



## A Case Study of Target in Brazil

The objective of this case study is to analyse the dataset from Target, a leading retailer in Brazil, and gain valuable insights into e-commerce trends and customer behaviour. The study aims to explore the growth of e-commerce in Brazil, identify seasonality patterns, understand the preferred purchasing times of Brazilian customers, analyse the evolution of e-commerce orders in different states, assess the impact of e-commerce on the economy, etc.

The ultimate goal is to provide Target with data-driven insights and recommendations that can enhance its operations, improve customer experiences, and drive business success in the Brazilian market.

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

- **Data type of columns in a table**

Extracting distinct Column names in all the tables and their Data types. It can be done for each table as well with filter for 'table\_name' using 'Where'.

**Query:**

```
select distinct column_name, data_type
from `target`.INFORMATION_SCHEMA.COLUMNS
```

**Result:**

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	column_name	data_type				
1	order_id	STRING				
2	order_item_id	INT64				
3	product_id	STRING				
4	seller_id	STRING				
5	shipping_limit_date	TIMESTAMP				
6	price	FLOAT64				
7	freight_value	FLOAT64				
8	seller_zip_code_prefix	INT64				
9	seller_city	STRING				
10	seller_state	STRING				
11	geolocation_zip_code_prefix	INT64				

Results per page: 50 1 ~ 43 of 43

- **Time period for which the data is given?**

**Query:**

```
select min(order_purchase_timestamp) as Start_date,
max(order_purchase_timestamp) as end_date,
from `target`.orders
```

**Result:**

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

- **Cities and States of customers ordered during the given period?**

#### Query:

```
select customer_id, customer_city, customer_state
from `target.customers`
order by customer_id
```

#### Result:

Query results SAVE RESULTS

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_id	customer_city	customer_state			
1	00012a2ce6f8dcd20d059ce9...	osasco	SP			
2	000161a058600d5901f007fab...	itapeperica	MG			
3	0001fd6190edaaf884bc3d49...	nova venecia	ES			
4	0002414f953443074040ace7...	mendonca	MG			
5	000379cdec625522490c315e7...	sao paulo	SP			
6	0004164d20a9e969af783496f...	valinhos	SP			
7	000419c5494106c306a97b563...	niteroi	RJ			
8	00046a560d407e99b969756e...	rio de janeiro	RJ			
9	00050bf6e01e69d5c0fd612f1b...	ijui	RS			
10	000598caf2ef4117407665ac3...	oliveira	MG			

Results per page: 50 1 - 50 of 9944

## 2. In-depth Exploration:

- **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?**

- **Count of orders**

- **Total orders in the given time period: 99441**

#### Query:

```
select customer_id, order_id,
count(*) as order_count
from `target.orders`
group by customer_id, order_id
```

#### Result:

Row	customer_id	order_id	order_count
1	725e9c75605414b21fd8c8d5a...	7a4df5d8c4ff4090e541401a20a...	1
2	4ee64f4bfc542546f422da0aeb...	35de4050331c6c644cddc86f4...	1
3	438449d4af8980d107bf04571...	b5359909123fa03c50bdb0cfe...	1
4	964a6df3d9bdf60fe3e7b8bb69...	dba5062fbd3af4fb6c33b1e04...	1
5	7d61b9f4f216052ba664f22e9c...	90ab3e7d52544ec7bc3363c82...	1

Results per page: 50 1 - 50 of 99441

- **Total orders delivered: 96478**

#### Query:

```
select
count(*) as order_count
from `target.orders`
group by order_status
having order_status = "delivered"
```

JOB INFORMATION		RESULTS
Row	order_count	
1	96478	

## ➤ Monthly count of Orders and Amount Spent

### Query:

```
select
EXTRACT(month FROM o.order_purchase_timestamp) as Order_Month,
EXTRACT(Year FROM o.order_purchase_timestamp) as Order_Year,
count(*) as Order_count,
round(sum(i.price),2) as amount_spent
from `target.orders` as o
join `target.order_items` as i
on o.order_id=i.order_id
group by Order_Month, Order_Year
order by Order_Month, Order_Year
```

