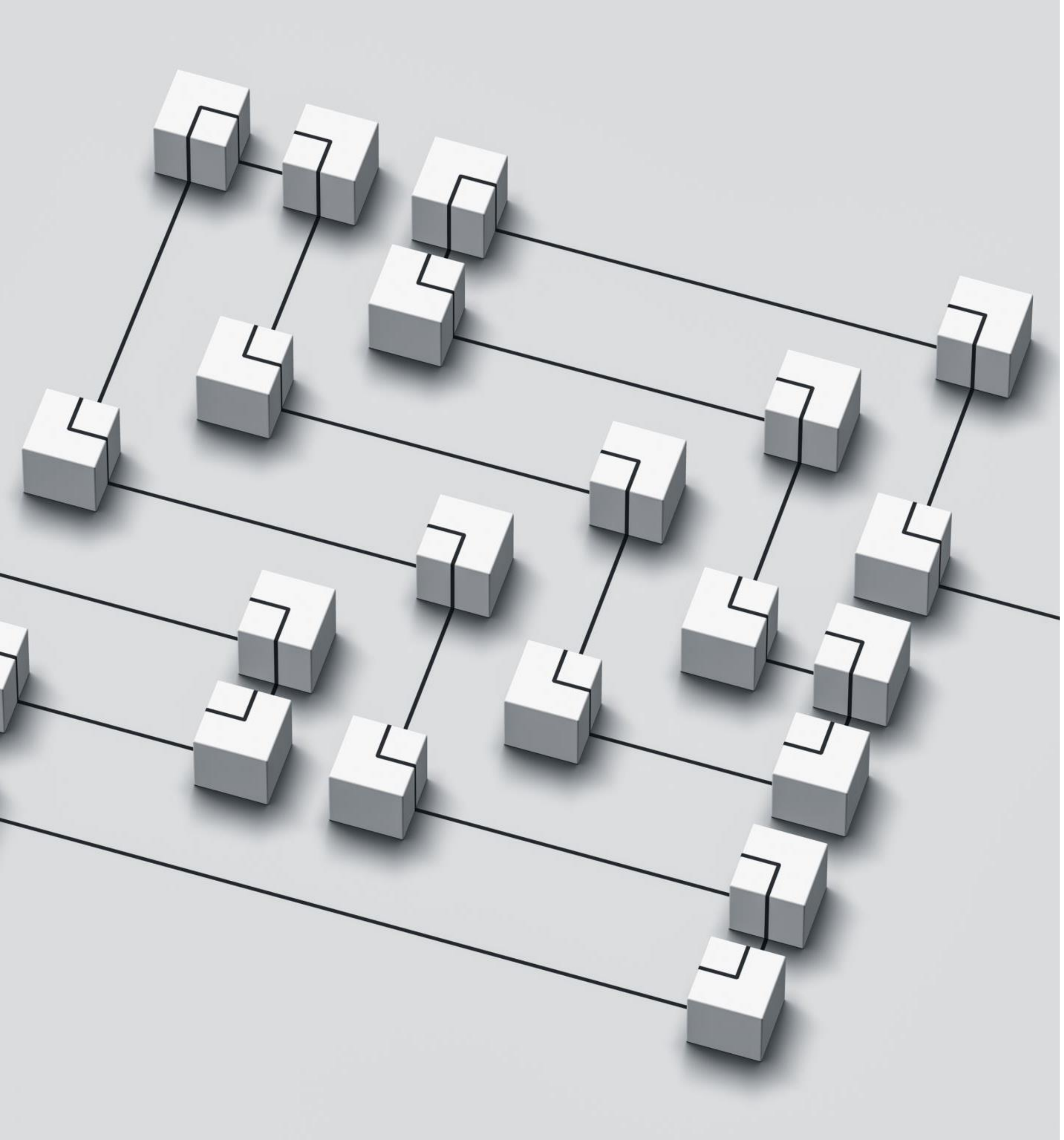
Anti-Money Laundering (AML) Analysis with MySQL

Detecting Suspicious Transactions | Smurfing| and Multi-Hop Money Flow

Presented by: Sneha

- **Introduction to AML**
- **What is AML?**
  - AML refers to regulations and techniques used to prevent financial crimes.
- **Why is it Important?**
  - Protects financial institutions from illicit activities.
  - Identifies suspicious transaction patterns.
- **Project Goal:**
  - Develop an end-to-end AML detection system using MySQL.



