**Target Case Study**

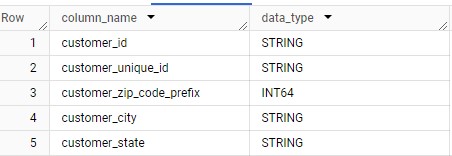
# Data type of all columns in the "customers" table.

# Assuming customers table already exists in the database

SELECT column\_name, data\_type

FROM target\_sales\_2016\_2018.INFORMATION\_SCHEMA.COLUMNS

WHERE table\_name = 'customers';



# Get the time range between which the orders were placed.

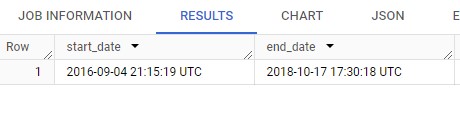
# Count the Cities & States of customers who ordered during the given period.

SELECT

  MIN(order\_purchase\_timestamp) AS start\_date,

  MAX(order\_purchase\_timestamp) AS end\_date

FROM target\_sales\_2016\_2018.orders;



SELECT

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

  COUNT(DISTINCT customer\_state) AS num\_states

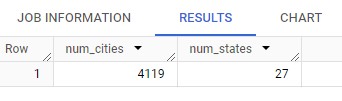
FROM target\_sales\_2016\_2018.customers

WHERE customer\_id IN

  (SELECT DISTINCT customer\_id

  FROM target\_sales\_2016\_2018.orders

  );



# In-depth Exploration:

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

SELECT

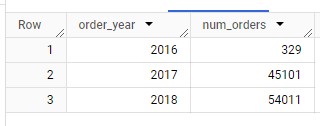
  EXTRACT(YEAR FROM order\_purchase\_timestamp) AS order\_year,

  COUNT(\*) AS num\_orders

FROM target\_sales\_2016\_2018.orders

GROUP BY order\_year

ORDER BY order\_year;



Inference: Yes, there has been major growth from 2016-17, then a small growth from 2017-18.

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

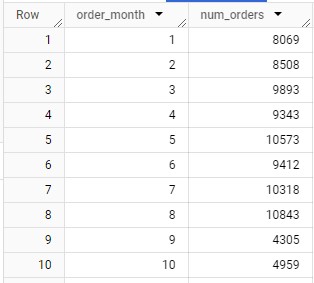
SELECT EXTRACT(MONTH FROM order\_purchase\_timestamp) AS order\_month,

       COUNT(\*) AS num\_orders

FROM target\_sales\_2016\_2018.orders

GROUP BY order\_month

ORDER BY order\_month;



Inference: From the results we can infer that peak seasons were mid-years May-Aug, Weak seasons were end of the year Sep-Dec.

# 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

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 'Morning'

        WHEN EXTRACT(HOUR FROM order\_purchase\_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'

        ELSE 'Night'

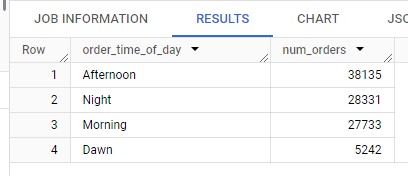
    END AS order\_time\_of\_day,

    COUNT(\*) AS num\_orders

FROM target\_sales\_2016\_2018.orders

GROUP BY order\_time\_of\_day

ORDER BY num\_orders DESC;



Inference: Most of the orders were placed during Afternoon time.

