***TARGET CASE STUDY - using big query setup***

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

***1.1 Data type of all columns in the "customers" table.***

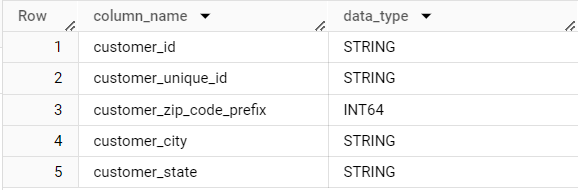
***ANSWER :***

SELECT column\_name, data\_type

FROM`scaler-dsml-sql-407614.TargetSQL.INFORMATION\_SCHEMA.COLUMNS`

WHERE table\_name= "customers"

**SNAP :**



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

**ANSWER :**

SELECT MIN(order\_purchase\_timestamp),

MAX(order\_purchase\_timestamp)

FROM`scaler-dsml-sql-407614.TargetSQL.orders`

**SNAP :**



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

**ANSWER :**

SELECT COUNT(c.customer\_city) AS count\_city,

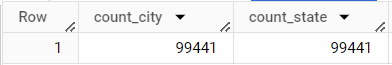
COUNT(c.customer\_state) AS count\_state

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

**SNAP :**

****

**2. In-depth Exploration:**

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

**ANSWER :**

SELECT

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

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

COUNT(DISTINCT o.order\_id) AS order\_count

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

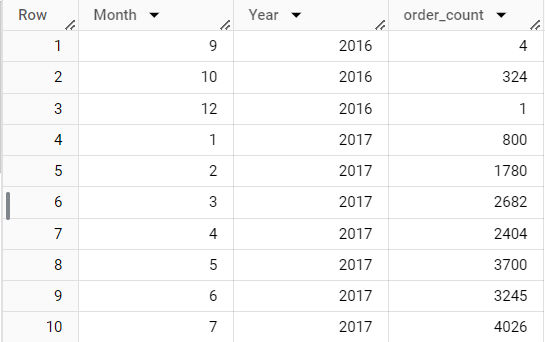
JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

GROUP BY Month, Year

ORDER BY Year, Month;

**SNAP :**



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

**ANSWER :**

SELECT

EXTRACT(Month FROM order\_purchase\_timestamp)AS Month,

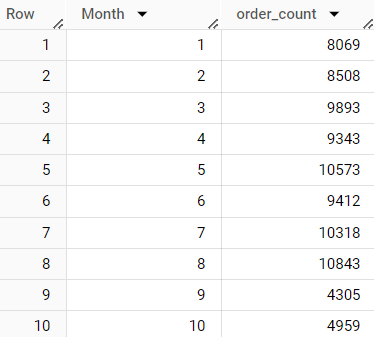
COUNT(DISTINCT order\_id) AS order\_count

FROM`scaler-dsml-sql-407614.TargetSQL.orders`

GROUP BY Month

ORDER BY Month;

**SNAP :**



**2.3 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**

**ANSWER :**

SELECT

CASE

WHEN EXTRACT(Hour FROM o.order\_purchase\_timestamp)BETWEEN 0 AND 5 THEN 'Dawn'

WHEN EXTRACT(Hour FROM o.order\_purchase\_timestamp)BETWEEN 6 AND 11 THEN 'Morning'

WHEN EXTRACT(Hour FROM o.order\_purchase\_timestamp)BETWEEN 12 AND 17 THEN 'Afternoon'

WHEN EXTRACT(Hour FROM o.order\_purchase\_timestamp)BETWEEN 18 AND 23 THEN 'Night'

END AS Hour,

COUNT(o.order\_id)AS order\_count

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

GROUP BY Hour

ORDER BY order\_count DESC;

