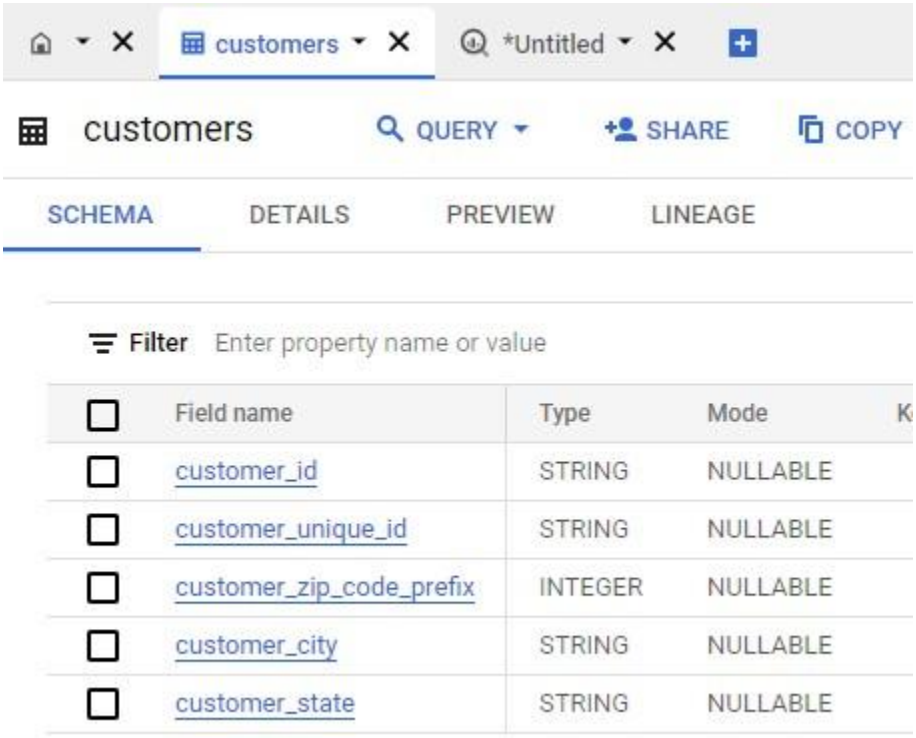
Analyze the given dataset provided by one of the leading Brazilian Clients to extract valuable insights and provide actionable recommendations.

*Note - *The screenshot contains only the top 10 results as in several queries I analyzed the trends and the output could be more than what some of the screenshot shows.*

1. Imported the dataset and did the usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1.1. The data type of all columns in the "customers" table.



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

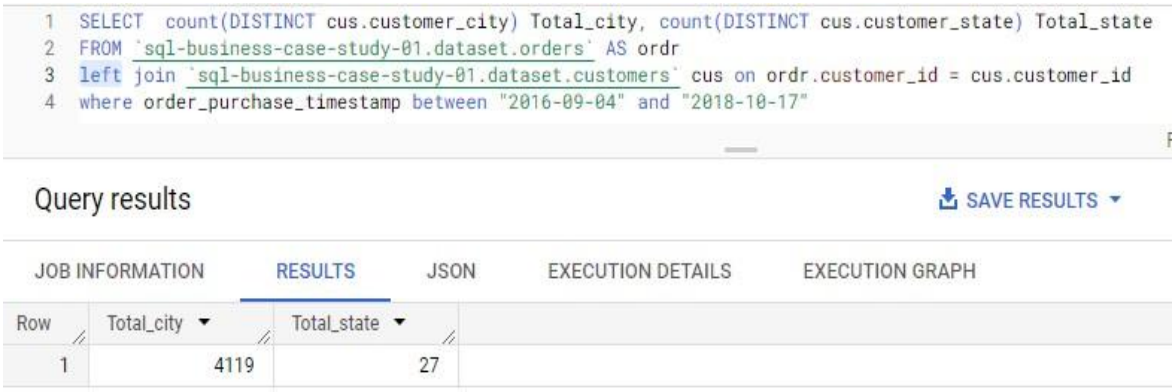
```
SELECT  min(order_purchase_timestamp) Orders_From,
max(order_purchase_timestamp) Orders_Till
FROM `brazilian-client.dataset.orders`
```

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

Comments For 1.2 - This dataset contains orders from 09 Sep 2016 to 17 Oct 2018

1.3. Counted the Cities and states of customers who ordered during the given period.

```
SELECT  count(DISTINCT cus.customer_city) Total_city, count(DISTINCT
cus.customer_state) Total_state
FROM `brazilian-client.dataset.orders` AS ordr
Left join `brazilian-client.dataset.customers` cus on ordr.customer_id = cus.customer_id
where order_purchase_timestamp between "2016-09-04" and "2018-10-17"
```



Comments For 1.3 - There are a total of 27 states and 4119 cities included.

