

Target Case Study

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

1A. Data type of all columns in the "customers" table.

```
select column_name, data_type  
from scaler-dsml-sql-nawaz.Targetcasestudy.INFORMATION_SCHEMA.COLUMNS  
where table_name = 'customers'
```

Query results		SAVE RESULTS	OPEN IN	▼	
JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	column_name	data_type			
1	customer_id	STRING			
2	customer_unique_id	STRING			
3	customer_zip_code_prefix	INT64			
4	customer_city	STRING			

Insight:-

1. Most columns in the "customers" table are stored as strings (VARCHAR), except for customer_zip_code_prefix, which is an integer.
2. This suggests that customer-related data is primarily textual, and numerical or date-related values might also be stored as strings.

Recommendations:-

1. Ensure that numerical and date fields are stored in appropriate data types for better performance and easier analysis.
2. Convert relevant string fields into proper formats (e.g., dates and numeric values) where needed.
3. Validate data consistency to prevent errors due to incorrect data types in calculations or queries.

1B. Get the time range between which the orders were placed.

```
select min(order_purchase_timestamp) as earliest_order_date,  
       max(order_purchase_timestamp) as latest_order_date  
from `Targetcasestudy.orders`
```

Query results		SAVE RESULTS	OPEN IN	▼	
JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	earliest_order_date	latest_order_date			
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC			

Insight:-

1. The orders span a period of approximately 2 years (or 773 days), providing a good dataset for trend analysis.
2. This time range allows for evaluating seasonal trends, peak order periods, and overall business growth.

Recommendations:

1. Analyze order trends over time to identify peak seasons and adjust marketing strategies accordingly.
2. Compare sales performance across different time frames to identify growth patterns.
3. Use this data to forecast demand and optimize inventory and logistics planning.

1C. Count the Cities & States of customers who ordered during the given period.

```
select count(distinct customer_city) as unique_cities,  
       count(distinct customer_state) as unique_states  
from `Targetcasestudy.customers`
```

Query results				SAVE RESULTS	OPEN IN	
JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH	
Row	unique_cities	unique_states				
1	4119	27				

Results per page: 50 ▾ 1 – 1 of 1 |< < > >|

Insights:-

- 1.The dataset includes customers from 27 states and 801 cities, indicating a broad geographic reach. This diversity suggests that the business serves a wide range of customers across multiple regions. Identifying high-order volume cities can help in targeting key markets more effectively.

Recommendations:

1. Focus marketing efforts on cities with high customer activity.
2. Expand logistics and supply chain support in high-order regions to improve efficiency.
3. Identify underperforming states/cities and explore strategies to boost engagement, such as promotions or localized campaigns.

2. In-depth Exploration:

2A. Is there a growing trend in the no. of orders placed over the past years?

```
select extract(Year from order_purchase_timestamp) as order_year,  
       count(order_id) as total_orders  
from `Targetcasestudy.orders`  
group by order_year
```

```
order by order_year
```

Query results

SAVE RESULTS ▾ OPEN IN ▾

JOB INFORMATION		RESULTS		CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	order_year	total_orders					
1	2016	329					
2	2017	45101					
3	2018	54011					

Results per page: 50 ▾ 1 – 3 of 3 |< < > >|

Insights:-

1. The query tracks the yearly trend in order placements.
2. It helps determine if there is a growth or decline in orders over time.
3. A consistent increase indicates business growth, while fluctuations may suggest seasonal or external influences.

Recommendations:

1. If there is growth, invest in scaling operations and inventory.
2. If orders fluctuate, analyze external factors like market trends or promotions.
3. In case of decline, revisit marketing strategies and customer engagement.
4. Use this trend data to forecast future demand and plan resources accordingly.

2B. Can we see some kind of monthly seasonality in terms of the no. of orders being placed

With cte AS

```
(  
    select *,  
    extract(DATE from order_purchase_timestamp) as order_date,  
    Extract(YEAR from order_purchase_timestamp) as order_year,  
    extract(MONTH from order_purchase_timestamp) as order_month,  
    FROM `Targetcasestudy.orders`  
)  
select  
    order_month,  
    order_year,  
    count(order_id) as TOTAL_ORDER  
from cte  
group by order_year, order_month
```

Query results

SAVE RESULTS
OPEN IN

Row	order_month	order_year	TOTAL_ORDER
1	11	2017	7544
2	12	2017	5673
3	2	2018	6728
4	4	2017	2404
5	7	2017	4026
6	5	2018	6873
7	10	2017	4631
8	1	2018	7269
9	6	2017	3245
10	9	2017	4285

Results per page: 50 | 1 - 25 of 25 | < < > >|

Insights:-

- 1.The query analyzes monthly seasonality in order placements.
2. It groups orders by month and year to identify trends over time.
3. Helps in understanding peak and low-demand months.

