TARGET BUSINESS STUDY

I. 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

QUERY :

```
SELECT column_name,
       data_type
From
`shailja-project-444415.TARGETBUSINESSSTUDY20250125.INFORMATION_SCHEMA.COLUMNS`
where table_name = 'CUSTOMER'
```

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

INSIGHT:

- Identifying data types helps determine if any columns need conversion (e.g., dates stored as strings).
- Checking for missing values ensures data completeness.

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

QUERY :

```
SELECT
    MIN(order_purchase_timestamp) AS earliest_order_time,
```

```
    MAX(order_purchase_timestamp) AS latest_order_time,
                                    TIMESTAMP_DIFF(MAX(order_purchase_timestamp),
MIN(order_purchase_timestamp), SECOND) AS time_difference_in_seconds
FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS`
```

OUTPUT:

Row	earliest_order_time	latest_order_time	time_difference_in_seconds
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	66773699

INSIGHT:

- Finding the earliest and latest order dates gives the business timeline.
- Seasonal trends can be inferred if data spans multiple years.

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

QUERY:

```
select count(distinct c.customer_city) as city_count,
count(distinct c.customer_state) as state_count
from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` c
join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
on c.customer_id = o.customer_id
```

OUTPUT:

Row	city_count	state_count
1	4119	27

INSIGHT:

- The most common cities and states help identify key markets.
- Comparing order frequency by state/city can highlight high-demand region.

II. In-depth Exploration:

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 past_year,
count(order_id) as order_number
from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS`
group by past_year
order by past_year
```

OUTPUT:

Row	past_year	order_number
1	2016	329
2	2017	45101
3	2018	54011

INSIGHT:

If we confirm a growing trend, businesses can invest in scaling operations.

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

QUERY:

```
WITH yearly_orders AS (
    SELECT
        FORMAT_TIMESTAMP('%Y', order_purchase_timestamp) AS order_year,
        FORMAT_TIMESTAMP('%B', order_purchase_timestamp) AS order_month,
        EXTRACT(MONTH FROM order_purchase_timestamp) AS month_number
        COUNT(DISTINCT order_id) AS total_orders
    FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS`
    GROUP BY order_year, order_month, month_number
```

```
    ORDER BY order_year, month_number),
trend_analysis AS (
    SELECT
        order_year,
        order_month,
        total_orders,
        month_number,
        LAG(total_orders) OVER (ORDER BY order_year, month_number)
        AS previous_orders,
        total_orders - LAG(total_orders) OVER (ORDER BY order_year, month_number)
        AS growth
    FROM yearly_orders)
SELECT
    order_year,
    order_month,
    total_orders,
    previous_orders,
    CASE
        WHEN growth > 0 THEN 'Increasing'
        WHEN growth < 0 THEN 'Decreasing'
        ELSE 'No Change'
    END AS trend
FROM trend_analysis
ORDER BY order_year, month_number
```

OUTPUT:

Row	order_year	order_month	total_orders	previous_orders	trend
1	2016	September	4	*null*	No Change
2	2016	October	324	4	Increasing
3	2016	December	1	324	Decreasing
4	2017	January	800	1	Increasing
5	2017	February	1780	800	Increasing
6	2017	March	2682	1780	Increasing
7	2017	April	2404	2682	Decreasing
8	2017	May	3700	2404	Increasing
9	2017	June	3245	3700	Decreasing
10	2017	July	4026	3245	Increasing

INSIGHT:

- If there's strong seasonality, marketing campaigns should align with peak shopping periods.

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**

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'
         ELSE 'Night'
    END AS Brazilian_Time,
    COUNT( distinct order_id) AS Order_Count
FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS`
GROUP BY Brazilian_Time
ORDER BY Order_Count desc
```

OUTPUT:

Row	Brazilian_Time	Order_Count
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

INSIGHT:

- Understanding time-of-day preferences helps in timing promotions and customer engagement effectively

---

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

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

QUERY:

```
WITH OrderCounts AS (
    SELECT
        EXTRACT(MONTH FROM O.order_purchase_timestamp) AS Order_Month,
        O.customer_id, COUNT( DISTINCT O.order_id) AS Order_Count
    FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` O
    GROUP BY Order_Month, O.customer_id)
SELECT
    OC.Order_Month,C.customer_state, SUM(OC.Order_Count) AS Order_Count_State
FROM OrderCounts OC
JOIN `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` C
    ON OC.customer_id = C.customer_id
GROUP BY OC.Order_Month,C.customer_state
ORDER BY Order_Count_State asc
```

OUTPUT:

Row	Order_Month	customer_state	Order_Count_State
1	1	RR	2
2	1	AC	8
3	1	AP	11
4	1	AM	12
5	1	TO	19
6	1	RO	23
7	1	SE	24
8	1	PB	33
9	1	AL	39
10	1	RN	51
11	1	PI	55
12	1	MA	66
13	1	MS	71

INSIGHT:

- Some states may show consistent growth, indicating increasing e-commerce penetration.
- If orders drop in certain months, it could indicate seasonal effects or economic factors.

