



# TradeGuard: SQL-Based Detection of Market Abuses

Presented by Sourav Pattanayak

# Table of Contents



- Introduction to Market Abuse
- Common Market Abuse Types
- The Analyst's Challenge
- The Datasets
- Business Questions
- Key Findings
- Conclusion and Recommendations
- Future Work



## Types of Market Abuse

- **Insider Trading:** Trading based on non-public, confidential information to gain an unfair advantage, sometimes the 'Insider' can be someone who already works in that organization.
- **Wash Trading:** Buying and selling the same stock simultaneously to create fake market activity.
- **Pump and Dump:** Spreading false information to inflate a stock's price, then selling it at a profit, leaving others with losses.
- **Spoofing:** Placing fake orders with the intent to cancel them before they are executed. The aim is to trick other traders into thinking there is higher demand or supply, thereby manipulating prices.
- **Front Running:** Trading ahead of a large order to benefit from the price movement it creates.
- **Layering:** Placing multiple fake orders to mislead the market about demand or supply.



## The Analyst's Challenge

- **Context:** As a Data Analyst at **EquiAnalytics Inc.**, you were tasked by the Risk Management team to investigate unusual trading patterns that could indicate market abuse. The goal was to identify any signs of manipulation in stock transactions and ensure regulatory compliance.
- **Challenge:** You were provided with three datasets: *surveillance\_optimization*, *regulatory\_enquiries*, and *market\_manipulation\_data*. Your job was to clean and analyze these datasets to uncover hidden patterns of abuse and provide actionable insights.
- **Approach:** Using SQL, you have to systematically clean the data, perform aggregations and calculations to detect anomalies, and analyze user behaviors to pinpoint potential manipulators, risks and areas to focus.



## The Datasets

- **Dataset 1:** Surveillance Optimization
- **Description:** Contains transactional data including transaction IDs, stock IDs, transaction prices, volumes, and dates.
- **Key Fields:** *Transaction\_ID, Stock, Transaction\_Price, Volume, Transaction\_Date*
- **Dataset 1:** Regulatory Enquiries
- **Description:** Logs of regulatory enquiries made by users, capturing details such as enquiry date, user ID, and enquiry price.
- **Key Fields:** *Enquiry\_ID, User\_ID, Enquiry\_Date, Enquiry\_Price*
- **Dataset 3:** Market Manipulation Data
- **Description:** Contains detailed records of trading activities that may be indicative of market manipulation
- **Key Fields:** *Transaction\_ID, User\_ID, Stock, Order\_Price, Order\_Volume, Date\_of\_Manipulation, Order\_Type*

# Queries with Easy Complexity





## Business Question-1

- Find the Total Transaction Value for Each Stock in the surveillance Data.



```
SELECT Stock_ID,  
ROUND(SUM(Transaction_Price * Volume), 2)  
AS Total_Transaction_Value  
FROM surveillance_optimization  
GROUP BY Stock_ID;
```

Stock_ID	Total_Transaction_Value
SID097	9351803.82
SID004	10809560.42
SID072	11726417.22
SID028	2457227.75
SID021	6085079.6



## Business Question-2

- Identify the top three manipulated Stocks with the Highest Average Order Price.



```
SELECT Stock_Name, ROUND(AVG(Order_Price), 2)
AS Avg_Order_Price
FROM market_manipulation
GROUP BY Stock_Name
ORDER BY Avg_Order_Price DESC
LIMIT 3;
```

Stock_Name	Avg_Order_Price
JPMorgan Chase	343.32
Berkshire Hathaway	334.69
Sony Group Corp.	327.5



## Business Question-3

- Find the Top five Enquiry\_Type and the users who requested it.



```
WITH UserEnquiries AS (
SELECT User_ID, Enquiry_Type, COUNT(*) AS Enquiry_Count
FROM regulatory_enquiries
GROUP BY User_ID, Enquiry_Type)
SELECT User_ID, Enquiry_Type, Enquiry_Count
FROM UserEnquiries
ORDER BY Enquiry_Count DESC
LIMIT 5;
```

User_ID	Enquiry_Type	Enquiry_Count
UID016	Fraudulent Reporting	4
UID024	Front Running	4
UID136	Fraudulent Reporting	3
UID003	Market Manipulation	3
UID137	Insider Trading	3