### Result :

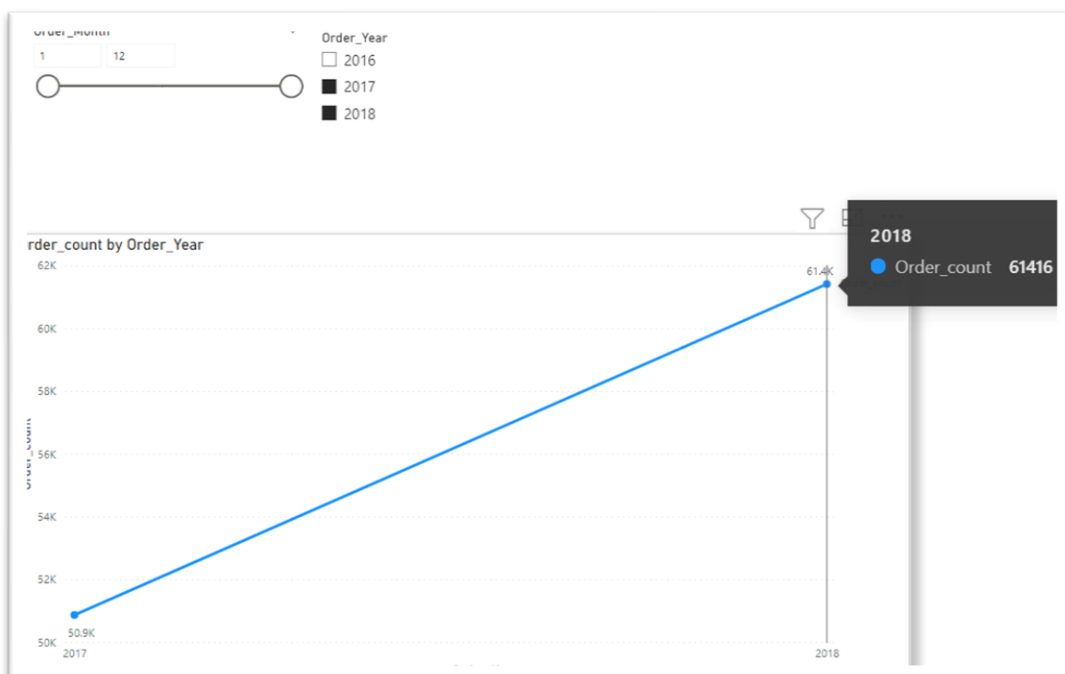
Query results [SAVE RESULTS](#) [VIEW](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS		EXECUTION GRAPH	PREVIEW
Row	Order_Month	Order_Year	Order_count	amount_spent			
1	1	2017	955	120312.87			
2	1	2018	8208	950030.36			
3	2	2017	1951	247303.02			
4	2	2018	7672	844178.71			
5	3	2017	3000	374344.3			
6	3	2018	8217	983213.44			
7	4	2017	2684	359927.23			
8	4	2018	7975	996647.75			
9	5	2017	4136	506071.14			
10	5	2018	7925	996517.68			
11	6	2017	3583	433038.6			
12	6	2018	7078	865124.31			
13	7	2017	4519	498031.48			
14	7	2018	7092	895507.22			
15	8	2017	4910	573971.68			
16	8	2018	7248	854686.33			