2. How are the customers distributed across all the states

QUERY:

```
SELECT
    customer_state,
    COUNT(customer_id) AS customer_count
FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER`
GROUP BY customer_state
ORDER BY customer_count DESC
limit 10
```

OUTPUT:

Row	customer_state	customer_count
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:

- Major cities (São Paulo, Rio de Janeiro) likely have the highest number of customers.
- Smaller states may show emerging trends, indicating new market potential.
- Some states may have high order volume but fewer customers, meaning customers place multiple order.

IV. 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.

QUERY:

```
with tbl as
(SELECT
 FORMAT_TIMESTAMP('%B', TIMESTAMP(o.order_purchase_timestamp))
    AS order_month,
 FORMAT_TIMESTAMP('%Y', TIMESTAMP(o.order_purchase_timestamp))
    AS order_year,
 COUNT( distinct o.order_id) AS total_orders,
     round(SUM(p.payment_value),2) AS total_cost
 FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
 JOIN `shailja-project-444415.TARGETBUSINESSSTUDY20250125.PAYMENTS` p
      on o.order_id = p.order_id
 GROUP order_year, order_month)
select tbl.order_year, tbl.order_month, tbl.total_cost, tbl.total_orders,
round(100*((tbl.total_cost - lag(tbl.total_cost) over(order by
tbl.order_year, tbl.order_month))/lag(tbl.total_cost)
over(order by tbl.order_year, tbl.order_month)),2) AS percentage_increase
from tbl
where tbl.order_year IN ('2017', '2018')
    AND tbl.order_month IN ('January', 'February', 'March', 'April', 'May',
'June', 'July', 'August')
ORDER BY
     tbl.order_year, tbl.order_month, percentage_increase
```

OUTPUT:

Row	order_year	order_month	total_cost	total_orders	percentage_increase
1	2017	April	417788.03	2404	null
2	2017	August	674396.32	4331	61.42
3	2017	February	291908.01	1780	-56.72
4	2017	January	138488.04	800	-52.56
5	2017	July	592382.92	4026	327.75
6	2017	June	511276.38	3245	-13.69
7	2017	March	449863.6	2682	-12.01
8	2017	May	592918.82	3700	31.8
9	2018	April	1160785.48	6939	95.77
10	2018	August	1022425.32	6512	-11.92
11	2018	February	992463.34	6728	-2.93
12	2018	January	1115004.18	7269	12.35
13	2018	July	1066540.75	6292	-4.35

INSIGHT:

- If the percentage increase is positive, it indicates higher spending on orders in 2018 compared to 2017.
- A high percentage increase suggests growth in e-commerce activity, possibly due to more customers, higher order values, or inflation

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

QUERY:

```
with tbl as
            (select c.customer_state, o.order_id
from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` c
join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
         on c.customer_id = o.customer_id)
select distinct tbl.customer_state,round(avg(oi.price) over(order by
tbl.customer_state),2) as avg_cost_per_state,
round(sum(oi.price) over(order by tbl.customer_state),2) as
total_cost_per_state
from tbl
join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERITEMS` oi
```

```
on tbl.order_id = oi.order_id
order by tbl.customer_state
```

OUTPUT:

Row	customer_state	avg_cost_per_state	total_cost_per_state
1	AC	173.73	15982.95
2	AL	179.66	96297.76
3	AM	169.26	118654.6
4	AP	168.75	132128.9
5	BA	140.44	643478.89
6	CE	143.69	870733.6
7	DF	138.59	1173337.54
8	ES	135.08	1448374.85
9	GO	133.51	1742966.8
10	MA	134.2	1862615.02

INSIGHT:

- If some less-populated states have high total order values, it suggests a high adoption rate of online shopping.
- These states represent **potential growth markets** for businesses to target with better logistics, promotions, and localized strategies.
- If certain states have a **high average but low total order value**, it means that **fewer customers are making large purchases**

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

QUERY:

```
with tbl as
          (
            select c.customer_state, o.order_id
            from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` c
            join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
            on c.customer_id = o.customer_id)
select distinct tbl.customer_state, freight_value,
round(avg(freight_value) over(order by tbl.customer_state), 2) as frieght_average,
round(sum(freight_value) over(order by tbl.customer_state), 2) as frieght_total
from tbl
join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERITEMS` oi
on tbl.order_id = oi.order_id
order by tbl.customer_state
```

OUTPUT :

Row	customer_state	avg_cost_per_state	total_cost_per_state
1	AC	173.73	15982.95
2	AL	179.66	96297.76
3	AM	169.26	118654.6
4	AP	168.75	132128.9
5	BA	140.44	643478.89
6	CE	143.69	870733.6
7	DF	138.59	1173337.54
8	ES	135.08	1448374.85
9	GO	133.51	1742966.8
10	MA	134.2	1862615.02

INSIGHT:

- States far from key logistics hubs (Amazonas, Acre, Roraima, Amapá) often have high total and average freight costs due to longer distances and higher delivery challenges.
- Customers in these regions might abandon carts due to high delivery fees.
- If some states have low total freight costs despite many orders, it indicates efficient shipping logistics or subsidized delivery costs.

