

TARGET SQL

Business Case

March 31st 2023

# 

# 

# Swapnil Sureshsing Rajput

Sept. 2022 TTS batch

# 

# TABLE OF CONTENTS

[Q. 1 Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset](#_dhtpx9b1jqxz)

[Q. 2 In-depth Exploration:](#_qz3hvabhdsjo)

[Q. 3 Evolution of E-commerce orders in the Brazil region:](#_qthw01u741hf)

[Q. 4 Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.](#_pjsu3t9vixkk)

[Q. 5 Analysis on sales, freight and delivery time](#_b8u917bvatmg)

[Q. 6 Payment type analysis:](#_vgkc1dy4dav1)

[Q. 7 Actionable Insights](#_35xl1xkpknr2)

[Q. 8 Recommendations](#_67oiaimlubxi)

# 

# 

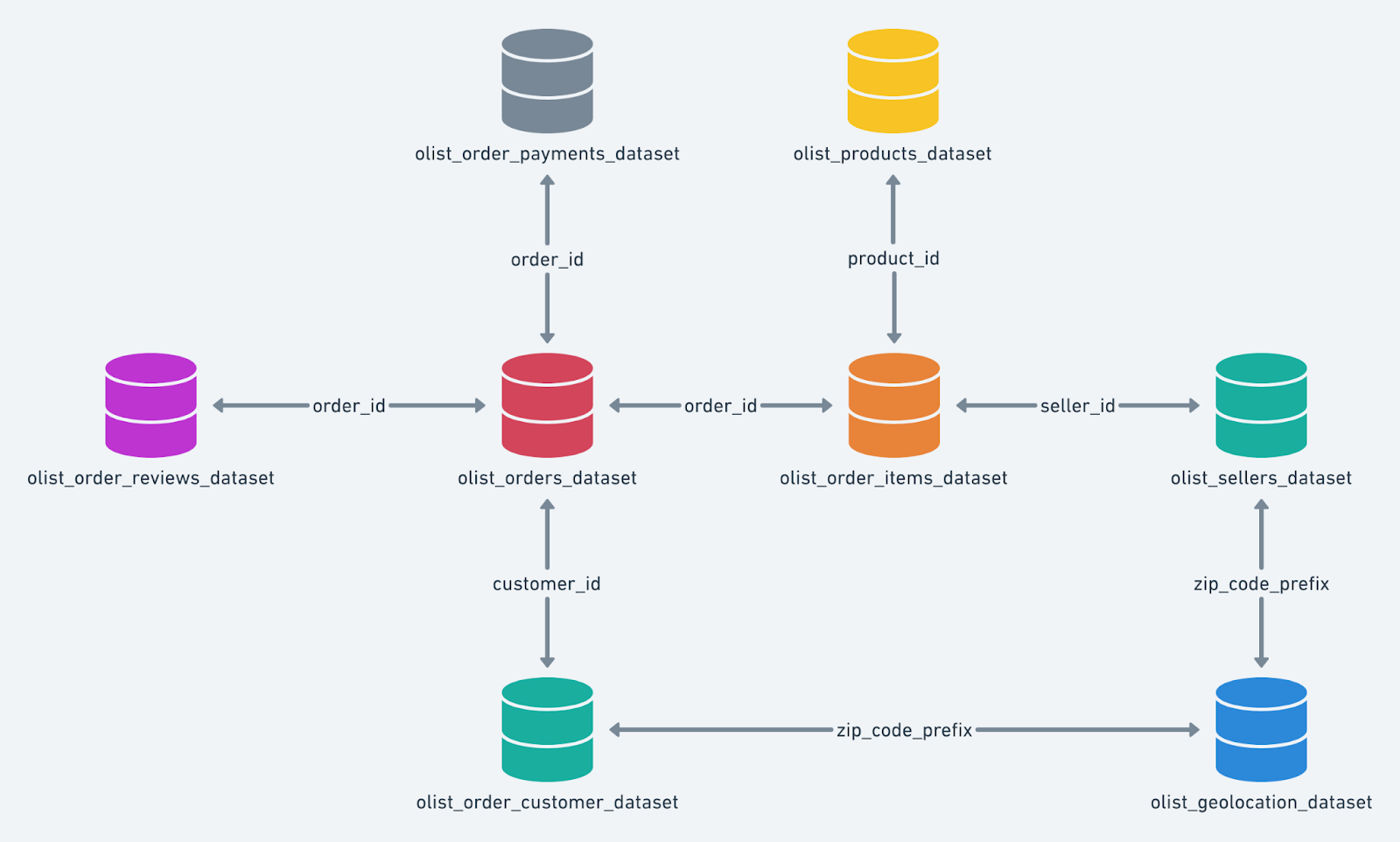
# Context

Target is one of the world’s most recognized brands and one of America’s leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allows viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