## Business Question-4

- Find Users Who Made More Than one Regulatory Enquiries in a single month.



```
SELECT User_ID, YEAR(Enquiry_Date) AS Year,  
MONTHNAME(Enquiry_Date) AS Month, COUNT(*)  
AS Enquiry_Count  
FROM regulatory_enquiries  
GROUP BY User_ID, YEAR(Enquiry_Date),  
MONTHNAME(Enquiry_Date)  
HAVING COUNT(*) > 1;
```

User_ID	Year	Month	Enquiry_Count
UID155	2020	February	2
UID138	2022	May	2
UID149	2023	February	2
UID037	2021	October	2
UID171	2023	February	2
UID189	2021	March	2
UID136	2020	July	2



## Business Question-5

- Write a SQL query to retrieve a list of users who have both conducted a transaction and made a corresponding enquiry having regulatory volume greater than 9000.



```
SELECT DISTINCT so.User_ID
FROM surveillance_optimization AS so
INNER JOIN regulatory_enquiries AS re
ON so.User_ID = re.User_ID
WHERE so.Volume >= 9000;
```

User_ID
UID178
UID005
UID102
UID106
UID097
UID078
UID069

## Business Question-6

- Find the total volume of orders for each stock with a corresponding regulatory enquiry.

```
SELECT mm.Stock_Name, SUM(mm.Order_Volume)
AS Total_Order_Volume
FROM market_manipulation AS mm
INNER JOIN regulatory_enquiries AS re
ON mm.Stock_ID = re.Stock_ID
GROUP BY mm.Stock_Name
ORDER BY SUM(mm.Order_Volume) DESC;
```

Stock_Name	Total_Order_Volume
Chevron Corp.	430154
Procter & Gamble	415464
Tesla Inc.	393859
Johnson & Johnson	362864
Samsung Electronics	356795
NVIDIA Corp.	352707
FlynnMobil	352388

# Queries with intermediate Complexity



# Business Question-7

- Which are the top 5 stocks with the highest positive price difference in average transaction prices between two consecutive months? Provide the stock names, the two consecutive months, and the corresponding price difference.



```
WITH MonthlyAvg AS (SELECT Stock_Name,
DATE_FORMAT(Transaction_Date, '%Y-%m')
AS Month,
AVG(Transaction_Price) AS
Avg_Price FROM surveillance_optimization
GROUP BY Stock_Name, DATE_FORMAT(Transaction_Date, '%Y-%m')),  
  
PriceDiff AS (SELECT Stock_Name, Month, Avg_Price,
LAG(Month) OVER (PARTITION BY Stock_Name ORDER BY Month)
AS Prev_Month,
Avg_Price - LAG(Avg_Price) OVER (PARTITION BY Stock_Name ORDER BY Month)
AS Price_Difference
FROM MonthlyAvg),  
  
RankedStocks AS (SELECT Stock_Name, Month, Prev_Month, Price_Difference,
ROW_NUMBER() OVER (ORDER BY Price_Difference DESC)
AS Price_Rank FROM PriceDiff
WHERE Price_Difference > 0)  
  
SELECT Stock_Name, Prev_Month AS Previous_Month,
Month AS Current_Month, Price_Difference FROM RankedStocks WHERE Price_Rank ≤ 5;
```

Stock_Name	Previous_Month	Current_Month	Price_Difference
JPMorgan Chase	2020-02	2022-04	409.54
Toyota Motor Corp.	2020-10	2022-03	407.87
Comcast Corp.	2022-04	2022-08	406.68
Walmart Inc.	2023-08	2023-09	400.75
Eli Lilly and Co.	2020-02	2020-04	393.22999999999996

## Business Question-8

➤ Give a count of the different Price Ranges for Each Stock.



```
SELECT Stock_Name,  
CASE WHEN Order_Price < 50 THEN '0-50'  
WHEN Order_Price BETWEEN 50 AND 100 THEN '50-100'  
ELSE '100+' END AS Price_Range,  
COUNT(*) AS Range_Count  
FROM market_manipulation  
GROUP BY Stock_Name, Price_Range  
ORDER BY Stock_Name, Range_Count DESC;
```