**SNAP :**



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

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

**ANSWER :**

SELECT

c.customer\_state,

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

COUNT(o.order\_purchase\_timestamp)AS order\_count

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

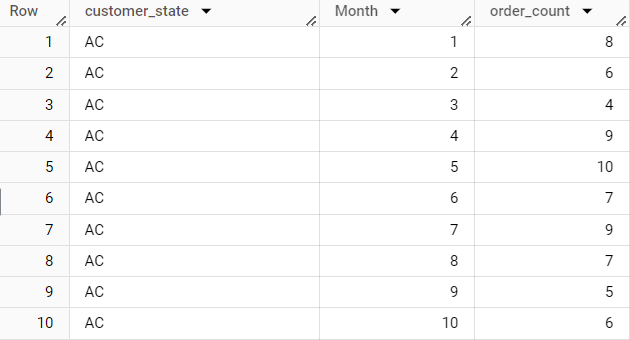
JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

GROUP BY c.customer\_state, Month

ORDER BY c.customer\_state, Month

**SNAP :**



**3.2 How are the customers distributed across all the states?**

**ANSWER :**

SELECT

customer\_state,

COUNT(customer\_id) AS no\_of\_customers

FROM`scaler-dsml-sql-407614.TargetSQL.customers`

GROUP BY customer\_state

ORDER BY customer\_state;

**SNAP :**



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

**4.1 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.**

**ANSWER :**

SELECT

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

((SUM(CASE

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

AND

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

THEN p.payment\_value END)

-

SUM(CASE

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

AND

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

THEN p.payment\_value END))/

SUM(CASE WHEN EXTRACT(YEAR FROM o.order\_purchase\_timestamp)= 2017 AND

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

THEN p.payment\_value END)) \* 100 AS percent\_increase

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.payments`AS p

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

WHERE

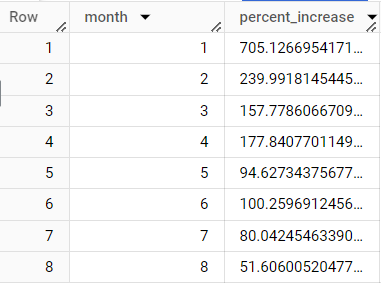
EXTRACT(YEAR FROM o.order\_purchase\_timestamp)IN(2017, 2018)AND

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

GROUP BY month

ORDER BY month;

**SNAP :**



**4.2Calculate the Total & Average value of order price for each state. and**

**4.3Calculate the Total & Average value of order freight for each state.**

**ANSWER :**

SELECT

c.customer\_state,

ROUND(AVG(i.price),2) AS mean\_price,

ROUND(SUM(i.price),2) AS total\_price,

ROUND(AVG(i.freight\_value),2) AS mean\_freight\_value,

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

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

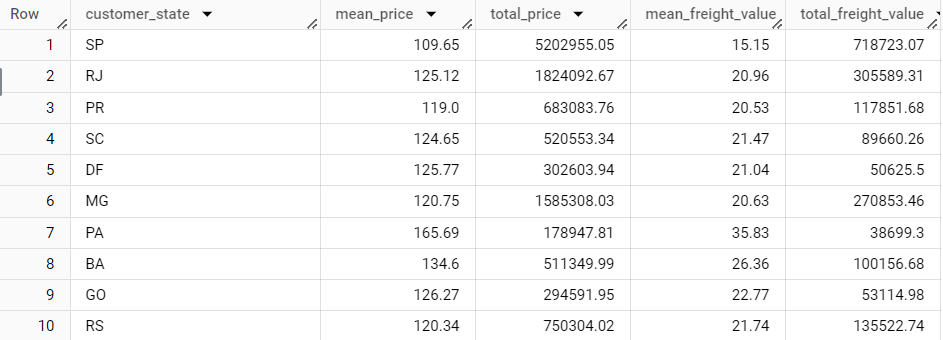
ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

