Case Study: E-Commerce Dataset Analysis (Target_SQL)

G. Barath Krishna

Question: What are the data types of all columns in the “customers” table?

```sql
SELECT
  column_name,
  data_type
FROM
  `target-sql-case-study-467309.Target_SQL.INFORMATION_SCHEMA.COLUMNS`
WHERE
  table_name = 'customers';
```

Query results

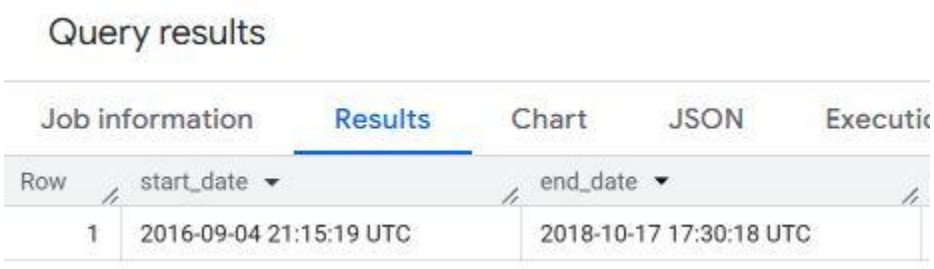
Job information | Results | Chart | JSON

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:** This helps understand the schema and structure before querying or transforming data.

Question: What is the time range during which orders were placed?

```sql
SELECT
  MIN(order_purchase_timestamp) AS start_date,
  MAX(order_purchase_timestamp) AS end_date
FROM
  `Target_SQL.orders`;
```



📈 **Insight:** Establishes the overall timeline of the dataset, key for trend analysis.

Question: Count the cities and states of customers who ordered during the period.

```sql
SELECT
  customer_state,
  customer_city,
  COUNT(customer_id) AS customer_count
FROM
  `Target_SQL.customers`
GROUP BY
  customer_state,
  Customer_city
```

```
ORDER BY
  Customer_count DESC
```

Query results

Job information | Results | Chart | JSON | Execution details | Exe

Row	customer_state	customer_city	customer_count
1	AC	epitaciolandia	1
2	AC	porto acre	1
3	AC	brasileia	1
4	AC	cruzeiro do sul	3
5	AC	manoel urbano	1

Query results

Job information | Results | Chart | JSON | Execution details | Ex

Row	customer_state	customer_city	customer_count
1	SP	sao paulo	15540
2	RJ	rio de janeiro	6882
3	MG	belo horizonte	2773
4	DF	brasilia	2131
5	PR	curitiba	1521

📈 **Insight:**

Shows regional concentration of customers, valuable for targeting and logistics.

The maximum number of orders were placed in the city of sao paulo

---

Question: Is there a growing trend in the number of orders over the years?

```sql
SELECT
  EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
  COUNT(order_id) AS no_of_orders_purchased
FROM
  `Target_SQL.orders`
GROUP BY
  EXTRACT(YEAR FROM order_purchase_timestamp)
ORDER BY
  Year;
```

Query results

Job information | Results | Chart

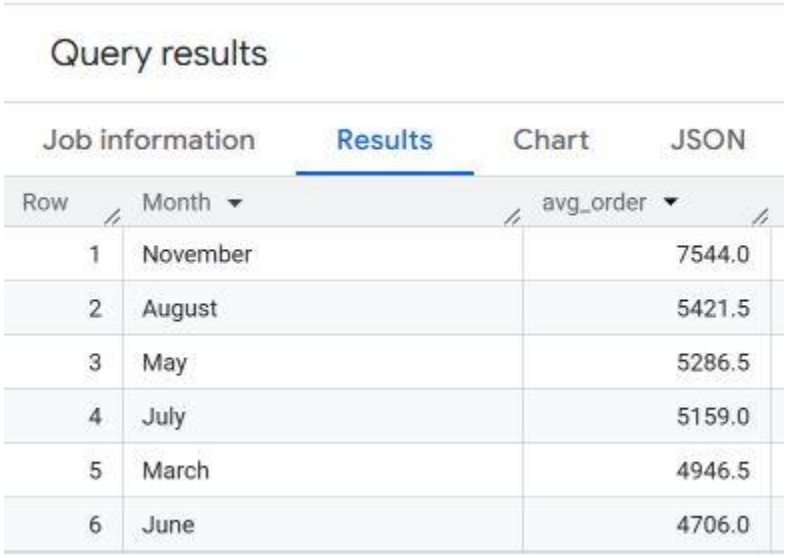
Row	Year	no_of_orders_pur...
1	2016	329
2	2017	45101
3	2018	54011