# Evolution of E-commerce orders in the Brazil region:

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

SELECT

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

  c.customer\_state,

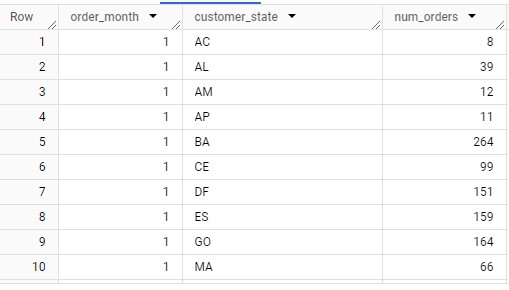
  COUNT(\*) AS num\_orders

FROM target\_sales\_2016\_2018.orders o

JOIN target\_sales\_2016\_2018.customers c ON o.customer\_id = c.customer\_id

GROUP BY order\_month, c.customer\_state

ORDER BY order\_month, c.customer\_state;



# How are the customers distributed across all the states?

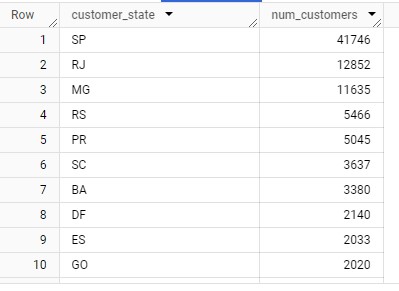
SELECT customer\_state,

       COUNT(DISTINCT customer\_id) AS num\_customers

FROM target\_sales\_2016\_2018.customers

GROUP BY customer\_state

ORDER BY num\_customers DESC;



Inference: State São Paulo (SP) has the most customers, while state Roraima (RR) has the least customer base.

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

# Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

WITH order\_costs\_2017 AS (

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

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

           SUM(p.payment\_value) AS total\_payment\_value

    FROM target\_sales\_2016\_2018.orders o

    JOIN target\_sales\_2016\_2018.payments p ON o.order\_id = p.order\_id

    WHERE EXTRACT(YEAR FROM o.order\_purchase\_timestamp)=2017

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

    GROUP BY order\_year, order\_month

), order\_costs\_2018 AS (

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

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

           SUM(p.payment\_value) AS total\_payment\_value

    FROM target\_sales\_2016\_2018.orders o

    JOIN target\_sales\_2016\_2018.payments p ON o.order\_id = p.order\_id

    WHERE EXTRACT(YEAR FROM o.order\_purchase\_timestamp)=2018

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

    GROUP BY order\_year, order\_month

)

SELECT

  order\_costs\_2017.order\_month as month,

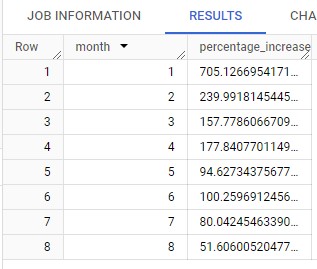
  (order\_costs\_2018.total\_payment\_value - order\_costs\_2017.total\_payment\_value) / order\_costs\_2017.total\_payment\_value \* 100 AS percentage\_increase

FROM order\_costs\_2017

JOIN order\_costs\_2018 ON

order\_costs\_2017.order\_month = order\_costs\_2018.order\_month

ORDER BY month;



Inference: Month of January had a huge boost of 700% compared to other months

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

SELECT c.customer\_state,

      ROUND(SUM(p.payment\_value),2) AS total\_order\_price,

      ROUND(AVG(p.payment\_value),2) AS avg\_order\_price

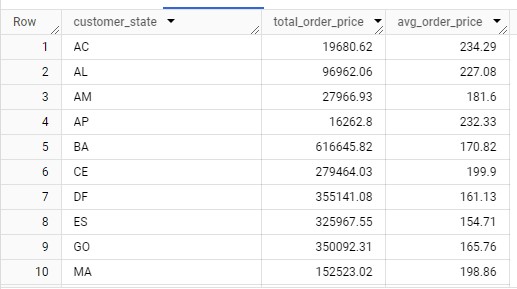
FROM target\_sales\_2016\_2018.orders o

JOIN target\_sales\_2016\_2018.customers c ON o.customer\_id = c.customer\_id

JOIN target\_sales\_2016\_2018.payments p ON o.order\_id = p.order\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state;