# 

# Schema



# 

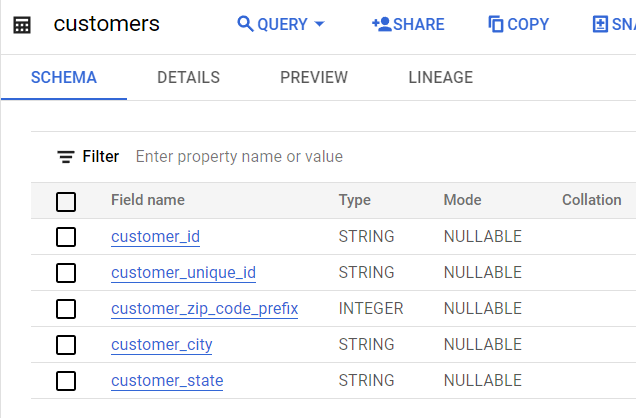
# Goal

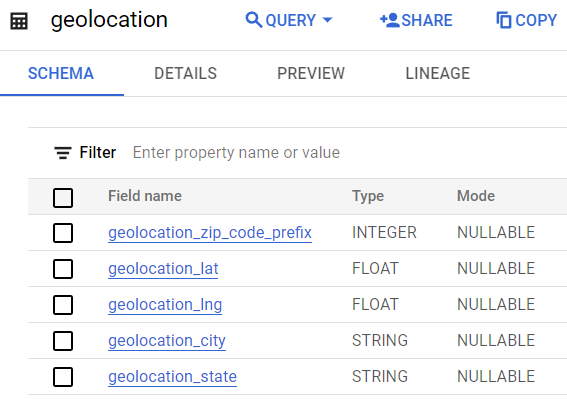
**To analyse the given dataset and provide insights and recommendations.**

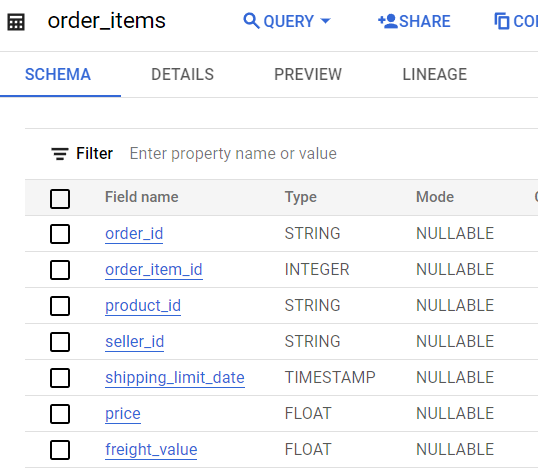
# 

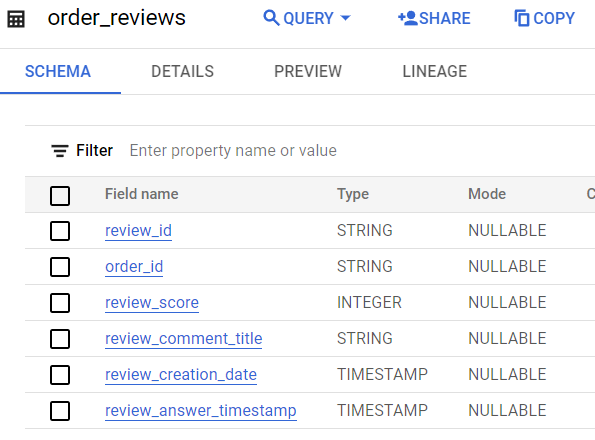
# Q. 1 Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