📈 **Insight:** Tracks growth of the e-commerce platform across calendar years.

Yes, Year-wise analysis also indicates that there is a growing trend when compare to past years.

Question: Can we see monthly seasonality in order placement?

```sql
WITH monthly_orders AS (
  SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
    FORMAT_DATE('%B', order_purchase_timestamp) AS month,
    COUNT(order_id) AS no_of_orders_purchased
  FROM
    `Target_SQL.orders`
  GROUP BY
    EXTRACT(YEAR FROM order_purchase_timestamp),
    FORMAT_DATE('%B', order_purchase_timestamp),
    EXTRACT(MONTH FROM order_purchase_timestamp)
)
SELECT
  month,
  AVG(no_of_orders_purchased) AS avg_order
FROM
  monthly_orders
GROUP BY
  month
ORDER BY
  avg_order DESC;
```



Query results

Job information | Results | Chart | JSON

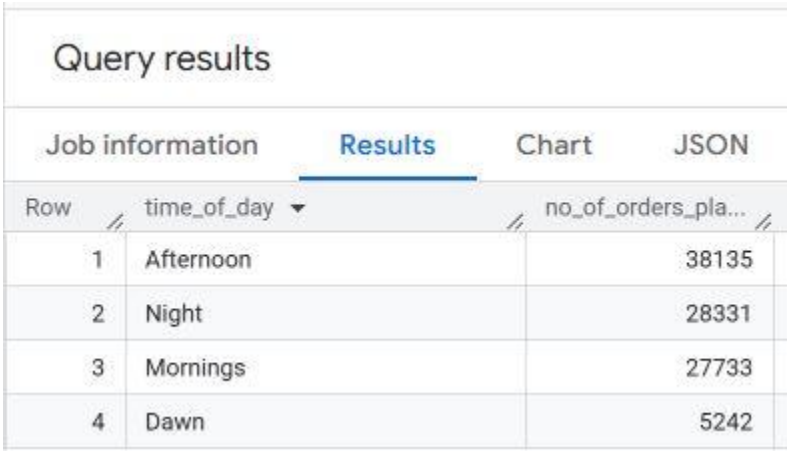
Row	Month	avg_order
1	November	7544.0
2	August	5421.5
3	May	5286.5
4	July	5159.0
5	March	4946.5
6	June	4706.0

📈 **Insight:** Highlights seasonal peaks in shopping behavior (e.g., holiday spikes).

In the month of **November** the orders were placed more when compared to other months, it is due to the **Black friday Sale,** and Sale Spike also seen Prior to the **Christmas eve.**

Question: During what time of day do customers place orders?

```sql
WITH cte AS (
  SELECT
    *,
    CASE
      WHEN EXTRACT(HOUR FROM order_purchase_timestamp) <= 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 time_of_day
  FROM
    `Target_SQL.orders`
)
SELECT
  time_of_day,
  COUNT(order_id) AS no_of_orders_placed
FROM
  cte
GROUP BY
  time_of_day
ORDER BY
  no_of_orders_placed DESC;
```



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

📈 **Insight:** Informs about customer behavior by time block, useful for scheduling ads/promotions.

Most of the orders were placed in the **Afternoon** time in the day and least orders were placed in the **Dawn** time.