Results per page: 50 1 - 24 of 24

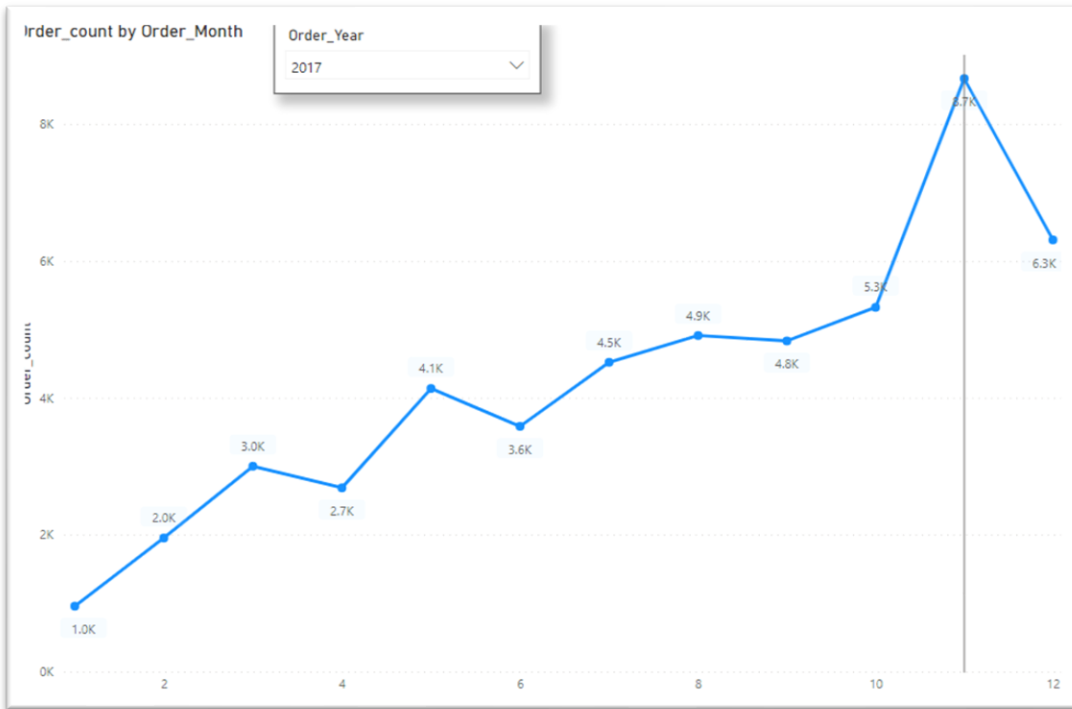
### Visualization and Insights:

The below graph shows a steep rise in orders from 2017 to 2018 which describes a growing trend of e-commerce in Brazil.



➤ **Seasonality with peaks at specific months:**

Including data for 2017 as it contains data for all the months. We can see from the below graph that there is a continuous growth in orders from January to November but the **order-count peaks in November** which might be because of various festivities.



➤ **Recommended Actions:**

- Since there is a growth in ecommerce, Target should focus more on fulfilling online orders while maintaining their retail stores. Target should be continuously building strategies for acquiring new online customers as the trend is growing.
- Since the growth in orders and amount spent peaks in November, Target should increase their workforce and supply of goods in the later months of the year. They can also have festive sales around the same period to attract more customers.

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

- **Query:** Getting count of orders for each Hour Bucket using Subquery.

```
SELECT Hour_Bucket,
count(*) as Order_Count
from
(select
EXTRACT(HOUR FROM order_purchase_timestamp) as Order_Hour,
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'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN 'Night'
END AS Hour_Bucket,
from `target.orders`)
group by Hour_Bucket
```

▪ **Result:**

Query results			
JOB INFORMATION		RESULTS	JSON
Row	Hour_Bucket	Order_Count	EXE
1	Morning	27733	
2	Dawn	5242	
3	Afternoon	38135	
4	Night	28331	

- **Insight:** Brazilian customers tend to do shopping in the **Afternoon period**.

▪ **Recommendations:**

- ♦ **Adjust Advertising:** Target can focus their advertising and promotional activities during the afternoon period to specifically target Brazilian customers.
- ♦ **Optimize Product Availability:** Since Brazilian customers are more active in the afternoon, it would be beneficial for Target to ensure that popular and in-demand products are readily available during this time.
- ♦ **Enhance Customer Service:** Target can allocate additional resources and staff during the afternoon period to provide excellent customer service to Brazilian customers.