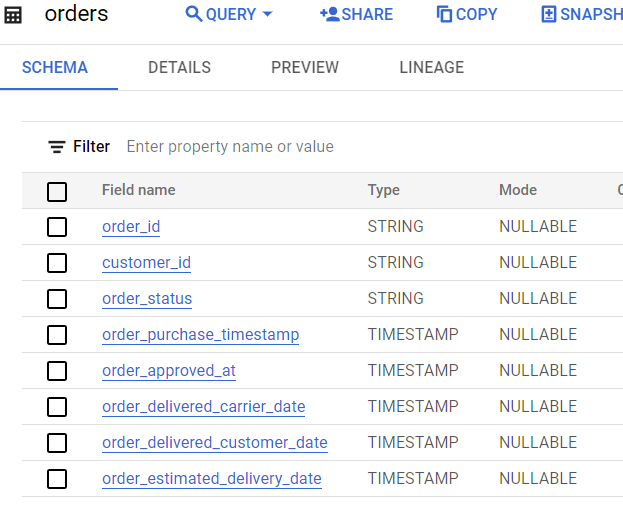
# Data type of columns in a table

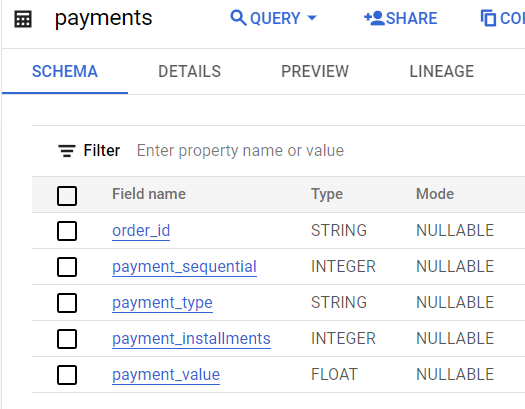


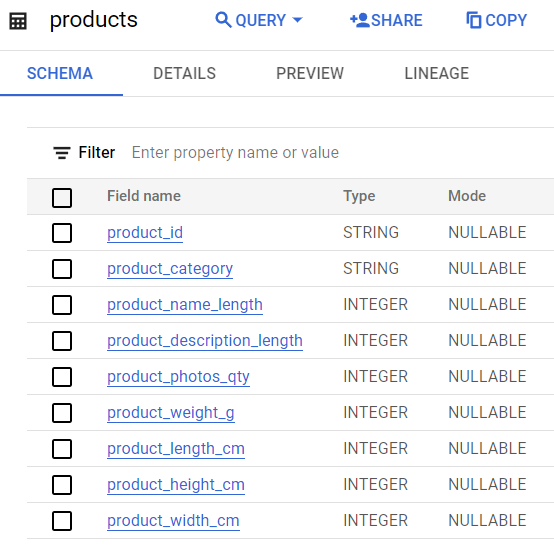


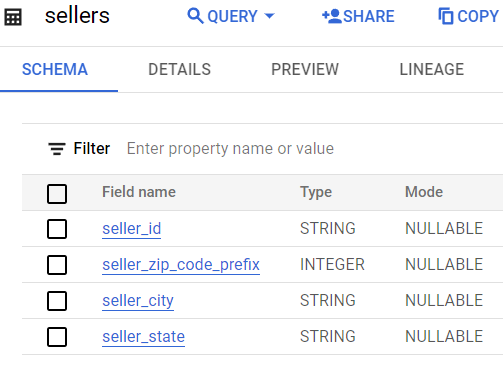












# Time period for which the data is given

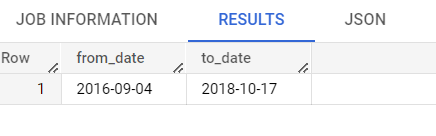
Data available for analysis is from Sept 2016 to Oct 2018.

SELECT

date(MIN(order\_purchase\_timestamp)) AS from\_date,

date(MAX(order\_purchase\_timestamp)) AS to\_date

FROM `scaler-target-sql-case.target\_sql.orders`;



**OR**

SELECT

DATE(MAX(order\_purchase\_timestamp)) AS date

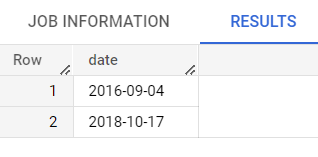
FROM `scaler-target-sql-case.target\_sql.orders`

UNION DISTINCT

SELECT

DATE(MIN(order\_purchase\_timestamp)) AS to\_date

FROM `scaler-target-sql-case.target\_sql.orders`;



# Cities and States of customers ordered during the given period

SELECT

DISTINCT INITCAP(customer\_city) AS CITY,

customer\_state AS States

FROM `scaler-target-sql-case.target\_sql.customers`;



# 

# Q. 2 In-depth Exploration:

# Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

There is a growing trend on e-commerce in Brazil year on year basis which is reflected through a number of parameters such as total number of unique orders, total sales and total order price as the years are passing by.