Question: Month-on-month number of orders in each state.

```sql
WITH cte AS (
  SELECT
    c.customer_state,
    FORMAT_DATE('%Y-%m', o.order_purchase_timestamp) AS year_month,
    COUNT(o.order_id) AS order_count
  FROM
    `Target_SQL.customers` c
  JOIN
    `Target_SQL.orders` o
  ON
    c.customer_id = o.customer_id
  GROUP BY
    1, 2
```

```
),
cte2 AS (
  SELECT
    customer_state,
    year_month,
    order_count,
    COALESCE(LAG(order_count, 1) OVER (PARTITION BY customer_state ORDER BY
year_month), 0) AS previous_month_count
  FROM
    cte
)
SELECT
  *,
  ROUND(SAFE_DIVIDE((order_count - previous_month_count) * 100,
previous_month_count), 2) AS mom_percentage_change
FROM
  cte2
ORDER BY
  mom_percentage_change DESC
LIMIT 40;
```

Query results

Save results

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	customer_state	year_Month	order_count	previous_month_...	mom_percentage...
1	PR	2017-01	65	1	6400.0
2	SP	2016-10	113	2	5550.0
3	RS	2016-10	24	1	2300.0
4	TO	2017-08	15	1	1400.0
5	MS	2017-02	11	1	1000.0

📈 **Insight:** Tracks dynamic growth or drop in order volume per state, month-to-month.

We can see the **PR state** has seen the most increase in order placement percentage has drastically increased in month of January (2017) compared to the previous month.

A similar trend can be seen for the state SP.

Question: How are the customers distributed across all the states?

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

Query results

Job information | Results | Chart | JSON

Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637

Query results

Job information | Results | Chart | JSON

Row	customer_state	customer_count
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

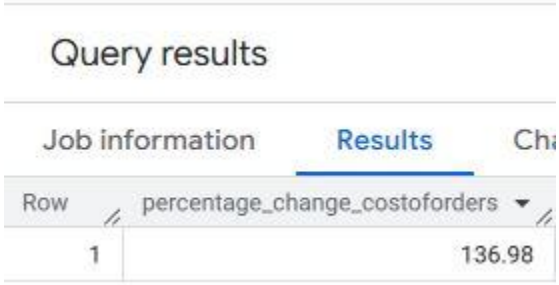
📈 **Insight:** Highlights which states dominate the customer base.

The state SP contains the maximum number of customers and the State RR has the minimum number of purchases by customers.

Question: Get the % increase in the cost of orders from 2017 to 2018 (Jan to Aug only).

```
WITH purchase_data AS (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
    SUM(p.payment_value) AS total_payment
  FROM
    `Target_SQL.orders` o
  JOIN
    `Target_SQL.payments` p
  ON
    o.order_id = p.order_id
  GROUP BY
    year, month
)
```

```sql
SELECT
  ROUND(
    SAFE_DIVIDE(
      SUM(CASE WHEN year = 2018 THEN total_payment ELSE 0 END) -
      SUM(CASE WHEN year = 2017 THEN total_payment ELSE 0 END),
      SUM(CASE WHEN year = 2017 THEN total_payment ELSE 0 END)
    ) * 100, 2
  ) AS percentage_change_costoforders
FROM
  purchase_data
WHERE
  year IN (2017, 2018)
  AND month BETWEEN 1 AND 8;
```



📈 **Insight:** Quantifies the economic growth of e-commerce within a defined seasonal window.

The total amount paid by customers more than doubled in the same 8-month period year-over-year.

