TARGET SQL CASE STUDY

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

A. Data type of all columns in the “customers” table.

1. **Query:**

```
SELECT column_name,
       data_type
FROM `Target_case_study.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers';
```

2. **Result:**

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

3. **Insights:**

customer_unique_id: Best for identifying unique customers across multiple purchases.
customer_zip_code_prefix: Stored as integer but should be treated as string to preserve formatting.

4. **Recommendations:**

Convert zip code to string to avoid losing leading zeros or misusing it in calculations.
Standardize city/state names to ensure accurate location-based reporting.

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

1. **Query:**

```
SELECT MIN(order_purchase_timestamp) AS first_order_date,
       MAX(order_purchase_timestamp) AS last_order_date
FROM `myfirstsandboxproject-395904.Target_case_study.orders`;
```

2. **Result:**

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

3. **Insights:**

Orders were placed between September 4, 2016 and October 17, 2018.

This gives us a 2+ year window of data which is enough to analyze trends over time (monthly, quarterly, seasonally).

The data covers a full cycle for year-over-year performance comparison.

4. **Recommendations:**

Use this time range to create monthly/seasonal trends, e.g., high-performing months or holiday spikes.

Segment your analyses into yearly cohorts (2016–2017, 2017–2018) for retention or performance tracking.

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

1. **Query:**

```
SELECT COUNT(DISTINCT c.customer_city) AS unique_customer_cities,
       COUNT(DISTINCT c.customer_state) AS unique_customer_states
FROM `myfirstsandboxproject-395904.Target_case_study.orders` o
JOIN `myfirstsandboxproject-395904.Target_case_study.customers` c
  ON o.customer_id = c.customer_id
WHERE o.order_purchase_timestamp BETWEEN '2016-09-04' AND '2018-10-17';
```

2. **Result:**

Row	unique_customer_cities	unique_customer_states
1	4119	27

3. **Insights:**

   Orders were placed from 4,119 unique cities, indicating a wide geographic spread across Brazil.

   Customers came from 27 different states, covering almost all of Brazil's federative units**.**

4. **Recommendations:**

   Focus marketing efforts on the top-performing cities within these 27 states to boost ROI.

   Identify underrepresented states and analyze potential market entry strategies there.

II. In-depth Exploration:

A. Is there a growing trend in the no. of orders placed over the past years?

Query:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       COUNT(order_id) AS total_orders
FROM `myfirstsandboxproject-395904.Target_case_study.orders`
WHERE order_purchase_timestamp BETWEEN '2016-09-04' AND '2018-10-17'
GROUP BY order_year
ORDER BY order_year;
```

Result:

Row	order_year	total_orders
1	2016	329
2	2017	45101
3	2018	54010

Insights:

Strong upward trend**:** Orders jumped from just 329 in 2016 to over 45,000 in 2017 and 54,000 in 2018**,** indicating rapid platform adoption and customer trust.

2018 growth (~19.9%) over 2017 suggests momentum was still building, though at a slower rate than the initial spike.

Recommendations:

Invest in retention**:** Use the growing user base to drive loyalty programs and cross-sell.

Analyze 2018 further**:** Identify key drivers behind the continued growth (e.g., campaigns, product categories).

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

Query:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
       COUNT(order_id) AS total_orders
FROM `myfirstsandboxproject-395904.Target_case_study.orders`
WHERE order_status = 'delivered'
GROUP BY order_year, order_month
ORDER BY order_year, order_month;
```

Result:

Row	order_year	order_month	total_orders
1	2016	9	1
2	2016	10	265
3	2016	12	1
4	2017	1	750
5	2017	2	1653
6	2017	3	2546
7	2017	4	2303
8	2017	5	3546
9	2017	6	3135
10	2017	7	3872
11	2017	8	4193
12	2017	9	4150
13	2017	10	4478
14	2017	11	7289

Insights:

With just 3 months data available in 2016, the number of orders peak in October. In 2017, the top 3 months are November, December and October. In 2018, January, March and April are the top 3 months. With the limited data available I don't see any monthly seasonality in terms of number of orders being placed.