SELECT

EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS Year,

COUNT(DISTINCT(o.order\_id)) AS orders,

ROUND(SUM(p.payment\_value),2) AS Sales,

ROUND(SUM(oi.price),2) AS order\_price

FROM `scaler-target-sql-case.target\_sql.orders` AS o

LEFT JOIN `scaler-target-sql-case.target\_sql.payments` AS p

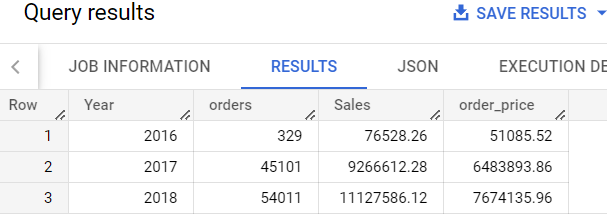
ON o.order\_id = p.order\_id

LEFT JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON oi.order\_id = p.order\_id

GROUP BY Year

ORDER BY Year;



**Monthly trend**

SELECT

EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS Year,

EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS Month,

COUNT(DISTINCT(o.order\_id)) AS orders,

ROUND(SUM(p.payment\_value),2) AS Sales,

ROUND(SUM(oi.price),2) AS order\_price

FROM `scaler-target-sql-case.target\_sql.orders` AS o

LEFT JOIN `scaler-target-sql-case.target\_sql.payments` AS p

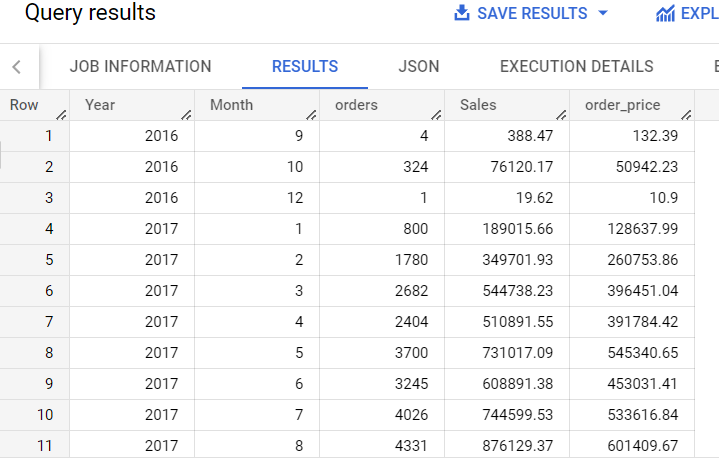
ON o.order\_id = p.order\_id

LEFT JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON oi.order\_id = p.order\_id

GROUP BY Year, Month

ORDER BY Year, Month;



**seasonality with peaks at specific months**

SELECT

EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS Month,

COUNT(DISTINCT(o.order\_id)) AS orders,

ROUND(SUM(p.payment\_value),2) AS Sales,

ROUND(SUM(oi.price),2) AS order\_price

FROM `scaler-target-sql-case.target\_sql.orders` AS o

LEFT JOIN `scaler-target-sql-case.target\_sql.payments` AS p

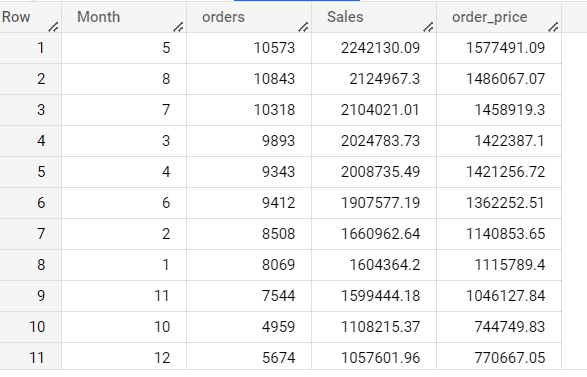
ON o.order\_id = p.order\_id

LEFT JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON oi.order\_id = p.order\_id

GROUP BY Month

ORDER BY Sales DESC, order\_price DESC;



# What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

SELECT

CASE

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 0 AND 7

THEN "Dawn"

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 7 AND 12

THEN "Morning"

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 12 AND 18

THEN "Afternoon"

WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 18 AND 24

THEN "Night"

END AS day\_time,

COUNT(DISTINCT(order\_id)) AS total\_orders

FROM `scaler-target-sql-case.target\_sql.orders` as o

GROUP BY day\_time

ORDER BY total\_orders DESC;