Recommendations:

1. Use seasonal trends to plan inventory and stock levels.
2. Align marketing campaigns with high-demand months.
3. Introduce promotions or discounts during low-demand periods.
4. Compare yearly trends to predict future demand fluctuations.

2C. 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**

```

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 order_time_interval,
count(*) as order_count
from `Targetcasestudy.orders`
group by order_time_interval
order by order_count desc

```

Query results

SAVE RESULTS ▾ OPEN IN ▾

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	order_time_interval		order_count			
1	AFTERNOON		38135			
2	NIGHT		28331			
3	MORNING		27733			
4	DAWN		5242			

Results per page: 50 ▾ 1 – 4 of 4 |< < > >|

Insights:-

1. The query categorizes order purchase timestamps into four time slots: Dawn, Morning, Afternoon, and Night.
2. This classification helps analyze customer purchasing behavior based on time.
3. It can identify peak shopping hours and trends in order placement.

Recommendations:

1. Use this data to optimize marketing strategies for different time slots.
2. Adjust inventory and staffing based on peak order times.
3. Enhance customer engagement by offering time-based discounts or promotions.
4. Further refine the time slots if more granularity is needed for better insights.
3. Evolution of E-commerce orders in the Brazil region:

3A. Get the month on month no. of orders placed in each state.

```
select extract(month from o.order_purchase_timestamp) as order_month, c.customer_state, count(*) as number_of_order
from `Targetcasestudy.orders` as o
join `Targetcasestudy.customers` as c
on o.customer_id = c.customer_id
group by order_month, c.customer_state
order by order_month, c.customer_state
```

Query results

SAVE RESULTS ▾ OPEN IN ▾

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	order_month	customer_state		number_of_order		
1	1	AC		8		
2	1	AL		39		
3	1	AM		12		
4	1	AP		11		
5	1	BA		264		
6	1	CE		99		
7	1	DF		151		
8	1	ES		159		
9	1	GO		164		
10	1	MA		66		

Results per page: 50 ▾ 1 – 50 of 322 |< < > >|

Insights:-

1. Order Trends by Month: The query extracts the number of e-commerce orders placed per month in different Brazilian states.
2. State-wise Order Volume: It shows the distribution of orders across various states.
3. Customer-State Mapping: By joining customer and order data, the query ensures accurate regional analysis.
4. Seasonality Check: The data can help identify peak and low sales months.

Recommendations:

1. Identify High-Performing States: Focus on states with high order volumes for targeted marketing.
2. Improve Logistics in Low-Order States: Strengthen supply chain and promotions in underperforming regions.
3. Seasonal Campaigns: Leverage peak order months for discounts and advertising.
4. Further Analysis: Extend to yearly trends and category-wise sales for deeper insights.

3B. How are the customers distributed across all the states?

```
select customer_state, count(distinct customer_id) as total_customers
from `Targetcasestudy.customers`
group by customer_state
order by total_customers desc
```

Query results

SAVE RESULTS ▾ OPEN IN ▾

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state	total_customers				
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:

1. Helps identify order trends over months.
2. Shows which states have the most orders.
3. Highlights peak and low months for sales.
4. Useful for marketing, inventory, and logistics planning.

Recommendations:

1. Optimize inventory in peak months to meet demand.
2. Target high-order states with promotions and better delivery options.

3. Boost marketing in low-order months to increase sales.
4. Improve logistics in states with high potential but low orders.
5. Analyze trends to plan seasonal discounts and campaigns.

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

4A. 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.

```
WITH payments_2017 AS (
  SELECT ROUND(SUM(p.payment_value), 2) AS total_2017
  FROM `Targetcasestudy.orders` o
  JOIN `Targetcasestudy.payments` p ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017
  AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
),
payments_2018 AS (
  SELECT ROUND(SUM(p.payment_value), 2) AS total_2018
  FROM `Targetcasestudy.orders` o
  JOIN `Targetcasestudy.payments` p ON o.order_id = p.order_id
  WHERE EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018
  AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
)
SELECT
  ROUND(((payments_2018.total_2018 - payments_2017.total_2017) / payments_2017.total_2017) * 100, 2) AS percentage_increase
FROM payments_2017, payments_2018;
```

Query results						
		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	percentage_increase					
1	136.98					

Results per page: 50 ▾ 1 – 1 of 1 |< < > >|

Insights:-

1. From January to August, there was a noticeable increase in payment values from 2017 to 2018.
2. The exact percentage increase can be calculated using your SQL query.

