**Business Case: Target SQL**

**Context:**

Target is a globally renowned brand and a prominent retailer in the United States. 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 particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

**Problem Statement:**

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

**What does 'good' look like?**

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.

select

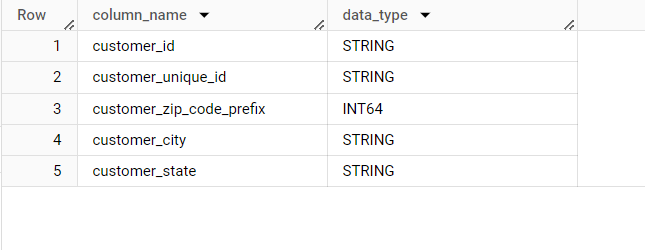
  column\_name, data\_type

from

  `sql-project-437307.Project\_DB.INFORMATION\_SCHEMA.COLUMNS`

where

  table\_name = 'customers';



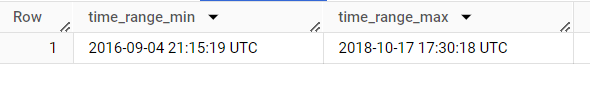
* 1. Get the time range between which the orders were placed.

select

min(order\_purchase\_timestamp) as time\_range\_min,

max(order\_purchase\_timestamp) as time\_range\_max

from `Project\_DB.orders`



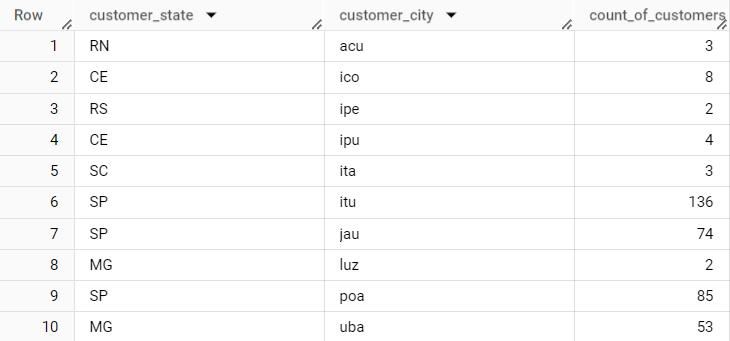
* 1. Count the Cities & States of customers who ordered during the given period.

select

customer\_state, customer\_city, count(\*) as count\_of\_customers

from `Project\_DB.customers`

group by customer\_state, customer\_city;



1. **In-depth Exploration:**  
   1. Is there a growing trend in the no. of orders placed over the past years?

select

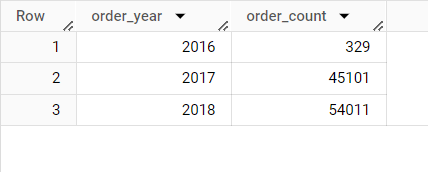
  EXTRACT(YEAR FROM order\_purchase\_timestamp) as order\_year,

  COUNT(\*) AS order\_count

from `Project\_DB.orders`

group by order\_year

order by order\_year



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

select

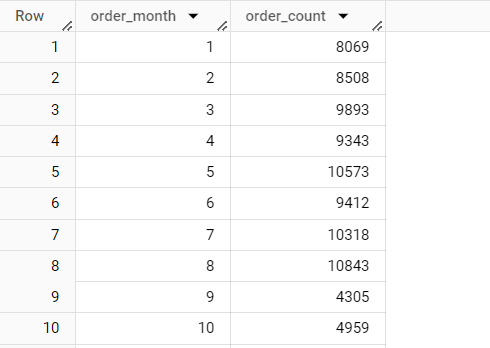
  EXTRACT(MONTH FROM order\_purchase\_timestamp) AS order\_month,

