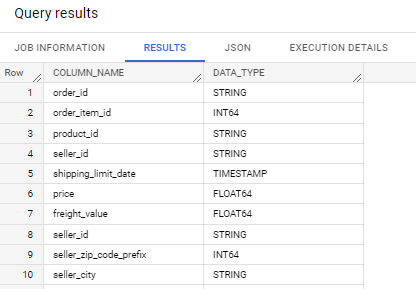
**TARGET CASE STUDY**

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

A. Data type of columns in a table

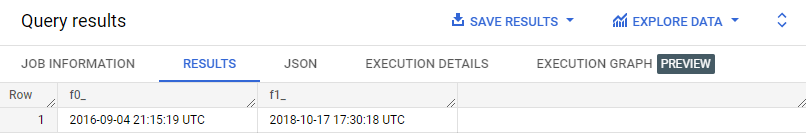
SELECT COLUMN\_NAME ,DATA\_TYPE

FROM `target.INFORMATION\_SCHEMA.COLUMNS`;



B. Time period for which the data is given

SELECT min(order\_purchase\_timestamp),max(order\_purchase\_timestamp) FROM `target.orders`;

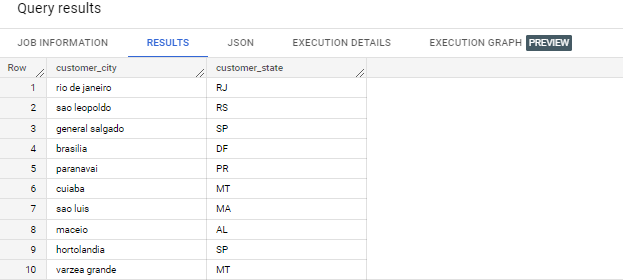


C. Cities and States of customers ordered during the given period

SELECT DISTINCT customer\_city,customer\_state from `target.customers` as c

JOIN `target.orders`as o

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



1. **In-depth Exploration:**

A. 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?

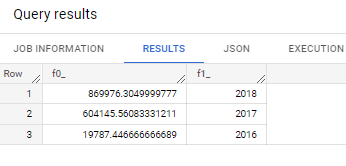
We have incomplete data for each year, for 2016 we have 3 month data and 2018 we have 10 month data so I calculated average sales per month for each year. And I found that there is an year on year increase in sales. So the trend is positive.

SELECT sum(p.payment\_value)/count(DISTINCT EXTRACT(MONTH FROM o.order\_purchase\_timestamp)), EXTRACT(YEAR FROM o.order\_purchase\_timestamp) from `target.orders` as o

JOIN `target.payments` as p

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

GROUP BY EXTRACT(YEAR FROM o.order\_purchase\_timestamp);



Aggregation by month to check the seasonality.

SELECT sales,month FROM

(SELECT sum(p.payment\_value) as sales, EXTRACT(MONTH FROM o.order\_purchase\_timestamp) as month from `target.orders` as o

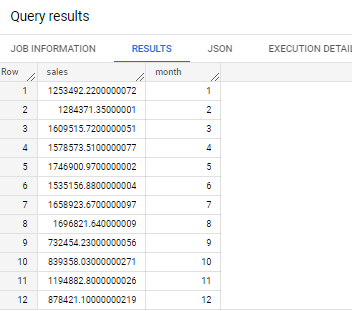
JOIN `target.payments` as p

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

GROUP BY EXTRACT(MONTH FROM o.order\_purchase\_timestamp)

)

ORDER BY month;



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

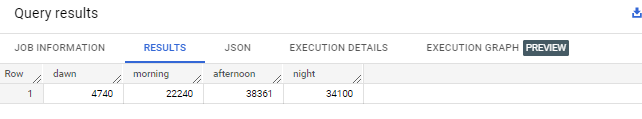
take into consideration the time zone and all four time zones

SELECT sum(if(time>=0 and time<=600, 1, 0)) as dawn,sum(if(time>=601 and time<=1200, 1, 0)) as morning,sum(if(time>=1201 and time<=1800, 1, 0)) as afternoon, sum(if(time>=1801 and time<=2359, 1, 0)) as night FROM