Recommendation:-

- 1.Investigate what factors contributed to the increase in payment values between these periods.
- 2.Consider replicating successful strategies or campaigns that led to the increase.
- 3.Gather and analyze customer feedback and market conditions during these months to identify key drivers of growth.
- 4.Regularly monitor performance to maintain growth and adapt strategies as needed.

4B. Calculate the Total & Average value of order price for each state.

```
select customer_state,
       ROUND(sum(p.payment_value),2) as total_order_price,
       round(avg(p.payment_value),2) as average_order_price
  from `Targetcasestudy.payments` as p
 join `Targetcasestudy.orders` as o
    on p.order_id = o.order_id
 join `Targetcasestudy.customers` as c
    on o.customer_id = c.customer_id
 group by customer_state
 order by total_order_price desc
```

Query results

SAVE RESULTS ▾ OPEN IN ▾

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state	total_order_price	average_order_price			
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			

Load more

Results per page: 50 ▾ 1 – 27 of 27 | < < > >|

Insight:-

- 1.The total and average order prices vary significantly by customer state.
- 2.States with higher total order prices indicate a larger volume of transactions or higher-value orders.

Recommendation:-

- 1.Focus marketing efforts on states with the highest total order prices to maximize revenue.
- 2.Identify common characteristics of customers in high-value states and target similar demographics in other states.

3.Offer promotions or incentives in states with lower total order prices to increase sales.

4.Regularly analyze state-wise data to track performance and adjust strategies as needed.

4C. Calculate the Total & Average value of order freight for each state.

```
select customer_state as state,
       round(sum(oi.freight_value),2) as total_freight,
       round(avg(oi.freight_value),2) as average_freight
  from `Targetcasestudy.orders` as o
  join `Targetcasestudy.order_items` as oi
    on o.order_id = oi.order_id
  join `Targetcasestudy.customers` as c
    on c.customer_id = o.customer_id
 group by state
 order by total_freight desc
```

Query results

RESULTS

Row	state	total_freight	average_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

Results per page: 50 ▾ 1 – 27 of 27 | < < > >|

Insight:-

1.Freight costs vary significantly by state, with some states incurring much higher total and average freight values.

2.States with higher total freight values might be associated with more significant shipping volumes or greater distances.

Recommendation:-

1.Analyze the factors contributing to high freight costs in certain states, such as shipping distances, order volumes, and logistics.

2.Optimize shipping routes and consider negotiating better rates with carriers for high-cost states.

3.Introduce incentives or promotions for customers in states with lower total freight costs to increase order volumes.

4. Regularly review and adjust shipping strategies to ensure cost-efficiency and customer satisfaction.

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

5.1 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_timestamp`
- `diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date`

```
select order_id,
       date_diff(date(order_delivered_customer_date),
                  date(order_purchase_timestamp), day) as delivery_time,
       date_diff(date(order_estimated_delivery_date),
                  date(order_delivered_customer_date), day) as diff_estimated_delivery
  from `Targetcasestudy.orders`
```

Query results

SAVE RESULTS ▾ OPEN IN ▾

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	order_id	delivery_time	diff_estimated_delivery			
1	1950d777989f6a877539f5379...	30	-12			
2	2c45c33d2f9cb8ff8b1c86cc28...	31	29			
3	65d1e226dfaeb8cdc42f66542...	36	17			
4	635c894d068ac37e6e03dc54e...	31	2			
5	3b97562c3aaee8bbedcb5c2e45...	33	1			
6	6bf47f50f04c4cb6774570cfde...	30	2			
7	276e9ec344d3bf029f83a161c...	44	-4			
8	54e1a3c2b97fb0809da548a59...	41	-4			
9	fd04fa4105ee8045f6a0139ca5...	37	-1			
10	302bb8109d097a9fc6e9cef5...	34	-5			

Results per page: 50 ▾ 1 – 50 of 99441 |< < > >|

Insight:-

1. The `delivery_time` column provides the actual delivery time for each order.
2. The `diff_estimated_delivery` column shows the difference between the actual delivery date and the estimated delivery date.
3. By examining these columns, we can identify how accurately the estimated delivery dates are being met and analyze delivery performance.

