TARGET SQL Business Case

Swati Singh

1. 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.

   i)QUERY:

```
SELECT column_name, data_type
FROM `scaler-dsml-sql-465413.target.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers';
```

   ii)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

   iii)INSIGHTS: NA

   iv)RECOMMENDATIONS: NA

   v)ASSUMPTIONS: NA

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

   i)QUERY

```
SELECT MIN(order_purchase_timestamp) AS first_order,
       MAX(order_purchase_timestamp) AS last_order,
FROM `target.orders`;
```

   ii)RESULT

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

   iii)INSIGHTS:NA

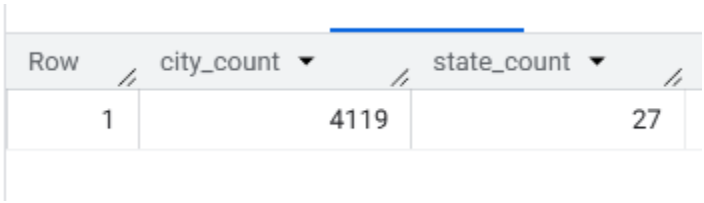
iv)RECOMMENDATIONS:NA
v)ASSUMPTIONS:NA

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

i)QUERY:

```
SELECT COUNT(DISTINCT customer_city) AS city_count,
      COUNT(DISTINCT customer_state) AS state_count
FROM`target.customers`;
```

ii)RESULT:



Row	city_count	state_count
1	4119	27

iii)INSIGHTS: There are 4119 cities and 27 states from where the customers have ordered.

iv)RECOMMENDATIONS:NA
v)ASSUMPTIONS:NA

2. In-depth Exploration:

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

i)QUERY:

```
SELECT *,
      LAG(num_of_orders)OVER(ORDER BY sub.year) AS
prev_year_num_of_orders
FROM
(
  SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
        COUNT(*) AS num_of_orders
  FROM `target.orders`
  GROUP BY year
)AS sub
ORDER BY sub.year;
```

ii)RESULT:

Row	year	num_of_orders	prev_year_num_o...
1	2016	329	*null*
2	2017	45101	329
3	2018	54011	45101

iii)INSIGHTS:
Year over Year growth: 2017 vs 2016= Massive spike—likely the first year of scaling or launch ,since the percentage growth is (45101 - 329)/(329)*100 approx 13,608%.
2018 vs 2017=Healthy growth, though much more stable since percentage growth is (54011-45101)/(45101)*100 approx 19.7%

iv)RECOMMENDATIONS:
The jump from 329 to 45,101 suggests a **major operational shift**—perhaps a product launch, marketing campaign, or platform expansion.
The 19.7% growth in 2018 indicates **continued momentum**, but the explosive phase may be stabilizing.
Therefore the same strategy must be continued.

v)ASSUMPTIONS: NA

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

i)QUERY:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
       EXTRACT(QUARTER FROM order_purchase_timestamp) AS quarter,
       COUNT(*) AS num_of_orders
  FROM `target.orders`
  GROUP BY year,quarter
  ORDER BY 1,3 DESC;
```

ii)RESULT:

Row	year	quarter	num_of_orders
1	2016	4	325
2	2016	3	4
3	2017	4	17848
4	2017	3	12642
5	2017	2	9349
6	2017	1	5262
7	2018	1	21208
8	2018	2	19979
9	2018	3	12820
10	2018	4	4

iii)INSIGHTS:

1.Rapid Growth in 2017

Orders jumped from **329 total in 2016** to **~45,101 in 2017**.

Growth was consistent across quarters:

Q1: 5,262

Q2: 9,349

Q3: 12,642

Q4: 17,848

This suggests strong **momentum and scaling**, possibly due to product-market fit or operational expansion.

2. **Peak in Early 2018**

Q1 2018: **21,208 orders** — highest single quarter.

Q2: Still strong at **19,979**.

Q3: Drops to **12,820** — still solid, but a clear decline.

Q4: Just **4 orders** — a dramatic collapse.

3.**Anomaly or Shutdown in Q4 2018**

From 12,820 to 4 orders is a **99.97% drop**.

Possible causes:System outage or data error,Business closure or pivot,Seasonal shutdown (if applicable),Filtering issue in data extraction

4.**Quarterly Trends**

2017 showed **steady quarter-over-quarter growth**.

2018 started strong but **declined each quarter**, ending in a cliff.

iv)RECOMMENDATIONS:

1. **Investigate Q4 2018 Collapse.**Validate the data — was this a reporting error, system failure, or real business event?If real:Conduct a **root cause analysis**: supply chain issues, customer churn, platform outage?Review **customer feedback**, support logs, and operational changes during Q3–Q4 2018.
2. Use the strong 2017–Q2 2018 data to build a **predictive model**.
3. The 2017 growth suggests successful scaling — replicate what worked.

v)ASSUMPTIONS:

Assumed one quarter as one season.

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

i)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'
WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN
'Night'
END AS time_interval,
COUNT(order_id) AS num_of_orders
FROM `target.orders`
GROUP BY 1
ORDER BY 2 DESC;
```