2. In-depth Exploration:

2.1. I found out if there is a growing trend in the number of orders placed over the past years.

```
SELECT
EXTRACT(year FROM order_purchase_timestamp) AS Year,
EXTRACT(month FROM order_purchase_timestamp) AS Month,
count(distinct order_id)Number_Of_Orders
FROM `brazilian-client.dataset.orders`
group by Year, Month
order by Year, Month
```

JOB INFORMATION | RESULTS | JSON | EXECUTION I

Row	Year	Month	Number_Of_Orders
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
10	2017	7	4026

***Comments For 2.1 - There is a general increase in the number of order trends up until September and October month of the year 2018.**

2.2. I found out if there is some kind of monthly seasonality in terms of the number of orders being placed.

I would like to answer this in two parts.

Maximum Order Trend in each year for given Data

```
SELECT
 FORMAT_DATE('%B', order_purchase_timestamp) AS Month,
 count(distinct order_id)Number_Of_Orders,
 FROM `brazilian-client.dataset.orders`
 group by Month
 order by Number_Of_Orders desc
```

Row	Month	Number_Of_Orders
1	August	10843
2	May	10573
3	July	10318
4	March	9893
5	June	9412
6	April	9343
7	February	8508
8	January	8069
9	November	7544
10	December	5674

Minimum Order Trend in each year for given Data

```
SELECT
 FORMAT_DATE('%B', order_purchase_timestamp) AS Month,
 count(distinct order_id)Number_Of_Orders,
 FROM `brazilian-client.dataset.orders`
 group by Month
 order by Number_Of_Orders
```

Row	Month	Number_Of_Orders
1	September	4305
2	October	4959
3	December	5674
4	November	7544
5	January	8069
6	February	8508
7	April	9343
8	June	9412
9	March	9893
10	July	10318

*Comments For 2.2 -
When commenting on seasonality in terms of the number of orders, it is crucial to look at the highest-order trends and lowest-order trends.

So mid months of August, May, and July generally see an increase in the number of orders.
Whereas the months of September, October, and December see a general trend of a lower number of orders.

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

2.3.1. 0-6 hrs: Dawn

2.3.2. 7-12 hrs: Mornings

2.3.3. 13-18 hrs: Afternoon

2.3.4. 19-23 hrs: Night

```
with cte as (SELECT distinct order_id Orders,
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 time_of_day
 FROM `brazilian-client.dataset.orders` )

 select time_of_day, count(Orders) Number_Of_Orders
 from cte
 group by time_of_day
 order by Number_Of_Orders desc
```

Row	time_of_day	Number_Of_Orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

Comments For 2.3 - As per the results, it could be safely inferred that the highest number of orders were placed in the afternoon.

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

3.1. I extracted the month-on-month number of orders placed in each state.

```
SELECT
cus.customer_state State,
EXTRACT(month FROM order_purchase_timestamp) AS Month_Number,
FORMAT_DATE("%b", order_purchase_timestamp) AS Month,
count(distinct order_id)Number_Of_Orders
FROM `brazilian-client.dataset.orders` odr
Left join `brazilian-client.dataset.customers` cus on  odr.customer_id =
cus.customer_id
group by State, Month_Number, Month
order by State, Month_Number
```

Row	State	Month_Number	Month	Number_Of_Orders
1	AC	1	Jan	8
2	AC	2	Feb	6
3	AC	3	Mar	4
4	AC	4	Apr	9
5	AC	5	May	10
6	AC	6	Jun	7
7	AC	7	Jul	9
8	AC	8	Aug	7
9	AC	9	Sep	5
10	AC	10	Oct	6

Comments For 3.1 - Analysis done as per the question asked by the client.

3.2. I found out the customer distribution across all the states.

```
SELECT
 customer_state State,
 count( distinct customer_id) Number_Of_Customers
 FROM `brazilian-client.dataset.customers`
 group by State
 order by Number_Of_Customers desc
```

Row	State	Number_Of_Custome
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

*__Comments For 3.2 - The top 3 states in terms of the number of unique customers are SP, RJ, MG__

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

4.1. Analyzed the % increase in the cost of orders from 2017 to 2018 (including the months between Jan to Aug only).

```
With cte as (Select
SUM(CASE WHEN EXTRACT(year FROM order_purchase_timestamp) =2017 and
EXTRACT(month FROM order_purchase_timestamp) between 1 and 8 THEN
payment_value ELSE 0 END) For_2017,
SUM(CASE WHEN EXTRACT(year FROM order_purchase_timestamp) =2018 and
EXTRACT(month FROM order_purchase_timestamp) between 1 and 8 THEN
payment_value ELSE 0 END) For_2018,
FROM `brazilian-client.dataset.orders` odr
join `brazilian-client.dataset.payments` p on odr.order_id = p.order_id)
select (((For_2018-For_2017)/For_2017)*100) AS Percent_Growth
from cte
```

Row	Percent_Growth
1	136.97687164665447

Comments For 4.1 - There has been a sweet 136.97% growth in the cost of orders.