Recommendation:-

- 1.Analyze the delivery_time data to identify patterns or factors affecting delivery times. This can help in optimizing the delivery process.
- 2.If there is a significant diff_estimated_delivery, work on improving the accuracy of estimated delivery dates. This could involve better coordination with logistics partners and more precise estimation algorithms.
- 3.Consider implementing a feedback loop with customers regarding delivery experiences to identify areas for improvement.
- 4.Regularly monitor and adjust delivery strategies to ensure timely delivery and enhance customer satisfaction.

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

```
select high.customer_state as high_state,
       high.average_freight_value as high_avg_freight,
       low.customer_state as low_state,
       low.average_freight_value as low_avg_freight
  from
  (
    select
      c.customer_state,
      round(avg(p.freight_value),2) as average_freight_value,
      row_number() over(order by
        (round(avg(p.freight_value),2))desc) as rowval1
     from `Targetcasestudy.orders` as o
     join `Targetcasestudy.order_items` as p
       on o.order_id = p.order_id
     join `Targetcasestudy.customers` as c
       on o.customer_id = c.customer_id
    group by
      c.customer_state
    order by
      average_freight_value desc
    limit 5
  ) as high
  join
  (
    select c.customer_state,
           round(avg(p.freight_value),2) as average_freight_value,
           row_number() over(order by (round(avg(p.freight_value),2))) as rowval2
      from `Targetcasestudy.orders` as o
      join `Targetcasestudy.order_items` as p
        on o.order_id = p.order_id
      join `Targetcasestudy.customers` as c
```

```

on o.customer_id = c.customer_id
group by c.customer_state
order by average_freight_value
limit 5
) as low
on high.rowval1 = low.rowval2

```

Query results

SAVE RESULTS OPEN IN ▼

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	high_state		high_avg_freight	low_state		low_avg_freight
1	RR		42.98	SP		15.15
2	PB		42.72	PR		20.53
3	RO		41.07	MG		20.63
4	AC		40.07	RJ		20.96
5	PI		39.15	DF		21.04

Results per page: 50 ▾ 1 - 5 of 5 |< < > >|>

Insight:-

- 1.The query identifies the states with the highest and lowest average freight values.
- 2.It compares the top 5 states with the highest average freight values to the 5 states with the lowest average freight values.

Recommendation:-

- 1.Investigate the factors contributing to the high average freight values in certain states. This might include shipping distances, volume of orders, or logistical challenges.
- 2.Consider optimizing shipping strategies for high-cost states to reduce freight expenses.
- 3.Identify best practices from states with low average freight values and apply them to high-cost states.
- 4.Negotiate better rates with shipping carriers for states with higher average freight values to reduce costs.
- 5.Monitor freight costs regularly and adjust strategies to maintain cost-efficiency and customer satisfaction.

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

```

with cte as
(
  select
    c.customer_state,
    round(avg(t1.delivery_time),2) as avg_delivery_time
  from
  (
    select
      *,

```

```

    timestamp_diff(order_delivered_customer_date, order_purchase_timestamp, day) as delivery_time,
    from `Targetcasestudy.orders`  

    where order_status = 'delivered' and order_delivered_carrier_date is not null  

    order by order_purchase_timestamp  

) as t1  

join `Targetcasestudy.customers` as c  

on t1.customer_id = c.customer_id  

group by c.customer_state  

order by avg_delivery_time  

)  

select  

c1.customer_state as low_state,  

c1.avg_delivery_time as low_avg_delivery_time,  

from  

(  

select  

*,  

row_number() over (order by cte.avg_delivery_time desc) as rowval2  

from  

cte  

order by rowval2  

) as c2  

join  

(  

select  

*,  

row_number() over (order by cte.avg_delivery_time) as rowval1  

from  

cte  

order by rowval1  

)as c1  

on c1.rowval1 = c2.rowval2  

limit 5

```

Query results

RESULTS

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	low_state			low_avg_delivery_time		
1	SP			8.3		
2	PR			11.53		
3	MG			11.54		
4	DF			12.51		
5	SC			14.48		

Insight:-

- 1.The query identifies states with the lowest average delivery times.

2.The avg_delivery_time metric provides insight into how quickly orders are delivered in each state.

Recommendation:-

- 1.Analyze the factors contributing to the low average delivery times in these states. This might include efficient logistics, shorter shipping distances, or better carrier performance.
- 2.Use the best practices from these states to improve delivery times in states with higher average delivery times.
- 3.Consider partnering with local delivery services in states with higher average delivery times to improve efficiency.
- 4.Continuously monitor delivery performance and gather customer feedback to enhance overall delivery experience and satisfaction.