- **Database Schema**

- **Tables in MySQL:**
  - **Customers_final:** Stores customer details.
  - **Transactions:** Contains all financial transactions.
  - **Risk_Parameters:** Defines thresholds for suspicious activities.

- **Indexes for Performance Optimization:**
  - CustomerID, SenderID, ReceiverID, Date indexed for efficient queries

- **Risk Parameters & Thresholds**
- **Risk Rules Implemented:**
  - **Suspicious Transaction:** Amount between **9000 - 9999**.
  - **High-Risk Customers:** Transactions exceed **100,000**.
  - **Smurfing Pattern:** More than **3 transactions**, total **>150,000** in 30 days

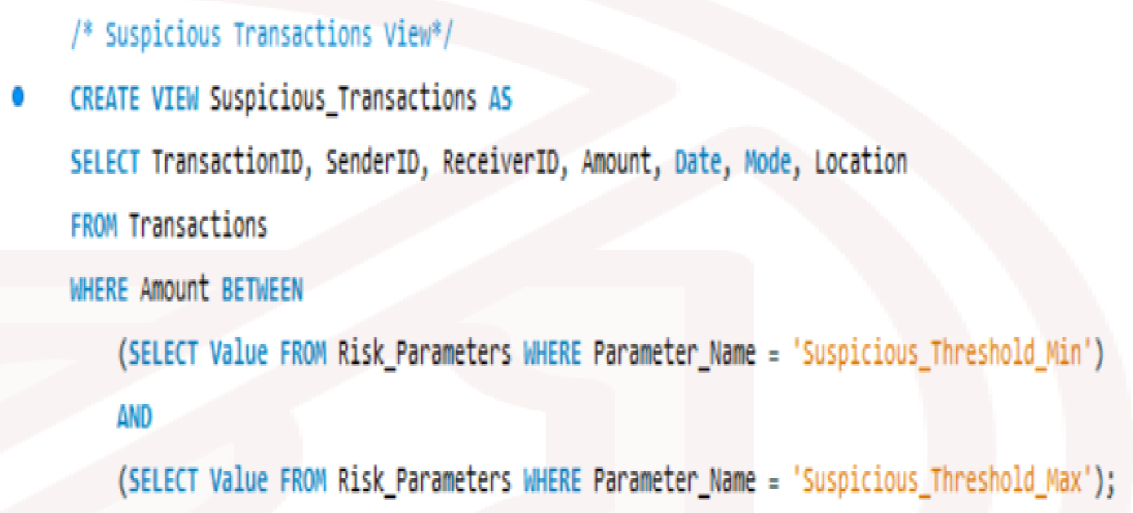
Suspicious Transactions View

•**Query Logic:**

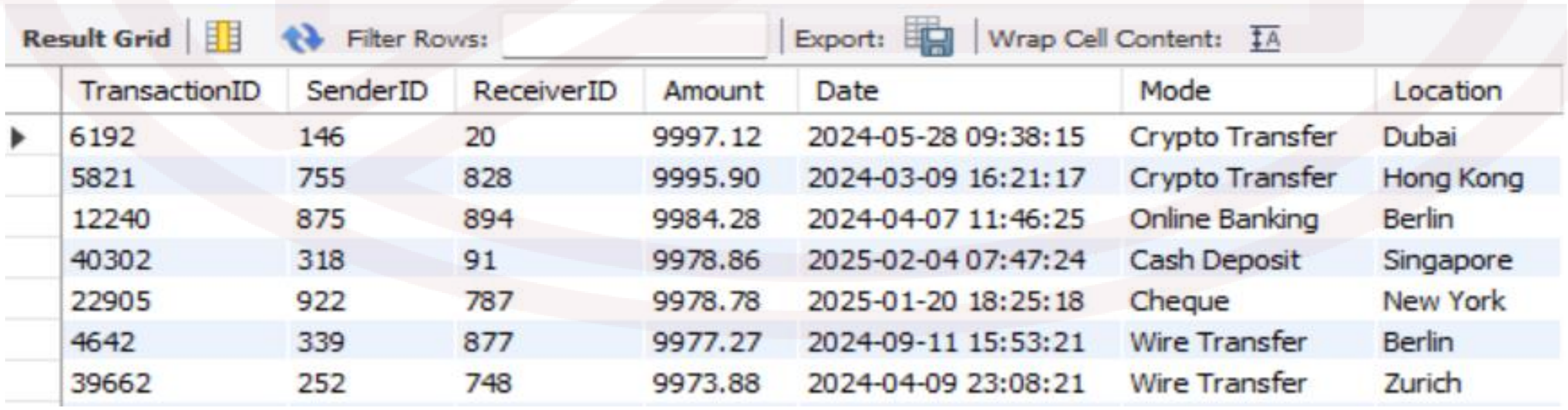
- **Extract transactions where amount is in suspicious range.**
- **Uses Risk_Parameters table for dynamic thresholding.**