4.2. Calculated the total and average value of the order price for each state.

```
SELECT
 cus.customer_state State,
 round(sum(price),3) Total_value,
 round(avg(price),3) Average_value
 FROM `sql-business-case-study-01.dataset.order_items` oi
join `sql-business-case-study-01.dataset.orders` odr on oi.order_id = odr.order_id
Left join `sql-business-case-study-01.dataset.customers` cus on  odr.customer_id =
cus.customer_id
group by state
order by Average_value desc
```

Row	State	Total_value	Average_value
1	PB	115268.08	191.475
2	AL	80314.81	180.889
3	AC	15982.95	173.728
4	RO	46140.64	165.974
5	PA	178947.81	165.692
6	AP	13474.3	164.321
7	PI	86914.08	160.358
8	TO	49621.74	157.529
9	RN	83034.98	156.966
10	CE	227254.71	153.758

***Comments For 4.2 - As per the above results, the best in terms of average order price is PB**

4.3. Calculated the Total and average value of order freight for each state.

```
SELECT
 cus.customer_state State,
 sum(freight_value) Total_Freight_Value,
 avg(freight_value) Average_Freight_Value
 FROM `brazilian-client.dataset.orders` odr
 join `brazilian-client.dataset.order_items` oit on odr.order_id = oit.order_id
Left join `brazilian-client.dataset.customers` cus on   odr.customer_id = cus.customer_id
group by state
order by Average_Freight_Value asc
```

Row	State	Total_Freight_Value	Average_Freight_Value
1	SP	718723.0699999...	15.147275390419248
2	PR	117851.6800000...	20.531651567944248
3	MG	270853.4600000...	20.630166806306541
4	RJ	305589.3100000...	20.96092393168248
5	DF	50625.49999999...	21.041354945968383
6	SC	89660.26000000...	21.470368773946436
7	RS	135522.7400000...	21.735804330392945
8	ES	49764.59999999...	22.058776595744682
9	GO	53114.97999999...	22.766815259322794
10	MS	19144.03000000...	23.374884004884006

***Comments For 4.3 - Based on the above data, the state with the least average freight value is SP**

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

5.1. - Extracted 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.
- Calculated the delivery time and the difference between the estimated & actual delivery date.

```
SELECT
    order_id,
    date_diff(order_delivered_customer_date,
    order_purchase_timestamp,day)time_to_deliver,
    date_diff(order_estimated_delivery_date,
    order_delivered_customer_date,day) diff_estimated_delivery
    FROM `brazilian-client.dataset.orders`
    where order_delivered_customer_date is not null and order_status
    ="delivered"
```

Row	order_id	time_to_deliver	diff_estimated_delive
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

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

```
with bottom5 as (SELECT
c.customer_state,
avg(freight_value) Fright_value,
FROM `brazilian-client.dataset.orders` o
join `brazilian-client.dataset.customers` c on o.customer_id = c.customer_id
```

```
join `brazilian-client.dataset.order_items` oi on o.order_id = oi.order_id
group by c.customer_state
order by Fright_value limit 5),

bottom5cte as (select *, row_number() over(order by bottom5.Fright_value) as NU

from bottom5),

top5 as (
 SELECT
c.customer_state,
avg(freight_value) Fright_value
FROM `brazilian-client.dataset.orders` o
join `brazilian-client.dataset.customers` c on o.customer_id = c.customer_id
join `brazilian-client.dataset.order_items` oi on o.order_id = oi.order_id
group by c.customer_state
order by Fright_value desc limit 5
),

top5cte as (select top5.customer_state, top5.Fright_value,
row_number() over (order by top5.Fright_value) as NU
from top5
order by top5.Fright_value)

select
top5cte.customer_state Top_5_States,
top5cte.Fright_value Freight_Value,
bottom5cte.customer_state Bottom_5_States,
bottom5cte.Fright_value Freight_Value
 from
bottom5cte join top5cte using (NU)
```