V. 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.**

QUERY:

```
select distinct order_id, 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 `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS`
where order_delivered_customer_date is not null
group by order_id, time_to_deliver, diff_estimated_delivery
order by order_id
```

OUTPUT

Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	-8
2	00018f77f2f0320c557190d7a1...	16	-2
3	000229ec398224ef6ca0657da...	7	-13
4	00024acbcdf0a6daa1e931b03...	6	-5
5	00042b26cf59d7ce69dfabb4e...	25	-15
6	00048cc3ae777c65dbb7d2a06...	6	-14
7	00054e8431b9d7675808bcb8...	8	-16
8	000576fe39319847cbb9d288c...	5	-15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	-18

INSIGHT:

- **If average delivery times increase over time, it may point to logistics delays, seasonal congestion, or supplier inefficiencies.**
- **Negative Difference (Early Deliveries): If `diff_estimated_delivery` is negative, it means orders were delivered before the estimated date, which is great for customer satisfaction.**
- **Zero Difference (On-Time Deliveries): If the difference is zero, the logistics system is accurate and efficient.**
- **Positive Difference (Late Deliveries): If there's a high positive difference, it suggests delays due to issues like weather, supply chain disruptions, or delivery inefficiencies**
- **Frequent late deliveries can reduce customer trust and lead to cart abandonment or negative reviews.**
- **Early/on-time deliveries increase customer satisfaction, leading to higher retention & repeat purchases.**
- **Companies might need better tracking, warehouse distribution, or delivery partnerships to improve delivery speed.**

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

QUERY:

```
WITH state_avg_freight AS (
    SELECT c.customer_state,AVG(oi.freight_value) AS avg_freight_value
    FROM
      `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` c
    JOIN
       `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
    ON
        c.customer_id = o.customer_id
    JOIN
        `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERITEMS` oi
    ON
        o.order_id = oi.order_id
    GROUP BY
        c.customer_state),
ranked_states AS (
    SELECT customer_state,avg_freight_value,
        ROW_NUMBER() OVER (ORDER BY avg_freight_value DESC) AS rank_highest,
        ROW_NUMBER() OVER (ORDER BY avg_freight_value ASC) AS rank_lowest
    FROM
        state_avg_freight)
```

```sql
SELECT
    highest_states.customer_state AS highest_state,
    lowest_states.customer_state AS lowest_state
FROM
    (SELECT customer_state, rank_highest FROM ranked_states WHERE rank_highest
<= 5) AS highest_states
LEFT JOIN
    (SELECT customer_state, rank_lowest FROM ranked_states WHERE rank_lowest
<= 5) AS lowest_states
ON
    highest_states.rank_highest = lowest_states.rank_lowest
WHERE
    highest_states.customer_state IS NOT NULL OR lowest_states.customer_state
IS NOT NULL
```

OUTPUT:

Row	customer_state	avg_cost_per_state	total_cost_per_state
1	AC	173.73	15982.95
2	AL	179.66	96297.76
3	AM	169.26	118654.6
4	AP	168.75	132128.9
5	BA	140.44	643478.89
6	CE	143.69	870733.6
7	DF	138.59	1173337.54
8	ES	135.08	1448374.85
9	GO	133.51	1742966.8
10	MA	134.2	1862615.02

INSIGHT:

- Customers in these states may abandon carts due to high shipping costs.
- Businesses may need to offer free shipping promotions to attract customers
- Businesses can offer fast & low-cost shipping, improving customer satisfaction & retention.
- These states are ideal for e-commerce warehouses & distribution centers.