Stock_Name	Price_Range	Range_Count
Abbott Laboratories	100+	12
Abbott Laboratories	50-100	1
Abbott Laboratories	0-50	1
Alpha Corp	100+	5
Alpha Corp	50-100	1
Alphabet Inc.	100+	12
Alphabet Inc.	0-50	2

# Business Question-9

- Detect users who have experienced more than three significant fluctuations in enquiry prices. A significant fluctuation is defined as a change in price that exceeds 10 units compared to the previous enquiry price.

```
SELECT User_ID, COUNT(*) AS Fluctuation_Count
FROM (SELECT User_ID, Enquiry_Price,
LAG(Enquiry_Price) OVER (PARTITION BY User_ID ORDER BY Enquiry_Date)
AS Prev_Enquiry_Price
FROM regulatory_enquiries
) AS PriceDiff
WHERE ABS(Enquiry_Price - Prev_Enquiry_Price) > 10
GROUP BY User_ID
HAVING COUNT(*) > 3;
```

User_ID	Fluctuation_Count
UID003	4
UID016	4
UID024	6
UID029	5
UID046	4
UID057	4
UID102	4

# Business Question-10

- Write a query to find the top 5 entities with the highest average order volume, excluding any extreme values. Calculate the average order volume for each entity, rounding the result to two decimal places.

```
WITH Stats AS (SELECT Stock_Name,
AVG(Order_Volume) AS Avg_Volume,
STDDEV(Order_Volume) AS StdDev_Volume
FROM market_manipulation
GROUP BY Stock_Name)

SELECT mm.Stock_Name, ROUND(AVG(mm.Order_Volume),2)
AS Avg_Order_Volume_Excluding_Outliers
FROM market_manipulation AS mm
JOIN Stats AS s ON mm.Stock_Name = s.Stock_Name
WHERE mm.Order_Volume BETWEEN s.Avg_Volume - 2 * s.StdDev_Volume
AND s.Avg_Volume + 2 * s.StdDev_Volume
GROUP BY mm.Stock_Name
ORDER BY Avg_Order_Volume_Excluding_Outliers DESC
LIMIT 5;
```

Stock_Name	Avg_Order_Volume_Excluding_Outliers
Sony Group Corp.	7510.40
NVIDIA Corp.	6440.00
JPMorgan Chase	6393.63
Comcast Corp.	6195.09
Microsoft Corp.	6174.67

# Business Question-11

➤ Identify the Stock with the Maximum Percentage Increase in Price Over a Year.

```
WITH YearlyPrices AS (SELECT Stock_Name,
YEAR(Transaction_Date) AS Year,
FIRST_VALUE(Transaction_Price)
OVER (PARTITION BY Stock_Name, YEAR(Transaction_Date))
ORDER BY Transaction_Date) AS Start_Price,
LAST_VALUE(Transaction_Price)
OVER (PARTITION BY Stock_Name, YEAR(Transaction_Date))
ORDER BY Transaction_Date
RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS End_Price
FROM surveillance_optimization)

SELECT Stock_Name, Year,
ROUND((End_Price - Start_Price) / Start_Price * 100, 2)
AS Percentage_Increase
FROM YearlyPrices
ORDER BY Percentage_Increase DESC
LIMIT 1;
```

Stock_Name	Year	Percentage_Increase
Microsoft Corp.	2021	1948.25

# Business Question-12

➤ Find Stocks with a Price Drop of More Than 90% Compared to the Previous Transaction.



```
SELECT Transaction_ID, Stock_Name, Transaction_Price, Prev_Price
FROM (SELECT Transaction_ID, Stock_Name, Transaction_Price,
LAG(Transaction_Price)
OVER (PARTITION BY Stock_Name ORDER BY Transaction_Date)
AS Prev_Price
FROM surveillance_optimization) AS Subquery
WHERE Prev_Price IS NOT NULL
AND (Transaction_Price / Prev_Price) < 0.1;
```

Transaction_ID	Stock_Name	Transaction_Price	Prev_Price
TID00297	Apple Inc.	30.86	447.69
TID00156	Coca-Cola Co.	46.91	469.69
TID00360	Eli Lilly and Co.	30.85	344.53
TID00246	Eli Lilly and Co.	25.71	372.46
TID00301	Honda Motor Co.	27.62	413.68
TID00424	Intel Corp.	25.65	445.45
TID00179	Mastercard Inc	29.35	305.77

