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

1.1. Data type of all columns in the "customers" table.

Query:

```
SELECT column_name, data_type

FROM `sql_business_case.INFORMATION_SCHEMA.COLUMNS`

WHERE table_name = 'customers'
```

Output:

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

1.2. Get the time range between which the orders were placed.

Query:

```
SELECT MIN(order_purchase_timestamp) First_date,
MAX(order_purchase_timestamp) Last_date
FROM `sql_business_case.orders` limit 10
```

Output:

Row	First_date	Last_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

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

Query:

```
SELECT
c.customer_city,
c.customer_state,
COUNT(o.customer_id) customer_count from `sql_business_case.orders` o
JOIN `sql_business_case.customers` c
ON o.customer_id=c.customer_id
GROUP BY c.customer_city,c.customer_state
ORDER BY COUNT(o.customer_id) DESC
```

Output:

Row	customer_city	customer_state	customer_count
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521
6	campinas	SP	1444
7	porto alegre	RS	1379
8	salvador	BA	1245
9	guarulhos	SP	1189
10	sao bernardo do campo	SP	938

Insight:

City "**sao paulo**" has high client base

1.4. What is the **distribution of customers across states** (customer_state)?

Query:

```
select customer_state, count(customer_id) Customer_distribution from
`sql_business_case.customers`
group by customer_state
order by count(customer_id) desc
```

output:

Row	customer_state	Customer_distribution
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Insight:

State "**SP**", **"RJ", "MG"** have exceptionally high clientele

2. In-depth Exploration:

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

Query:

```
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS purchase_year,
  COUNT(order_id) AS total_orders
FROM `sql_business_case.orders`
GROUP BY purchase_year
ORDER BY purchase_year
```

output:

Row	purchase_year	total_orders
1	2016	329
2	2017	45101
3	2018	54011

Insight:

Drastic growth in total orders placed can be observed in **year – 2017**, overall a **growing trend** can be observed.

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

Query:

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS Purchase_year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS Purchase_month,
COUNT(order_id) AS total_orders_placed
FROM `sql_business_case.orders`
group by Purchase_year,Purchase_month
order by Purchase_year,Purchase_month
```

output:

Row	Purchase_year	Purchase_month	total_orders_placed
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245

Insight:

End of years – **2016 and 2018 have less orders** but it is a **exception for year - 2017**

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

Query:

```
  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 as peak_order_window ,
  count(*)  order_volume
  from `sql_business_case.orders`
  group by peak_order_window
  order by order_volume desc
```

output:

Row	peak_order_window	order_volume
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Insight:

Clearly **Afternoon** emerges as the **peak** shopping window for Brazilian customers, with **minimal** activity observed during **Dawn** hours.

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

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

Query:

```
with count_ as (
select
c.customer_state State,
format_date('%B %Y', date(o.order_purchase_timestamp)) Month,
count(o.order_id) Total_Orders ,
EXTRACT(MONTH FROM DATE(o.order_purchase_timestamp)) AS month_num,
EXTRACT(YEAR FROM DATE(o.order_purchase_timestamp)) AS year
 from `sql_business_case.customers` as c
join `sql_business_case.orders` o on c.customer_id=o.customer_id
group by c.customer_state, month ,month_num,year
)

select State,   Month , Total_Orders ,
 (
  round(((    Total_Orders - lag(Total_Orders) over(partition by state
order by year,month_num,state)    )/lag(Total_Orders) over(partition by
state order by year,month_num,state))*100,2)
 ) MoM_Orders_Growth_Percent

 from count_
 order by state, year,month_num
```

Output:

Row	state	month	Total_Orders	MoM_Orders_Growth_Percent
1	AC	January 2017	2	null
2	AC	February 2017	3	50.0
3	AC	March 2017	2	-33.33
4	AC	April 2017	5	150.0
5	AC	May 2017	8	60.0
6	AC	June 2017	4	-50.0
7	AC	July 2017	5	25.0
8	AC	August 2017	4	-20.0
9	AC	September 2017	5	25.0
10	AC	October 2017	6	20.0

Insights:

Month on month no. of orders placed in each state is as above.

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

Query:

```
select customer_state State,
```

```
round((count(*)/sum(count(*)) over())*100 ,2)
Customer_Percentage_Distribution
from `sql_business_case.customers`
group by customer_state
order by Customer_Percentage_Distribution desc
```

Output:

Row	State	Customer_Percentage_Distribution
1	SP	41.98
2	RJ	12.92
3	MG	11.7
4	RS	5.5
5	PR	5.07
6	SC	3.66
7	BA	3.4
8	DF	2.15
9	ES	2.04
10	GO	2.03

Insights:

It can be observed that **SP** has exceptionally high distribution of **41.98%** and **RR** being the least distributed location with only **0.05%**.

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

4.1. 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.

Query:

```
with cte as(
select
extract(year from date(o.order_purchase_timestamp) ) year,
extract(month from date(o.order_purchase_timestamp)) month,
p.payment_value
from `sql_business_case.orders` o
join `sql_business_case.payments` p on o.order_id = p.order_id