5D. 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.

```
WITH delivery_speed as (
  select
    c.customer_state,
    avg(date_diff(o.order_delivered_customer_date,
      o.order_estimated_delivery_date, DAY)) as avg_delivery_speed,
    row_number() over (order by
      avg(date_diff(o.order_delivered_customer_date,
        o.order_estimated_delivery_date, DAY))) as rank_fastest
  from `Targetcasestudy.orders` as o
  join `Targetcasestudy.customers` as c
  on o.customer_id = c.customer_id
  where o.order_delivered_customer_date is not null and
    o.order_estimated_delivery_date is not null
  group by c.customer_state
)
select customer_state, avg_delivery_speed
from delivery_speed
where rank_fastest <= 5
order by avg_delivery_speed
```

Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	customer_state			avg_delivery_speed		
1	AC			-19.762500000...		
2	RO			-19.1316872427...		
3	AP			-18.7313432835...		
4	AM			-18.6068965517...		
5	RR			-16.4146341463...		

Results per page: 50 ▾ 1 – 5 of 5 |< < > >|

Insight:-

- 1.This query identifies the top 5 states with the fastest average delivery speeds, based on the difference between the actual delivery date and the estimated delivery date.
- 2.States with lower average delivery speeds indicate better performance in meeting or exceeding estimated delivery dates.

Recommendation:-

- 1.Analyze the logistics and delivery processes in the states with the fastest average delivery speeds to identify best practices.
- 2.Apply these best practices to states with slower delivery speeds to improve overall delivery performance.
- 3.Consider factors such as carrier efficiency, local infrastructure, and warehouse locations that might be contributing to faster deliveries in certain states.
- 4.Regularly monitor delivery speed data to ensure continued improvement and adjust strategies as needed for optimal performance.

6.Analysis based on the payments:

6A. Find the month on month no. of orders placed using different payment types.

```
select extract(month from o.order_purchase_timestamp) as month_no,
format_date("%B", o.order_purchase_timestamp) as month,
count(distinct o.order_id) as order_count,
p.payment_type,
from `Targetcasestudy.orders` AS o
join `Targetcasestudy.payments` as p
on o.order_id = p.order_id
group by 1,2,4
order by 1
```

Query results

SAVE RESULTS OPEN IN

JOB INFORMATION		RESULTS		CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	month_no	month			order_count	payment_type	
1	1	January			6093	credit_card	
2	1	January			1715	UPI	
3	1	January			337	voucher	
4	1	January			118	debit_card	
5	2	February			1723	UPI	
6	2	February			6582	credit_card	
7	2	February			288	voucher	
8	2	February			82	debit_card	
9	3	March			7682	credit_card	
10	3	March			1942	UPI	

Results per page: 50 ▾ 1 – 50 of 50 |< < > >|

Insights:-

- Monthly Sales:** Track the number of orders each month to see trends.
- Payment Types:** Identify which payment methods customers prefer.
- Order Count:** Monitor the total number of orders monthly.

Recommendations:-

- Marketing:** Focus on peak months for promotions.
- Payments:** Highlight popular payment methods.
- Inventory:** Ensure stock is adequate during high sales periods.
- Engagement:** Boost sales in slower months with discounts or offers.

6B. Find the no. of orders placed on the basis of the payment installments that have been paid.

```
SELECT Payment_installments, count(distinct order_id) as order_count
from `Targetcasestudy.payments`
group by payment_installments
```

Query results

SAVE RESULTS OPEN IN

JOB INFORMATION		RESULTS		CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	Payment_installment	order_count					
1	0	2					
2	1	49060					
3	2	12389					
4	3	10443					
5	4	7088					
6	5	5234					
7	6	3916					
8	7	1623					
9	8	4253					
10	9	644					

Results per page: 50 ▾ 1 – 24 of 24 |< < > >|

Insight:-

- 1.This query provides a count of distinct orders grouped by the number of payment installments.
- 2.The data reveals customer preferences for different installment options when making payments.

Recommendation:

- 1.Analyze the popularity of various installment plans to understand customer preferences and financial behavior.
- 2.Consider promoting or offering incentives for the most popular installment plans to encourage more orders.
- 3.If certain installment plans have low uptake, investigate potential barriers and consider adjusting the options to better meet customer needs.
- 4.Use the insights to optimize payment installment options and improve overall customer satisfaction.