Target Business Case

1 Objective:

The objective of this case study to drive meaningful insights on target business situation and to identify and analyse potential solutions for any problem area identified during case study.

By using data and insights provided in this case study, decision-makers can make more informed and strategic decisions in their own business operations.

2 Exploratory Analysis:

2.1 Data type of columns in a table:

I have explored data base schema of each table, Screenshot attached below for table "Order_items" from Target SQL Dataset.

Query:

```
SELECT column_name,data_type
FROM target.INFORMATION_SCHEMA.COLUMNS
WHERE table_name='order_items'
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

ow	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

Above query can be performed on all table to understand the schema of table.

2.2 Time period for which the data is given:

Based on query performed on orders dataset, Time period of data analysis is starting from 2016 to 2018.

Query:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS yrs,
COUNT(order_id) AS No_of_orders,
FROM `target.orders`
GROUP BY yrs
```

Result:

Query results

JOB INFORMATION | RESULTS

Row	yrs	No_of_orders
1	2017	45101
2	2018	54011
3	2016	329

2.3 Cities and States of customers ordered during the given period

Query:

```
SELECT DISTINCT customer_city,
customer_state
FROM `target.customer` as c
JOIN  `target.orders` AS o
ON c.customer_id = o.customer_id
WHERE EXTRACT(YEAR FROM order_purchase_timestamp) BETWEEN 2016 AND 2018
LIMIT 10
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

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

3 In-depth Exploration:

3.1 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?

Query:

```
SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS purchase_year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS purchase_month,
COUNT(order_id) AS no_of_orders
FROM `target.orders`
GROUP BY purchase_year, purchase_month
ORDER BY purchase_year, purchase_month
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXEC

Row	purchase_year	purchase_month	no_of_orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673

Complete Scenario and seasonal peak:

- Looking at the data, it seems that the company's order volume increased from April 2017 to November 2017, with the highest number of orders in November 2017. Then, the order volume decreased from November 2017 to March 2018, with the lowest number of orders in October and November 2018.
- Overall, the trend seems to show a cyclical pattern with peaks and valleys in order volume. However, it is worth noting that the table only includes data for the first 10 months of 2018, so it is difficult to draw any conclusions about the company's performance for the entire year.
- *Yes, there is a seasonal pattern in the data*, with peaks occurring in the months of November and December for both 2017 and 2018. This could potentially be due to holiday shopping or end-of-year sales.
- Additionally, there is a peak in July 2017, which could potentially be due to mid-year sales or promotions.
- Overall, there is a general trend of higher order volumes in the second half of the year compared to the first half, which could be due to a variety of factors such as seasonality, consumer behaviour, or marketing efforts or repeat customers.

Recommendations:

- As mentioned earlier, there seems to be a clear seasonal pattern in the data, with peaks in November and December. If the company hasn't already done so, they can develop strategies to capitalize on them by conducting in-depth analysis on what's driving it.
- Identify the causes of the low-order months: The data also shows some months with lower order volumes, such as October and November 2018. The company should analyze the causes of these lower-order months and develop strategies to address them, such as marketing campaigns, sales promotions or getting more reviews by previous customers.
- Focus on customer experience: The customer experience is very critical in business in driving repeat business and positive reviews. The company should focus on providing excellent customer service, ensuring timely delivery, and creating a seamless checkout process to enhance the overall customer experience.

3.2 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or 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'
  WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 24
  THEN 'Night'
 END
 AS Buying_time,
 COUNT(DISTINCT order_id) AS No_of_Orders
FROM `target.orders`
GROUP BY Buying_time
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECU

Row	Buying_time	No_of_Orders
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

Actionable Insights:

- Looking at the data, it seems that the largest number of orders were made in the afternoon, with 38,135 orders, followed by the morning with 27,733 orders, and the night with 28,331 orders. The lowest number of orders were made during the dawn period, with 5,242 orders.
- One possible reason for this could be that people are more active during the day and have more time to shop or browse products.
- To better understand this trend, the company could conduct further analysis to determine whether there are any particular products or customer segments that are driving the higher order volumes during specific times of the day. This analysis could help the company to optimize its marketing efforts and potentially increase sales by targeting customers during peak order times.