(

SELECT (EXTRACT(HOUR FROM order\_purchase\_timestamp)\*100+EXTRACT(HOUR FROM order\_purchase\_timestamp)) as time FROM `target.orders`

);



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

A. Get month-on-month orders by states

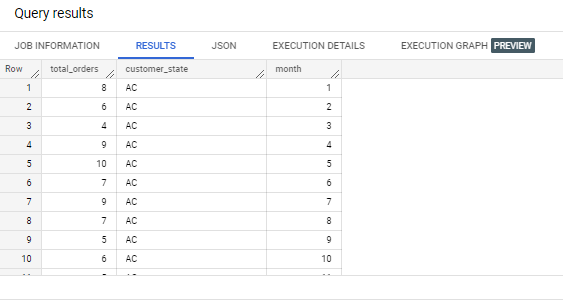
SELECT count(\*) as total\_orders,c.customer\_state,

EXTRACT(MONTH FROM order\_purchase\_timestamp) as month

from `target.customers`as c

JOIN `target.orders`as o

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



B. Distribution of customers across the states in Brazil

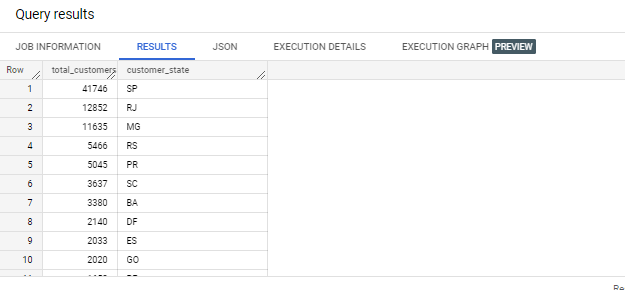
SELECT count(DISTINCT c.customer\_id) as total\_customers,c.customer\_state from `target.customers` as c

JOIN `target.orders`as o

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

GROUP BY c.customer\_state

ORDER BY count(DISTINCT c.customer\_id) desc;



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

A. 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(((sales2018-sales2017)/sales2017)\*100,2) as percentage\_ordervaluechange FROM

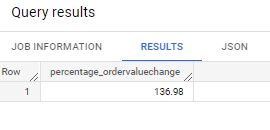
(SELECT sum(if(EXTRACT(YEAR FROM o.order\_purchase\_timestamp)=2017,p.payment\_value,0)) as sales2017,

sum(if(EXTRACT(YEAR FROM o.order\_purchase\_timestamp)=2018,p.payment\_value,0)) as sales2018 from `target.orders` o

JOIN `target.payments` p

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

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



B. Mean & Sum of price and freight value by customer state

SELECT c.customer\_state,round(sum(oi.price),2) as price\_sum,round(sum(oi.freight\_value),2) as freight\_sum, round(avg(oi.price),2) as price\_mean,

round(avg(oi.freight\_value),2) as freight\_mean

FROM `target.orders` o

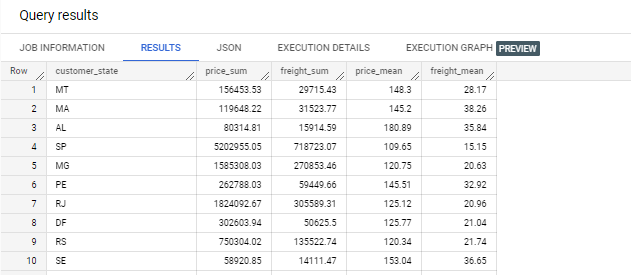
JOIN `target.order\_items` oi

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

JOIN `target.customers` c

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

GROUP BY c.customer\_state;



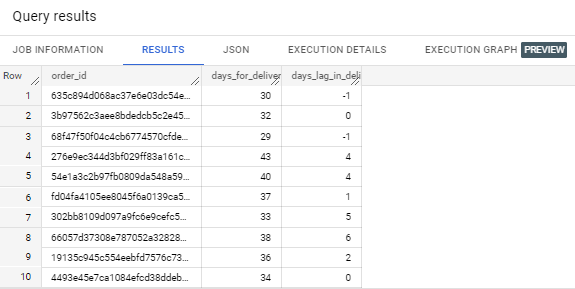
1. **Analysis on sales, freight and delivery time**