Row	Top_5_States	Freight_Value	Bottom_5_States	Freight_Value_1
1	PI	39.147970479704767	SP	15.147275390419248
2	AC	40.073369565217405	PR	20.531651567944248
3	RO	41.069712230215842	MG	20.630166806306541
4	PB	42.723803986710941	RJ	20.96092393168248
5	RR	42.984423076923093	DF	21.041354945968383

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

```
with bottom5 as (SELECT
c.customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,day))time_to_deliver
FROM `brazilian-client.dataset.orders` o
join `brazilian-client.dataset.customers` c on o.customer_id = c.customer_id
group by c.customer_state
order by time_to_deliver limit 5),

bottom5cte as (select *, row_number() over(order by bottom5.time_to_deliver) as NU
from bottom5),


top5 as (SELECT
c.customer_state,
avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,day))time_to_deliver
FROM `brazilian-client.dataset.orders` o
join `brazilian-client.dataset.customers` c on o.customer_id = c.customer_id
group by c.customer_state
order by time_to_deliver desc limit 5),

 top5cte as (select *, row_number() over(order by top5.time_to_deliver) as NU
from top5)

select
top5cte.customer_state Top_5_States,
top5cte.time_to_deliver Time_To_Deliver,
bottom5cte.customer_state Bottom_5_States,
bottom5cte.time_to_deliver Time_To_Deliver
 from
bottom5cte join top5cte using (NU)
```

Row	Top_5_States	Time_To_Deliver	Bottom_5_States	Time_To_Deliver_1
1	PA	23.316067653276981	SP	8.2980935447227022
2	AL	24.040302267002513	PR	11.526711354864908
3	AM	25.986206896551728	MG	11.54218777523343
4	AP	26.731343283582085	DF	12.509134615384616
5	RR	28.975609756097562	SC	14.475183305132528

5.4. Extracted the top 5 states where the order delivery is really fast compared to the estimated delivery date.
I used the difference between the averages of actual & estimated delivery dates to figure out how fast the delivery was for each state.

```
with cte as (SELECT
 C.customer_state,
avg(date_diff(order_delivered_customer_date,order_purchase_timestamp, day))
No_of_actual_average_delivery_time,
avg(date_diff(order_estimated_delivery_date,order_purchase_timestamp, day))
No_of_estimated_average_delivery_time,
FROM `brazilian-client.dataset.orders` o
join `brazilian-client.dataset.customers` c on o.customer_id = c.customer_id
where o.order_status = "delivered"
group by c.customer_state
order by c.customer_state)


select customer_state,
(No_of_estimated_average_delivery_time - No_of_actual_average_delivery_time)
Difference
from cte
order by Difference limit 5
```

Row	customer_state	Difference
1	AL	8.1687657430730631
2	MA	8.9665271966527413
3	SE	9.4537313432835681
4	ES	9.8882205513784669
5	CE	10.186864738076622

6. Analysis based on the payments:

6.1. Analyzed the month-on-month number of orders placed using different payment types.

```
SELECT
payment_type Payment_Type,
EXTRACT(month FROM order_purchase_timestamp) AS Month_Number,
FORMAT_DATE("%b", order_purchase_timestamp) AS Month,
count(odr.order_id)Number_Of_Orders
FROM `brazilian-client.dataset.orders` odr
Left join `brazilian-client.dataset.payments` pay on  odr.order_id = pay.order_id
group by Payment_Type, Month_Number, Month
order by Payment_Type, Month_Number
```

Row	Payment_Type	Month_Number	Month	Number_Of_Orders
1	*null*	9	Sep	1
2	UPI	1	Jan	1715
3	UPI	2	Feb	1723
4	UPI	3	Mar	1942
5	UPI	4	Apr	1783
6	UPI	5	May	2035
7	UPI	6	Jun	1807
8	UPI	7	Jul	2074
9	UPI	8	Aug	2077
10	UPI	9	Sep	903

Comments For 6.1 - The above data shows the payment distribution by payment type in each state.

However, 1 payment has payment_type = null, we need clarification from the business on how exactly to treat the null value in this case.

6.2. Extracted the number of orders placed on the basis of the payment installments that have been paid.

```
SELECT payment_sequential No_Of_Installments,
count(distinct order_id) No_of_orders
 FROM `brazilian-client.dataset.payments`
 where payment_sequential>=2
 group by payment_sequential
 order by payment_sequential desc
```

Row	No_Of_Installments	No_of_orders
1	29	1
2	28	1
3	27	1
4	26	2
5	25	2
6	24	2
7	23	2
8	22	3
9	21	4
10	20	4

Comments For 6.2 - I have calculated only those orders that had payment sequential equal to or greater than 2 so that I can exclude any single payment order that is not on EMI and has been paid in a single go.