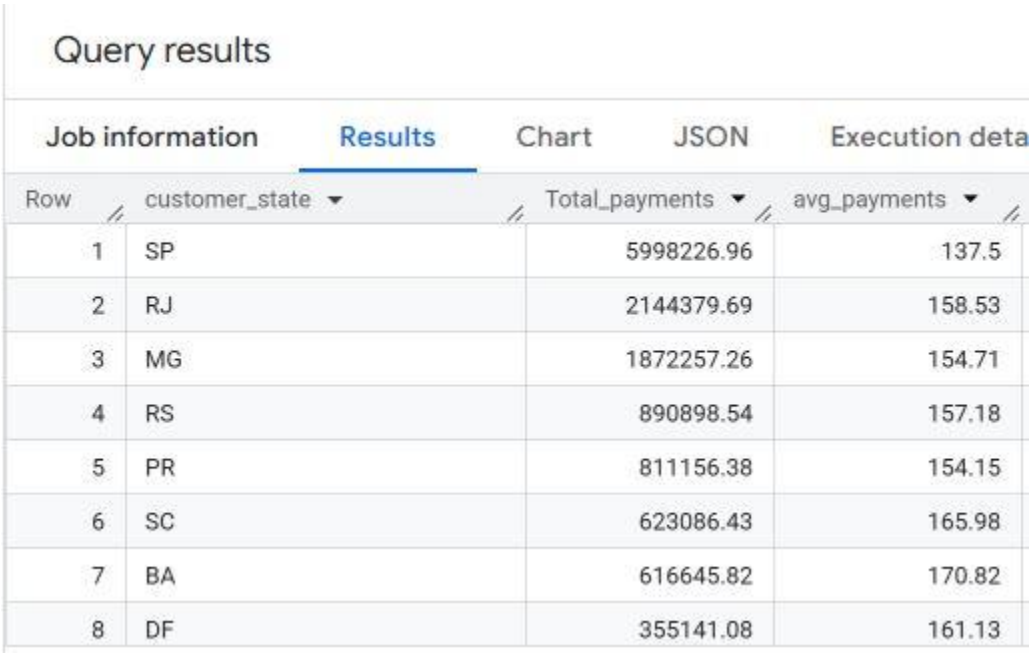
Indicates strong growth in business performance and customer spending.

Question: Calculate the total and average value of order price for each state.

```
SELECT
  c.customer_state,
  ROUND(SUM(p.payment_value), 2) AS total_payments,
  ROUND(AVG(p.payment_value), 2) AS avg_payments
FROM
  `Target_SQL.customers` c
JOIN
  `Target_SQL.orders` o ON c.customer_id = o.customer_id
JOIN
  `Target_SQL.payments` p ON o.order_id = p.order_id
GROUP BY
  C.customer_state

ORDER BY

  Total_payments desc, avg_payments;
```



Query results

Job information | Results | Chart | JSON | Execution deta

Row	customer_state	Total_payments	avg_payments
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13

📈 **Insight:** Shows how different states contribute to overall revenue and average transaction size.

You can see states SP & RJ have the highest total order values, indicating the states with the most significant sales volume.

Question: Calculate the total and average value of order freight for each state.

```sql
SELECT
  c.customer_state,
  ROUND(SUM(oi.freight_value), 2) AS total_freight_value,
  ROUND(AVG(oi.freight_value), 2) AS avg_freight_value
FROM
  `Target_SQL.customers` c
JOIN
  `Target_SQL.orders` o ON c.customer_id = o.customer_id
JOIN
  `Target_SQL.order_items` oi ON o.order_id = oi.order_id
GROUP BY
  C.customer_state;

ORDER BY

  Total_freight_value Desc, avg_freight_value;
```

Query results

Job information | Results | Chart | JSON | Execution detai

Row	customer_state	Total_frieght_value	avg_frieght_value
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92

📈 **Insight:** Identifies states with the highest freight costs, indicating shipping effort or distance.

The state SP & RJ have high total_frieght_value correlated to Total_payments as seen in the previous question.

Question: Delivery time and gap from estimated delivery per order.

```
SELECT
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date,
DAY) AS diff_estimated_delivery
FROM
  `Target_SQL.orders`;
```

Query results

Job information | Results | Chart

Row	time_to_deliver	diff_estimated_d...
1	35	16
2	30	28
3	30	-12
4	54	-36
5	43	6
6	36	14
7	29	20
8	30	19

📈 **Insight:** Measures actual delivery performance against estimated schedules.