Recommendations:

- Optimize marketing and social media campaigns: The company should analyse their marketing strategies and identify ways to promote its products during the times of the day when customers tend to place more orders. This could include targeted social media advertising campaigns or email marketing promotions during peak order times.

- Offer promotions during low-order times: The company could consider offering promotions or discounts during the dawn or night periods to encourage more orders during these times. This could help to balance out the order volumes across different times of the day.
- Optimize website and checkout process: The company should ensure that its website and checkout process are optimized for ease of use and efficiency. This could help to increase the likelihood of customers completing their orders, particularly during peak order times.

4 Evolution of E-commerce orders in the Brazil region:

4.1 Get month on month orders by states

Query:

```
SELECT customer_state,
EXTRACT(MONTH FROM order_purchase_timestamp) AS purchase_month,
COUNT(order_id) AS no_of_orders
FROM `target.orders` AS o
JOIN `target.customer` AS c
ON c.customer_id = o.customer_id
GROUP BY customer_state, purchase_month
ORDER BY no_of_orders DESC
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

4.2 Distribution of customers across the states in Brazil

Query:

```
SELECT customer_state,
COUNT(customer_id) AS No_of_customers
FROM `target.customer`
GROUP BY customer_state
ORDER BY No_of_customers DESC
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EX

Row	customer_state	No_of_customer
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

Insights:

- The query result shows the number of customers for each state, with the highest number of customers in SP (41746) and the lowest number in RR (46).
- It's worth noting that without additional information about the business, the product or service being sold, it's difficult to draw any meaningful insights or conclusions from this data alone. However, this data could be useful for identifying which states have a higher concentration of customers and may be more profitable markets for the business to target in the future.

Recommendations:

- Identify top-performing states: By analysing the state- wise sales data, businesses can identify which states have a higher concentration of customers or higher sales revenue. These states may represent more profitable markets or may indicate a greater demand for the product or service.
- Identify trends: By analysing sales data over time, businesses can identify market trends and adjust their strategies accordingly. For example, if sales are declining in a particular state, the business may need to adjust its pricing, marketing, or product offerings to remain competitive in that area/state.
- Conduct competitor analysis: By analysing sales data for competitors in the market, businesses can identify gaps in the market and potential areas for growth. This information can help businesses refine their marketing and product strategies and better target their audience.
- Performance against goals: By setting specific state wise sales goals and tracking progress against that, businesses can identify areas of improvement and adjust their strategies to achieve better results and thrive in low sales states.

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

5.1 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
  Order_value_for_2017,
  Order_value_for_2018,
  (((Order_value_for_2018 - Order_value_for_2017)/ Order_value_for_2017)* 100) As percentage_increase_in_cost_of_orders
FROM (SELECT
      SUM(IF(EXTRACT(year FROM o.order_purchase_timestamp) = 2017, p.payment_value,0)) AS Order_value_for_2017,
      SUM(IF(EXTRACT(year FROM o.order_purchase_timestamp) = 2018, p.payment_value,0)) AS Order_value_for_2018
    FROM `target.orders` AS o
    JOIN `target.payments` AS p
    ON o.order_id = p.order_id
    WHERE EXTRACT(month FROM o.order_purchase_timestamp) BETWEEN 1 AND 8)
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTI

Row	Order_value_for_2017	Order_value_for_2018	percentage_increase_in_cost_of_orders
1	3669022.1199999228	8694733.83999986...	136.97687164666226

