```sql
#Exploratory Analysis:-

#Data type of all columns in the "customers" table:

SELECT column_name, data_type
FROM `targetbusinesscasestudy-425313.Target_Dataset.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers';
```

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

```sql
#Get the time range between which the orders were placed:

SELECT MIN(order_purchase_timestamp) AS min_order_timestamp,
       MAX(order_purchase_timestamp) AS max_order_timestamp
FROM Target_Dataset.orders;
```

JOB INFORMATION | RESULTS | CHART | JSON | E

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

```sql
#Count the Cities & States of customers who ordered during the given period:

SELECT customer_city, customer_state, COUNT(*) AS order_count
FROM Target_Dataset.orders o
JOIN Target_Dataset.customers c ON o.customer_id = c.customer_id
GROUP BY customer_city, customer_state;
```

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	customer_city	customer_state	order_count
1	acu	RN	3
2	ico	CE	8
3	ipe	RS	2
4	ipu	CE	4
5	ita	SC	3
6	itu	SP	136
7	jau	SP	74
8	luz	MG	2
9	poa	SP	85
10	uba	MG	53

```
#In-depth Exploration:-

#Growing trend in the no. of orders placed over the past years:

SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
       COUNT(*) AS order_count
FROM Target_Dataset.orders
GROUP BY year
ORDER BY year;
```

Row	year	order_count
1	2016	329
2	2017	45101
3	2018	54011

```
#Monthly seasonality in terms of the no. of orders being placed:

SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
       COUNT(*) AS order_count
FROM Target_Dataset.orders
GROUP BY month
ORDER BY order_count desc;
```

Row	month	order_count
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674

```sql
#Time of the day Brazilian customers mostly place their orders:

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 'Morning'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
            ELSE 'Night'
        END AS order_time,
        COUNT(*) AS order_count
FROM Target_Dataset.orders
GROUP BY order_time
ORDER BY order_count desc;
```

Row	order_time	order_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

```sql
#Evolution of E-commerce orders in Brazil region:-

#Month-on-month no. of orders placed in each state:

SELECT EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
        customer_state,
        COUNT(*) AS order_count
FROM Target_Dataset.orders o
JOIN Target_Dataset.customers c ON o.customer_id = c.customer_id
GROUP BY month, customer_state
ORDER BY order_count desc;
```

Row	month	customer_state	order_count
1	8	SP	4982
2	5	SP	4632
3	7	SP	4381
4	6	SP	4104
5	3	SP	4047
6	4	SP	3967
7	2	SP	3357
8	1	SP	3351
9	11	SP	3012
10	12	SP	2357

```sql
#Distribution of customers across all states:

SELECT customer_state, COUNT(DISTINCT customer_id) AS customer_count
FROM Target_Dataset.customers
GROUP BY customer_state
ORDER BY customer_count desc;
```

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

```sql
#Impact on Economy:-

#% increase in the cost of orders from year 2017 to 2018:

SELECT ((SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2018 THEN payment_value END) -
        SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 THEN payment_value END)) /
        SUM(CASE WHEN EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 THEN payment_value END)) * 100 AS percent_increase
FROM Target_Dataset.payments
JOIN Target_Dataset.orders ON payments.order_id = orders.order_id
WHERE EXTRACT(YEAR FROM order_purchase_timestamp) IN (2017, 2018)
  AND EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8;
```

Row	percent_increase
1	136.9768716466…

```sql
#Total & Average value of order price and freight for each state:

SELECT customer_state,
       SUM(price) AS total_order_price,
       AVG(price) AS avg_order_price,
       SUM(freight_value) AS total_freight_value,
       AVG(freight_value) AS avg_freight_value
FROM Target_Dataset.orders o
JOIN Target_Dataset.order_items oi ON o.order_id = oi.order_id
JOIN Target_Dataset.customers c ON o.customer_id = c.customer_id
GROUP BY customer_state;
```

Row	customer_state	total_order_price	avg_order_price	total_freight_value	avg_freight_value
1	SP	5202955.050002…	109.6536291597…	718723.0699999…	15.14727539041…
2	RJ	1824092.669999…	125.1178180945…	305589.3100000…	20.96092393168…
3	PR	683083.7600000…	119.0041393728…	117851.6800000…	20.53165156794…
4	SC	520553.3400000…	124.6535775862…	89660.26000000…	21.47036877394…
5	DF	302603.9399999…	125.7705486284…	50625.49999999…	21.04135494596…
6	MG	1585308.029999…	120.7485741488…	270853.4600000…	20.63016680630…
7	PA	178947.8099999…	165.6924166666…	38699.30000000…	35.83268518518…
8	BA	511349.9900000…	134.6012082126…	100156.6799999…	26.36395893656…
9	GO	294591.9499999…	126.2717316759…	53114.97999999…	22.76681525932…
10	RS	750304.0200000…	120.3374530874…	135522.7400000…	21.73580433039…

```sql
#Analysis based on sales, freight, and delivery time:-

#No. of days taken to deliver each order and difference between estimated & actual delivery date:

SELECT order_id,
       order_delivered_customer_date - order_purchase_timestamp AS time_to_deliver,
       order_delivered_customer_date - order_estimated_delivery_date AS diff_estimated_delivery
FROM Target_Dataset.orders;
```