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

- **Get month on month orders by states.**

• **Query:**

```
select
c.customer_state,
EXTRACT(month FROM o.order_purchase_timestamp) as Order_Month,
count(*) as Order_count
from `target.orders` as o
join `target.customers` as c
on o.customer_id=c.customer_id
group by customer_state,Order_Month
order by Order_Month, Order_count desc
```

# We can also add Order\_Year in group by to show the data in a better way:

```
select
c.customer_state,
EXTRACT(month FROM o.order_purchase_timestamp) as Order_Month,
EXTRACT(Year FROM o.order_purchase_timestamp) as Order_Year,
count(*) as Order_count
from `target.orders` as o
join `target.customers` as c
on o.customer_id=c.customer_id
group by customer_state,Order_Month, Order_Year
order by Order_Month,Order_Year
```

- Result for Queries above:

Row	customer_state	Order_Month	Order_count
1	SP	1	3351
2	RJ	1	990
3	MG	1	971
4	PR	1	443
5	RS	1	427
6	SC	1	345
7	BA	1	264
8	GO	1	164
9	ES	1	159
10	DF	1	151
11	PF	1	113

Row	customer_state	Order_Month	Order_Year	Order_count
1	PR	1	2017	65
2	MG	1	2017	108
3	SP	1	2017	299
4	RJ	1	2017	97
5	PA	1	2017	12
6	SC	1	2017	31
7	MA	1	2017	9
8	BA	1	2017	25
9	MT	1	2017	11
10	ES	1	2017	12
11	CE	1	2017	9

- Insights:** Sao Paulo(SP) seems to have the highest orders for each month followed by RJ and MG.

- Recommendations:** Target can allocate additional resources and staff during to the states who needs it most to provide excellent customer service and should increase their capacity there.

- Distribution of customers across the states in Brazil.

- Query:

```
SELECT customer_state,
count(customer_unique_id) as Unique_Customers
from `target.customers`
group by customer_state
order by Unique_Customers desc
```

- Results:

Query results

JOB INFORMATION		RESULTS	JSON	EXE
Row	customer_state	Unique_Customers		
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		
11	PE	1652		
12	CE	1336		

- Insights:** Sao Paulo has the largest customer base as compared to other states.

- **Recommendations:**

- ♦ **Market Expansion:** Given the significant customer base in São Paulo, Target should consider expanding its presence in this state. This can include opening more physical stores or establishing strategic partnerships to reach a wider customer audience.
- ♦ **Localized Marketing:** Target can develop targeted marketing campaigns specifically tailored to the São Paulo market.
- ♦ **Local Partnerships:** Collaborating with local sellers, manufacturers, or suppliers in São Paulo can allow Target to offer regionally relevant products or exclusive partnerships.

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

- **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.**

- **Query:**

```
select
round((SUM(case when year = 2018 then payment_value END) - SUM(case when year =
2017 then payment_value END)) / SUM(case when year = 2017 THEN payment_value END)
* 100,2) AS percentage_increase_in_Cost
from
(select
EXTRACT(YEAR from o.order_purchase_timestamp) AS year,
p.payment_value
from
`target.orders` as o
join `target.payments` as p ON o.order_id = p.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)
```

- **Result: 136.98%**

Row	percentage_increase_in_Cost
1	136.98

- **Insight:** The amount spent by customers in 2018 has more than doubled from 2017, it indicates significant growth and increased purchasing power among customers.

- **Recommendations:**

- ❖ **Capitalize on the Growth:** Target should seize the opportunity presented by the substantial increase in customer spending in 2018.
- ❖ **Enhance Product Offerings:** With customers spending more in 2018, Target should assess the demand and preferences of customers during this period. By identifying popular product categories and items, Target can expand its product offerings and ensure they align with customer preferences.
- ❖ **Strengthen E-commerce Capabilities:** As customers are spending more, Target should prioritize its e-commerce capabilities. This includes investing in a user-friendly website, optimizing mobile experiences, and streamlining the online purchase process. Offering flexible delivery options, fast shipping, and hassle-free returns.