  COUNT(\*) AS order\_count

from `Project\_DB.orders`

group by order\_month

order by order\_month



* 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

with demo1 as

(with demo as

(select

  order\_id, extract(HOUR from order\_purchase\_timestamp) as order\_hour

from `Project\_DB.orders`)

select

  \*, case when order\_hour >=0 and order\_hour <=6 then 'Dawn (0-6hrs)'

  when order\_hour >=7 and order\_hour <=12 then 'Mornings (7-12hrs)'

  when order\_hour >=13 and order\_hour <=18 then 'Afternoon (13-18)'

  when order\_hour >=19 and order\_hour <=23 then 'Night (19-23)'

  end as order\_timings

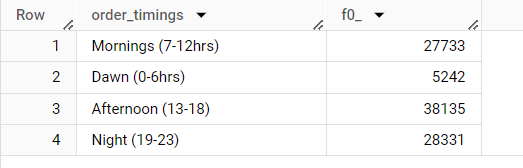
from demo)

select

  order\_timings, count (\*)

from demo1

group by order\_timings



1. **Evolution of E-commerce orders in the Brazil region:**
   1. Get the month on month no. of orders placed in each state.

select

  c.customer\_state,

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

  count(\*) as order\_count

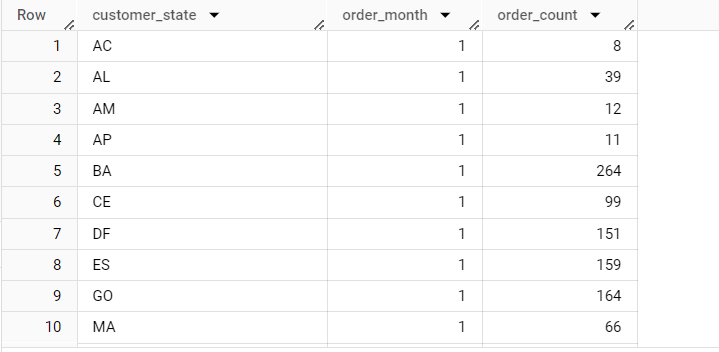
from `Project\_DB.orders` as o

join `Project\_DB.customers` as c

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

group by c.customer\_state,order\_month

order by order\_month, c.customer\_state



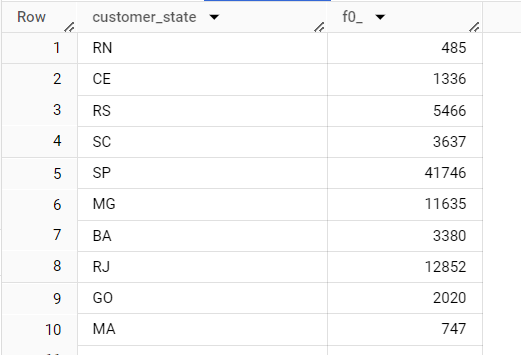
* 1. How are the customers distributed across all the states?

select

  customer\_state, count(\*)

from `Project\_DB.customers`

group by customer\_state



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.

with demo2 as

(with demo1 as

(with demo 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 cost\_of\_orders

from `Project\_DB.orders` as o

join `Project\_DB.payments` as p

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

group by order\_year,order\_month

having order\_month <= 8

order by order\_year,order\_month)

select

  order\_year, sum(cost\_of\_orders) as order\_cost

from demo

group by order\_year

order by order\_year)

select

  order\_year,order\_cost,

  lag(order\_cost) over(order by order\_year) as first\_,

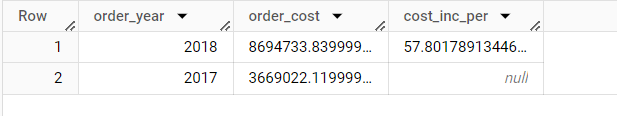
from demo1)