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

QUERY:

```
WITH state_avg_delivery_time AS (
    SELECT c.customer_state,
    AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY) AS avg_delivery_time)
    FROM `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` c
    JOIN `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
    ON c.customer_id = o.customer_id
    WHERE o.order_delivered_customer_date IS NOT NULL
    GROUP BY c.customer_state)
,ranked_states AS (
    SELECT
        customer_state,
        avg_delivery_time,
        ROW_NUMBER() OVER (ORDER BY avg_delivery_time ASC) AS rank_lowest,
        ROW_NUMBER() OVER (ORDER BY avg_delivery_time DESC) AS rank_highest
    FROM state_avg_delivery_time)
SELECT
    highest_states.customer_state AS highest_state,
    lowest_states.customer_state AS lowest_state
FROM
    (SELECT customer_state, rank_highest FROM ranked_states WHERE rank_highest
<= 5) AS highest_states
FULL OUTER JOIN
    (SELECT customer_state, rank_lowest FROM ranked_states WHERE rank_lowest
<= 5) AS lowest_states
ON highest_states.rank_highest = lowest_states.rank_lowest
WHERE highest_states.customer_state IS NOT NULL OR
lowest_states.customer_state IS NOT NULL
```

OUTPUT:

Row	highest_state	lowest_state
1	AM	MG
2	AP	PR
3	PA	SC
4	AL	DF
5	RR	SP

INSIGHT:

- Long delivery times can cause customer dissatisfaction and order cancellations.
- Businesses might need regional warehouses or faster shipping options
- Faster deliveries increase customer satisfaction and retention.
- Businesses can promote same-day or next-day delivery options to gain a competitive edge.

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 tbl as
    (select
    round(avg(date_diff(o.order_delivered_customer_date,
    o.order_estimated_delivery_date, DAY)),2) as avg_diff_estimated_delivery,
    c.customer_state
    from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.CUSTOMER` c
    join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` o
    on c.customer_id = o.customer_id
    group by c.customer_state
```

```
having avg_diff_estimated_delivery < 0)
select tbl.customer_state,tbl.avg_diff_estimated_delivery
From tbl
order by tbl.avg_diff_estimated_delivery
limit 5
```

OUTPUT:

Row	customer_state	avg_diff_estimated_delivery
1	AC	-19.76
2	RO	-19.13
3	AP	-18.73
4	AM	-18.61
5	RR	-16.41

INSIGHT:

📌 For Fast-Delivery States:

- Promote "Fast Delivery" as a selling point (e.g., same-day or next-day delivery offers).
- Maintain and scale warehouse efficiency to sustain fast shipping speeds.

📌 For Slower States:

- Analyze logistics bottlenecks and optimize fulfillment strategies.
- Improve inventory placement by setting up regional warehouses in slow-delivery states

---

VI. Analysis based on the payments:

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

QUERY:

```
SELECT
FORMAT_TIMESTAMP('%Y', TIMESTAMP(O.order_purchase_timestamp)) AS order_year,
FORMAT_TIMESTAMP('%B', TIMESTAMP(O.order_purchase_timestamp)) AS order_month,
payment_type,count(distinct O.order_id) as order_count
from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` O
join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.PAYMENTS` P
on O.order_id = P.order_id
group by order_year,order_month,payment_type
order by PARSE_TIMESTAMP('%Y',order_year),PARSE_TIMESTAMP('%B',order_month)
```

OUTPUT:

Row	order_year	order_month	payment_type	order_count
1	2016	September	credit_card	3
2	2016	October	credit_card	253
3	2016	October	UPI	63
4	2016	October	voucher	11
5	2016	October	debit_card	2
6	2016	December	credit_card	1
7	2017	January	credit_card	582
8	2017	January	UPI	197
9	2017	January	voucher	33
10	2017	January	debit_card	9

INSIGHT:

- Credit cards are likely the dominant payment method, followed by boletos (bank slips), debit cards, and vouchers.
- If credit card usage increases over time, it suggests growing consumer confidence in online payments.
- If boleto (bank slip) usage is high in certain months, it may indicate that some customers prefer non-instant payment methods due to financial planning or lack of credit access.
- **Spike in credit card & installment payments** due to big purchases.
- Possible increase in orders, especially via credit cards and installment payments.

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

QUERY:

```
select count(DISTINCT O.order_id) as order_count, payment_installments
from `shailja-project-444415.TARGETBUSINESSSTUDY20250125.ORDERS` O
join `shailja-project-444415.TARGETBUSINESSSTUDY20250125.PAYMENTS` P
on O.order_id = P.order_id
group by payment_installments
order by payment_installments
```

OUTPUT:

Row	order_count	payment_installment
1	2	0
2	49060	1
3	12389	2
4	10443	3
5	7088	4
6	5234	5
7	3916	6
8	1623	7
9	4253	8
10	644	9
11	5315	10
12	23	11
13	133	12

INSIGHT:

- Encourage longer installment options **before big sales events** to maximize revenue.
- Offer **discounts for upfront payments** in slower months to boost immediate cash flow.
- Offer **interest-free installments** on high-ticket items to **increase conversion rates**.
- Provide **BNPL (Buy Now, Pay Later) options** to attract customers who prefer small, flexible payments.
- If **most customers prefer full payment**, offering **small discounts for one-time payments** can boost immediate cash flow.
- If **installments are popular**, businesses should provide **more flexible installment plans (interest-free options)** to attract customers

---