- **Mean & Sum of price and freight value by customer state**

## ▪ Query:

```
select
customer_state,
round(sum(it.price),2) as Price_total_Sum,
round(avg(it.price),2) as Price_mean_value,
round(sum(it.freight_value),2) as Freight_total_Sum,
round(avg(it.freight_value),2) as Freight_mean_value
from `target.orders` as o
join `target.customers` as c
on o.customer_id=c.customer_id
join `target.order_items` as it
on o.order_id=it.order_id
group by customer_state
order by Price_total_Sum desc, Freight_total_Sum desc
```

## ▪ Result:

Row	customer_state	Price_total_Sum	Price_mean_value	Freight_total_Sum	Freight_mean_value
1	SP	5202955.05	109.65	718723.07	15.15
2	RJ	1824092.67	125.12	305589.31	20.96
3	MG	1585308.03	120.75	270853.46	20.63
4	RS	750304.02	120.34	135522.74	21.74
5	PR	683083.76	119.0	117851.68	20.53
6	SC	520553.34	124.65	89660.26	21.47
7	BA	511349.99	134.6	100156.68	26.36
8	DF	302603.94	125.77	50625.5	21.04
9	GO	294591.95	126.27	53114.98	22.77
10	ES	275037.31	121.91	49764.6	22.06
11	PE	262788.03	145.51	59449.66	32.92

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

## • Calculate days between purchasing, delivering and estimated delivery

## ▪ Query:

```
SELECT
order_id,
order_purchase_timestamp,
order_delivered_customer_date,
order_estimated_delivery_date,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
days_to_delivery,
DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY) AS
delivery_delay
FROM
`target.orders`
order by days_to_delivery desc, delivery_delay desc
LIMIT 10
```

## ▪ Result:

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	days_to_delivery	delivery_delay
1	ca07593549f181...	2017-02-21 23:31:27 UTC	2017-09-19 14:36:39 UTC	2017-03-22 00:00:00 UTC	209	181
2	1b3190b2dfa9d7...	2018-02-23 14:57:35 UTC	2018-09-19 23:24:07 UTC	2018-03-15 00:00:00 UTC	208	188
3	440d0d17af5528...	2017-03-07 23:59:51 UTC	2017-09-19 15:12:50 UTC	2017-04-07 00:00:00 UTC	195	165
4	285ab9426d6982...	2017-03-08 22:47:40 UTC	2017-09-19 14:00:04 UTC	2017-04-06 00:00:00 UTC	194	166
5	0f4519c5f1c541d...	2017-03-09 13:26:57 UTC	2017-09-19 14:38:21 UTC	2017-04-11 00:00:00 UTC	194	161
6	2fb597c2f772eca...	2017-03-08 18:09:02 UTC	2017-09-19 14:33:17 UTC	2017-04-17 00:00:00 UTC	194	155
7	47b40429ed8cce...	2018-01-03 09:44:01 UTC	2018-07-13 20:51:31 UTC	2018-01-19 00:00:00 UTC	191	175
8	2fe324feb907e3...	2017-03-13 20:17:10 UTC	2017-09-19 17:00:07 UTC	2017-04-05 00:00:00 UTC	189	167
9	2d7561026d542c...	2017-03-15 11:24:27 UTC	2017-09-19 14:38:18 UTC	2017-04-13 00:00:00 UTC	188	159
10	c27815f7e3dd0b...	2017-03-15 23:23:17 UTC	2017-09-19 17:14:25 UTC	2017-04-10 00:00:00 UTC	187	162



- Find `time_to_delivery` & `diff_estimated_delivery`.

