1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset**

   a. Data type of columns in a table

   Datatypes of columns in customer's table -

Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

   Datatypes of all the tables are generally string and integers and timestamps

   b. Time period for which the data is given:

   The problem statement states that this business case has information of orders from 2016 to 2018. This can be validated from the orders table -

```
SELECT DISTINCT EXTRACT(YEAR FROM order_purchase_timestamp) as year
FROM `targetcase.orders`;
```

Row	year
1	2017
2	2018
3	2016

   c. Cities and States of customers ordered during the given period

```
SELECT distinct customer_state, customer_city
FROM `targetcase.customers`
order by customer_state
```

13	AL	canapi

The customers are from 27 different states and 4119 cities across Brazil.

2. **In-depth Exploration:**
   a. 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?

      The growing trend of ecommerce in Brazil can be analyzed from the number of purchase orders placed over the years. The data gives us an idea about orders placed at Brazil's largest Ecommerce platform over 3 years.

      SQL query :

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) as year,
COUNT(EXTRACT(YEAR FROM order_purchase_timestamp)) AS number_of_orders
FROM `targetcase.orders`
GROUP by year
ORDER by year DESC
```


      NOTE – 2016 data has only 3 months of data in the dataset.

      We can also see that specific seasons in the year attract more customers,

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) as year,
       EXTRACT(MONTH FROM order_purchase_timestamp) as month,
       COUNT(EXTRACT(YEAR FROM order_purchase_timestamp)) AS
number_of_orders
FROM `targetcase.orders`
GROUP by year, month
ORDER by year DESC, month ASC
```

      INSIGHTS:

As we can see there has been an exponential growth in number of orders in 2017 compared to 2016. (although 2016 data has only 3 months). 2017 to 2018 also shows a significant growth in the number of orders.

Although we can group by the count of orders based on month, that doesn't give us the whole picture as 2016 has only 3 months of data.

From the 2nd query result, we can say that the number of orders show an increase during the festive seasons of November , December, Jan (stays through March). Peaks can be observed during these months.

RECOMMENDATIONS:

We could try to run targeted marketing campaigns during the off-season based on demographic information to increase the sales, we could also get data on the products lying in customer carts and offer discounts to complete the sale. We could also give more attractive offers during the festive seasons to boost revenue.

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

Considering :

Dawn – 3AM to 6AM

Morning – 6AM to 11.59AM

Afternnoon – 12PM to 5PM

Night – 6PM to 3AM

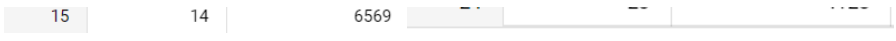
```
SELECT DISTINCT EXTRACT(HOUR FROM order_purchase_timestamp) as
hour_of_the_day,

       COUNT(order_id) AS number_of_orders

FROM `targetcase.orders`

GROUP BY hour_of_the_day

Order by hour_of_the_day ASC;
```



INSIGHTS :

From the following result we can say that the orders seem to be going up late morning (10 to 12PM) to night time around 9PM. Post which the number starts going down. Post 1AM, there are very few orders.

RECOMMENDATIONS : We could run more advertisements on sales, discount, offers during the active hours to boost sales.

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

a. Get month on month orders by states

```
SELECT c1.customer_state AS state,EXTRACT(MONTH FROM
o1.order_purchase_timestamp) as month,

       COUNT(o1.order_id) AS number_of_orders

FROM `targetcase.orders` as o1

LEFT JOIN `targetcase.customers`as c1

on o1.customer_id = c1.customer_id

GROUP BY state, month

ORDER BY state ASC, month ASC
```


By joining the tables order and customers, we can get data for month-on-month sales in all the states.

INSIGHTS:

If we consider data for year 2017, we can see that the order in many states go up in nov and dec. For example -

RECOMMENDATIONS:

We can form clusters of states based on sales and run marketing campaigns to get more sales from states with low orders.

b. Distribution of customers across the states in Brazil

```
SELECT customer_state, COUNT(customer_id) AS no_of_customer
```

```
FROM `targetcase.customers`

GROUP BY customer_state;
```

Row	customer_state	no_of_customer
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747
11	PE	1652
12	PB	536
13	ES	2033
14	PR	5045

Row	customer_state	no_of_customer
15	RO	253
16	MS	715
17	PA	975
18	TO	280
19	MT	907
20	PI	495
21	AL	413
22	AM	148
23	DF	2140
24	SE	350
25	RR	46
26	AP	68
27	AC	81

INSIGHTS :

By analyzing this, we can infer which states have a big customer base and which don't. By joining this table with orders table, we can also check which states contribute to revenue growth and whether that has any correlation with the number of customers.