GROUP BY c.customer\_state

**SNAP :**



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

***5.1 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***

**ANSWER :**

***SELECT***

***order\_id,***

***DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp,***

***DAY)***

***AS delivered\_in\_days,***

***DATE\_DIFF(order\_estimated\_delivery\_date,order\_purchase\_timestamp,***

***DAY)***

***AS estimated\_delivery\_in\_days,***

***DATE\_DIFF(order\_estimated\_delivery\_date,***

***order\_delivered\_customer\_date, DAY)***

***AS estimated\_minus\_actual\_delivery\_days***

***FROM`scaler-dsml-sql-407614.TargetSQL.orders`***

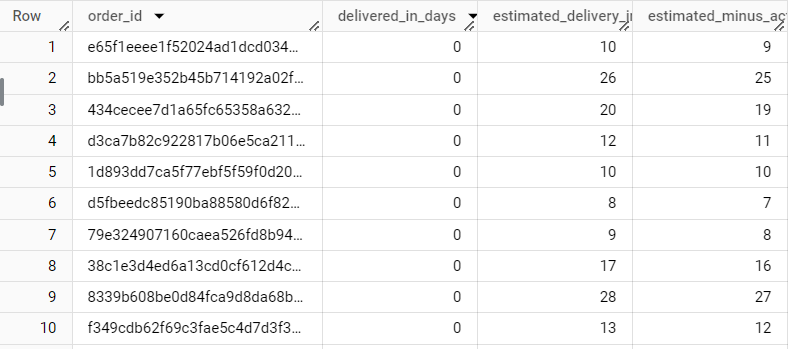
***WHERE***

***DATE\_DIFF(order\_delivered\_customer\_date,***

***order\_purchase\_timestamp, DAY)IS NOT NULL***

***ORDER BY delivered\_in\_days;***

**SNAP :**



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

***ANSWER :***

SELECT customer\_state,

MAX(CASE WHEN ranking = "TOP"

THEN avg\_freight END) AS top\_avg\_freight,

MAX(CASE WHEN ranking = "BOTTOM"

THEN avg\_freight END) AS bottom\_avg\_freight

FROM((SELECT c.customer\_state,

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

"TOP" AS ranking

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

GROUP BY c.customer\_state

ORDER BY avg\_freightDESC

LIMIT 5)

UNION DISTINCT

(SELECT c.customer\_state,

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

"BOTTOM" AS ranking

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

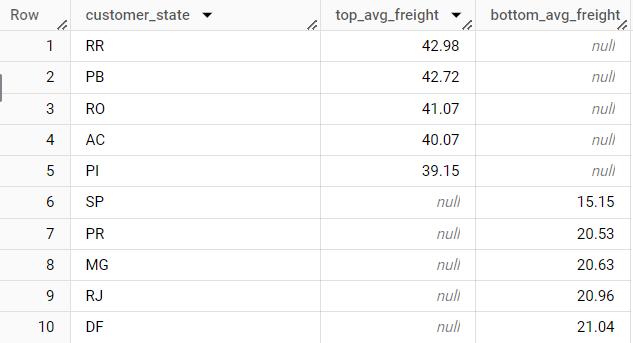
GROUP BY c.customer\_state

ORDER BY avg\_freight ASC

LIMIT5)) AS subquery

GROUP BY customer\_state;

**SNAP :**



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

***ANSWER :***

SELECT customer\_state,

MAX(CASE WHEN ranking = "TOP"

THEN avg\_delivery\_time END) AS top\_avg\_delivery\_time,

MAX(CASE WHEN ranking = "BOTTOM"

THEN avg\_delivery\_time END) AS bottom\_avg\_delivery\_time

FROM(

(SELECT c.customer\_state,

ROUND(AVG(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY)), 2)AS avg\_delivery\_time,

"TOP"AS ranking

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

GROUP BY c.customer\_state

ORDER BY avg\_delivery\_time DESC LIMIT 5)

UNION DISTINCT

(SELECT c.customer\_state,

ROUND(AVG(DATE\_DIFF(o.order\_delivered\_customer\_date, o.order\_purchase\_timestamp, DAY)), 2) AS avg\_delivery\_time,