Query:

```
SELECT
order_id,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY) AS
time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM
`target.orders`
```

Result:

Row	order_id	time_to_delivery	diff_estimated_delivery
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

- Insight:** If some `diff_estimated_delivery` values are coming out as **negative**, it means that the `order_delivered_customer_date` is later than the `order_estimated_delivery_date`. **This can happen if the order was delivered earlier than expected.**

- Group data by state, take mean of `freight_value`, `time_to_delivery`, `diff_estimated_delivery`
- Query:

```
SELECT
c.customer_state,
ROUND(AVG(oi.freight_value),2) AS mean_freight_value,
ROUND(AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp,
HOUR)),2) AS mean_time_to_delivery,
ROUND(AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,
HOUR)),2) AS mean_diff_estimated_delivery
FROM
`target.customers` AS c
JOIN
`target.orders` AS o
ON
c.customer_id = o.customer_id
JOIN
`target.order_items` AS oi
ON
o.order_id = oi.order_id
GROUP BY
c.customer_state
order by mean_time_to_delivery
LIMIT 10
```

- **Result:**

Row	customer_state	mean_freight_value	mean_time_to_delive	mean_diff_estimated
1	SP	15.15	208.87	251.89
2	PR	20.53	286.24	306.57
3	MG	20.63	287.11	302.91
4	DF	21.04	310.52	275.42
5	SC	21.47	359.53	260.55
6	RJ	20.96	363.06	271.04
7	RS	21.74	364.03	321.95
8	GO	22.77	369.18	277.89
9	MS	23.37	372.34	252.29
10	ES	22.06	375.08	238.41

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

```
SELECT customer_state, ROUND(AVG(freight_value)) as avg_freight_value
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
JOIN `target.order_items` oi
ON o.order_id = oi.order_id
GROUP BY customer_state
ORDER BY avg_freight_value DESC
LIMIT 5
```

- **Result:**

Row	customer_state	avg_freight_value
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0

- **Query: lowest**

```
SELECT customer_state, round(AVG(freight_value),2) as avg_freight_value
FROM `target.customers` c
JOIN `target.orders` o
ON c.customer_id = o.customer_id
JOIN `target.order_items` oi
ON o.order_id = oi.order_id
GROUP BY customer_state
ORDER BY avg_freight_value ASC
LIMIT 5
```

- **Result:**

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

- **Top 5 states with highest/lowest average time to delivery**

- **Query: highest**

```
SELECT customer_state, ROUND(AVG(time_to_delivery),2) AS avg_time_to_delivery
FROM (
SELECT
c.customer_state,
TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)
AS
time_to_delivery
FROM `target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
) AS subquery
GROUP BY customer_state
ORDER BY avg_time_to_delivery DESC
LIMIT 5
```

- **Result:**

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

- **Query: lowest**

```
SELECT customer_state, ROUND(AVG(time_to_delivery),2) AS avg_time_to_delivery
FROM (
SELECT
c.customer_state,
TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)
AS
time_to_delivery
FROM `target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
) AS subquery
GROUP BY customer_state
ORDER BY avg_time_to_delivery ASC
LIMIT 5
```

- **Result:**

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

- **Top 5 states where delivery is really fast/ not so fast compared to estimated date**

- **Query:**

```
WITH delivery_duration AS (  
  SELECT  
    c.customer_state,  
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,  
    DAY) AS  
    time_to_delivery,  
    TIMESTAMP_DIFF(o.order_estimated_delivery_date,  
    o.order_delivered_customer_date, DAY) AS  
    diff_estimated_delivery  
  FROM  
    `target.customers` AS c  
  JOIN  
    `target.orders` AS o  
  ON  
    c.customer_id = o.customer_id  
)  
SELECT  
  customer_state  
FROM  
  delivery_duration  
WHERE  
  time_to_delivery <= diff_estimated_delivery  
GROUP BY  
  customer_state  
ORDER BY  
  COUNT(*) DESC  
LIMIT 5
```

