```
Select * from `Target.customers`
Select * from `Target.geolocation`
Select * from `Target.order_items`
Select * from `Target.order_reviews`
Select * from `Target.orders`
Select * from `Target.payments`
Select * from `Target.products`
Select * from `Target.sellers`
```

1. Import the dataset and do usual exploratory analysis steps like checking the structure

& characteristics of the dataset:

a. b. c. Data type of all columns in the "customers" table.

Get the time range between which the orders were placed.

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

```
Select count(distinct customer_city) as city_count,
count(distinct customer_state) as state_count
from `Target.orders` o join `Target.customers` c on o.customer_id = c.customer_id
where extract(year from order_purchase_timestamp) between 2016 and 2017
```

2. In-depth Exploration:

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

```
Select
extract(year from order_purchase_timestamp) as order_year,
count(order_id) as number_of_order
from `Target.orders`
group by order_year
order by order_year
```

b. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
Select
format_datetime('%Y-%m', order_purchase_timestamp) as Month_of_order,
count(order_id) as Num_of_order
from `Target.orders`
group by Month_of_order
order by Month_of_order
```

c. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

i. 0-6 hrs : Dawn

ii. 7-12 hrs : Mornings

iii. 13-18 hrs : Afternoon

iv. 19-23 hrs : Night

```
Select
case
when extract(hour from order_purchase_timestamp) between 0 and 6 then 'Dawn'
when extract(hour from order_purchase_timestamp) between 7 and 12 then 'Mornings'
when extract(hour from order_purchase_timestamp) between 13 and 18 then 'Afternoon'
```

```
when extract(hour from order_purchase_timestamp) between 19 and 23 then 'Night'
end order_time_range,
count (*) as number_of_orders
from `Target.orders`
group by order_time_range
order by number_of_orders
```

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

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

```
Select customer_state,
format_date('%Y-%m', order_purchase_timestamp) as month_wise,
count (*) as number_of_order
from `Target.orders` o join `Target.customers` c on o.customer_id = c.customer_id
group by customer_state,month_wise
order by month_wise,customer_state
```

b. How are the customers distributed across all the states?

```
Select customer_state,
count(customer_unique_id) as customer_count
from `Target.customers`
group by customer_state
order by customer_count desc
```

4. 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.

```
with year_info as (Select payment_value,
```

```
extract(year from order_purchase_timestamp) as order_Year
from `Target.payments` p join `Target.orders` o on p.order_id =o.order_id
where extract(YEAR FROM order_purchase_timestamp) in (2017, 2018)
and extract(MONTH FROM order_purchase_timestamp) between 1 and 8),
value_info as (
select order_Year,
sum(payment_value) as overall_value
from year_info
group by order_Year
)

select
max(case when order_Year =2017 then overall_value end) as total_2017,
max(case when order_Year =2018 then overall_value end) as total_2018,
round(((max(case when order_Year =2018 then overall_value end) - max(case when order_Year
=2017 then overall_value end)) / max(case when order_Year =2017 then overall_value end))*100,2)
as Difference
from value_info
```

b. Calculate the Total & Average value of order price for each state.

```
Select distinct customer_state,
sum(price) over (partition by customer_state) as State_total,
round(avg(price) over (partition by customer_state),2) as State_avg
from `Target.order_items` oi join `Target.orders` o on oi.order_id = o.order_id join
`Target.customers` c on o.customer_id =c.customer_id
order by State_avg
```

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

```
Select
```

```
distinct customer_state,
sum(freight_value) over (partition by customer_state) as State_total,
round(avg(freight_value) over (partition by customer_state),2) as State_avg
from
`Target.order_items` oi
join
`Target.orders` o on oi.order_id = o.order_id
join
`Target.customers` c on o.customer_id =c.customer_id
order by
State_avg
```

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

a. 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:

i. time_to_deliver = order_delivered_customer_date - order_purchase_timestamp

ii. diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date

```
select
order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_deliver,
```

```
date_diff(order_estimated_delivery_date,order_purchase_timestamp, day) as
diff_estimated_delivery,
date_diff (order_estimated_delivery_date,order_delivered_customer_date, day) as day_diff
from `Target.orders`
where
order_delivered_customer_date is not null
```

b. c. d. Find out the top 5 states with the highest & lowest average freight value.

```
select
customer_state,
avg(freight_value) as sate_avg
from `Target.order_items` oi
join `Target.orders` o on oi.order_id = o.order_id
join `Target.customers` c on o.customer_id = c.customer_id
group by customer_state
order by sate_avg desc
limit 5
```

```
select
customer_state,
avg(freight_value) as sate_avg
from `Target.order_items` oi
join `Target.orders` o on oi.order_id = o.order_id
join `Target.customers` c on o.customer_id = c.customer_id
group by customer_state
order by sate_avg
limit 5
```

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

Fastest

```
select customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as del_days
from `Target.order_items` oi
join `Target.orders` o on oi.order_id = o.order_id
join `Target.customers` c on o.customer_id = c.customer_id
group by customer_state
order by del_days
limit 5
```

Slowest

```
select customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as del_days
from `Target.order_items` oi
join `Target.orders` o on oi.order_id = o.order_id
join `Target.customers` c on o.customer_id = c.customer_id
group by customer_state
order by del_days desc
limit 5
```

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.

```
Select customer_state,
avg (date_diff(order_estimated_delivery_date, order_delivered_customer_date, day)) as delivered_in
from `Target.orders` o join `Target.customers` c on o.customer_id = c.customer_id
group by customer_state
order by delivered_in
limit 5
```

6.Analysis based on the payments:

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

```
Select payment_type,
format_timestamp('%Y-%m', o.order_purchase_timestamp) as order_Month,
count(*) as number_of_payment
from `Target.payments` p join `Target.orders` o on p.order_id = o.order_id
group by payment_type,
format_timestamp('%Y-%m', o.order_purchase_timestamp)
order by order_Month,payment_type
```

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

```
Select payment_installments,
count(distinct p.order_id) as order_count
from `Target.payments` p join `Target.orders` o on p.order_id = o.order_id
group by payment_installments
order by payment_installments
```