),
filtered_year as (
```

```
select year,
sum(payment_value) cost_of_orders
from cte
where month<9
group by year

)

select Year, round(cost_of_orders,2) Cost_of_Orders,
round(((cost_of_orders-lag(cost_of_orders) over(order by year)
)/lag(cost_of_orders) over(order by year) )*100,2) percent_change
from filtered_year
order by year
```

Output:

Row	Year	Cost_of_Orders	percent_change
1	2017	3669022.12	*null*
2	2018	8694733.84	136.98

Insights:

We can observe a **136.98% increase** in order costs for year **2018.**

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

Query:

```
select c.customer_state, count(c.customer_id) Tot_customers ,
round(sum(ot.price),2) Tot_Order_Price,round( avg(ot.price),2)
Avg_Order_Price

from `sql_business_case.customers` c
join `sql_business_case.orders` o on o.customer_id=c.customer_id
join `sql_business_case.order_items` ot on ot.order_id =o.order_id
group by c.customer_state
order by Tot_Order_Price desc, Avg_Order_Price desc
```

Output:

Row	customer_state	Tot_customers	Tot_Order_Price	Avg_Order_Price
1	SP	47449	5202955.05	109.65
2	RJ	14579	1824092.67	125.12
3	MG	13129	1585308.03	120.75
4	RS	6235	750304.02	120.34
5	PR	5740	683083.76	119.0
6	SC	4176	520553.34	124.65
7	BA	3799	511349.99	134.6
8	DF	2406	302603.94	125.77
9	GO	2333	294591.95	126.27
10	ES	2256	275037.31	121.91

Insights:

States with the highest customer volumes like SP, RJ, and MG contribute the most to total revenue, though their average order prices are moderate. Smaller states like PB, AL, and AC have fewer customers but significantly higher average order prices, indicating high-value purchases per customer.

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

Query:

```
select c.customer_state, count(c.customer_id) Tot_customers ,
round(sum(ot.freight_value),2) Tot_Freight_Value,round(
avg(ot.freight_value),2) Avg_Freight_Value

from `sql_business_case.customers` c
join `sql_business_case.orders` o on o.customer_id=c.customer_id
join `sql_business_case.order_items` ot on ot.order_id =o.order_id
group by c.customer_state
order by Tot_Order_Price desc, Avg_Order_Price desc
```

Output:

Row	customer_state	Tot_customers	Tot_Freight_Value	Avg_Freight_Value
1	SP	47449	718723.07	15.15
2	RJ	14579	305589.31	20.96
3	MG	13129	270853.46	20.63
4	RS	6235	135522.74	21.74
5	PR	5740	117851.68	20.53
6	BA	3799	100156.68	26.36
7	SC	4176	89660.26	21.47
8	PE	1806	59449.66	32.92
9	GO	2333	53114.98	22.77

Insights:

Heavily populated states like SP and RJ have lower average freight values, suggesting economies of scale or better logistics infrastructure. In contrast, less populated and remote states such as RR, PB, and AC face significantly higher freight costs, likely due to distance and limited transportation access.

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

5.1. 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

Query:

```
select order_id,
date_diff(order_delivered_customer_date, order_purchase_timestamp ,DAY)
Delivery_Time,
date_diff(order_delivered_customer_date , order_estimated_delivery_date
,DAY) Delivery_Date_Accuracy
 from `sql_business_case.orders`
```

Output:

Row	order_id	Delivery_Time	Delivery_Date_Accuracy
1	ecab90c9933c58908d3d6add7...	30	-16
2	8563039e855156e48fccee4d61...	30	0
3	6ea2f835b4556291ffdc53fa0b3...	33	7
4	6a0a8bfbbe700284feb0845d95...	36	17
5	9d531c565e28c3e0d756192f8...	56	32
6	8fc207e94fa91a7649c5a5dab6...	54	32
7	f31535f21d145b2345e2bf7f09...	81	49
8	06ae7271902bbb087fc093137f...	31	-1
9	4906eeadde5f70b308c20c4a8f...	32	6
10	bca3dc20a3ec02261c5b17dc2...	30	0

Insights:

Delivery times range from **0 to 81 days.**

Most deliveries fall in the **30–40** day range.

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

Query:

```
select c.customer_state, round(avg(ot.freight_value),2)
Tot_Freight_Value

from `sql_business_case.customers` c
join `sql_business_case.orders` o on o.customer_id=c.customer_id
join `sql_business_case.order_items` ot on ot.order_id =o.order_id
group by c.customer_state
order by Tot_Freight_Value desc
limit 5
```

Output:

Row	customer_state	Highest_Avg_Freight_Value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

Row	customer_state	Lowest_Avg_Freight_Value
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

Insights:

Lowest Freight Value: **(SP)** Stands Out meaning higher order density, or closer proximity to fulfillment centers.
expanding warehousing in or near SP could further reduce delivery costs across neighboring regions.

Highest Freight Value: Freight costs are **significantly higher in** RR, RO, AC , These areas may lack local warehouses or distribution centers, leading to **longer and costlier shipments**.

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

Query:

```
select c.customer_state,
round(avg(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp ,DAY)),2) Highest_Avg_Delivery_Time
from `sql_business_case.customers` c
join `sql_business_case.orders` o on o.customer_id=c.customer_id
group by c.customer_state
 order by  Highest_Avg_Delivery_Time desc
 limit 5
```

Output:

Row	customer_state	Highest_Avg_Delivery_Time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

Row	customer_state	Lowest_Avg_Delivery_Time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

Insights:

- **RR** has **the longest average delivery time at 28.98 days**, indicating major logistical challenges, likely due to remote location and limited infrastructure.
- These states suggest a need for **logistics investment, better infrastructure, or optimized distribution centers** to improve service in these underserved regions.

5.4. 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.

Query:

```
with cte as(
select c.customer_state
State,round(date_diff(o.order_delivered_customer_date,
o.order_purchase_timestamp,day),2) Actual_Delivery_Time,
round(date_diff(o.order_estimated_delivery_date,
o.order_purchase_timestamp,day),2) Estimated_Delivery_Time
from `sql_business_case.customers` c
join `sql_business_case.orders` o on o.customer_id=c.customer_id

)


select State,round( avg(Estimated_Delivery_Time-Actual_Delivery_Time
),2) Avg_Delivery_Time from cte
group by State
order by Avg_Delivery_Time
```

Output:

Row	State	Avg_Delivery_Time
1	AL	8.17
2	MA	8.97
3	SE	9.45
4	ES	9.89
5	CE	10.19

Insights:

- **AL** has recorded with **shortest average delivery time gap** of **8.17 days**. This suggests highly efficient logistics or conservative delivery estimates.
- **All five states deliver ahead of schedule**, which is a positive sign for customer satisfaction.

6. Analysis based on the payments:

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

Query:

```
select p.payment_type , format_date('%B
%Y',date(o.order_purchase_timestamp)) Time_period,count(o.order_id)
Orders
from `sql_business_case.orders` o
join `sql_business_case.payments` p on o.order_id=p.order_id
group by p.payment_type,time_period
order by Time_period
```

Output:

Row	payment_type	Time_period	Orders
1	credit_card	April 2017	1846
2	UPI	April 2017	496
3	debit_card	April 2017	27
4	voucher	April 2017	202
5	credit_card	April 2018	5455
6	voucher	April 2018	370
7	UPI	April 2018	1287
8	debit_card	April 2018	97
9	UPI	August 2017	938
10	credit_card	August 2017	3284

Insights:

- **Credit Card** is consistently the **most used payment method** across all months, with significant growth
- **UPI usage grew rapidly**, showing a clear adoption trend — from 197 (Jan 2017) to 1518 (Jan 2018), indicating increasing user trust and platform integration.

- All payment types (especially credit card and UPI) show **strong month-on-month and year-on-year growth**, reflecting:
    - Growing e-commerce penetration
    - Increasing digital payment adoption in Brazil

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

Query:

```
select Payment_Installments,COUNT(*) Total_Orders from
`sql_business_case.payments`
group by payment_installments
order by payment_installments
```

Output:

Row	Payment_Installments	Total_Orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

Insights:

- **Majority of customers (over 52,000)** choose to pay in **1 installment** — indicating a preference for **full payment upfront**.
- **gradual drop** can be observed in order volume as the number of installments increases

Recommendations Summary:

1. **Expand Logistics Infrastructure in Remote States:**
   States like RR, AC, and PB experience high freight costs and long delivery times. Investing in local warehouses or distribution hubs can reduce shipping costs and improve customer satisfaction.
2. **Leverage SP as a Strategic Fulfillment Hub:**
   SP has high order density and the lowest average freight costs. Enhancing warehousing and fulfillment capabilities in this region can optimize nationwide logistics.
3. **Optimize Delivery Promises Using Data:**
   States like AL consistently deliver ahead of estimated dates. Update estimated delivery algorithms to reflect actual performance for improved transparency and trust.

4. **Incentivize High-Value Regions with Targeted Promotions:**
   States with high average order prices but fewer customers (e.g., PB, AL, AC) show potential for premium product targeting and personalized marketing strategies.
5. **Promote Afternoon Deals or Campaigns:**
   Since the peak ordering window is in the afternoon, scheduling flash sales or marketing pushes during this time can capitalize on user behavior.
6. **Encourage Full Upfront Payments:**
   Over 52,000 customers prefer one-time payments. Reinforcing this behavior with small incentives (e.g., cashback for full payment) can reduce transaction processing costs.
7. **Monitor and Enhance UPI Adoption:**
   UPI usage is rising steadily. Ensure seamless UPI integration and provide incentives for its use to align with Brazil's digital payment evolution.