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

SELECT c.customer\_state,

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

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

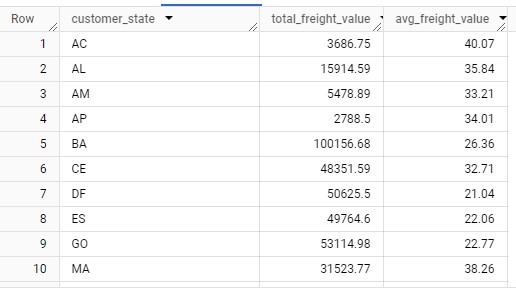
FROM target\_sales\_2016\_2018.orders o

JOIN target\_sales\_2016\_2018.customers c ON o.customer\_id = c.customer\_id

JOIN target\_sales\_2016\_2018.order\_items oi ON oi.order\_id = o.order\_id

GROUP BY c.customer\_state

ORDER BY c.customer\_state;



# Analysis based on sales, freight and delivery time.

# 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.

WITH delivery\_times AS (

    SELECT

        order\_id,

        DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY) AS delivery\_time,

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

    FROM

    target\_sales\_2016\_2018.orders

    WHERE

        order\_delivered\_customer\_date IS NOT NULL

        AND order\_estimated\_delivery\_date IS NOT NULL

)

SELECT

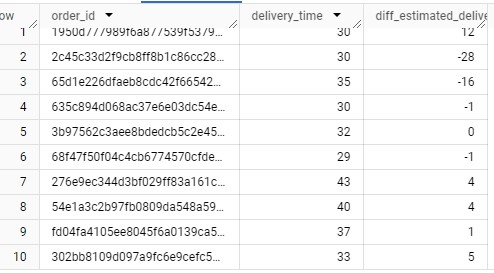
    order\_id,

    delivery\_time,

    diff\_estimated\_delivery

FROM

    delivery\_times;



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

WITH state\_avg\_freight AS (

    SELECT

      c.customer\_state,

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

    FROM target\_sales\_2016\_2018.orders o

    JOIN target\_sales\_2016\_2018.customers c ON o.customer\_id = c.customer\_id

    JOIN target\_sales\_2016\_2018.order\_items oi ON oi.order\_id = o.order\_id

    GROUP BY c.customer\_state

), state\_ranks as (

  SELECT

    customer\_state,

    avg\_freight,

    ROW\_NUMBER() OVER (ORDER BY avg\_freight DESC) AS rank\_high,

    ROW\_NUMBER() OVER (ORDER BY avg\_freight ASC) AS rank\_low

  FROM state\_avg\_freight

)

SELECT

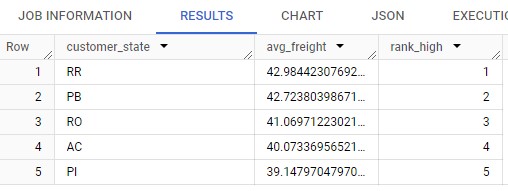
 customer\_state,

 avg\_freight,

 rank\_high

FROM state\_ranks

WHERE rank\_high<=5



SELECT

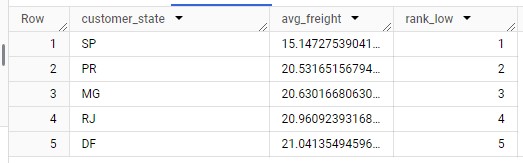
  customer\_state,

  avg\_freight,

  rank\_low

FROM state\_ranks

WHERE rank\_low<=5;



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

WITH state\_delivery\_time AS (

    SELECT

      c.customer\_state,

      AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) AS avg\_delivery\_time,

      ROW\_NUMBER() OVER (ORDER BY AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) DESC) AS rank\_high,

      ROW\_NUMBER() OVER (ORDER BY AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_purchase\_timestamp, DAY)) ASC) AS rank\_low