## Business Question-13

- Retrieve Orders with Manipulated Prices Higher Than the Average Manipulated Price for Each Stock.

```
SELECT Stock_Name, Order_Price  
FROM market_manipulation AS mm  
WHERE Order_Price >  
(SELECT AVG(Order_Price)FROM market_manipulation  
WHERE Stock_Name = mm.Stock_Name);
```

Stock_Name	Order_Price
Tesla Inc.	420.61
Visa Inc.	396.86
Comcast Corp.	417.18
NVIDIA Corp.	211.49
Pfizer Inc.	460.85
Home Depot Inc.	360.53
Walmart Inc.	440.67

## Business Question-14

- Find the stocks with the top 5 highest percentages of total transactions and their corresponding percentages.



```
SELECT Stock_Name,  
ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*))  
FROM surveillance_optimization),2)  
AS Percentage_Of_Total_Transactions  
FROM surveillance_optimization  
GROUP BY Stock_Name  
ORDER BY ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*))  
FROM surveillance_optimization),2) DESC  
LIMIT 5 ;
```

Stock_Name	Percentage_Of_Total_Transactions
PepsiCo Inc.	4.67
Abbott Laboratories	4.67
Alpha Corp	4.00
Gamma Inc	3.78
Johnson & Johnson	3.56

# Business Question-15

- Determine the top 3 Most Common market manipulation Type and Its Average Price for Each Stock.

```
WITH CommonOrderType AS (SELECT Order_Price, Stock_Name, Manipulation_Type,  
COUNT(*) AS manipulation_Type_Count,  
ROW_NUMBER() OVER (PARTITION BY Stock_Name ORDER BY COUNT(*) DESC) AS RowNum  
FROM market_manipulation  
GROUP BY Stock_Name, Manipulation_Type, Order_Price)  
  
SELECT Manipulation_Type, ROUND(AVG(Order_Price), 2) AS Avg_Price  
FROM CommonOrderType  
WHERE RowNum = 1  
GROUP BY Manipulation_Type  
ORDER BY Avg_Price DESC  
LIMIT 3;
```

Manipulation_Type	Avg_Price
Spoofing	307.74
Pump and Dump	272.87
Wash Trading	253.6

# Queries with Advanced Complexity



# Business Question-16

➤ Calculate the Yearly original Price Increase and Identify Stocks with Over 500% Growth.

```
WITH YearlyPrices AS (SELECT Stock_Name,
YEAR(Transaction_Date) AS Year,
FIRST_VALUE(Transaction_Price) OVER (PARTITION BY Stock_Name, YEAR(Transaction_Date)
ORDER BY Transaction_Date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS Start_Price,
LAST_VALUE(Transaction_Price) OVER (PARTITION BY Stock_Name, YEAR(Transaction_Date)
ORDER BY Transaction_Date ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING) AS End_Price
FROM surveillance_optimization
GROUP BY Stock_Name, YEAR(Transaction_Date), Transaction_Date, Transaction_Price),

PriceGrowth AS (SELECT Stock_Name, Year,
ROUND((End_Price - Start_Price) / Start_Price * 100,2) AS Percentage_Growth
FROM YearlyPrices
GROUP BY Stock_Name, Year, Start_Price, End_Price)

SELECT Stock_Name, year, Percentage_Growth
FROM PriceGrowth
WHERE Percentage_Growth > 500
ORDER BY Percentage_Growth DESC;
```

Stock_Name	year	Percentage_Growth
Microsoft Corp.	2021	1948.25
Eli Lilly and Co.	2022	1605.45
Eli Lilly and Co.	2020	1577.36
Eli Lilly and Co.	2021	1107.33
UnitedHealth Group	2021	1104.2
Beta Ltd	2021	811.78
Cisco Systems	2023	517.92

