1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:
    1. Data type of all columns in the "customers" table.
    2. Get the time range between which the orders were placed.
    3. Count the Cities & States of customers who ordered during the given period.

```
select COUNT(citycount),sum(statecount)
from(
select c.customer_id,
count(c.customer_city)as citycount,
count(c.customer_state) as statecount,
max(order_purchase_timestamp) as last_,
min(order_purchase_timestamp) as first_
from `Business_case_study.orders` as o
right join `Business_case_study.customers` as c
on o.customer_id=c.customer_id
group by 1)
```

2. In-depth Exploration:

    1. 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 total_orders
FROM
  `Business_case_study.orders`
GROUP BY
  order_year
ORDER BY
  order_year;
```

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

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
  COUNT(order_id) AS total_orders
FROM
  `Business_case_study.orders`
GROUP BY
  order_month
```

```
ORDER BY
  order_month;
```

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
   - 0-6 hrs : Dawn
   - 7-12 hrs : Mornings
   - 13-18 hrs : Afternoon
   - 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
'Morning'
    WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'
    ELSE 'Night'
  END AS time_of_day,
  COUNT(order_id) AS total_orders
FROM
  `Business_case_study.orders`
GROUP BY
  time_of_day
ORDER BY
  total_orders DESC;
```

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

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

```
select distinct extract(month from o.order_purchase_timestamp) as
month_,
c.customer_state,
count(o.order_id) as count_,
from `Business_case_study.orders` as o
right join `Business_case_study.customers` as c
on o.customer_id=c.customer_id
group by month_,c.customer_state
```

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

```
select
count(customer_id) as count_,
```

```
Customer_state
from `Business_case_study.customers`
group by 2
```

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

1. 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.

```
SELECT
  ROUND(
    (MAX(total_payment) - MIN(total_payment)) / MIN(total_payment) *
100, 2
  ) AS percent_increase
FROM (
SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
  SUM(p.payment_value) AS total_payment
FROM
  `Business_case_study.payments` AS p
INNER JOIN
  `Business_case_study.orders` AS o
ON
  p.order_id = o.order_id
WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018)
  AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY
  order_year
ORDER BY
  order_year)
```

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

```
select  distinct c.customer_state,sum(p.payment_value) over(partition
by c.customer_state) as total_order_value,
avg(p.payment_value) over(partition by c.customer_state) as avg_value
from `Business_case_study.payments` as p
inner join `Business_case_study.orders` as o on p.order_id=o.order_id
inner join `Business_case_study.customers` as c on
o.customer_id=c.customer_id
```

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

```
select  distinct c.customer_state,sum(oi.freight_value) over(partition
by c.customer_state) as total_order_value,
avg(oi.freight_value) over(partition by c.customer_state) as avg_value
from `Business_case_study.order_item` as oi
inner join `Business_case_study.orders` as o on oi.order_id=o.order_id
inner join `Business_case_study.customers` as c on
o.customer_id=c.customer_id
```

1. Analysis based on sales, freight and delivery time.
    1. 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:
        - time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
        - diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date

```
select
date_diff(order_delivered_customer_date,order_purchase_timestamp,day)
as time_to_deliver,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,
day) as diff_estimated_delivery
from `Business_case_study.orders`
```

    2. Find out the top 5 states with the highest & lowest average freight value.

```
WITH state_avg_freight AS (
  SELECT
    c.customer_state,
    ROUND(AVG(oi.freight_value), 2) AS avg_freight
  FROM
    `Business_case_study.order_item` AS oi
  INNER JOIN
    `Business_case_study.orders` AS o ON oi.order_id = o.order_id
  INNER JOIN
```

```
    `Business_case_study.customers` AS c ON o.customer_id =
c.customer_id
  GROUP BY
    c.customer_state
),

ranked_states AS (
  SELECT
    customer_state,
    avg_freight,
    RANK() OVER (ORDER BY avg_freight DESC) AS high_rank,
    RANK() OVER (ORDER BY avg_freight ASC) AS low_rank
  FROM
    state_avg_freight
)

SELECT *
FROM ranked_states
WHERE high_rank <= 5 OR low_rank <= 5
ORDER BY avg_freight DESC;
```

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

```
WITH state_time_delivery AS (
  SELECT
    c.customer_state,

ROUND(AVG(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)), 2) AS avg_time_delivery
  FROM
    `Business_case_study.orders` AS o
  INNER JOIN
    `Business_case_study.customers` AS c ON o.customer_id =
c.customer_id
  GROUP BY
    c.customer_state
),

ranked_states AS (
  SELECT
    customer_state,
    avg_time_delivery,
```

```
    RANK() OVER (ORDER BY avg_time_delivery DESC) AS high_rank,
    RANK() OVER (ORDER BY avg_time_delivery ASC) AS low_rank
  FROM
    state_time_delivery
)

SELECT *
FROM ranked_states
WHERE high_rank <= 5 OR low_rank <= 5
ORDER BY avg_time_delivery DESC;
```

4. 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.

```
with avg_delivery_time as(
select
c.customer_state,avg(date_diff(order_delivered_customer_date,order_est
imated_delivery_date,day)) as avg_delivery
from `Business_case_study.orders` as o
inner join `Business_case_study.customers` as c
on o.customer_id=c.customer_id
group by c.customer_state),

ranked_state as(
  select
  customer_state,
  avg_delivery,
  rank() over(order by avg_delivery) as ranker
  from  avg_delivery_time
)

select * from ranked_state
where ranker<=5
order by ranker;
```

6. Analysis based on the payments:

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

```
select extract(month from o.order_purchase_timestamp) as month_,
p.payment_type,
count(o.order_id) as count_
from `Business_case_study.payments` as p
inner join `Business_case_study.orders` as o
on p.order_id=o.order_id
group by 1,2
```

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

```
select payment_sequential,payment_installments,
count(order_id)
from `Business_case_study.payments`
group by payment_sequential,payment_installments
having payment_installments=payment_sequential
```