

Business Case: Analysis and fetching insights of data for E-commerce site using SQL

Time period for which the data is given

```
select min(order_purchase_timestamp) as min_date,
       max(order_purchase_timestamp) as max_date
from `target.orders`
```

Cities and States of customers ordered during the given period

```
select distinct(geolocation_city) as city,
       (geolocation_state) as state
from target.geolocation
```

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?

```
select extract(month from order_purchase_timestamp) as month,
       extract(year from order_purchase_timestamp) as year ,
       count(order_id) as no_of_orders
from `target.orders` group by year, month
order by no_of_orders , year ,month
```

As we can observe no of orders are increasing almost on monthly basis with some slight dip oin no of orders but overall it is increasing

As we cannot observe any outlier in any month, we fail to accept that there is any peak sale due to season.

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

```
select case when extract(hour from order_purchase_timestamp) between 4 and 5 then "dawn"
           when extract(hour from order_purchase_timestamp) between 5 and 12 then "morning"
           when extract(hour from order_purchase_timestamp) between 12 and 18 then "afternoon"
           else "night"
       end as time_of_day,
       count(order_id) as no_of_orders
from `target.orders`
group by time_of_day
```

Most brazillian do shopping in afternoon i.e from 12 pm to 6 pm so if we want to run some offer that will be perfect time and we can consider this time as feasible for reaching out to customers

Get month on month orders by states

```
select c.customer_state,
       extract(month from o.order_purchase_timestamp) as month,
       count(order_id) as no_of_orders
from `target.orders` o join `target.customer` c on o.customer_id = c.customer_id
group by c.customer_state, month
order by c.customer_state, month
```

Trying month name

```
select c.customer_state,
       format_date("%B", extract(date from o.order_purchase_timestamp)) as month,
       count(order_id) as no_of_order from `target.orders` o
join `target.customer` c on o.customer_id = c.customer_id
group by c.customer_state, month
order by c.customer_state, month
```

Distribution of customers across the states in Brazil

```
select customer_state,
       count(customer_unique_id) as no_of_customer
from `target.customer`
group by customer_state
order by no_of_customer desc
```

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

```
with base as(
  select extract(year from
    order_purchase_timestamp ) as year,
    extract(month from order_purchase_timestamp ) as month,
    p.payment_value total from `target.payments` p join `target.orders` o on p.o
    rder_id = o.order_id
  )
, base2 as(
  select base.year, sum(total) as total from base where (base.year between 2017
    and 2018) and (base.month between 1 and 8) group by base.year)
, base3 as(
  select base2.year as year, total, lead(base2.total) over (order by base2.year
    ) as next_year from base2)
select year, ( (next_year - base3.total ) /
total ) * 100 as percent_increase from base3
```

Mean & Sum of price and freight value by customer state

```
select customer_state,
       avg(price) as avg_price,
       avg(freight_value) as avg_freight_value,
       sum(price) as sum_price,
       avg(freight_value) as avg_frieght
from `target.orders` o join `target.order_items` as ot on o.order_id = ot.order_id
                        join target.customer c on o.customer_id = c.customer_id
group by customer_state
```

Calculate days between purchasing, delivering and estimated delivery

```
select order_id,
       date_diff(order_estimated_delivery_date, order_purchase_timestamp, day)
       as estimated_delivery_day,
       date_diff(order_estimated_delivery_date, order_delivered_customer_date, day)
       as delivered_day
from `target.orders`
order by delivered_day desc
```

Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- $\text{time_to_delivery} = \text{order_purchase_timestamp} - \text{order_delivered_customer_date}$
- $\text{diff_estimated_delivery} = \text{order_estimated_delivery_date} - \text{order_delivered_customer_date}$

```
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`
```

Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
select c.customer_state,
       avg(ot.freight_value) as freight_value,
       avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))
       as time_to_delivery ,
       avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date,
day))
       as diff_estimated_delivery
from `target.orders` o join `target.order_items` ot on o.order_id = ot.order_id
join `target.customer` c on o.customer_id = c.customer_id
group by c.customer_state
```

Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
select c.customer_state,
       avg(ot.freight_value) as freight_value,
       avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))
         as time_to_delivery ,
       avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date, day))
         as diff_estimated_delivery
from `target.orders` o join `target.order_items` ot on o.order_id = ot.order_id
join `target.customer` c on o.customer_id = c.customer_id
group by c.customer_state
order by freight_value desc
limit 5
```

Top 5 states with highest/lowest average time to delivery

```
select c.customer_state,
       avg(ot.freight_value) as freight_value,
       avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))
         as time_to_delivery ,
       avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date,
day))
         as diff_estimated_delivery
from `target.orders` o join `target.order_items` ot on o.order_id = ot.order_id
join `target.customer` c on o.customer_id = c.customer_id
group by c.customer_state
order by time_to_delivery desc
limit 5
```

--

Top 5 states where delivery is really fast/ not so fast compared to estimated date

```
select c.customer_state,
       avg(ot.freight_value) as freight_value,
       avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))
         as time_to_delivery ,
       avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date,
day))
         as diff_estimated_delivery
from `target.orders` o join `target.order_items` ot on o.order_id = ot.order_id
join `target.customer` c on o.customer_id = c.customer_id
group by c.customer_state
order by diff_estimated_delivery desc
limit 5
```

Month over Month count of orders for different payment types

```
select extract(year from o.order_purchase_timestamp) as year,
       format_date("%B",extract(date from o.order_purchase_timestamp)) as month_name,
       extract(month from o.order_purchase_timestamp) as month_number,
       p.payment_type,
       count(o.order_id) as no_of_transactions from `target.orders` o
```

```
join target.payments p on o.order_id = p.order_id
group by year, month_name, month_number, p.payment_type
order by year, month_number asc, p.payment_type
```

Count of orders based on the no. of payment installments

```
select payment_installments,
       count(order_id) as no_of_payments
from `target.payments`
group by payment_installments
```

Business Recommendation

Any product launch or any marketing should be done between 12 noon to 6 evening to gain maximum number of customers .

There should maximum staffing between 12 noon to 6 evening to provide best services to customers .

From 2017 -2018 there is increment of 136 % in cost of purchase .

The final delivery time is been done before the estimated delivery time . Hence, we could optimise the the estimated delivery time .

We could bring the more schemes and offer to provide our customers for the best services, also we could optimise the EMI option into 3 transactions