# Business Question-17

➤ Identify Users with Significant Price Fluctuations in Both Transaction and Enquiry Prices.

```
WITH TransactionFluctuations AS (SELECT User_ID,
Transaction_ID,
LAG(Transaction_Price) OVER (PARTITION BY User_ID ORDER BY Transaction_Date)
AS Previous_Transaction_Price,
Transaction_Price,
ABS(Transaction_Price - LAG(Transaction_Price) OVER (PARTITION BY User_ID
ORDER BY Transaction_Date)) / LAG(Transaction_Price) OVER (PARTITION BY User_ID ORDER BY Transaction_Date) * 100
AS Price_Change_Percentage
FROM surveillance_optimization),

EnquiryFluctuations AS (
SELECT User_ID,
LAG(Enquiry_Price) OVER (PARTITION BY User_ID ORDER BY Enquiry_Date) AS Previous_Enquiry_Price,
Enquiry_Price,
ABS(Enquiry_Price - LAG(Enquiry_Price) OVER (PARTITION BY User_ID
ORDER BY Enquiry_Date)) / LAG(Enquiry_Price) OVER (PARTITION BY User_ID ORDER BY Enquiry_Date) * 100
AS Enquiry_Change_Percentage
FROM regulatory_enquiries)

SELECT DISTINCT t.User_ID
FROM TransactionFluctuations AS t
JOIN EnquiryFluctuations AS e ON t.User_ID = e.User_ID
WHERE t.Price_Change_Percentage > 90
AND e.Enquiry_Change_Percentage > 90
AND t.Previous_Transaction_Price IS NOT NULL
AND e.Previous_Enquiry_Price IS NOT NULL;
```

User_ID
UID004
UID011
UID021
UID029
UID042
UID053
UID079

# Business Question-18

- Find the average manipulated prices for the top 3 stocks with the highest total manipulated volumes due to insider trading.

```
WITH ManipulatedVolumes AS (SELECT Stock_Name, SUM(Order_Volume) AS Total_Manipulated_Volume
FROM market_manipulation
WHERE Manipulation_Type = 'Insider Trading'
GROUP BY Stock_Name),
```

```
TopStocks AS (
SELECT Stock_Name
FROM ManipulatedVolumes
ORDER BY Total_Manipulated_Volume DESC
LIMIT 3)
```

```
SELECT mm.Stock_Name, AVG(mm.Order_Price) AS Avg_Manipulated_Price
FROM market_manipulation AS mm
JOIN TopStocks AS ts ON mm.Stock_Name = ts.Stock_Name
WHERE mm.Manipulation_Type = 'Insider Trading'
GROUP BY mm.Stock_Name;
```

Stock_Name	Avg_Manipulated_Price
Delta Co	364.526
Epsilon LLC	203.91
Gamma Inc	191.99

# Business Question-19

➤ Identify The Top 10 Cross-Market Manipulation Patterns and Calculate Potential Gains.

```
WITH CrossMarketManipulations AS (SELECT so.User_ID, so.Stock_Name AS Stock_Name,
so.Transaction_Price AS Surveillance_Price,
mm.Order_Price AS Manipulation_Price,
mm.Order_Volume,
ROW_NUMBER() OVER (PARTITION BY so.User_ID, so.Stock_Name
ORDER BY so.Transaction_Date DESC, mm.Date_of_Manipulation DESC) AS Row_Num
FROM surveillance_optimization AS so
INNER JOIN market_manipulation AS mm
ON so.User_ID = mm.User_ID AND so.Stock_Name = mm.Stock_Name
WHERE so.Transaction_Date < mm.Date_of_Manipulation),
PotentialGains AS (SELECT User_ID, Stock_Name AS Stock_Name,
ROUND((Manipulation_Price - Surveillance_Price) * Order_Volume),2) AS Total_Potential_Gain
FROM CrossMarketManipulations
WHERE Row_Num = 1
GROUP BY User_ID, Stock_Name)

SELECT User_ID, Stock_Name, Total_Potential_Gain
FROM PotentialGains
ORDER BY Total_Potential_Gain DESC
LIMIT 10;
```

User_ID	Stock_Name	Total_Potential_Gain
UID163	UnitedHealth Group	1493530.5
UID104	Pfizer Inc.	1369392.25
UID167	JPMorgan Chase	1264745.42
UID171	Home Depot Inc.	870320.78
UID118	Gamma Inc	533021.82
UID100	Comcast Corp.	512108.8
UID157	Apple Inc	71535.76