Row	order_id	time_to_deliver	diff_estimated_delivery
1	7a4df5d8cff4090e541401a20a...	null	null
2	35de4050331c6c644cddc86f4...	null	null
3	b5359909123fa03c50bdb0cfe...	null	null
4	dba5062fbda3af4fb6c33b1e04...	null	null
5	90ab3e7d52544ec7bc3363c82...	null	null
6	fa65dad1b0e818e3ccc5cb0e3...	null	null
7	1df2775799eecdf9dd8502425...	null	null
8	6190a94657e1012983a274b8...	null	null
9	58ce513a55c740a3a81e8c8b7...	null	null
10	088683f795a3d30bfd61152c4f...	null	null

```
#Top 5 states with the highest & lowest average freight value:

SELECT c.customer_state,
       AVG(oi.freight_value) AS avg_freight_value
FROM Target_Dataset.order_items AS oi
JOIN Target_Dataset.orders AS o ON oi.order_id = o.order_id
JOIN Target_Dataset.customers AS c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY avg_freight_value DESC
LIMIT 5;

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

Row	customer_state	avg_freight_value
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

Row	customer_state	avg_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

```sql
#Top 5 states with the highest & lowest average delivery time:

SELECT customer_state,
       AVG(order_delivered_customer_date - order_purchase_timestamp) AS avg_delivery_time
FROM Target_Dataset.orders o
JOIN Target_Dataset.customers c ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY avg_delivery_time DESC
LIMIT 5;

SELECT customer_state,
       AVG(order_delivered_customer_date - order_purchase_timestamp) AS avg_delivery_time
FROM Target_Dataset.orders o
JOIN Target_Dataset.customers c ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY avg_delivery_time
LIMIT 5;
```

Row	customer_state	avg_delivery_time
1	RR	0-0 0 705:18:3.975609
2	AP	0-0 0 652:26:29.850746
3	AM	0-0 0 634:13:25.613793
4	AL	0-0 0 589:3:9.103274
5	PA	0-0 0 570:33:0.021141

Row	customer_state	avg_delivery_time
1	SP	0-0 0 210:16:21.207111
2	PR	0-0 0 287:47:52.704448
3	MG	0-0 0 288:14:46.320827
4	DF	0-0 0 311:13:17.884615
5	SC	0-0 0 359:1:23.299971

```sql
#Top 5 states where the order delivery is really fast compared to the estimated date of delivery:

SELECT customer_state,
       AVG(order_delivered_customer_date - order_estimated_delivery_date) AS avg_delivery_diff
FROM Target_Dataset.orders o
JOIN Target_Dataset.customers c ON o.customer_id = c.customer_id
GROUP BY customer_state
ORDER BY avg_delivery_diff
LIMIT 5;
```

Row	customer_state	avg_delivery_diff
1	AC	0-0 0 -481:50:53.400
2	RO	0-0 0 -465:31:25.802469
3	AP	0-0 0 -457:25:34.119402
4	AM	0-0 0 -452:26:36.986206
5	RR	0-0 0 -398:16:13.243902

```sql
#Month-on-month no. of orders placed using different payment types:

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

Row	month	payment_type	order_count
1	1	voucher	477
2	1	credit_card	6103
3	1	debit_card	118
4	1	UPI	1715
5	2	credit_card	6609
6	2	voucher	424
7	2	UPI	1723
8	2	debit_card	82
9	3	voucher	591
10	3	credit_card	7707

```sql
#No. of orders placed on the basis of the payment installments that have been paid:

SELECT payment_installments,
       COUNT(*) AS order_count
FROM Target_Dataset.payments
GROUP BY payment_installments;
```

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

Insights and Recommendations:

Insights:

Highest Average Freight Value States:

1. Identify states with the highest average freight value to understand potential logistical challenges or higher delivery costs in these areas.

Lowest Average Freight Value States:

1. Identify states with the lowest average freight value to understand areas with efficient delivery systems or lower shipping costs.

Recommendations:

Optimize Shipping Costs:

1. For states with high average freight values, consider negotiating better rates with carriers or optimizing delivery routes to reduce costs.

Improve Logistics:

1. Investigate the logistics infrastructure in states with high freight values to identify areas for improvement, such as warehouse locations or transportation methods.

Targeted Marketing:

1. Use the insights to adjust pricing strategies or offer promotions in states with higher shipping costs to offset the freight value for customers.