ii)RESULT:

Row	time_interval	num_of_orders
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

iii)INSIGHTS:

1.**Afternoon Dominates**

38,135 orders — the highest volume.

Suggests peak user activity or business transactions occur post-lunch.

Could be ideal for promotions, staffing, or system optimization.

2. **Night & Morning Are Competitive**

Night: 28,331

Morning: 27,733

Very close — indicates **steady demand throughout the day**, possibly from different user segments (e.g., working professionals vs early risers).

3.**Dawn Is Quiet**

Only 5,242 orders — just **7% of total**.

Could reflect:Low user engagement,Operational downtime,Natural sleep cycle alignment

iv)RECOMMENDATIONS:

1.Focus on Afternoon Peak

Run high-impact campaigns (ads, discounts, new launches) during afternoon hours.

Use this window for push notifications or email outreach — users are most responsive.

2.Leverage Night & Morning

These slots are nearly equal in volume — ideal for:

Segmented offers (e.g., breakfast deals in morning, late-night snacks or services at night)

A/B testing of promotions across time slots

3.Rethink Dawn Strategy

With only 5.3% of orders, consider:

Reducing ad spend or promotional efforts during this time.

Using it for **system maintenance**, backups, or low-priority tasks.

v)ASSUMPTIONS: NA

3. Evolution of E-commerce orders in the Brazil region:
   A) Get the month on month no. of orders placed in each state.

i)QUERY:

```
SELECT c.customer_state,
        EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month_no,
        COUNT(order_id) AS num_orders
FROM `target.customers` AS c
JOIN
`target.orders` AS o
USING(customer_id)
GROUP BY 1,2
ORDER BY 1,2;
```

ii)RESULT:

Row	customer_state	month_no	num_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6
11	AC	11	5

iii)INSIGHTS: NA
iv)RECOMMENDATIONS: NA
v)ASSUMPTIONS: NA

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

i)QUERY:

```
SELECT customer_state,
        COUNT(DISTINCT customer_unique_id) AS num_customers
FROM `target.customers`
GROUP BY 1
ORDER BY 2 DESC;
```

ii)RESULT:

Row	customer_state	num_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

iii)INSIGHTS:

1. São Paulo (SP) Dominates

- 40,302 customers — nearly half of total.
- Indicates strong brand presence, market penetration, or urban concentration.
- SP should be the primary focus for retention, upselling, and logistics.

2. Regional Clustering

- Top 5 states (SP, RJ, MG, RS, PR) account for ~72% of all customers.
- These are mostly southeastern and southern states — suggests geographic bias or urban targeting.

3. North & Northeast Underrepresented

- States like RR, AP, AC, AM, TO have <1% share each.

