TARGET CASE STUDY - The Biggest Retailer

```sql
-- 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
```

```sql
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
```

```sql
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
```

```sql
  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'
```

```sql
-- 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;
```