•**Use Case:**

- **Identify transactions just below regulatory reporting limits.**



```
/* Suspicious Transactions View*/
CREATE VIEW Suspicious_Transactions AS
SELECT TransactionID, SenderID, ReceiverID, Amount, Date, Mode, Location
FROM Transactions
WHERE Amount BETWEEN
    (SELECT Value FROM Risk_Parameters WHERE Parameter_Name = 'Suspicious_Threshold_Min')
    AND
    (SELECT Value FROM Risk_Parameters WHERE Parameter_Name = 'Suspicious_Threshold_Max');
```



TransactionID	SenderID	ReceiverID	Amount	Date	Mode	Location
6192	146	20	9997.12	2024-05-28 09:38:15	Crypto Transfer	Dubai
5821	755	828	9995.90	2024-03-09 16:21:17	Crypto Transfer	Hong Kong
12240	875	894	9984.28	2024-04-07 11:46:25	Online Banking	Berlin
40302	318	91	9978.86	2025-02-04 07:47:24	Cash Deposit	Singapore
22905	922	787	9978.78	2025-01-20 18:25:18	Cheque	New York
4642	339	877	9977.27	2024-09-11 15:53:21	Wire Transfer	Berlin
39662	252	748	9973.88	2024-04-09 23:08:21	Wire Transfer	Zurich

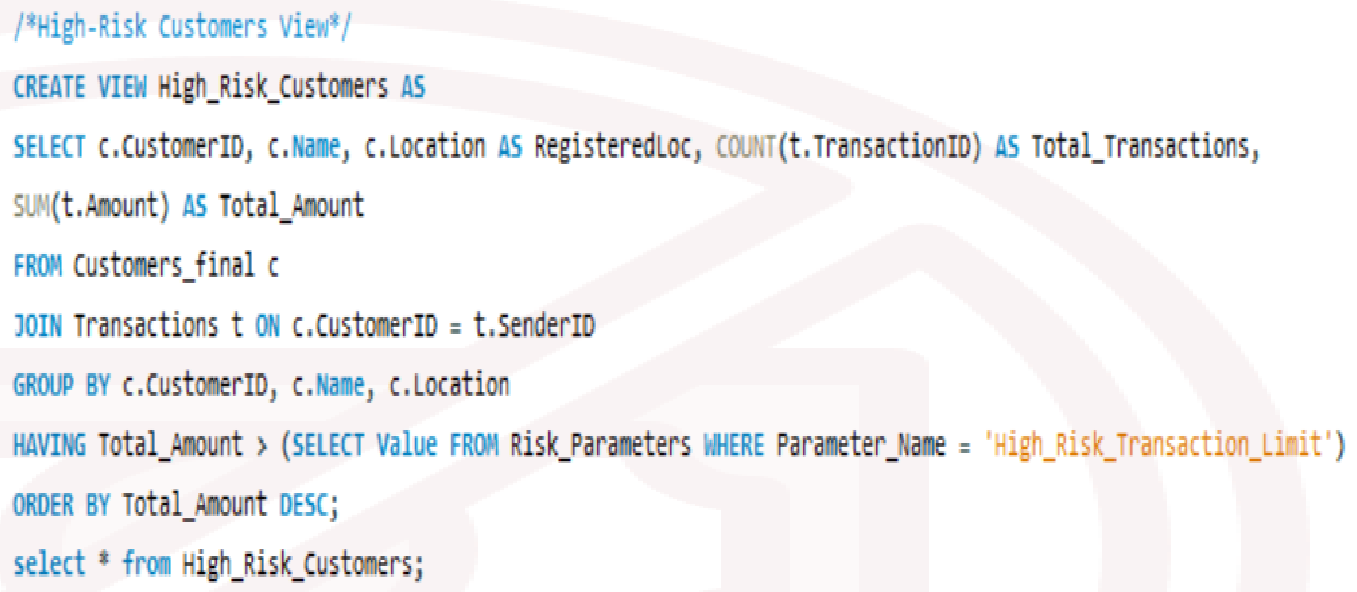
High-Risk Customers Analysis

•Query Logic:

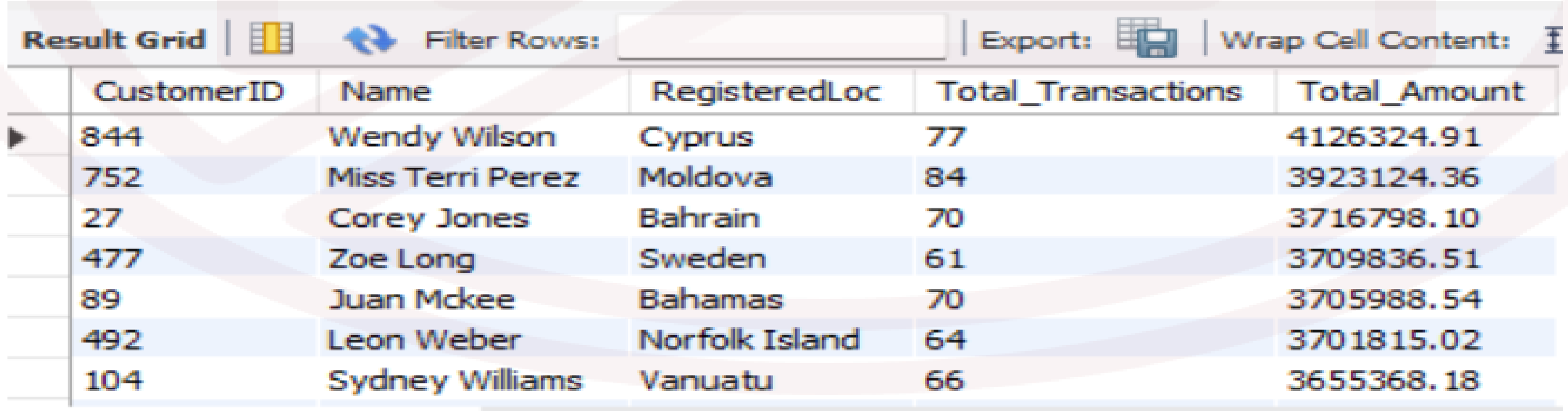
- **Aggregates transaction amounts per customer.**
- **Flags customers exceeding 100,000 in transactions.**

•Use Case:

- **Identifies individuals engaging in high-risk financial activity.**



```
/*High-Risk Customers View*/
CREATE VIEW High_Risk_Customers AS
SELECT c.CustomerID, c.Name, c.Location AS RegisteredLoc, COUNT(t.TransactionID) AS Total_Transactions,
SUM(t.Amount) AS Total_Amount
FROM Customers_final c
JOIN Transactions t ON c.CustomerID = t.SenderID
GROUP BY c.CustomerID, c.Name, c.Location
HAVING Total_Amount > (SELECT Value FROM Risk_Parameters WHERE Parameter_Name = 'High_Risk_Transaction_Limit')
ORDER BY Total_Amount DESC;
select * from High_Risk_Customers;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	Name	RegisteredLoc	Total_Transactions	Total_Amount
844	Wendy Wilson	Cyprus	77	4126324.91
752	Miss Terri Perez	Moldova	84	3923124.36
27	Corey Jones	Bahrain	70	3716798.10
477	Zoe Long	Sweden	61	3709836.51
89	Juan Mckee	Bahamas	70	3705988.54
492	Leon Weber	Norfolk Island	64	3701815.02
104	Sydney Williams	Vanuatu	66	3655368.18

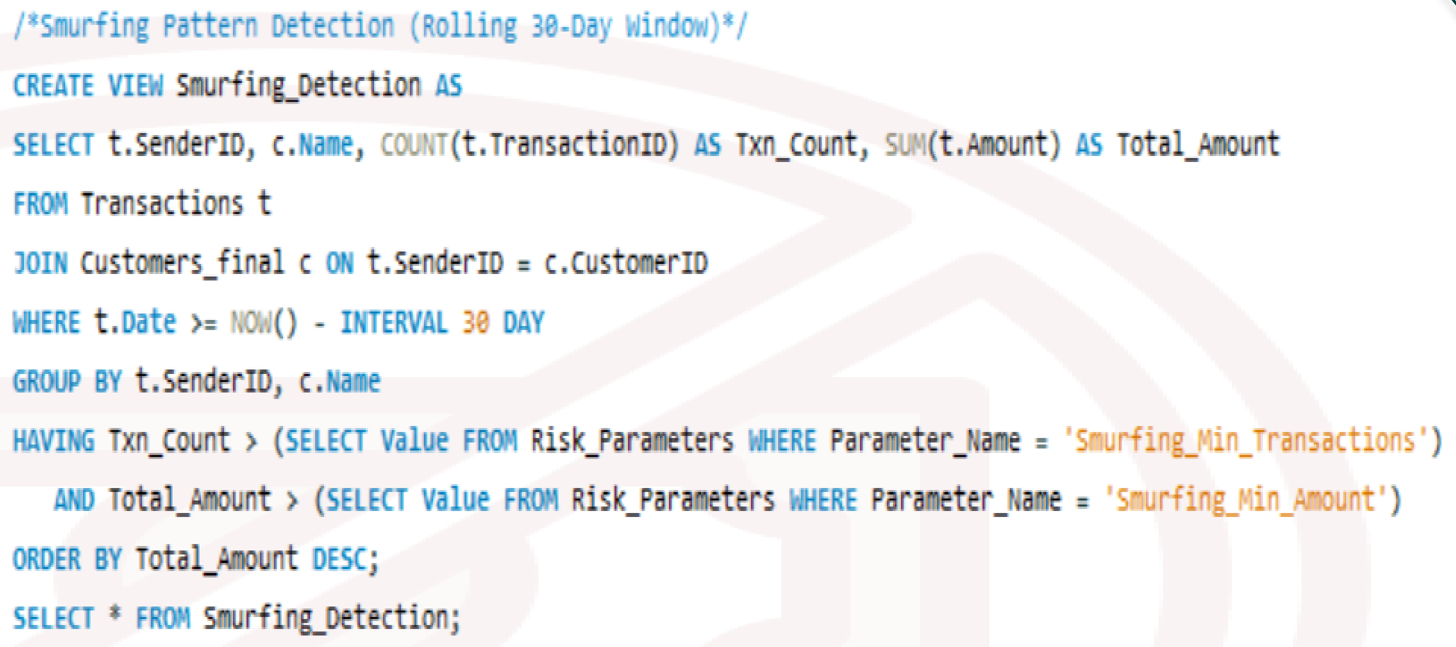
Smurfing Pattern Detection

•**Query Logic:**

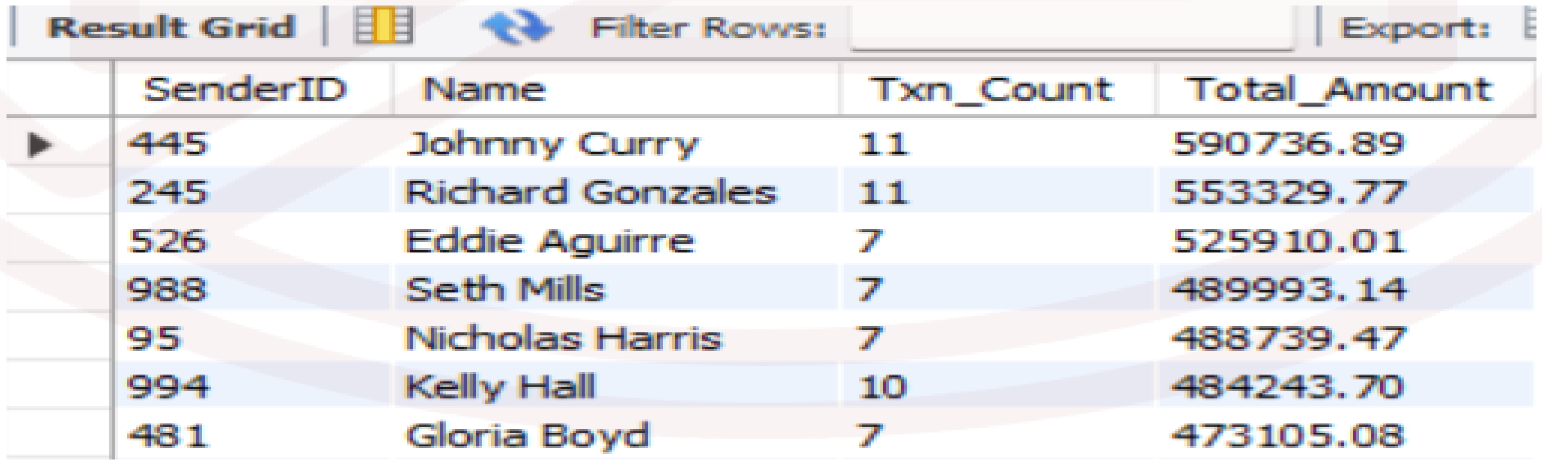
- **Detects frequent small transactions** adding up to large sums.
- Rolling 30-day window to ensure real-time tracking.

•**Use Case:**

- Identifies potential structuring to avoid detection