5.2 Mean & Sum of price and freight value by customer state

Query:

```
SELECT
  c.customer_state AS cust_state,
  SUM(oit.price) AS Sum_all_price,
  AVG(oit.price) AS Mean_of_price,
  SUM(oit.freight_value) AS Sum_all_freight,
  AVG(oit.freight_value) AS Mean_of_freight
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	cust_state	Sum_all_price	Mean_of_price	Sum_all_freight	Mean_of_freight
1	AC	15982.95	173.73	3686.75	40.07
2	AL	80314.81	180.89	15914.59	35.84
3	AM	22356.84	135.5	5478.89	33.21
4	AP	13474.3	164.32	2788.5	34.01
5	BA	511349.99	134.6	100156.68	26.36
6	CE	227254.71	153.76	48351.59	32.71
7	DF	302603.94	125.77	50625.5	21.04
8	ES	275037.31	121.91	49764.6	22.06
9	GO	294591.95	126.27	53114.98	22.77
10	MA	119648.22	145.2	31523.77	38.26

Insights:

- Based on 1st query results, it appears the order value for a business in 2017 and 2018, as well as the percentage increase in the cost of orders between those two years.
- The percentage increase in the cost of orders is shown as 136.98%, indicating that the cost of orders increased by over 100% between 2017 and 2018. Overall, this table suggests that the business has experienced substantial growth in order value between 2017 and 2018, which could be a positive indicator of business success.
- Base on 2nd query results, The state with the highest total price is SP with a value of 5,202,955.05, while the state with the lowest total price is Roraima RR with a value of 7,829.43.
- The state with the highest average freight cost per order is PB with a value of 42.72, while the state with the lowest average freight cost per order is SP with a value of 15.15.

Recommendations:

- Review pricing strategy: With such a large increase in the cost of orders, it may be worth for the business to review its pricing strategy to ensure that it remains competitive while still generating a profit.
- Analyse spending patterns: Businesses can use the average spending data to analyse spending patterns within each state. For example, states with higher average spending may indicate that customers are willing to pay more for certain products or services, and businesses could adjust their offerings or marketing strategies accordingly.
- Evaluate freight costs: The data also includes information on total freight costs and average freight costs for each state. Businesses could evaluate these costs to identify potential opportunities for cost savings or process improvements, such as optimizing shipping routes or negotiating better rates with carriers.

6 Analysis on sales, freight and delivery time

6.1 Calculate days between purchasing, delivering and estimated delivery

Query:

```
SELECT order_id,
TIMESTAMP_DIFF(order_estimated_delivery_date, order_purchase_timestamp, day) AS Estimated_del
ivery_time,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS Actual_delive
ry_time,
ABS(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date, day)) AS Esti
mated_Vs_Actual_days
FROM `target.orders`
WHERE order_status='delivered'
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	order_id	Estimated_delivery_time	Actual_delivery_time	Estimated_Vs_Actual_days
1	635c894d068ac37e6e03dc54e...	32	30	1
2	3b97562c3aee8bdedcb5c2e45...	33	32	0
3	68f47f50f04c4cb6774570cfde...	31	29	1
4	276e9ec344d3bf029ff83a161c...	39	43	4
5	54e1a3c2b97fb0809da548a59...	36	40	4
6	fd04fa4105ee8045f6a0139ca5...	35	37	1
7	302bb8109d097a9fc6e9cefc5...	28	33	5
8	66057d37308e787052a32828...	32	38	6
9	19135c945c554eebfd7576c73...	33	36	2
10	4493e45e7ca1084efcd38ddeb...	33	34	0

6.2 Find time_to_delivery & diff_estimated_delivery.

(Formula for the same given below:
time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date)