# Business Question-20

- Find Stocks with High Volatility using Manipulation and Enquiry Tables.

```
WITH CombinedData AS (SELECT mm.Stock_Name AS Stock_Name,
mm.Order_Price AS Price, mm.Date_of_Manipulation AS Date
FROM market_manipulation AS mm
UNION ALL
SELECT re.Stock_Name AS Stock_Name,
re.Enquiry_Price AS Price,
re.Enquiry_Date AS Date
FROM regulatory_enquiries AS re),
StockVolatility AS (
SELECT Stock_Name,
ROUND(STDDEV(Price),2) AS Price_StdDev
FROM CombinedData
GROUP BY Stock_Name)
SELECT Stock_Name, Price_StdDev
FROM StockVolatility
WHERE Price_StdDev > (SELECT AVG(Price_StdDev) FROM StockVolatility)
ORDER BY Price_StdDev DESC;
```

Stock_Name	Price_StdDev
Microsoft Corp.	158.33
Beta Ltd	153.27
Procter & Gamble	153.13
Amazon.com Inc.	151.43
Honda Motor Co.	151.02
Samsung Electronics	150.6
Mastercard Inc.	148.31

# Business Question-21

➤ Calculate the Monthly Average Price and Volume for Each Stock and Compare with Global Averages.

```
WITH MonthlyAverages AS (
SELECT Stock_Name,
DATE_FORMAT(Transaction_Date, '%Y-%m') AS Month,
ROUND(AVG(Transaction_Price), 2) AS Avg_Price,
ROUND(AVG(Volume), 2) AS Avg_Volume
FROM surveillance_optimization
GROUP BY Stock_Name, DATE_FORMAT(Transaction_Date, '%Y-%m')),

GlobalAverages AS (
SELECT ROUND(AVG(Transaction_Price), 2) AS Global_Avg_Price,
ROUND(AVG(Volume), 2) AS Global_Avg_Volume
FROM surveillance_optimization)

SELECT ma.Stock_Name, ma.Month, ma.Avg_Price, ma.Avg_Volume,
ga.Global_Avg_Price, ga.Global_Avg_Volume,
ROUND(ma.Avg_Price - ga.Global_Avg_Price, 2) AS Price_Difference,
ROUND(ma.Avg_Volume - ga.Global_Avg_Volume, 2) AS Volume_Difference
FROM MonthlyAverages AS ma
INNER JOIN GlobalAverages AS ga;
```

Stock	Month	Avg_Price	Avg_Volume	Global_Avg_Price	Global_Avg_Volume	Price_Difference	Volume_Difference
ABC Corp.	2022-02	159.21	4868.00	246.02	4822.24	-86.81	45.76
ABC Corp.	2022-06	284.66	3367.00	246.02	4822.24	38.64	-1455.24
ABC Corp.	2021-08	449.84	7955.00	246.02	4822.24	203.82	3132.76
ACME Inc.	2020-08	148.65	2251.00	246.02	4822.24	-97.37	-2571.24
ACME Inc.	2022-03	464.01	4375.00	246.02	4822.24	217.99	-447.24

# Business Question-22

- Determine the impact on stock prices where the average post-manipulation price is at least twice the average pre-manipulation price.



```
WITH PreManipulationPrices AS (
SELECT mm.Stock_Name, mm.Date_of_Manipulation,
ROUND(AVG(so.Transaction_Price),2)AS Avg_Pre_Manipulation_Price
FROM market_manipulation AS mm
JOIN surveillance_optimization AS so ON mm.Stock_Name = so.Stock_Name
WHERE so.Transaction_Date < mm.Date_of_Manipulation
GROUP BY mm.Stock_Name, mm.Date_of_Manipulation),

PostManipulationPrices AS (
SELECT mm.Stock_Name, mm.Date_of_Manipulation,
ROUND(AVG(so.Transaction_Price),2) AS Avg_Post_Manipulation_Price
FROM market_manipulation AS mm
JOIN surveillance_optimization AS so ON mm.Stock_Name = so.Stock_Name
WHERE so.Transaction_Date > mm.Date_of_Manipulation
GROUP BY mm.Stock_Name, mm.Date_of_Manipulation)

SELECT pre.Stock_Name, pre.Date_of_Manipulation,
pre.Avg_Pre_Manipulation_Price, post.Avg_Post_Manipulation_Price,
ROUND(post.Avg_Post_Manipulation_Price - pre.Avg_Pre_Manipulation_Price, 2) AS Price_Impact
FROM PreManipulationPrices AS pre
JOIN PostManipulationPrices AS post ON pre.Stock_Name = post.Stock_Name
AND pre.Date_of_Manipulation = post.Date_of_Manipulation
WHERE post.Avg_Post_Manipulation_Price >= 2 * pre.Avg_Pre_Manipulation_Price;
```