# 

# Q. 3 **Evolution of E-commerce orders in the Brazil region:**

# Get month on month orders by states

SELECT

c.customer\_state AS state,

EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS Year,

EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS Month,

COUNT(o.order\_id) AS total\_orders

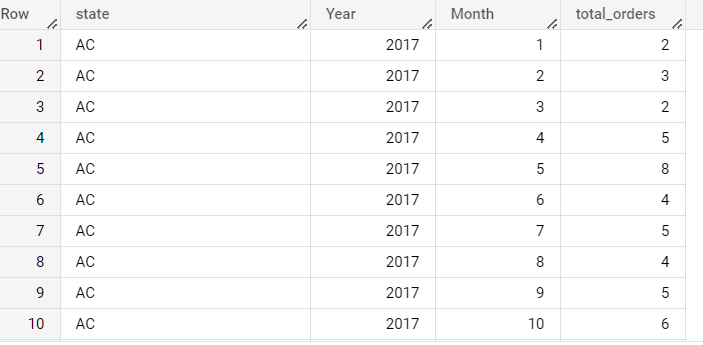
FROM `scaler-target-sql-case.target\_sql.orders` AS o

INNER JOIN `scaler-target-sql-case.target\_sql.customers` AS c

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

GROUP BY state, Year,Month

ORDER BY state,Year,Month ;



# Distribution of customers across the states in Brazil

SELECT

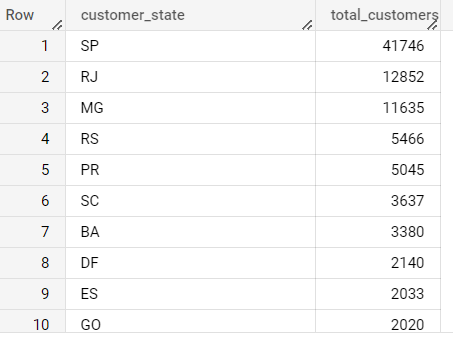
c.customer\_state,

COUNT(c.customer\_id) AS total\_customers

FROM `scaler-target-sql-case.target\_sql.customers` AS c

GROUP BY c.customer\_state

ORDER BY total\_customers DESC;



# 

# Q. 4 **Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.**

# Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment\_value” column in payments table

SELECT

ROUND((SUM(CASE WHEN DATE\_TRUNC(order\_purchase\_timestamp, MONTH) BETWEEN '2018-01-01' AND '2018-08-31' THEN payment\_value ELSE 0 END)

- SUM(CASE WHEN DATE\_TRUNC(order\_purchase\_timestamp, MONTH) BETWEEN '2017-01-01' AND '2017-08-31' THEN payment\_value ELSE 0 END))

/ SUM(CASE WHEN DATE\_TRUNC(order\_purchase\_timestamp, MONTH) BETWEEN '2017-01-01' AND '2017-08-31' THEN payment\_value ELSE 0 END) \* 100,2) AS percent\_increase

FROM `scaler-target-sql-case.target\_sql.orders` AS o

JOIN `scaler-target-sql-case.target\_sql.payments` AS p

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

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



# Mean & Sum of price and freight value by customer state

SELECT

c.customer\_state,

SUM(oi.price) AS total\_price,

ROUND(AVG(oi.price)) AS average\_price,

SUM(oi.freight\_value) AS total\_freight\_value,

ROUND(AVG(oi.freight\_value)) AS average\_freight\_value

FROM `scaler-target-sql-case.target\_sql.order\_items`AS oi

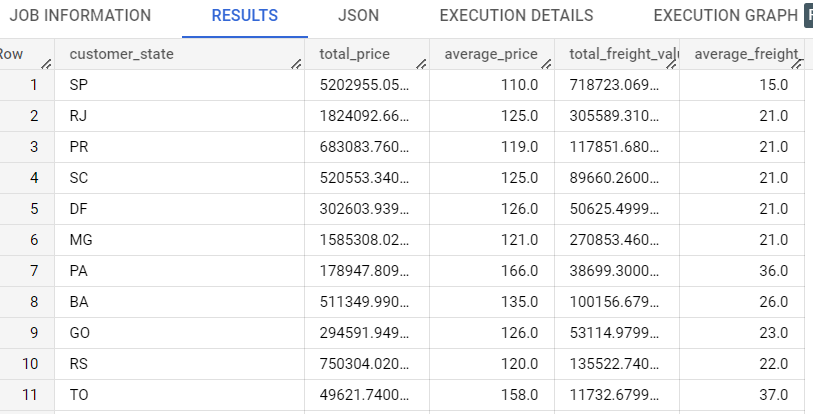
LEFT JOIN `scaler-target-sql-case.target\_sql.orders` AS o