- **Result:**

Row	customer_state ▼
1	SP
2	MG
3	RJ
4	PR
5	RS

- **Query:**

```
WITH delivery_duration AS (  
  SELECT  
    c.customer_state,  
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,  
    DAY) AS  
    time_to_delivery,  
    TIMESTAMP_DIFF(o.order_estimated_delivery_date,  
    o.order_delivered_customer_date, DAY) AS  
    diff_estimated_delivery  
  FROM  
    `target.customers` AS c  
  JOIN  
    `target.orders` AS o
```

```
ON
c.customer_id = o.customer_id
)
SELECT
customer_state
FROM
delivery_duration
WHERE
time_to_delivery <= diff_estimated_delivery
GROUP BY
customer_state
ORDER BY COUNT(*) DESC LIMIT 5
```

▪ **Result:**

Row	customer_state
1	SP
2	MG
3	RJ
4	PR
5	RS

- **Payment type analysis:**
- **Month over Month count of orders for different payment types.**

▪ **Query:**

```
SELECT
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
p.payment_type,
COUNT(*) AS order_count
FROM
`target.orders` AS o
JOIN
`target.payments` AS p
ON
o.order_id = p.order_id
GROUP BY
month,
payment_type
ORDER BY
month ASC
LIMIT 10
```

▪ **Result:**

Row	month	payment_type	order_count
1	1	debit_card	118
2	1	voucher	477
3	1	UPI	1715
4	1	credit_card	6103
5	2	voucher	424
6	2	debit_card	82
7	2	credit_card	6609
8	2	UPI	1723
9	3	UPI	1942
10	3	voucher	591

- **Count of orders based on the no. of payment installments.**

- **Query:**

```
SELECT
payment_installments,
COUNT(*) AS order_count
FROM
`target.payments`
GROUP BY
payment_installments
ORDER BY
payment_installments ASC
LIMIT 10
```

- **Result:**

Row	payment_installment	order_count
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

- In addition to insights and recommendations mentioned above:

### ❖ Overall Actionable Insights:

**Sales Forecasting and Planning:** Use monthly order data to forecast sales and effectively manage inventory. Identify peak and lowest activity months for resource allocation and inventory optimization.

**Targeted Marketing Campaigns:** Segment customers by location, purchase history, and preferences. Build personalized services and support to increase efficiency and engagement.

**Increase Customer Satisfaction:** Review ratings and feedback to identify areas for improvement. Prioritize customer issues, improve product quality, and improve the overall purchasing experience.

**Geographic Expansion Strategies:** Use geolocation data to identify customers in multiple locations and low-cost markets. Prepare growth plans and distribute marketing activities accordingly.

**Payment Improvement:** Check the payment information to understand the preferred payment method. Improve the checkout process to improve checkouts and reduce shipping costs.

**Product Performance Analysis:** Evaluate product performance and identify best-selling products. Use this information for inventory management and advertising strategies.

## ❖ Overall Recommendations:

**Sales Analysis and Planning:** Use annual sales reports to create accurate sales forecasts and develop sales strategies.

**Targeted Marketing Strategy:** Focus on marketing activities during peak months to increase customer reach. Allocate resources and use multiple marketing strategies.

**Customer Engagement and Protection:** Increase customer engagement and loyalty through service retention, personalized service and exceptional customer service.

**Inventory Management:** Optimizing inventory levels based on sales data and trends. Increase the efficiency of delivery and logistics management.

**Geographical Expansion:** Consider expanding your business into an area with a large customer base. Conduct market research and adapt marketing strategies to meet local needs.

**Payment and payment optimization:** Continue to identify payment options and streamline payment processes to ensure a seamless payment experience.

**Sales Collaboration and Operations:** Build relationships with suppliers by providing support and overseeing sales activities.

**Competitive Analysis:** Stay up to date on business trends, pricing strategies and products. Differentiate the company's products from its competitors.

**Continuous Improvement:** Analyze sales data, customer feedback, and business models to improve strategies, improve processes, and drive innovation.

**By implementing these recommendations, Target can increase sales, increase customer satisfaction, increase quality, and maintain profitability in e-commerce.**