- Could reflect:Limited delivery coverage,Lower digital adoption,Untapped market potential

iv)RECOMMENDATIONS:

1.Double Down on High-Density States (SP, RJ, MG)

SP alone contributes ~44% of customers — make it your **core market**:

Launch **exclusive loyalty programs**

Offer **same-day delivery** or premium services

Prioritize **customer retention** and upselling

2.Expand in Underpenetrated Regions (North & Northeast)

States like RR, AP, AC, AM have **<1% share** — untapped potential:

Run **localized awareness campaigns**

Partner with **regional influencers or delivery services**

Offer **first-time buyer discounts** or free shipping

3.Optimize Logistics

Use customer density to:

Strategically place warehouses near SP, RJ, MG

Improve **last-mile delivery** in high-volume areas

Test **hub-and-spoke models** for remote regions

4.Regional Staffing

Scale support and fulfillment teams in high-demand states

Train regional teams to handle **local dialects, preferences, and cultural nuances**

5.Geo-Targeted Ads

Run state-specific campaigns.

v)ASSUMPTIONS: NA

4. 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.

i)QUERY:

```
SELECT *,

ROUND(((Y.total_payment2018-Y.total_payment2017)/Y.total_payment2017)*100
,2) AS percent_inc
```

```
FROM
(
  SELECT 2017 AS year_2017,
        ROUND(SUM(CASE WHEN EXTRACT(YEAR FROM
order_purchase_timestamp)=2017 THEN payment_value END)) AS
total_payment2017,
        2018 AS year_2018,
        ROUND(SUM(CASE WHEN EXTRACT(YEAR FROM
order_purchase_timestamp)=2018 THEN payment_value END)) AS
total_payment2018
FROM `target.orders` AS o
JOIN
`target.payments` AS p
USING(order_id)
WHERE EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8
)AS Y;
```

ii)RESULT:

Row	year_2017	total_payment2017	year_2018	total_payment2018	percent_inc
1	2017	3669022.0	2018	8694734.0	136.98

iii)INSIGHTS:

Massive Year-over-Year Growth:

Total Payment in 2017: ₹3,669,022

Total Payment in 2018: ₹8,694,734

Percent Increase: +136.98

This is **more than double** the previous year's value — a strong indicator of growth in either customer base, transaction volume, or pricing.

iv)RECOMMENDATIONS:

1.**Identify key growth drivers**: Was it a product launch, marketing campaign, or regional expansion?

2.Replicate success: Scale the winning strategies to other regions or customer segments.

3.Retention focus: Ensure new customers from 2018 are retained in 2019 and beyond.

v)ASSUMPTIONS: NA

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

i)QUERY:

```
SELECT c.customer_state,
        ROUND(SUM(payment_value),2) AS total_price,
        ROUND(AVG(payment_value),2) AS avg_price
FROM `target.orders` AS o
JOIN
`target.payments` AS p
ON o.order_id=p.order_id
JOIN
`target.customers` AS c
ON o.customer_id=c.customer_id
GROUP BY 1
ORDER BY 3 DESC;
```

ii)RESULT:

Row	customer_state	total_price	avg_price
1	PB	141545.72	248.33
2	AC	19680.62	234.29
3	RO	60866.2	233.2
4	AP	16262.8	232.33
5	AL	96962.06	227.08
6	RR	10064.62	218.8
7	PA	218295.85	215.92
8	SE	75246.25	208.44
9	PI	108523.97	207.11
10	TO	61485.33	204.27
11	CE	279464.03	199.9

iii)INSIGHTS:

1.**São Paulo (SP): Highest Total, Lowest Average**

Total Price: 5.99M (highest by far)

Avg Price: 137.50 (lowest in dataset)

SP likely has **high transaction volume** but **low-value orders** — possibly due to:Dense urban market with frequent small purchases,Strong logistics enabling micro-orders,Price-sensitive customer base.

2. Smaller States Show Higher Avg Prices

Top Avg Prices: PB (248.33), AC (234.29), RO (233.20)

These states have **lower total prices**, suggesting:

Fewer customers or orders

But **higher-value purchases per transaction.T**hese may be **niche markets** with specialized or bundled purchases.

iv)RECOMMENDATIONS:

1. Regional Pricing Strategy

High Avg Price States (e.g., PB, AC, RO, AP):Customers are willing to pay more per order.Introduce **premium products**, **bundled offers**, or **exclusive services**.

Consider **localized promotions** that emphasize quality or exclusivity.

Low Avg Price States (e.g., SP, RJ, MG):High volume, low-value orders.