ON oi.order\_id = o.order\_id

LEFT JOIN `scaler-target-sql-case.target\_sql.customers` as c

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

GROUP BY c.customer\_state;



# 

# 

# 

# 

# 

# Q. 5 **Analysis on sales, freight and delivery time**

# Calculate days between purchasing, delivering and estimated delivery

SELECT

order\_purchase\_timestamp,

order\_estimated\_delivery\_date,

order\_delivered\_customer\_date,

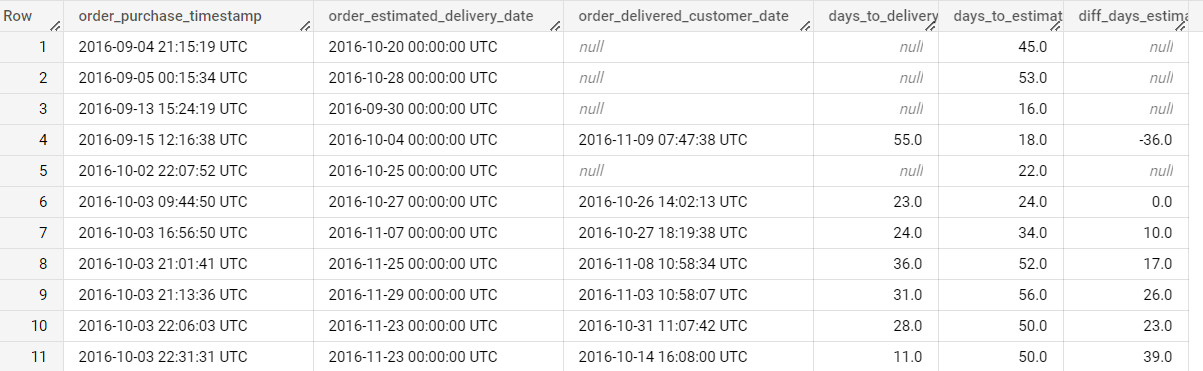
ROUND(EXTRACT(HOUR FROM (order\_delivered\_customer\_date - order\_purchase\_timestamp)) /24) AS days\_to\_delivery,-- difference in days between delivery date & order date. The longer the difference, it leaves customers in a dilemma and discourages future purchases.

ROUND(EXTRACT(HOUR FROM (order\_estimated\_delivery\_date - order\_purchase\_timestamp)) / 24) AS days\_to\_estimated\_delivery, -- difference in days between estimated delivery date & order date. The longer the difference, it discourages customers from purchasing..

ROUND(EXTRACT(HOUR FROM (order\_estimated\_delivery\_date - order\_delivered\_customer\_date)) / 24) AS diff\_days\_estimated\_delivery -- difference in days between delivery date & estimated delivery date. If the difference is negative, then the product was delivered later than the estimated date. If the difference is positive, the product was delivered earlier than the estimated date.

FROM `scaler-target-sql-case.target\_sql.orders`

ORDER BY order\_purchase\_timestamp;



# Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:

# time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date

# diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

SELECT

order\_purchase\_timestamp,

order\_delivered\_customer\_date,

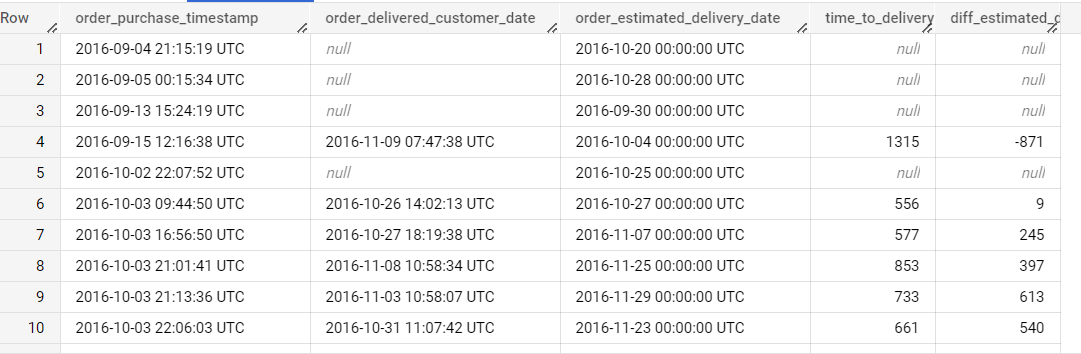
order\_estimated\_delivery\_date,

EXTRACT(HOUR FROM (order\_delivered\_customer\_date - order\_purchase\_timestamp)) AS time\_to\_delivery, -- difference in hours