select

 order\_year, order\_cost,

 ((order\_cost - first\_) / order\_cost) \* 100 as cost\_inc\_per

from demo2



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

select

  c.customer\_state, sum(p.payment\_value) as total\_price,

  avg(p.payment\_value) as average\_price

from `Project\_DB.customers` as c

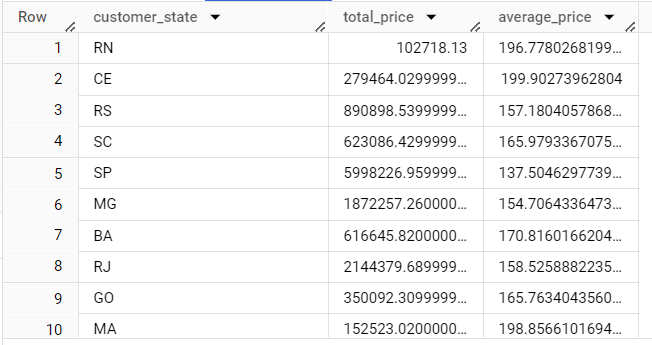
join `Project\_DB.orders` as o

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

join `Project\_DB.payments` as p

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

group by c.customer\_state



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

select

  c.customer\_state, sum(ot.freight\_value) as total\_order\_freight,

  avg(ot.freight\_value) as average\_order\_freight

from `Project\_DB.customers` as c

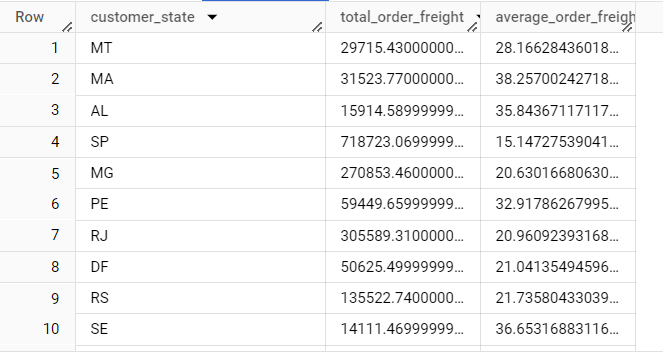
join `Project\_DB.orders` as o

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

join `Project\_DB.order\_items` as ot

on o.order\_id = ot.order\_id

group by c.customer\_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\_delivered\_customer\_date - order\_estimated\_delivery\_date

select

  order\_id,

  DATE(order\_purchase\_timestamp) as purchase\_date,

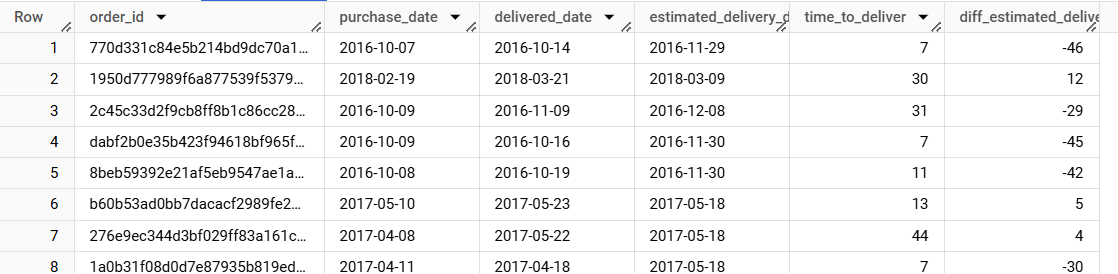
  DATE(order\_delivered\_customer\_date) as delivered\_date,

  DATE(order\_estimated\_delivery\_date) as estimated\_delivery\_date,

  DATE\_DIFF(DATE(order\_delivered\_customer\_date), DATE(order\_purchase\_timestamp), DAY) as time\_to\_deliver,

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

from `Project\_DB.Orders`;

****

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

select

  c.customer\_state, avg(oi.freight\_value) as avg\_freight\_value

from `Project\_DB.customers` as c

join `Project\_DB.orders` as o

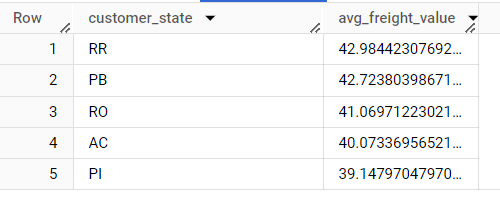
on c.customer\_id = o.customer\_id

join `Project\_DB.order\_items` as oi

on o.order\_id = oi.order\_id

group by customer\_state

order by avg\_freight\_value desc limit 5



select

  c.customer\_state, avg(oi.freight\_value) as avg\_freight\_value

from `Project\_DB.customers` as c

join `Project\_DB.orders` as o

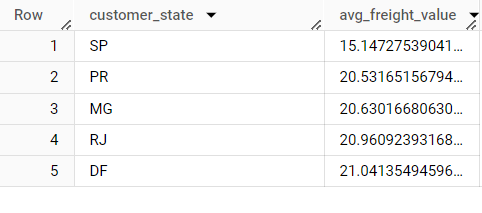
on c.customer\_id = o.customer\_id

join `Project\_DB.order\_items` as oi

on o.order\_id = oi.order\_id

group by customer\_state

order by avg\_freight\_value limit 5



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

with state\_delivery\_times as (

 select

    c.customer\_state,

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

  from `Project\_DB.customers` as c

  join `Project\_DB.Orders` as o

  on c.customer\_id = o.customer\_id

  group by

    customer\_state)