A. Calculate days between purchasing, delivering and estimated delivery

SELECT order\_id, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day) AS days\_for\_delivery,

DATE\_DIFF(order\_delivered\_customer\_date,order\_estimated\_delivery\_date, day) AS days\_lag\_in\_delivery FROM `target.orders`

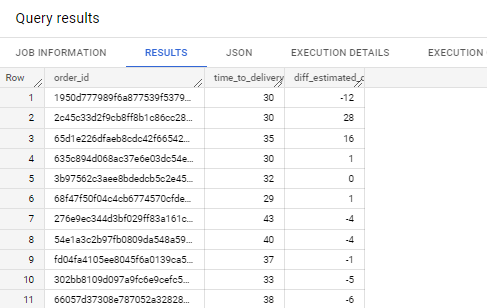
WHERE order\_status='delivered';



B. Find time\_to\_delivery & diff\_estimated\_delivery

SELECT order\_id, DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day) AS time\_to\_delivery,

DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, day) AS diff\_estimated\_delivery FROM `target.orders`



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

SELECT c.customer\_state,avg(oi.freight\_value) as avg\_freight\_value ,avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day)) AS avg\_time\_to\_delivery,

avg(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, day)) AS avg\_diff\_estimated\_delivery FROM `target.orders` o

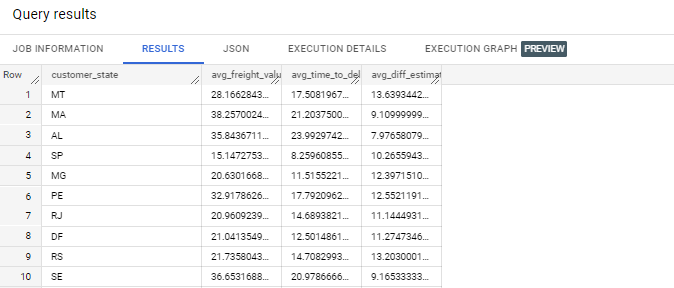
JOIN `target.customers` c

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

JOIN `target.order\_items` oi

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

GROUP BY c.customer\_state;



D.

Top 5 lowest

SELECT c.customer\_state,avg(oi.freight\_value) as avg\_freight\_value FROM `target.orders` o

JOIN `target.customers` c

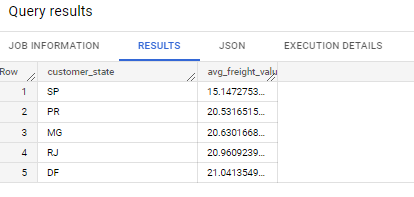
ON o.customer\_id=c.customer\_id

JOIN `target.order\_items` oi

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

GROUP BY c.customer\_state

ORDER BY avg(oi.freight\_value) LIMIT 5;



Top 5 highest

SELECT c.customer\_state,avg(oi.freight\_value) as avg\_freight\_value FROM `target.orders` o

JOIN `target.customers` c

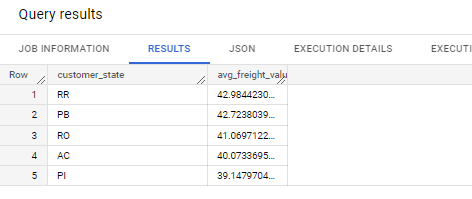
ON o.customer\_id=c.customer\_id

JOIN `target.order\_items` oi

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

GROUP BY c.customer\_state

ORDER BY avg(oi.freight\_value) DESC LIMIT 5;



E.

