**TARGET CASE STUDY - The Biggest Retailer**

-- the level of the table

select count(\*) as count\_of\_all from `scale-ds-ml.Target.order\_items`;

select count(\*) from

(select order\_id, order\_item\_id

from `scale-ds-ml.Target.order\_items`

group by 1,2);

--- Getting the time period for which the data is given

select \* from `scale-ds-ml.Target.orders`;

select

min(order\_purchase\_timestamp) as first\_order,

max(order\_purchase\_timestamp) as last\_order

from `scale-ds-ml.Target.orders`;

-- Number of cities and states in our dataset

select

count(distinct (geolocation\_city)) as city\_count,

count(distinct (geolocation\_state)) as state\_count

from `scale-ds-ml.Target.geolocation`;

-- Is there a growing trend in e-commerce in Brazil? How can we describe a complete scenario

select

extract(year from timestamp(order\_purchase\_timestamp)) as Year,

extract(month from timestamp(order\_purchase\_timestamp)) as Month,

count(1) as num\_orders

from `scale-ds-ml.Target.orders`

group by Year, Month

order by Year, Month;

-- can we see some seasonality with peaks at specific months

select

extract(month from timestamp(order\_purchase\_timestamp)) as Month,

count(1) as num\_orders

from `scale-ds-ml.Target.orders`

group by Month

order by Month;

-- What time do brazilian customers tends to buy (Dawn, Morning, Afternoon or Night)

select

case

when extract( hour from timestamp(order\_purchase\_timestamp)) between 0 and 6 then "dawn"

when extract(hour from timestamp(order\_purchase\_timestamp)) between 7 and 12 then "morning"

when extract(hour from timestamp(order\_purchase\_timestamp)) between 13 and 18 then "Afternoon"

when extract(hour from timestamp(order\_purchase\_timestamp)) between 19 and 23 then "Night"

end as time\_of\_day,

count(distinct order\_id) as counter

from `scale-ds-ml.Target.orders`

group by 1

order by 2 desc;

-- Get month on month orders by state/Region

select

extract(month from timestamp(order\_purchase\_timestamp)) as month,

g.geolocation\_state,

count(1) as number\_order

from `scale-ds-ml.Target.orders` o

join `scale-ds-ml.Target.Customers` c

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

join `scale-ds-ml.Target.geolocation` g

on g.geolocation\_zip\_code\_prefix = c.customer\_zip\_code\_prefix

group by g.geolocation\_state, month

order by g.geolocation\_state desc, month asc;

-- How are customers distributed in Brazil

select g.geolocation\_state, count(distinct (c.customer\_unique\_id)) as num\_customers

from `scale-ds-ml.Target.geolocation` g

join `scale-ds-ml.Target.Customers` c

on g.geolocation\_zip\_code\_prefix = c.customer\_zip\_code\_prefix

group by g.geolocation\_state

order by num\_customers desc;

-- Analyze the money movemnet by e-commerce by looking at order price, freight and others.

-- Create CTE table and new column

with base as

(

select

extract(month from timestamp(o.order\_purchase\_timestamp)) as month,

extract(year from timestamp(o.order\_purchase\_timestamp)) as year,

(sum(price)/ count(distinct o.order\_id)) as price\_per\_order,

(sum(freight\_value)/count(distinct o.order\_id)) as freight\_per\_order

from `scale-ds-ml.Target.orders` o

join `scale-ds-ml.Target.order\_items` i

on i.order\_id = o.order\_id

group by month, year

)

select price\_per\_order, freight\_per\_order

from base

order by year asc, month asc;

-- total amount sold in 2017 between jan and august (because data is available from 2017 01 to 2018 01)

-- compare YoY on monthly level

with base as

(

select

extract(month from timestamp(o.order\_purchase\_timestamp)) as month,

extract(year from timestamp(o.order\_purchase\_timestamp)) as year,

sum(price) as total\_price,

sum(freight\_value) as total\_freight

from `scale-ds-ml.Target.orders` o

join `scale-ds-ml.Target.order\_items` i

on i.order\_id = o.order\_id

group by month, year

order by year asc, month asc

)

select

month,

price\_2017, price\_2018, round((price\_2018-price\_2017)/price\_2017 \* 100,2) as year\_over\_year

from

(

select

month,

sum(case when year = 2017 then total\_price else 0 end) as price\_2017,

sum(case when year = 2018 then total\_price else 0 end) as price\_2018,

from base

where (year = 2017 or year = 2018) and month between 1 and 8

group by month

order by month

);

-- MoM increase in the year 2017

select

month, orders, lagger\_orders,

(orders - coalesce(lagger\_orders,0))/coalesce(lagger\_orders,1) \* 100 from

(

select \*,

lag(orders,1) over(order by month asc) as lagger\_orders from

(

select

extract(month from timestamp(o.order\_purchase\_timestamp)) as month,

count(distinct o.order\_id) as orders,

count(distinct c.customer\_unique\_id) as customers

from `scale-ds-ml.Target.orders` o

join `scale-ds-ml.Target.Customers` c

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

where extract(year from timestamp(o.order\_purchase\_timestamp)) = 2017

group by

1

)base\_1

) base2

--sum and mean price by customer state

--It is very intresting to see how states have a high total amount and a low price per order

with base as

(

select

c.customer\_state as state,

sum(price) as total\_price,

count(distinct(o.order\_id)) as num\_orders

from `scale-ds-ml.Target.orders` o

join `scale-ds-ml.Target.order\_items` i

on i.order\_id = o.order\_id

inner join `scale-ds-ml.Target.Customers` c

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

group by state

)

select

state, total\_price, num\_orders, (total\_price/num\_orders) as avg\_price

from base

order by total\_price desc;

-- Analysis on sales, freight nd delivery time

-- create new columns for time to delivery and difference in estimated vs actual\_delivery

select

order\_id,

date\_diff(

date(order\_estimated\_delivery\_date),

date(order\_purchase\_timestamp),

DAY

) as time\_to\_delivery

from `scale-ds-ml.Target.orders`

where order\_status = 'delivered'

-- Top 5 States with highest/lowest average time to delivery

select g.geolocation\_state as state,

sum(timestamp\_diff( timestamp(order\_estimated\_delivery\_date),timestamp(order\_purchase\_timestamp),DAY ))/count(order\_id) as avg\_time

from `scale-ds-ml.Target.orders` o

join `scale-ds-ml.Target.Customers` c

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

join `scale-ds-ml.Target.geolocation` g

on g.geolocation\_zip\_code\_prefix = c.customer\_zip\_code\_prefix

where order\_status = 'delivered'

group by state

order by avg\_time

limit 5;