EXTRACT(HOUR FROM (order\_estimated\_delivery\_date - order\_delivered\_customer\_date)) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.orders`

ORDER BY order\_purchase\_timestamp;



# Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

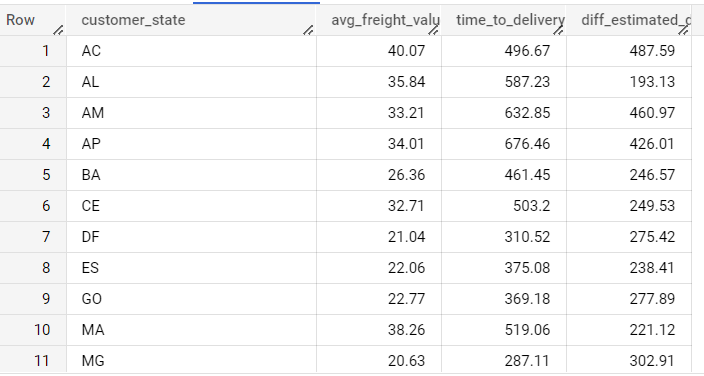
ON c.customer\_id = o.customer\_id

JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state;



# Sort the data to get the following:

# Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

* **Top 5 states with highest average freight value**

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

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

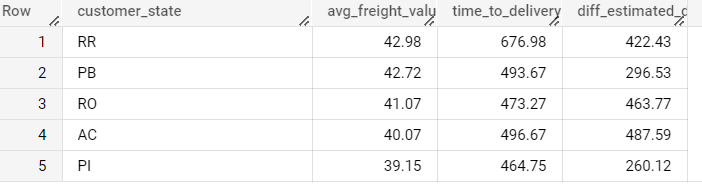
JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_freight\_value DESC

LIMIT 5;



* **Top 5 states with lowest average freight value**

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

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

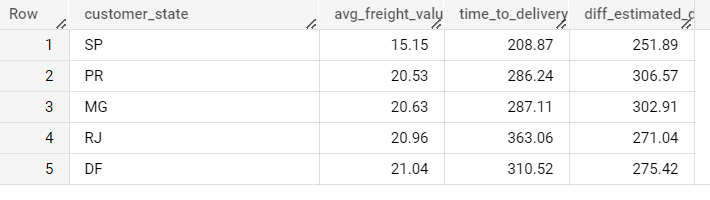
JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY avg\_freight\_value ASC

LIMIT 5;



# Top 5 states with highest/lowest average time to delivery

* **Top 5 states with highest time to delivery**

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

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

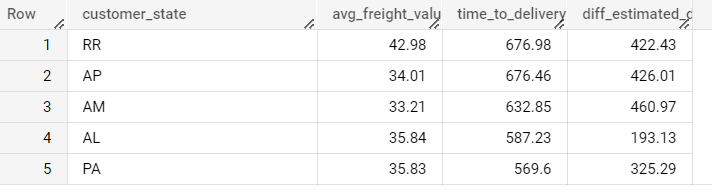
JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY time\_to\_delivery DESC

LIMIT 5;



* **Top 5 states with lowest time to delivery**

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

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

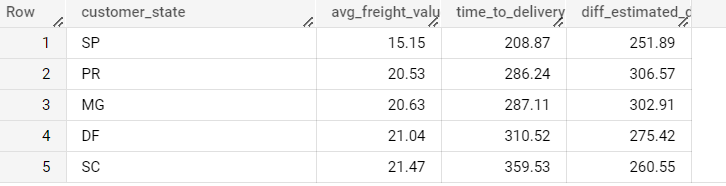
JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY time\_to\_delivery ASC

LIMIT 5;



# Top 5 states where delivery is really fast/ not so fast compared to estimated date

* **Top 5 states where delivery is really fast compared to estimated date**

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

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

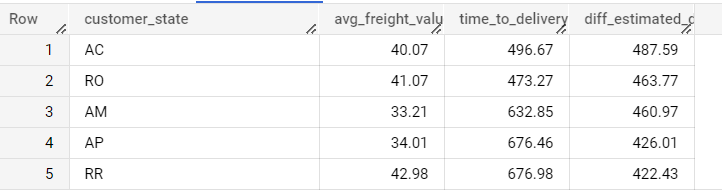
JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY diff\_estimated\_delivery DESC