select

  customer\_state, avg\_delivery\_time

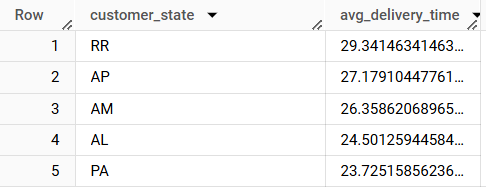
from

  state\_delivery\_times

order by

  avg\_delivery\_time desc

limit 5;



with state\_delivery\_times as (

 select

    c.customer\_state,

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

  from `Project\_DB.customers` as c

  join `Project\_DB.Orders` as o

  on c.customer\_id = o.customer\_id

  group by

    customer\_state

)

select

  customer\_state, avg\_delivery\_time

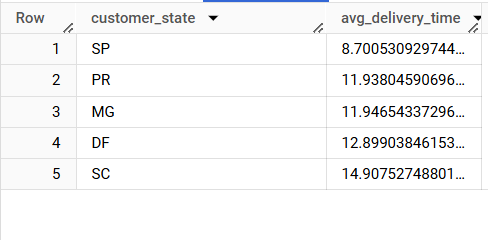
from

  state\_delivery\_times

order by

  avg\_delivery\_time asc

limit 5;



* 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.

with state\_delivery\_speed as (

  select

    c.customer\_state,

    AVG(DATE\_DIFF(DATE(o.order\_delivered\_customer\_date), DATE(o.order\_estimated\_delivery\_date), DAY)) as avg\_diff\_actual\_estimated

 from

    `Project\_DB.customers` as c

  join `Project\_DB.Orders` as o

  on c.customer\_id = o.customer\_id

group by

    customer\_state)

select

  customer\_state,

  avg\_diff\_actual\_estimated

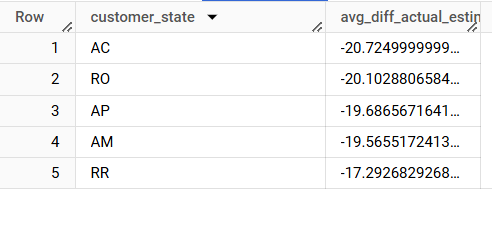
from

  state\_delivery\_speed

order by

  avg\_diff\_actual\_estimated asc

limit 5;



1. **Analysis based on the payments:**
   1. Find the month on month no. of orders placed using different payment types.

select

  p.payment\_type,

extract(MONTH from o.order\_purchase\_timestamp) as month, count(\*)

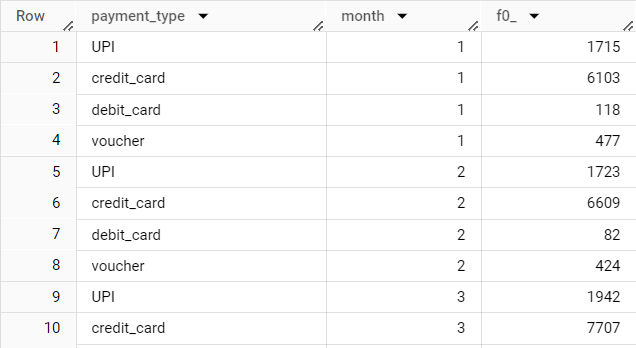
from `Project\_DB.orders` as o

join `Project\_DB.payments` as p

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

group by p.payment\_type, month

order by month, p.payment\_type



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

select

  p.payment\_installments, count(\*)

from `Project\_DB.orders` as o

join `Project\_DB.payments` as p

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

group by p.payment\_installments

order by p.payment\_installments