RECOMMENDATIONS:

Based on the above analysis, we can target the correct states with correct strategies (Marketing Campaigns, More offers)

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

   a. 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

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) as years,

ROUND(SUM(p1.payment_value),2) AS total_amt

from `targetcase.orders` as o1

LEFT JOIN `targetcase.payments` as p1
```

```
ON o1.order_id = p1.order_id
WHERE EXTRACT(YEAR FROM order_purchase_timestamp) IN (2017,2018)
AND EXTRACT(MONTH FROM o1.order_purchase_timestamp) NOT IN (9,10,11,12)
GROUP BY EXTRACT(YEAR FROM order_purchase_timestamp);
```


INSIGHTS :

The percentage increase in orders can be calculated from the above total sales: the **increase** in sales in 2018 is around **42%** compared to 2017

b. Mean & Sum of price and freight value by customer state

```
SELECT customer_state AS state, ROUND(SUM(price),2) as Sum_price,
ROUND(AVG(price),2) as Mean_price, ROUND(SUM(freight_value),2) AS
Sum_freight,
ROUND(AVG(freight_value),2) AS Mean_freight
FROM `targetcase.order_items` as o1
LEFT JOIN `targetcase.orders` as o2
ON o1.order_id = o2.order_id
LEFT JOIN `targetcase.customers` as c1
ON o2.customer_id = c1.customer_id
GROUP BY state;
```

INSIGHTS:

From this analysis we can infer which states cost low in terms of transportation costs. Some states have pretty low freight values and high average orders which amounts to higher profits.

RECOMMENDATIONS:

We can do some research on states which have high transportation costs and low average order values, and try to increase the average order value by running offers and discounts to maximize the profits and cover up the transporation costs hence increasing the revenue.

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```
SELECT o2.order_id,

ROUND(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY),2) AS Time_to_delivery,

ROUND(DATE_DIFF( order_estimated_delivery_date, order_purchase_timestamp,
DAY),2) AS Estimated_delivery,

ROUND(DATE_DIFF( order_estimated_delivery_date, order_delivered_customer_date,
DAY),2) AS diff_days

FROM `targetcase.order_items` as o1

LEFT JOIN `targetcase.orders` as o2

ON o1.order_id = o2.order_id;
```


INSIGHTS:

This tables gives us an idea about the journey of a product from purchase to delivery.

The number of days between order placement and delivery could be lowered based on in depth further analysis

RECOMMENDATIONS :

We can try to minimize the number of days between order placement and delivery by improving the carrier network. Try to analyze why delivery takes long time for eg , is it because of bad infrastructure, product availability, carrier incompetency. More reasoning and analysis on the above points would help us improve the customer experience.

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

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

diff_estimated_delivery = order_estimated_delivery_date-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_purchase_timestamp, DAY) AS
diff_estimated_delivery

FROM `targetcase.orders`

WHERE order_delivered_customer_date IS NOT NULL
```

10	302bb8109d097a9fc6e9cefc5...	33	28

INSIGHTS :

This table would give us an overall idea on our estimates on delivery period. Also, try to see why some products are delayed – do these products belong to certain areas or category? If yes, we can deep dive to find out ways to improve it.

RECOMMENDATIONS:

Analyze the orders delayed and find out a pattern based on the product category, or customer state. The pattern can be used to focus on delivery network of states which are facing issues. If the product of particular categories are delayed, then we can make sure these are well in stock and dispatched on time.

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

```
SELECT customer_state AS state,

ROUND(AVG(freight_value),2) AS Mean_freight,

ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)),2) AS Mean_time_to_delivery,

ROUND(AVG(DATE_DIFF( order_estimated_delivery_date, order_purchase_timestamp,
DAY)),2) AS Mean_estimated_delivery

FROM `targetcase.order_items` as o1

LEFT JOIN `targetcase.orders` as o2

ON o1.order_id = o2.order_id

LEFT JOIN `targetcase.customers` as c1

ON o2.customer_id = c1.customer_id

GROUP BY state;
```

Row	state	Mean_freight	Mean_time_to_delivery	Mean_estimated_delivery
1	SP	15.15	8.26	18.9
2	RJ	20.96	14.69	26.1
3	PR	20.53	11.48	24.38
4	SC	21.47	14.52	25.51
5	DF	21.04	12.5	24.19
6	MG	20.63	11.52	24.31
7	PA	35.83	23.3	36.96
8	BA	26.36	18.77	29.14
9	GO	22.77	14.95	26.62
10	RS	21.74	14.71	28.31

4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