Recommendations:

Ramp up marketing before Q4: Given the November surge, plan campaigns and inventory ahead of October to maximize sales.

Introduce early-year incentives: To combat the post-holiday slump, consider discounts or loyalty perks in Jan–Feb.

Monitor Q2-Q3 growth: Gradual increases mid-year hint at steady organic growth.

C. 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 'Morning'
            WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 and 18 THEN 'Afternoon'
            ELSE 'Night' END AS time_of_day,
            COUNT(*) AS total_orders
FROM `myfirstsandboxproject-395904.Target_case_study.orders`
GROUP BY time_of_day
ORDER BY total_orders DESC;
```

Result:

Row	time_of_day	total_orders
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Insights:

Afternoon is the peak period – Most orders are placed between 13:00–18:00, suggesting high us engagement post-lunch hours.

Night and Morning are also active – A significant portion of users order between 7:00–12:00 and 19:00 - 23:00, showing strong activity throughout the day.

Dawn sees the least activity – Very few users place orders between 00:00–06:00, indicating minimal overnight shopping behavior.

Recommendations:

Schedule marketing campaigns or flash sales in the afternoon to capitalize on peak user activity.

Use dawn hours for system maintenance or backend updates as user traffic is lowest during this time.

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

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

Query:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year,
       EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
       c.customer_state,
       COUNT(*) AS total_orders
FROM `myfirstsandboxproject-395904.Target_case_study.orders` o
JOIN `myfirstsandboxproject-395904.Target_case_study.customers` c
  ON o.customer_id = c.customer_id
GROUP BY order_year, order_month, c.customer_state
ORDER BY order_year, order_month, total_orders DESC;
```

Results:

Row	order_year	order_month	customer_state	total_orders
1	2016	9	SP	2
2	2016	9	RR	1
3	2016	9	RS	1
4	2016	10	SP	113
5	2016	10	RJ	56
6	2016	10	MG	40
7	2016	10	RS	24
8	2016	10	PR	19
9	2016	10	SC	11
10	2016	10	GO	9
11	2016	10	CE	8
12	2016	10	PE	7
13	2016	10	DF	6
14	2016	10	BA	4
15	2016	10	ES	4

Insights:

States like SP and RJ consistently show high order volumes, indicating a strong customer base.
Even low-volume states like RR and AC show initial engagement, suggesting gradual market penetration in less active regions.
Early data (e.g., from late 2016) shows rapid acceleration in order volume, particularly in states like SP, as marketing and logistics mature.

Recommendations:

Focus on improving delivery speed and customer experience in top-performing states like SP, RJ, and MG to boost retention.
Invest in awareness and logistics infrastructure in states with low engagement (like RR, AP) to expand national coverage.

B. How are the customers distributed across all the states?

Query:

```
SELECT customer_state,
       COUNT(DISTINCT customer_unique_id) AS total_customers
FROM `myfirstsandboxproject-395904.Target_case_study.customers`
GROUP BY customer_state
ORDER BY total_customers DESC;
```

Result:

Row	customer_state	total_customers
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952
11	PE	1609
12	CE	1313
13	PA	949
14	MT	876
15	MA	726

Insights:

A majority of the customer base is concentrated in a few states such as SP, RJ, and MG alone account for over 50% of all customers.
Several states like PA, MT, and MA have significantly fewer customers (under 1,000).
Mid-tier states like RS, PR, and SC show moderate customer presence and growth potential.

Recommendations:

Focus marketing and logistics efforts on high-tier states to maximize reach and efficiency.

Investigate barriers to entry in the low-tier states regions such as delivery coverage or brand awareness and explore targeted campaigns or partnerships.

Consider mid-tier states for regional expansion strategies with localized promotions or service enhancements.