Identify orders with long delivery time, indicates the delays or issue in the delivering the order.

Question: Top and bottom 5 states by average freight value.

```
WITH avg_freight AS (
  SELECT
    c.customer_state,
    ROUND(AVG(oi.freight_value), 2) AS avg_freight_value
  FROM
    `Target_SQL.customers` c
  JOIN
    `Target_SQL.orders` o ON c.customer_id = o.customer_id
  JOIN
    `Target_SQL.order_items` oi ON o.order_id = oi.order_id
  GROUP BY
    c.customer_state
)
(
  SELECT
    customer_state,
```

```
    avg_freight_value,
    'Top 5' AS remark
  FROM
    avg_freight
  ORDER BY
    avg_freight_value DESC
  LIMIT 5
)
UNION ALL
(
  SELECT
    customer_state,
    avg_freight_value,
    'Bottom 5' AS remark
  FROM
    avg_freight
  ORDER BY
    avg_freight_value
  LIMIT 5
);
```

Query results

Job information | Results | Chart | JSON | Execution detail

Row	customer_state	avg_freight_value	remark
1	RR	42.98	TOP 5
2	PB	42.72	TOP 5
3	RO	41.07	TOP 5
4	AC	40.07	TOP 5
5	PI	39.15	TOP 5
6	SP	15.15	Bottom 5
7	PR	20.53	Bottom 5
8	MG	20.63	Bottom 5
9	RJ	20.96	Bottom 5
10	DF	21.04	Bottom 5

📈 **Insight:** Identifies logistical outliers in terms of freight costs.

States with lower average freight values may benefit from improved logistics and supply chain management to reduce costs and increase competitiveness.

Question: Top and bottom 5 states by average delivery time.

```
WITH avg_delivery AS (
  SELECT
    c.customer_state,
    ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)), 2) AS avg_delivery_time
  FROM
    `Target_SQL.customers` c
  JOIN
    `Target_SQL.orders` o ON c.customer_id = o.customer_id
  GROUP BY
    c.customer_state
)
(
  SELECT
    customer_state,
    avg_delivery_time,
    'Top 5' AS category
  FROM
    avg_delivery
  ORDER BY
    avg_delivery_time DESC
  LIMIT 5
)
UNION ALL
(
  SELECT
    customer_state,
    avg_delivery_time,
    'Bottom 5' AS category
  FROM
    avg_delivery
  ORDER BY
    avg_delivery_time
  LIMIT 5
);
```

Query results

Job information | Results | Chart | JSON | Executi

Row	customer_state	avg_delivery_time	category
1	RR	28.98	Top 5
2	AP	26.73	Top 5
3	AM	25.99	Top 5
4	AL	24.04	Top 5
5	PA	23.32	Top 5
6	SP	8.3	Bottom 5
7	PR	11.53	Bottom 5
8	MG	11.54	Bottom 5
9	DF	12.51	Bottom 5
10	SC	14.48	Bottom 5

📈 **Insight:** Reveals states with fastest and slowest deliveries.

Top 5 States (Highest Delivery Time)

- These states have **the slowest delivery speeds**.
- Possible inferences:
    - **Geographically remote or less accessible** (e.g., northern regions like Acre, Roraima).
    - **Weaker logistics infrastructure**.
    - Fewer distribution hubs or longer last-mile distances.
    - Lower order volume leading to **less frequent shipping routes**.

Business impact: Poor customer experience in these states. May result in lower satisfaction, reduced retention, or bad reviews.

Bottom 5 States (Lowest Delivery Time)

- These states have the **fastest average delivery** times.
- Possible reasons:
    - **Proximity to distribution centers or warehouses**.
    - Better road, air, and urban infrastructure.
    - High order volumes leading to optimized logistics.
    - Often include **São Paulo (SP)**, **Rio de Janeiro (RJ)**, or **Minas Gerais (MG)**.