Stock_Name	Date_of_Manipulation	Avg_Pre_Manipulation_Price	Avg_Post_Manipulation_Price	Price_Impact
Comcast Corp.	2020-05-06	50.76	287.21	236.45
Comcast Corp.	2020-03-07	50.76	287.21	236.45
Tesla Inc.	2020-08-29	106.3	234.37	128.07
Cisco Systems	2022-08-16	93.58	250.27	156.69
Tesla Inc.	2021-01-10	88.49	262.51	174.02
Tesla Inc.	2020-09-08	88.49	262.51	174.02

## Key Findings-Transaction & Price

- ✓ Out of all the surveillance data, **SID072** has the greatest overall transaction value.
- ✓ **Chevron Corp.** had the largest order volume (430,154), with **Tesla Inc.** (393,859) and **Procter & Gamble** (415,464) following.
- ✓ The greatest price range (100+) has the majority of transactions.
- ✓ The top 5 stocks by transaction percentage are PepsiCo Inc. (4.67%), Abbott Laboratories (4.67%), Alpha Corp (4.00%), Gamma Inc. (3.78%), and Johnson & Johnson (3.56%).
- ✓ With a positive price difference of \$409.54, **JPMorgan Chase** displayed the largest positive differential, followed by **Toyota Motor Corp.** (\$207.87) and **Walmart Inc.** (\$400.75).
- ✓ **Microsoft Corp.** topped the growth chart in 2021 with a 1948.25% rise.
- ✓ There were **fifteen transactions** involving different equities that saw price declines exceeding 90%.
- ✓ At 301.91%, **JPMorgan Chase** experienced the largest post-manipulation price increase.



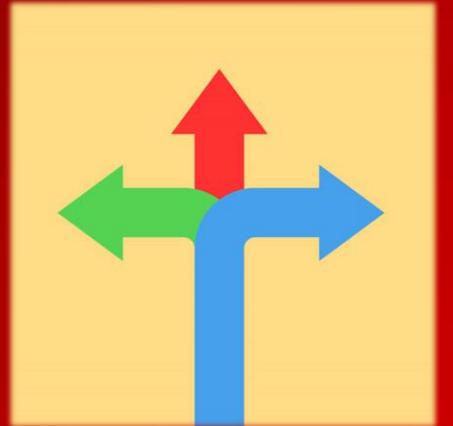
## Key Findings-Manipulation & User Analysis

- ✓ **Top 3 Manipulated Stocks** by Avg. Order Price: JPMorgan Chase (\$343.32), Berkshire Hathaway (\$334.69), Sony Group Corp. (\$327.5).
- ✓ **Common Manipulation Types:** Spoofing (\$307.74), Pump and Dump (\$272.87), Wash Trading (\$253.60).
- ✓ **Highest Insider Trading Manipulated Prices:** Delta Co. (\$364.53), Epsilon LLC (\$203.91), Gamma Inc. (\$191.99).
- ✓ Significant deviations from global trading patterns suggest suspicious activity.
- ✓ **Common enquiry types:** Fraudulent Reporting, Front Running, Market Manipulation, Insider Trading.
- ✓ **16 users** made multiple regulatory enquiries; 25 users had high regulatory volume transactions and enquiries.
- ✓ **13 users** had 4 to 6 significant price fluctuations in enquiry prices.
- ✓ Users like UID004, UID011, UID021, UID029 showed notable price fluctuations in both transaction and enquiry prices.
- ✓ **Top Gains from Cross-Market Manipulation:** UnitedHealth Group (\$1.49M), Pfizer Inc. (\$1.37M), JPMorgan Chase (\$1.26M); Top Losses: Abbott Laboratories (-\$316K), Visa Inc. (-\$276K).



