

COUNT(DISTINCT o.order\_id) > 1

ORDER BY

total\_orders DESC

LIMIT 5



* **Avg time for delivery**

SELECT

ROUND(SUM(TIMESTAMPDIFF(DAY, order\_purchase\_timestamp, order\_delivered\_customer\_date))/COUNT(order\_id),2) avg\_delivery

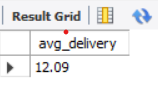
FROM

target.orders

WHERE

order\_status = 'delivered'

LIMIT 1

****