Query:

```
SELECT order_id,
ABS(TIMESTAMP_DIFF(order_purchase_timestamp,order_delivered_customer_date,day)) AS Time_to
_delivery,
ABS(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day)) AS Diff_
estimated_delivery
FROM `target.orders`
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	order_id	Time_to_delivery	Diff_estimated_d
1	770d331c84e5b214bd9dc70a...	7	45
2	dabf2b0e35b423f94618bf965f...	7	44
3	8beb59392e21af5eb9547ae1a...	10	41
4	1a0b31f08d0d7e87935b819ed...	6	29
5	cec8f5f7a13e5ab934a486ec9e...	20	40
6	58527ee4726911bee84a0f42c...	10	48
7	10ed5499d1623638ee810eff1...	28	29
8	818996ea247803ddc123789f2...	9	35
9	d195cac9ccaa1394ede717d38...	10	41
10	64eeb35d3ade7fcdff9fbb1ca5...	6	41

6.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Query:

```
SELECT
  c.customer_state AS cust_state,
  ROUND(AVG(oit.freight_value),2) AS Mean_of_freight,
  ROUND(AVG(ABS(TIMESTAMP_DIFF(order_purchase_timestamp,order_delivered_customer_date,day))),2) AS Time_to_delivery,
  ROUND(AVG(ABS(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day))),2) AS Diff_estimated_delivery
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRA

Row	cust_state	Mean_of_freight	Time_to_delivery	Diff_estimated_delivery
1	AC	40.07	20.33	21.24
2	AL	35.84	23.99	12.06
3	AM	33.21	25.96	20.47
4	AP	34.01	27.75	24.58
5	BA	26.36	18.77	12.92
6	CE	32.71	20.54	14.26
7	DF	21.04	12.5	12.22
8	ES	22.06	15.19	12.09
9	GO	22.77	14.95	12.9
10	MA	38.26	21.2	12.66

6.4 Sort the data to get the following:

6.4.1 Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

A) Top 5 States with highest average freight value

Query:

```
SELECT
  c.customer_state AS cust_state,
  ROUND(AVG(oit.freight_value),2) AS Mean_of_freight
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Mean_of_freight DESC
LIMIT 5
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON

Row	cust_state	Mean_of_freight
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

B) Top 5 States with Lowest average freight value

Query:

```
SELECT
  c.customer_state AS cust_state,
  ROUND(AVG(oit.freight_value),2) AS Mean_of_freight
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Mean_of_freight ASC
LIMIT 5
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | E

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

6.4.2 Top 5 states with highest/lowest average time to delivery

A) Top 5 States with Highest Average time to delivery

Query:

```
SELECT
  c.customer_state AS cust_state,
  ROUND(AVG(ABS(TIMESTAMP_DIFF(order_purchase_timestamp,order_delivered_customer_date,day))),2) AS Time_to_delivery
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Time_to_delivery DESC
LIMIT 5
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON

Row	cust_state	Time_to_delivery
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3

B) Top 5 States with lowest Average time to delivery

Query:

```
SELECT
  c.customer_state AS cust_state,
  ROUND(AVG(ABS(TIMESTAMP_DIFF(order_purchase_timestamp,order_delivered_customer_date,day))),2) AS Time_to_delivery
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY Time_to_delivery ASC
LIMIT 5
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EX

Row	cust_state	Time_to_delivery
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

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

A) Top 5 states where delivery is really fast compared to estimated date

Query:

```
SELECT c.customer_state AS cust_state,
ROUND(AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day)),2) AS Estimated_vs_actual_delivery_days
FROM `target.orders` AS o
```

```
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
WHERE order_status='delivered'
GROUP BY c.customer_state
ORDER BY Estimated_vs_actual_delivery_days DESC
LIMIT 5
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTI

Row	cust_state	Estimated_vs_actual_delivery_days
1	AC	20.01
2	RO	19.08
3	AM	18.98
4	AP	17.44
5	RR	17.43

B) Top 5 states where delivery is not so fast compared to estimated date

Query:

```
SELECT c.customer_state AS cust_state,
ROUND(AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_custom
er_date,day)),2) AS Estimated_vs_actual_delivery_days
FROM `target.orders` AS o
JOIN `target.order_items` AS oit
ON o.order_id = oit.order_id
JOIN `target.customer` AS c
ON o.customer_id = c.customer_id
WHERE order_status='delivered'
GROUP BY c.customer_state
ORDER BY Estimated_vs_actual_delivery_days ASC
LIMIT 5
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECU

Row	cust_state	Estimated_vs_actual_delivery_days
1	AL	7.98
2	MA	9.11
3	SE	9.17
4	ES	9.77
5	BA	10.12

Insights:

- The mean freight cost ranges from as low as 15.15 in SP to as high as 42.98 in RR, suggesting that freight costs can vary significantly depending on the location of the customer.
- Time to delivery varies across customer states: The time to delivery ranges from 8.26 days in SP to 27.83 days in RR, which is more than 3 times longer. This indicates that delivery times can vary widely depending on the customer's location, and factors such as distance, transportation infrastructure, etc.
- Estimated delivery times are not always accurate: The difference between estimated and actual delivery times ranges from as low as 10.99 in SP to as high as 25.35 in RR. This suggests that estimated delivery times may not always be reliable and that customers may experience delays in receiving their shipments.

Recommendations:

- Analyse the factors that are driving the variations in freight costs and delivery times across different customer states. This could involve looking at factors such as transportation infrastructure, and distance, and identifying ways to optimize these to reduce costs and improve delivery times.
- Improve the accuracy of estimated delivery times to enhance customer satisfaction. This could involve using more online/accurate tracking and delivery systems, improving communication with customers, and implementing processes to better predict delivery times.
- By analysing historical data and identifying patterns and trends, these methods can help identify potential bottlenecks or inefficiencies in the logistics process and suggest ways to optimize them.

7 Payment type analysis:

7.1 Month over Month count of orders for different payment types

Query:

```
SELECT EXTRACT(MONTH FROM o.order_purchase_timestamp) AS purchase_month,
p.payment_type,
COUNT(DISTINCT o.order_id) AS No_of_orders
FROM `target.orders` AS o
JOIN `target.payments` AS p
ON o.order_id =p.order_id
GROUP BY purchase_month, payment_type
ORDER BY purchase_month
```

Result:

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION

Row	purchase_month	payment_type	No_of_orders
1	1	credit_card	6093
2	1	UPI	1715
3	1	voucher	337
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6582
7	2	voucher	288
8	2	debit_card	82
9	3	credit_card	7682
10	3	UPI	1942
11	3	debit_card	109
12	3	voucher	395
13	4	voucher	353
14	4	credit_card	7276
15	4	UPI	1783
16	4	debit_card	124
17	5	credit_card	8308
18	5	UPI	2035
19	5	debit_card	81
20	5	voucher	374

7.2 Count of orders based on the no. of payment installments

Query:

```
SELECT p.payment_installments,
COUNT(o.order_id) AS No_of_orders
FROM `target.orders` AS o
JOIN `target.payments` AS p
ON o.order_id =p.order_id
GROUP BY payment_installments
ORDER BY payment_installments
```

Result:

Query results

JOB INFORMATION | RESULTS | JS

Row	payment_installments	No_of_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:

- Credit card and UPI are the most popular payment types for online orders. During the months of October and November, the highest number of orders were placed using these payment methods.
- Vouchers are a popular payment method during the months of November and December, indicating that customers may be more inclined to use discounts or promotions during the holiday season.
- Debit card usage is relatively low throughout the year, with the exception of the month of December. This may suggest that customers prefer to use credit cards or other payment methods for online orders.

Recommendations:

- Offer a wider range of payment options to cater to different customer preferences. This could include popular payment methods like credit cards and UPI, as well as less popular options like debit cards or online wallets.
- Companies should consider offering more flexible payment options, such as the ability to pay in installments over a longer period of time or the ability to delay payments until a later date. This may help to attract more customers and improve the overall customer experience by offering more flexibility in the payment process.