Business advantage: Higher potential for **customer satisfaction**, **repeat purchases**, and operational efficiency.

Question: Top 5 states where delivery is faster than estimated.

```
SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)), 2) AS avg_days_early
FROM
  `Target_SQL.customers` c
JOIN
  `Target_SQL.orders` o ON c.customer_id = o.customer_id
WHERE
  order_delivered_customer_date IS NOT NULL
  AND order_estimated_delivery_date IS NOT NULL
GROUP BY
  c.customer_state
ORDER BY
  avg_days_early DESC
LIMIT 5;
```

Query results

Job information | Results | Chart | JSON

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

📈 **Insight:** Measures best-performing states in beating delivery estimates.

- These top 5 states consistently **receive deliveries earlier than estimated**, improving customer satisfaction.
- Indicates **efficient logistics and buffer in estimated dates**, possibly due to proximity to warehouses or optimized routes.

Question: Month-on-month number of orders by payment type.

```
SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  p.payment_type,
  COUNT(o.order_id) AS order_count
FROM
  `Target_SQL.payments` p
JOIN
  `Target_SQL.orders` o ON p.order_id = o.order_id
GROUP BY
```

```
  1, 2, 3
ORDER BY
  year, month, payment_type;
```

Query results

Job information | Results | Chart | JSON | Execution details | Execution

Row	year	month	payment_type	order_count
1	2016	9	credit_card	3
2	2016	10	UPI	63
3	2016	10	credit_card	254
4	2016	10	debit_card	2
5	2016	10	voucher	23
6	2016	12	credit_card	1
7	2017	1	UPI	197

📈 **Insight:** Tracks the popularity of different payment modes over time.

Majority of the orders are placed using credit card followed by UPI

Question: Number of orders based on payment installments.

```
SELECT
  payment_installments,
  COUNT(DISTINCT order_id) AS number_of_orders
FROM
```

```
  `Target_SQL.payments`
WHERE
  payment_installments >= 1
GROUP BY
  payment_installments
ORDER BY
  Payment_installments;
```

Query results

Job information | Results | Chart

Row	payment_installm...	number_of_orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253

📈 **Insight:** Analyzes how often customers choose to pay in installments, indicating affordability preferences.

Around 50% of the orders placed have at least one installment paid.

Recommendations for Target E-commerce

1. **Expand Logistics to Slow Delivery Regions**

   States like Acre and Roraima have high average delivery times. Consider **opening micro-fulfillment centers** or **partnering with local carriers** to improve delivery

speed and customer satisfaction.

2. **Focus on High-Performing States (SP, RJ)**

   São Paulo (SP) and Rio de Janeiro (RJ) contribute significantly to both **sales volume and payment value**. Invest in **exclusive promotions, faster shipping, or loyalty programs** in these regions to retain high-value customers.

3. **Leverage Seasonal Trends More Aggressively**

   With **November seeing the highest order volume**, optimize **Black Friday and holiday campaigns** with advance inventory planning, discount bundling, and personalized marketing.

4. **Promote Digital and Installment Payments**

   Credit card and UPI are dominant payment methods, and nearly **50% of orders involve installments**. Offer more **installment options**, particularly for high-ticket items, to increase cart size and affordability.

5. **Improve Delivery Accuracy in Underperforming States**

   Focus on reducing the gap between **estimated and actual delivery times**, especially in states not among the “Top 5 faster-than-estimated” performers. Consider **dynamic delivery estimation models** based on region.

6. **Monitor Freight vs. Revenue Efficiency**

   In states with **high freight costs but low order volume**, analyze cost-benefit of shipping policies. You might consider **minimum order thresholds** for free shipping or **zone-based pricing models**.

7. **Target Marketing by Time-of-Day**

   Most orders are placed in the **afternoon**, so schedule **email, SMS, and app push promotions** during that window for maximum conversion.