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

A. 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 cte AS (
SELECT EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
       EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
       ROUND(SUM(p.payment_value), 2) AS total_payment_value
FROM `myfirstsandboxproject-395904.Target_case_study.orders` o
JOIN `myfirstsandboxproject-395904.Target_case_study.payments` p
  ON o.order_id = p.order_id
WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) BETWEEN 2017 AND 2018
  AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY order_year, order_month)

SELECT T1.order_year AS Year1, T1.total_payment_value AS Total1,
       T2.order_year AS Year2, T2.total_payment_value AS Total2,
ROUND(((T2.total_payment_value -
T1.total_payment_value)/T1.total_payment_value)*100,2) AS percent_increase
FROM cte AS T1, cte AS T2
WHERE T1.order_year < T2.order_year;
```

Result:

Row	Year1	Total1	Year2	Total2	percent_increase
1	2017	511276.38	2018	1153982.15	125.71
2	2017	511276.38	2018	992463.34	94.11
3	2017	511276.38	2018	1160785.48	127.04
4	2017	511276.38	2018	1023880.5	100.26
5	2017	511276.38	2018	1115004.18	118.08
6	2017	511276.38	2018	1066540.75	108.6
7	2017	511276.38	2018	1159652.12	126.82
8	2017	511276.38	2018	1022425.32	99.98
9	2017	449863.6	2018	1153982.15	156.52
10	2017	449863.6	2018	992463.34	120.61
11	2017	449863.6	2018	1160785.48	158.03
12	2017	449863.6	2018	1023880.5	127.6
13	2017	449863.6	2018	1115004.18	147.85
14	2017	449863.6	2018	1066540.75	137.08
15	2017	449863.6	2018	1159652.12	157.78

Insights:

Across both values of Total1 (~511K and ~449K), the Total2 values remain consistently higher (all > 990K), reinforcing that the growth wasn't a fluke but a consistent upward trend**.**

The multiple rows imply different segments or customer/payment types might be included. Yet in every row**,** there's a noticeable increase in total payments, implying broad-based improvement**.**

Recommendations:

Perform segmentation analysis on customer type, product category, or region.

Use 2018 high-activity user data to identify repeat buyers, loyal customers, or successful cohorts for retargeting and upselling campaigns.
Since the growth is based on Jan–Aug data, consider whether seasonal effects or a one-time spike occurred.

B. Calculate the Total & Average value of order price for each state.
Query:

```
SELECT c.customer_state,
       ROUND(SUM(p.payment_value), 2) AS total_order_value,
       ROUND(AVG(p.payment_value), 2) AS avg_order_value
FROM `myfirstsandboxproject-395904.Target_case_study.payments` p
JOIN `myfirstsandboxproject-395904.Target_case_study.orders` o
  ON p.order_id = o.order_id
JOIN `myfirstsandboxproject-395904.Target_case_study.customers` c
  ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY total_order_value DESC;
```

Result:

Row	customer_state	total_order_value	avg_order_value
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
9	GO	350092.31	165.76
10	ES	325967.55	154.71
11	PE	324850.44	187.99
12	CE	279464.03	199.9
13	PA	218295.85	215.92
14	MT	187029.29	195.23
15	MA	152523.02	198.86

Insights:

Highest total order value at R$ 5.99M**,** approximately 3x higher than the next state. Indicates a large customer base and strong demand. Average order value is relatively modest (R$137.5), implying volume drives revenue.
PA, CE, and MA have avg order values > R$198**,** topping the list. These states may have fewer orders but high-ticket purchases. Potential for luxury/premium product targeting.
RJ: High total (R$2.14M) and strong avg (R$158.5). MG: Total of R$1.87M and consistent avg value (R$154.7). These are strategic markets for both scale and value**.**

Recommendations:

Expand logistics & inventory in SP: Scale operations to support the volume-heavy market. Improve delivery speed and promotions to retain market dominance. Consider loyalty programs to increase avg order value.
Target high-value states with exclusive campaigns: Launch premium product lines in CE, PA, and MA. Explore partnerships with local influencers to boost visibility.
Grow underperforming but promising states: States like MT, MA, and PA have low total but high avg values. Invest in awareness and market penetration strategies. Offer discounts on minimum cart value to increase order frequency**.**

C. Calculate the Total & Average value of order freight for each state.
Query:

```
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 `myfirstsandboxproject-395904.Target_case_study.order_items` oi
JOIN `myfirstsandboxproject-395904.Target_case_study.orders` o
  ON oi.order_id = o.order_id
JOIN `myfirstsandboxproject-395904.Target_case_study.customers` c
  ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY total_freight_value DESC;
```

Result:

Row	customer_state	total_freight_value	avg_freight_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
9	GO	53114.98	22.77
10	DF	50625.5	21.04
11	ES	49764.6	22.06
12	CE	48351.59	32.71
13	PA	38699.3	35.83
14	MA	31523.77	38.26
15	MT	29715.43	28.17

Insights:

SP has the highest total freight value (~718k), but one of the lowest average freight costs (~15.15), indicating high volume and efficient logistics.

MA and PA have lower total freight values, but highest average freight costs (~38.26 and ~35.83), suggesting fewer orders with expensive deliveries.

PE and CE also show high average freight values (~32+), which may reflect regional delivery challenges or longer distances.

Recommendations:

Optimize Logistics in High-Cost States: Focus on improving delivery efficiency in states like MA, PA and CE to reduce average freight costs.

Leverage SP's Efficiency: Use SP's model as a logistics benchmark and consider regional warehouses or optimized routes in similar urban centers.

Dynamic Pricing or Incentives: Consider freight subsidies or free shipping thresholds in states with high average freight to increase order volume and reduce per-order shipping cost.

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

A. 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_timestam.
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date.

Query:

```
SELECT order_id,
       DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp), DAY) AS time_to_deliver,
       DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date), DAY) AS diff_estimated_delivery
FROM `myfirstsandboxproject-395904.Target_case_study.orders`
WHERE order_status = 'delivered'
ORDER BY order_id;
```

Result:

Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792...	7	9
2	00018f77f2f0320c557190d7a1...	16	3
3	000229ec398224ef6ca0657da...	8	14
4	00024acbcdf0a6daa1e931b03...	6	6
5	00042b26cf59d7ce69dfabb4e...	25	16
6	00048cc3ae777c65dbb7d2a06...	7	15
7	00054e8431b9d7675808bcb8...	8	17
8	000576fe39319847cbb9d288c...	5	16
9	0005a1a1728c9d785b8e2b08...	10	0
10	0005f50442cb953dcd1d21e1f...	2	19
11	00061f2a7bc09da83e415a52d...	5	11
12	00063b381e2406b52ad42947...	11	0
13	0006ec9db01a64e59a68b2c34...	7	22
14	0008288aa423d2a3f00fcb17c...	13	8
15	0009792311464db532ff765bf...	8	6

Insights:

Most orders are delivered within 7–10 days, which indicates a reasonably efficient delivery process.

A few orders (like the one with 25 days delivery time) show significant delays, pointing to potential outliers or logistical issues.

The difference between estimated and actual delivery is often positive (customer receives order earlier), suggesting conservative delivery estimates.

Recommendations:

Investigate **orders with long delivery times (e.g., >15 days)** to identify bottlenecks like location, product type or carrier issues.

Consider **improving the accuracy of estimated delivery dates**, especially where actual deliveries are consistently earlier over the week.

Implement **customer feedback collection** for orders with large gaps between actual and estimated dates to asses satisfaction and adjust estimates accordingly

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

Query:

```sql
WITH avg_freight_per_state AS (
  SELECT c.customer_state AS customer_state,
         ROUND(AVG(oi.freight_value), 2) AS avg_freight_value
  FROM `myfirstsandboxproject-395904.Target_case_study.customers` c
  JOIN `myfirstsandboxproject-395904.Target_case_study.orders` o
    ON c.customer_id = o.customer_id
  JOIN `myfirstsandboxproject-395904.Target_case_study.order_items` oi
    ON o.order_id = oi.order_id
  GROUP BY customer_state
),
top5_states AS (
  SELECT customer_state, avg_freight_value, 'Top5' AS remark
  FROM avg_freight_per_state
  ORDER BY avg_freight_value DESC
  LIMIT 5
),
bottom5_states AS (
  SELECT customer_state, avg_freight_value, 'Bottom5' AS remark
  FROM avg_freight_per_state
  ORDER BY avg_freight_value ASC
  LIMIT 5
)
SELECT customer_state, avg_freight_value, remark
FROM top5_states
UNION ALL
SELECT customer_state, avg_freight_value, remark
FROM bottom5_states
ORDER BY remark DESC;
```

Result:

Row	customer_state	avg_freight_value	remark
1	RR	42.98	Top5
2	PB	42.72	Top5
3	RO	41.07	Top5
4	AC	40.07	Top5
5	PI	39.15	Top5
6	SP	15.15	Bottom5
7	PR	20.53	Bottom5
8	MG	20.63	Bottom5
9	RJ	20.96	Bottom5
10	DF	21.04	Bottom5

Insights:

The top 5 states with the highest average freight value (RR, PB, RO, AC, PI) are likely geographically remote, indicating longer delivery routes and higher shipping costs.

The bottom 5 states (SP, PR, MG, RJ, DF) are among Brazil's most populous and urbanized, implying better logistics infrastructure and lower delivery costs.

There is a clear regional disparity in freight expenses, likely influenced by accessibility and proximity to major distribution centers.

Recommendations:

Optimize logistics in high-cost states by exploring localized warehouses or regional partnerships to reduce freight expenses.

Continue leveraging existing infrastructure in low-cost states for faster and more economical deliveries.

Use freight value data to re-evaluate pricing strategy or offer freight subsidies/discounts in high-cost areas to improve customer satisfaction and retention.

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

Query:

```sql
WITH delivery_times AS (
  SELECT customer_state,
         ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)), 2) AS avg_delivery_time
  FROM `myfirstsandboxproject-395904.Target_case_study.orders` o
  JOIN `myfirstsandboxproject-395904.Target_case_study.customers` c
    ON c.customer_id = o.customer_id
  WHERE order_status = 'delivered'
  GROUP BY customer_state
),
ranked_states AS (
  SELECT customer_state,
         avg_delivery_time,
         ROW_NUMBER() OVER(ORDER BY avg_delivery_time DESC) AS rnk_desc,
         ROW_NUMBER() OVER(ORDER BY avg_delivery_time ASC) AS rnk_asc
  FROM delivery_times
)
SELECT customer_state,
       avg_delivery_time,
       CASE WHEN rnk_desc <= 5 THEN 'Top5'
            WHEN rnk_asc <= 5 THEN 'Bottom5'
            ELSE NULL
       END AS remark
FROM ranked_states
WHERE rnk_desc <= 5 OR rnk_asc <= 5;
```

Result:

Row	customer_state	avg_delivery_time	remark
1	SP	8.3	Bottom5
2	PR	11.53	Bottom5
3	MG	11.54	Bottom5
4	DF	12.51	Bottom5
5	SC	14.48	Bottom5
6	PA	23.32	Top5
7	AL	24.04	Top5
8	AM	25.99	Top5
9	AP	26.73	Top5
10	RR	28.98	Top5

Insights:

Bottom 5 States (Fastest Delivery): SP (8.3 days) leads with the shortest average delivery time. PR, MG, DF, and SC all fall under 15 days, indicating efficient local supply chains and potentially high order volumes aiding faster delivery. These states likely benefit from urban density.

Top 5 States (Slowest Delivery): RR (28.98 days) has the longest delivery time, followed by AP, AM**,** and AL, suggesting infrastructure limitations. PA and other northern states face significantly higher delivery times, which could impact customer satisfaction and order frequency. Lower population density and less optimized delivery routes may be contributing to these delays.

Recommendations:

For Bottom 5 States: Use these states as benchmarks for operational efficiency and consider piloting same-day or next-day delivery services here. Market quick delivery in these regions to attract more customers and encourage higher cart values.

For Top 5 States: Consider partnering with local couriers to cut down delivery time. Invest in delivery route optimization tools to better handle long-distance or cross-region logistics. Set realistic delivery expectations and explore incentives (like discounts or loyalty points) for customers in slow-delivery zones.

D. 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 del_diff_per_order AS (
  SELECT c.customer_state,
         DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS early_days
  FROM `myfirstsandboxproject-395904.Target_case_study.orders` o
  JOIN `myfirstsandboxproject-395904.Target_case_study.customers` c
    ON c.customer_id = o.customer_id
  WHERE order_status = 'delivered'
),
avg_early_del_by_state AS (
  SELECT customer_state,
         ROUND(AVG(early_days), 2) AS avg_early_days
  FROM del_diff_per_order
  GROUP BY customer_state
),
top5_early_del_states AS (
  SELECT customer_state,
         avg_early_days,
         ROW_NUMBER() OVER(ORDER BY avg_early_days DESC) AS rnk
  FROM avg_early_del_by_state
)
SELECT customer_state,
       avg_early_days,
       'Top5_fastdelivery' AS remark
FROM top5_early_del_states
WHERE rnk <= 5;
```