Focus on **economies of scale**, **subscription models**, or **loyalty discounts** to increase retention and frequency.

2.Marketing Personalization by Region: Deploy **A/B tested ads** tailored to regional buying patterns.

v)ASSUMPTIONS: NA

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

i)QUERY:

```
SELECT c.customer_state,
        ROUND(SUM(i.freight_value),2) AS total_freight,
        ROUND(AVG(i.freight_value),2) AS avg_freight
FROM `target.orders` AS o
JOIN
`target.order_items` AS i
ON o.order_id=i.order_id
JOIN
`target.customers` AS c
ON o.customer_id=c.customer_id
GROUP BY 1
ORDER BY 2 DESC;
```

ii)RESULT:

Row	customer_state	total_freight	avg_freight
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

iii)INSIGHTS:
São Paulo (SP): High Total, Lowest Average
Total Freight: ₹718,723 — highest overall
Avg Freight: ₹15.15 — lowest in dataset
SP benefits from:Dense infrastructure,Proximity to fulfillment centers,High order volume diluting per-order freight

iv)RECOMMENDATIONS:
Negotiate regional carrier contracts: Partner with local logistics firms for better rates. **Introduce freight subsidies or discounts**: Absorb part of the cost to boost order volume.
Consolidate shipments: Use batch delivery or regional hubs to reduce per-order cost.
Explore alternative transport modes: Rail or waterway options where feasible.
Set up micro-fulfillment centers: Even small warehouses can drastically cut last-mile costs.
Use drop-shipping or third-party logistics (3PL): Leverage existing networks to reduce overhead.
Analyze demand density: If pockets of high demand exist, prioritize those for infrastructure.

v)ASSUMPTIONS: NA

5. 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.

i)QUERY:

```
SELECT 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 `target.orders`
WHERE LOWER(order_status)='delivered'
ORDER BY order_id;
```

ii)RESULT:

Row	order_id	time_to_deliver	diff_estimated_d...
1	00010242fe8c5a6d1ba2dd792...	7	-8
2	00018f77f2f0320c557190d7a1...	16	-2
3	000229ec398224ef6ca0657da...	7	-13
4	00024acbcdf0a6daa1e931b038...	6	-5
5	00042b26cf59d7ce69dfabb4e5...	25	-15
6	00048cc3ae777c65dbb7d2a06...	6	-14
7	00054e8431b9d7675808bcb81...	8	-16
8	000576fe39319847cbb9d288c...	5	-15
9	0005a1a1728c9d785b8e2b08b...	9	0
10	0005f50442cb953dcd1d21e1fb...	2	-18
11	00061f2a7bc09da83e415a52d...	4	-10

iii)INSIGHTS: NA

iv)RECOMMENDATIONS: NA
v)ASSUMPTIONS: NA

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

i)QUERY:

```
WITH freight_stats AS (
  SELECT
    c.customer_state,
    ROUND(AVG(oi.freight_value), 2) AS avg_freight
  FROM `target.order_items` AS oi
  JOIN `target.orders` AS o ON oi.order_id = o.order_id
  JOIN `target.customers` AS c ON o.customer_id = c.customer_id
  GROUP BY c.customer_state
),
top5 AS (
  SELECT customer_state AS top_state, avg_freight AS top_avg_freight,
         ROW_NUMBER() OVER (ORDER BY avg_freight DESC) AS rank
  FROM freight_stats
  LIMIT 5
),
bottom5 AS (
  SELECT customer_state AS bottom_state, avg_freight AS
bottom_avg_freight,
         ROW_NUMBER() OVER (ORDER BY avg_freight ASC) AS rank
  FROM freight_stats
  LIMIT 5
)

SELECT
  t.rank,
  t.top_state, t.top_avg_freight,
  b.bottom_state, b.bottom_avg_freight
FROM top5 t
JOIN bottom5 b ON t.rank = b.rank;
```

ii)RESULT:

Row	rank	top_state	top_avg_freight	bottom_state	bottom_avg_freight
1	1	RR	42.98	SP	15.15
2	2	PB	42.72	PR	20.53
3	3	RO	41.07	MG	20.63
4	4	AC	40.07	RJ	20.96
5	5	PI	39.15	DF	21.04

iii)INSIGHTS:

Freight Cost Disparity

- Top states have freight costs nearly 2x higher than bottom states.
- This suggests logistical challenges or remote delivery zones in RR, PB, RO, AC, and PI.

iv)RECOMMENDATIONS:

1. Operational Optimization

- High freight zones may benefit from:
- Regional warehouses or distribution centers
- Bulk shipping strategies or route optimization
- Carrier renegotiation for remote deliveries

2. Pricing Strategy

- Consider dynamic pricing or freight surcharges for high-cost states to maintain margins.
- Alternatively, subsidize shipping in strategic regions to boost customer acquisition.

3. Customer Segmentation

- If customer demand is strong in high-cost states, it may justify the expense.
- If demand is low, consider marketing efforts to increase volume and dilute freight cost per order.

v)ASSUMPTIONS: NA

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

i)QUERY:

```
WITH delivery_stats AS (
  SELECT
    c.customer_state,
    ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)), 2) AS avg_delivery_days
```

```
  FROM `target.orders` AS o
  JOIN `target.customers` AS c ON o.customer_id = c.customer_id
  WHERE o.order_delivered_customer_date IS NOT NULL
  GROUP BY c.customer_state
),
top5 AS (
  SELECT
    customer_state AS top_state,
    avg_delivery_days AS top_avg_days,
    ROW_NUMBER() OVER (ORDER BY avg_delivery_days DESC) AS rank
  FROM delivery_stats
  LIMIT 5
),
bottom5 AS (
  SELECT
    customer_state AS bottom_state,
    avg_delivery_days AS bottom_avg_days,
    ROW_NUMBER() OVER (ORDER BY avg_delivery_days ASC) AS rank
  FROM delivery_stats
  LIMIT 5
)

SELECT
  t.rank,
  t.top_state, t.top_avg_days,
  b.bottom_state, b.bottom_avg_days
FROM top5 t
JOIN bottom5 b ON t.rank = b.rank;
```

ii)RESULT:

Row	rank	top_state	top_avg_days	bottom_state	bottom_avg_days
1	1	RR	28.98	SP	8.3
2	2	AP	26.73	PR	11.53
3	3	AM	25.99	MG	11.54
4	4	AL	24.04	DF	12.51
5	5	PA	23.32	SC	14.48

iii)INSIGHTS:

1.**RR (Roraima)** stands out with nearly **29 days**—suggesting serious delays, possibly due to limited transport routes or low delivery density.

2.High delivery times may correlate with **higher freight costs**, **lower customer satisfaction**, and **greater return risk**.
3.**SP (São Paulo)** leads with just **8.3 days**, likely due to its central role in Brazil's e-commerce and supply chain.
4.These regions may benefit from **same-day or next-day delivery pilots**, **lower operational costs**, and **higher customer retention**.

iv)RECOMMENDATIONS:
1.**Establish or expand warehouses** in Northern states like RR, AP, and AM.
2.Partner with **regional logistics providers** who specialize in remote deliveries.
3.Track **average delivery time**, **variance**, and **on-time delivery rate**.
4.For high-delay regions, offer **discounts**, **loyalty points**, or **free returns** to offset dissatisfaction.

v)ASSUMPTIONS: NA

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.

i)QUERY:

```
SELECT c.customer_state,

ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delivered_custome
r_date,DAY))) AS avg_diff_actual_vs_estimate
FROM `target.orders` AS o
JOIN
`target.customers` AS c
USING(customer_id)
WHERE LOWER(order_status)='delivered'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

ii)RESULT:

Row	customer_state	avg_diff_actual_vs_estimate
1	AC	20.0
2	AM	19.0
3	RO	19.0
4	AP	19.0
5	RR	16.0

iii)INSIGHTS:

1.Deliveries in these states are arriving **16–20 days earlier** than promised.
2.Early deliveries can lead to **positive surprises**, boosting **customer satisfaction**, **trust**, and **brand perception**.

iv)RECOMMENDATIONS:

1. **Refine Delivery Time Estimates**

Use historical delivery data to **recalibrate estimates** per state.

2. **Highlight Fast Delivery in Marketing**

Promote these states as **"fast delivery zones"** to boost conversion.
Offer **express shipping upgrades** or **next-day delivery trials** where feasible.

3.**Optimize Inventory Allocation**

Consider **regional demand forecasting** and **stock placement** to further reduce delivery time and cost.

4.**Customer Communication**

Update checkout messaging to reflect **more accurate delivery windows**, improving transparency and trust.

v)ASSUMPTIONS: NA

6. Analysis based on the payments:
   A) Find the month on month no. of orders placed using different payment types.

i)QUERY:

```
SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
    p.payment_type,
    COUNT(*) AS num_of_orders
FROM `target.orders` AS o
```

```
JOIN
`target.payments` AS p
USING(order_id)
GROUP BY 1,2,3
ORDER BY 1,2,3;
```

ii)RESULT:

Row	year	month	payment_type	num_of_orders
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
8	2017	1	credit_card	583
9	2017	1	debit_card	9
10	2017	1	voucher	61
11	2017	2	UPI	398

iii)INSIGHTS:

1. Credit Card Dominance:

Consistently the most used payment method across all months.
Indicates strong customer trust in card payments and possibly better integration with e-commerce platforms.

2. Rapid Growth of UPI:

Started modestly with **63 orders in Oct 2016**, then surged to **1,509 orders by Nov 2017**.
UPI is becoming a **strong secondary channel**, especially among mobile-first users.

3. Debit Card Usage Remains Low:

Orders range from **2 to 70 per month**, with no significant growth.
May reflect **limited incentives**, **lower credit limits**, or **user preference for credit rewards**.

4. Voucher Usage is Steady but Niche:

Indicates use in **promotional campaigns**, **gift cards**, or **corporate purchases**.
Could be leveraged more aggressively for **customer acquisition or loyalty programs**.

iv)RECOMMENDATIONS:
1.Optimize checkout UX for credit card and UPI, as they dominate usage.
2.**Promote UPI incentives** to accelerate adoption among mobile users.
3.**Reevaluate debit card positioning**—consider bundling with offers or simplifying authentication.
4.Use **voucher campaigns** for targeted promotions, especially during festive seasons.

v)ASSUMPTIONS: NA

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

i)QUERY:

```
SELECT payment_sequential,payment_installments,
        COUNT(DISTINCT order_id) AS num_of_orders
FROM `target.payments`
GROUP BY 1,2
ORDER BY 1,2;
```

ii)RESULT:

Row	payment_sequent...	payment_installm...	num_of_orders
1	1	1	48236
2	1	2	12360
3	1	3	10422
4	1	4	7066
5	1	5	5221
6	1	6	3904
7	1	7	1619
8	1	8	4242
9	1	9	644
10	1	10	5305
11	1	11	23

iii)INSIGHTS:

1.Nearly **half of all orders** are paid in a **single installment** (`payment_installments = 1` and `payment_sequential = 1`).
Indicates strong **preference for upfront payments**, possibly due to:
Small ticket sizes
Lack of interest-free EMI offers
Customer aversion to debt

2. **2 to 6 installments** are common, suggesting customers are comfortable with **short-term EMIs**.

3. Very few customers opt for **12+ installments**, indicating:
Limited availability of long-term financing
Preference for faster payoff
Possibly higher interest rates or lack of awareness

iv)RECOMMENDATIONS:

1. **Promote Short-Term EMI Options:**
Focus on 2–6 installment plans with **zero-interest offers**.
Highlight these options during checkout for **high-value items**.

2. **Analyze Product-Level EMI Adoption**
Identify which categories drive longer installment usage.
Tailor financing offers accordingly.

3. **Improve Installment Visibility**

Ensure customers understand installment options clearly.
Consider **installment calculators** or **monthly breakdowns** on product pages.
4. **Monitor Sequential Payment Behavior:**
Investigate why `payment_sequential > 1` drops off—could be due to **data gaps**, **customer drop-off**, or **payment failures**

v)ASSUMPTIONS: NA