```
SELECT customer_state AS state,

ROUND(AVG(freight_value),2) AS Mean_freight,

ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)),2) AS Mean_time_to_delivery,

ROUND(AVG(DATE_DIFF( order_estimated_delivery_date, order_purchase_timestamp,
DAY)),2) AS Mean_estimated_delivery

FROM `targetcase.order_items` as o1

LEFT JOIN `targetcase.orders` as o2

ON o1.order_id = o2.order_id

LEFT JOIN `targetcase.customers` as c1

ON o2.customer_id = c1.customer_id

GROUP BY state

ORDER BY Mean_freight ASC

LIMIT 5;
```

Top 5 states with lowest freight value:

5	DF	21.04	12.5	24.19

Top 5 states with highest freight value:

5	PI	39.15	18.93	29.92

INSIGHTS :

The tables give us an idea on the states which have low/high transportation costs. We can use this to maximize profits by trying to increase average order values in states with higher transportation costs.

6. Top 5 states with highest/lowest average time to delivery

```
SELECT customer_state AS state,
ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)),2) AS Avg_time_to_delivery
FROM `targetcase.order_items` as o1
LEFT JOIN `targetcase.orders` as o2
ON o1.order_id = o2.order_id
LEFT JOIN `targetcase.customers` as c1
ON o2.customer_id = c1.customer_id
GROUP BY state
ORDER BY Avg_time_to_delivery ASC
LIMIT 5;
```

Top 5 states with lowest average time to delivery:

Row	state	Avg_time_to_delivery
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

Top 5 states with highest average time to delivery:

INSIGHTS:

This gives us an idea on the states where the delivery time is low/ high. Based on this, we can try to do an analysis on why the orders get delayed, and how to improve.

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

```
SELECT customer_state AS state,

ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)),2) AS Avg_time_to_delivery,

ROUND(AVG(DATE_DIFF( order_estimated_delivery_date, order_purchase_timestamp,
DAY)),2) AS Mean_estimated_delivery,

ROUND(AVG(DATE_DIFF( order_estimated_delivery_date,
order_delivered_customer_date, DAY)),2) AS diff_days

FROM `targetcase.order_items` as o1

LEFT JOIN `targetcase.orders` as o2

ON o1.order_id = o2.order_id

LEFT JOIN `targetcase.customers` as c1

ON o2.customer_id = c1.customer_id

GROUP BY state

ORDER BY diff_days DESC

LIMIT 5;
```

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

Top 5 states where delivery is really fast compared to estimated date :

INSIGHTS :

Based on this analysis, we can try to focus on the states where the delivery of a

product takes longer or equal number of days as estimated time and find out the reason (carrier incompetence, infrastructure issues or product availability)

6. Payment type analysis:

1. Month over Month count of orders for different payment types

```sql
SELECT p1.payment_type AS payment_type, EXTRACT(month FROM
o1.order_purchase_timestamp) AS Month,

COUNT(o1.order_id) AS No_of_orders

FROM `targetcase.orders` as o1

LEFT JOIN `targetcase.payments` as p1

ON o1.order_id = p1.order_id

GROUP by payment_type, month

Order by payment_type, month;
```

				25	credit_card	12	4378
37	debit_card	12	64				

INSIGHTS:

From this we can see that customers are more inclined towards taking credit. Credit card is the most popular payment gateway, followed by debit cards and UPI.

RECOMMENDATIONS:

More offers could be given out on credit cards to increase average order values and sales. For eg, 10% off on orders above 1000 and so on.

2. Count of orders based on the no. of payment installments

```sql
SELECT  payment_installments, COUNT(order_id) No_of_orders

from `targetcase.payments`

group by payment_installments

order by payment_installments ASC
```

11	10	5328

INSIGHTS:

We can infer that customers tend to prefer at least 1 to 3 payment installments. We can tie up

And introduce more options for installments and credit offers to increase sales.

RECOMMENDATIONS:

Giving out more credit card and installement options such as 10% off over a certain order value which will contribute to increase in average order values and covering up others costs which will have a good impact on overall sales.