# Conclusions

- ✓ **Stock Dominance:** Stock ID SID072 and large-cap companies such as Chevron Corp. dominate the market in terms of trading volume, indicating market concentration.
- ✓ **Price Sensitivity:** Large price movements in the stock prices of companies such as JPMorgan Chase and Microsoft indicate that these stocks are highly sensitive to market forces and manipulation.
- ✓ **Pervasive Manipulation:** The high average order prices of manipulated stocks and common manipulative tactics such as spoofing, pumping and dumping, and wash trading indicate that market manipulation is widespread.
- ✓ **User Involvement:** A significant number of users have been involved in regulatory investigations, highlighting the need for stricter market oversight.
- ✓ **Multiple Market Vulnerabilities:** The potential for large gains and losses in multiple stocks indicates that manipulation in multiple markets poses significant risks to market stability.
- ✓ **Significant Fluctuations:** The existence of price volatility among some users suggests that some users may be engaging in activities that could disrupt the market or profit from price fluctuations.



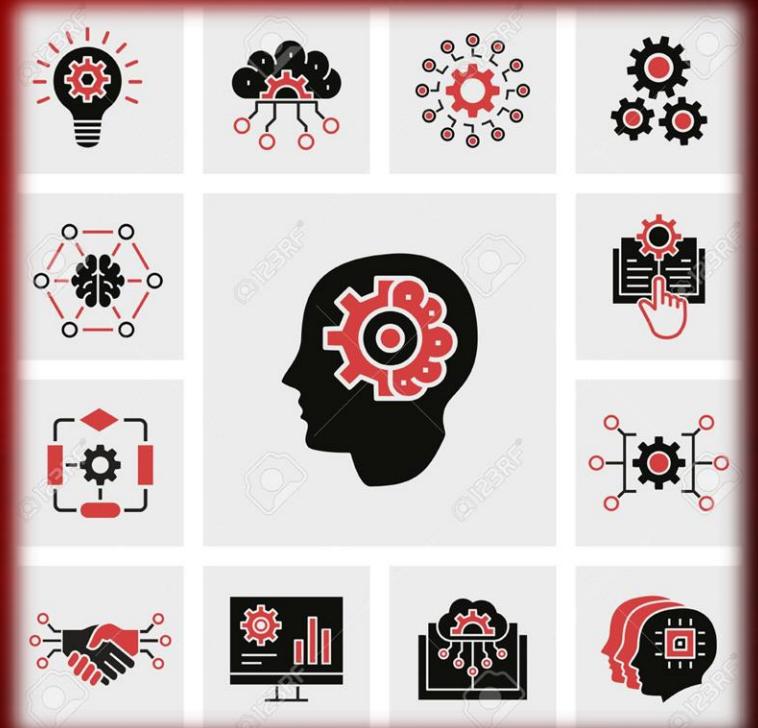
# Recommendations

- ✓ **Enhanced Monitoring:** A Strong Focus on stocks with high transaction values and price volatility for potential manipulation.
- ✓ **Targeted Audits:** It is required to Prioritize investigation into Spoofing, Pump and Dump, and Wash Trading activities.
- ✓ **User Activity Review:** Close monitoring users with multiple regulatory enquiries and significant price fluctuations.
- ✓ **Risk Mitigation:** An Implementation of stronger controls to mitigate risks from cross-market manipulation, especially for high-impact stocks.
- ✓ **Proactive Reporting:** Strengthen fraud detection mechanisms for common enquiry types like Fraudulent Reporting and Insider Trading.



## Future Work

- ✓ **Enhanced Data Collection:** Incorporate real-time data feeds, social media sentiment, and news analysis to better detect manipulation patterns.
- ✓ **Advanced Analytics:** Implement machine learning models for anomaly detection and predictive insights to identify subtle manipulation strategies.
- ✓ **Real-time Monitoring:** Develop tools for real-time detection of suspicious activities, allowing for immediate response to potential fraud.
- ✓ **Collaboration with Regulators:** Partner with regulatory bodies to share insights and improve monitoring techniques for market compliance.
- ✓ **User Behavior Analysis:** Profile trader behaviors to better predict and mitigate potential manipulation activities.
- ✓ **Blockchain for Transparency:** Explore blockchain technology to enhance transparency and reduce manipulation opportunities.



# THANK YOU