    FROM target\_sales\_2016\_2018.orders o

    JOIN target\_sales\_2016\_2018.customers c ON o.customer\_id = c.customer\_id

    WHERE order\_delivered\_customer\_date IS NOT NULL

    GROUP BY c.customer\_state

)

SELECT

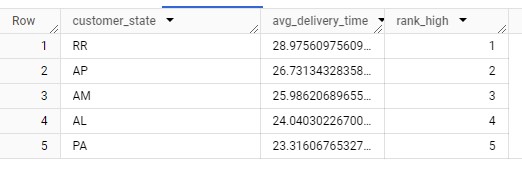
    customer\_state,

    avg\_delivery\_time,

    rank\_high

FROM state\_delivery\_time

WHERE rank\_high <= 5;



# SELECT

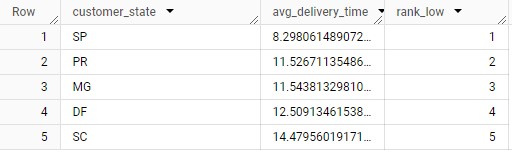
#    customer\_state,

#    avg\_delivery\_time,

#    rank\_low

# FROM state\_delivery\_time

# WHERE rank\_low <= 5



# 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.

WITH state\_delivery\_speed AS (

    SELECT

        c.customer\_state,

        AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date, DAY)) AS avg\_delivery\_speed,

        ROW\_NUMBER() OVER (ORDER BY AVG(DATE\_DIFF(order\_delivered\_customer\_date, order\_estimated\_delivery\_date, DAY)) ASC) AS rank\_fastest

    FROM target\_sales\_2016\_2018.orders o

    JOIN target\_sales\_2016\_2018.customers c ON o.customer\_id = c.customer\_id

    WHERE

        order\_delivered\_customer\_date IS NOT NULL

        AND order\_estimated\_delivery\_date IS NOT NULL

    GROUP BY c.customer\_state

)

SELECT

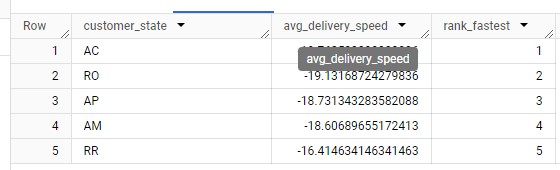
    customer\_state,

    avg\_delivery\_speed,

    rank\_fastest

FROM state\_delivery\_speed

WHERE rank\_fastest <= 5;



# Analysis based on the payments:

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

SELECT

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

    p.payment\_type,

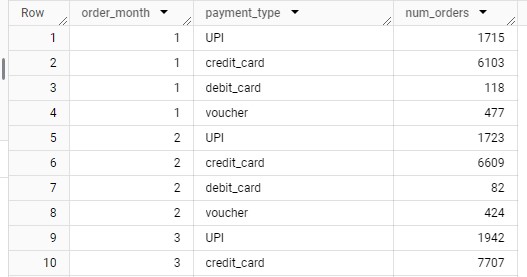
    COUNT(\*) AS num\_orders

FROM target\_sales\_2016\_2018.orders o

JOIN target\_sales\_2016\_2018.payments p ON o.order\_id = p.order\_id

GROUP BY order\_month, p.payment\_type

ORDER BY order\_month, p.payment\_type;



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

SELECT

    payment\_installments,

    COUNT(\*) AS num\_orders

FROM target\_sales\_2016\_2018.payments

GROUP BY payment\_installments

ORDER BY payment\_installments;



-- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Actionable Insights and recommendations:

1. Introducing special offers, discounts, events, etc., in the time of peak season May-Aug.
2. Taking special care and measures to maintain the order servers in Afternoon period.
3. Designating some states like SP as high value due to large customer base and creating business plans to improve sales push in these regions.
4. Upgrading transport and infrastructure to improve the Delivery days taken for states like RR, AP, who has smaller customer base.
5. Introducing offers catered to the credit card users due to its popularity as a preferred payment method.