Result:

Row	customer_state	avg_early_days	remark
1	AC	19.76	Top5_fastdelivery
2	RO	19.13	Top5_fastdelivery
3	AP	18.73	Top5_fastdelivery
4	AM	18.61	Top5_fastdelivery
5	RR	16.41	Top5_fastdelivery

Insights:

Exceptional Early Delivery: States like AC (19.76 days early) and RO (19.13 days early) are delivering orders significantly ahead of the estimated date, nearly 2–3 weeks early on average.

Recommendations:

Recalibrate Delivery Estimates: Consider adjusting estimated delivery dates to be more realistic for these states, which can improve customer satisfaction by reducing the perception of "padding" in timelines.
Customer Feedback Integration: Leverage this delivery speed in marketing communications and customer reviews, highlighting early deliveries in these states can enhance brand trust and loyalty.

VI. Analysis based on the payments:

A. Find the month on month no. of orders placed using different payment types**.**
Query:

```
SELECT FORMAT_TIMESTAMP('%Y-%m', o.order_purchase_timestamp) AS order_month,
       p.payment_type,
       COUNT(DISTINCT o.order_id) AS total_orders
FROM `myfirstsandboxproject-395904.Target_case_study.orders` o
JOIN `myfirstsandboxproject-395904.Target_case_study.payments` p
  ON o.order_id = p.order_id
GROUP BY order_month, p.payment_type
ORDER BY order_month, p.payment_type;
```

Result:

Row	order_month	payment_type	total_orders
1	2016-09	credit_card	3
2	2016-10	UPI	63
3	2016-10	credit_card	253
4	2016-10	debit_card	2
5	2016-10	voucher	11
6	2016-12	credit_card	1
7	2017-01	UPI	197
8	2017-01	credit_card	582
9	2017-01	debit_card	9
10	2017-01	voucher	33
11	2017-02	UPI	398
12	2017-02	credit_card	1347
13	2017-02	debit_card	13
14	2017-02	voucher	69
15	2017-03	UPI	590

Insights:

Credit Card orders have seen a significant rise, growing from just 3 orders in September 2016 to 1347 orders by February 2017. UPI payments are also steadily increasing, starting at 63 orders in October 2016 and reaching 590 orders in March 2017.

January 2017 saw a notable spike across all payment methods, especially credit card (582 orders) and UPI (197 orders). This could be due to seasonal shopping trends or promotions. February 2017 had the highest number of total orders, with credit card usage peaking at 1347 orders and UPI at 398 orders.

Recommendations:

Strengthen security and optimize checkout experiences for these payment methods. Consider loyalty programs or cashback incentives for credit card and UPI users.

Introduce discounts or exclusive deals for debit card transactions. Promote voucher-based payments with bundled offers or limited-time promotions.

B. Find the no. of orders placed on the basis of the payment installments that have been paid.
Query:

```
SELECT payment_installments,
       COUNT(order_id) AS total_orders
FROM `myfirstsandboxproject-395904.Target_case_study.payments`
GROUP BY payment_installments
ORDER BY payment_installments;
```

Result:

Row	payment_installments	total_orders
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
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15

Insights:

Orders with 5+ installments are the most frequent, showing that customers prefer spreading payments over time rather than paying in full.
3-installment orders are significantly high, suggesting it might be a common preference for mid-tier purchases.
Single-payment orders are comparatively low, indicating a need for incentives to encourage upfront payments.

Recommendations:

Offer tailored financing options based on frequently chosen installment ranges.
Introduce cashback or discounts for single-payment transactions to increase adoption.
Provide promotional offers or loyalty rewards for customers opting for 5+ installment plans to boost retention.