BOTTOM 5

SELECT c.customer\_state,avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day)) AS avg\_time\_to\_delivery

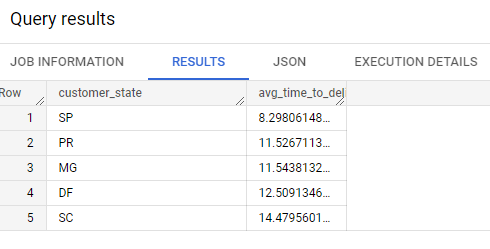
FROM `target.orders` o

JOIN `target.customers` c

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

GROUP BY c.customer\_state

ORDER BY avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day)) LIMIT 5;



TOP 5

SELECT c.customer\_state,avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day)) AS avg\_time\_to\_delivery

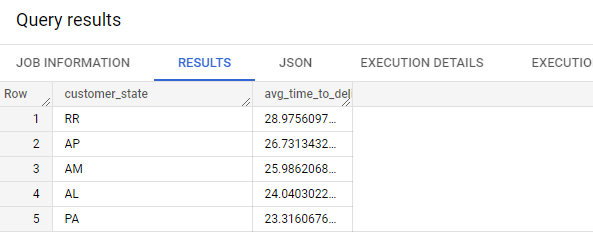
FROM `target.orders` o

JOIN `target.customers` c

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

GROUP BY c.customer\_state

ORDER BY avg(DATE\_DIFF(order\_delivered\_customer\_date,order\_purchase\_timestamp, day)) DESC LIMIT 5;



F.

TOP 5 fastest delivery states

SELECT c.customer\_state,avg(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, day))

AS avg\_diff\_estimated\_delivery

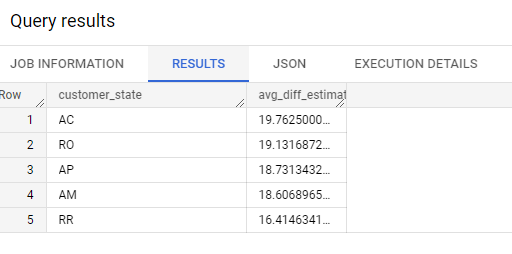
FROM `target.orders` o

JOIN `target.customers` c

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

GROUP BY c.customer\_state

ORDER BY avg(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, day)) DESC LIMIT 5;



TOP 5 Slowest delivery states

SELECT c.customer\_state,avg(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, day))

AS avg\_diff\_estimated\_delivery

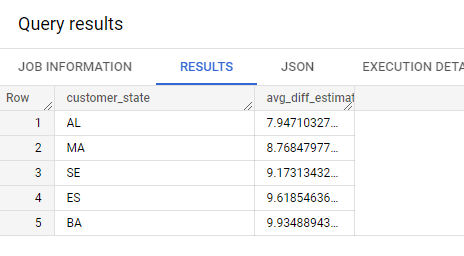
FROM `target.orders` o

JOIN `target.customers` c

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

GROUP BY c.customer\_state

ORDER BY avg(DATE\_DIFF(order\_estimated\_delivery\_date,order\_delivered\_customer\_date, day)) LIMIT 5;



1. **Payment type analysis:**

A. Month over Month count of orders for different payment types

SELECT orders,month,payment\_type FROM

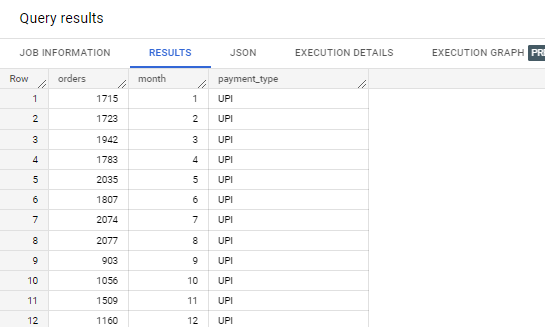
(SELECT count(\*) as orders,EXTRACT(MONTH FROM order\_purchase\_timestamp) as month,p.payment\_type FROM `target.orders`o

JOIN `target.payments` p

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

GROUP BY EXTRACT(MONTH FROM order\_purchase\_timestamp),p.payment\_type)

ORDER BY payment\_type,month;



B. Count of orders based on the no. of payment installments

SELECT payment\_installments, count(\*) as no\_of\_orders

FROM `target.payments` p JOIN `target.orders` o

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

GROUP BY payment\_installments

ORDER BY payment\_installments;