LIMIT 5;



* **Top 5 states where delivery is not so fast compared to estimated date**

SELECT

c.customer\_state,

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

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_delivered\_customer\_date - o.order\_purchase\_timestamp))),2) AS time\_to\_delivery, -- difference in hours

ROUND(AVG(EXTRACT(HOUR FROM (o.order\_estimated\_delivery\_date - o.order\_delivered\_customer\_date))),2) AS diff\_estimated\_delivery -- difference in hours

FROM `scaler-target-sql-case.target\_sql.customers`AS c

JOIN `scaler-target-sql-case.target\_sql.orders` AS o

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

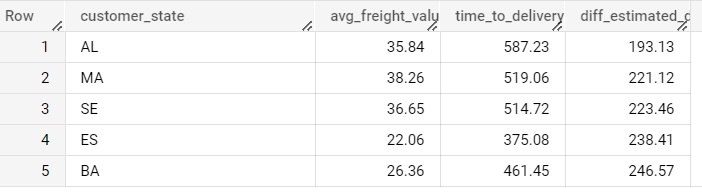
JOIN `scaler-target-sql-case.target\_sql.order\_items` AS oi

ON o.order\_id = oi.order\_id

GROUP BY c.customer\_state

ORDER BY diff\_estimated\_delivery ASC

LIMIT 5;



# 

# Q. 6 **Payment type analysis:**

# Month over Month count of orders for different payment types

It’s been observed that customer might have used multiple payment options for the same order, so same order is counted in the order count for different payment options

SELECT

EXTRACT(YEAR FROM o.order\_purchase\_timestamp) AS Year,

EXTRACT(MONTH FROM o.order\_purchase\_timestamp) AS Month,

p.payment\_type AS type,

COUNT(o.order\_id) AS total\_orders

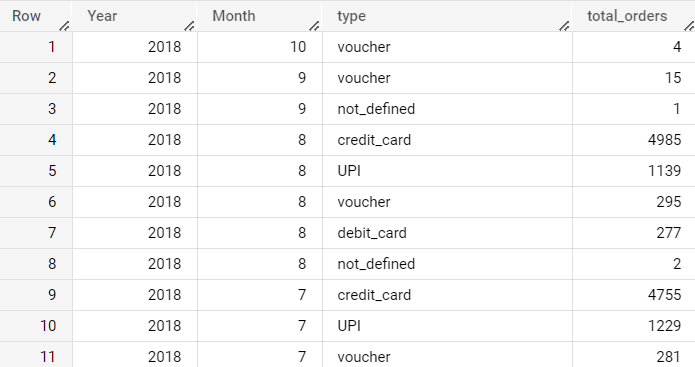
FROM `scaler-target-sql-case.target\_sql.orders` AS o

JOIN `scaler-target-sql-case.target\_sql.payments` AS p

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

GROUP BY Year, Month, type

ORDER BY Year DESC, Month DESC, total\_orders DESC;



# Count of orders based on the no. of payment instalments

SELECT

payment\_type,

COUNT(DISTINCT(order\_id)) AS total\_distinct\_orders

FROM `scaler-target-sql-case.target\_sql.payments`

WHERE payment\_installments > 1

GROUP BY payment\_type;



# 

# Q. 7 Actionable Insights

1. Trend for online purchases is increasing.
2. Need to improve time to delivery for top 5 states
3. Cost of order is increasing year on year basis. One of the reasons seems to be increasing Freight value.
4. Time to delivery is affecting customer experience as there is a lot of unpredictability on the delivery date. There is too much gap in estimated delivery and actual time to deliver.
5. Delay in time to delivery affects the future business and acts as a hindrance for future purchases.

# 

# Q. 8 Recommendations

There is a growing trend on e-commerce in Brazil year on year basis which is reflected through a number of parameters such as total number of unique orders, total sales and total order price as the years are passing by. By increasing the product offering, more customers can be attracted and overall business can be grown.

Time to delivery needs to be improved and estimated delivery needs to be calculated correctly to avoid unnecessary confusion for the customers. Delayed time to delivery increases dissatisfaction, hence it needs to be fixed.

Average Freight value per order is on a higher side, it certainly needs to be reevaluated in order to attract more customers.