"BOTTOM" AS ranking

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`ASc

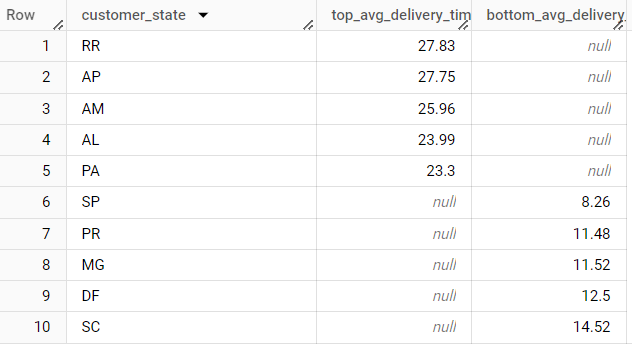
ON o.customer\_id= c.customer\_id

GROUP BY c.customer\_state

ORDERBY avg\_delivery\_time ASC LIMIT 5))AS subquery

GROUP BY customer\_state;

**SNAP :**

****

**5.4 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.**

***ANSWER :***

SELECT customer\_state,

MAX(CASE WHEN ranking = "TOP"

THEN diff\_estimated\_delivery END) AS top\_diff\_estimated\_delivery,

MAX(CASE WHEN ranking = "BOTTOM"

THEN diff\_estimated\_delivery END) AS bottom\_diff\_estimated\_delivery

FROM(

(SELECT

c.customer\_state,

ROUND(AVG(DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date, DAY)), 2)

AS diff\_estimated\_delivery,

"TOP" AS ranking

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

ONo.customer\_id = c.customer\_id

GROUP BY c.customer\_state

ORDER BY diff\_estimated\_delivery DESC

LIMIT 5)

UNION DISTINCT

(SELECT

c.customer\_state,

ROUND(AVG(DATE\_DIFF(o.order\_estimated\_delivery\_date, o.order\_delivered\_customer\_date, DAY)), 2)

As diff\_estimated\_delivery,

"BOTTOM" AS ranking

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.order\_items`AS i

ON o.order\_id = i.order\_id

JOIN`scaler-dsml-sql-407614.TargetSQL.customers`AS c

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

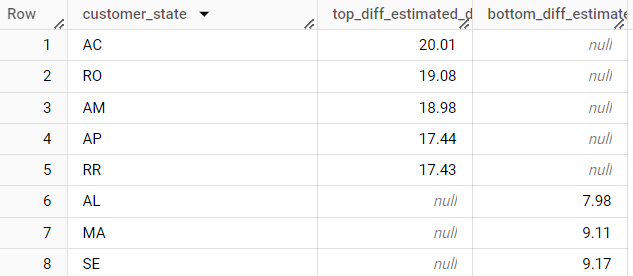
GROUP BY c.customer\_state

ORDER BY diff\_estimated\_delivery ASC

LIMIT 5)) AS subquery

GROUP BY customer\_state;

**SNAP :**

****

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

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

**ANSWER :**

SELECT

p.payment\_type,

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

COUNT(DISTINCT o.order\_id) AS order\_count

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

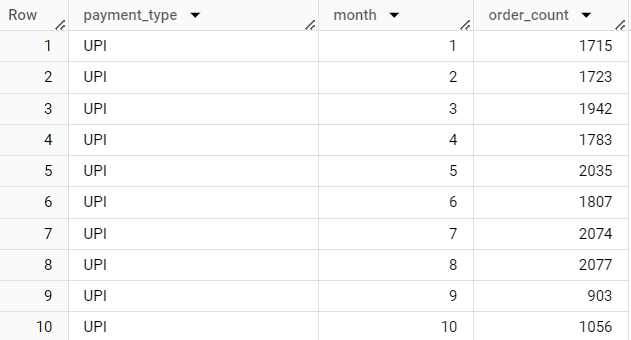
JOIN`scaler-dsml-sql-407614.TargetSQL.payments`AS p

ONo.order\_id= p.order\_id

GROUP BY p.payment\_type, month

ORDER BY p.payment\_type, month

**SNAP :**

****

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

**ANSWER :**

SELECT

p.payment\_installments,

COUNT(o.order\_id)AS order\_count

FROM`scaler-dsml-sql-407614.TargetSQL.orders`AS o

JOIN`scaler-dsml-sql-407614.TargetSQL.payments`AS p

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

GROUP BY p.payment\_installments

ORDER BY order\_count DESC

**SNAP :**