```sql
/*Smurfing Pattern Detection (Rolling 30-Day Window)*/
CREATE VIEW Smurfing_Detection AS
SELECT t.SenderID, c.Name, COUNT(t.TransactionID) AS Txn_Count, SUM(t.Amount) AS Total_Amount
FROM Transactions t
JOIN Customers_final c ON t.SenderID = c.CustomerID
WHERE t.Date >= NOW() - INTERVAL 30 DAY
GROUP BY t.SenderID, c.Name
HAVING Txn_Count > (SELECT Value FROM Risk_Parameters WHERE Parameter_Name = 'Smurfing_Min_Transactions')
   AND Total_Amount > (SELECT Value FROM Risk_Parameters WHERE Parameter_Name = 'Smurfing_Min_Amount')
ORDER BY Total_Amount DESC;
SELECT * FROM Smurfing_Detection;
```

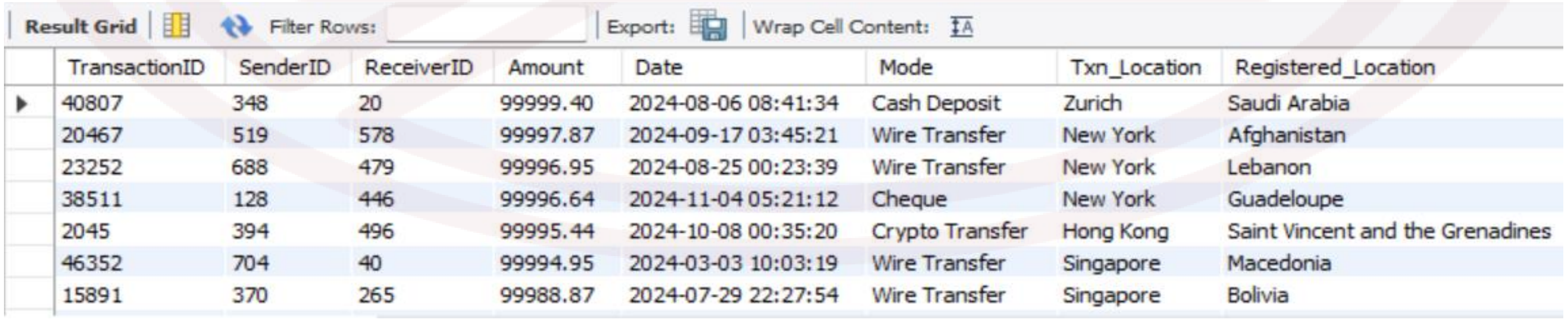


SenderID	Name	Txn_Count	Total_Amount
445	Johnny Curry	11	590736.89
245	Richard Gonzales	11	553329.77
526	Eddie Aguirre	7	525910.01
988	Seth Mills	7	489993.14
95	Nicholas Harris	7	488739.47
994	Kelly Hall	10	484243.70
481	Gloria Boyd	7	473105.08

Cross-Location Transaction Analysis

- **Query Logic:**
  - **Flags transactions where sending & receiving locations differ.**
  - **Filters for high-value transactions (>80,000).**
- **Use Case:**
  - **Identifies possible cross-border money laundering activities**

```sql
/*Cross-Location Transaction Analysis (Geospatial Risk Insight)*/
CREATE VIEW Cross_Location_Transactions AS
SELECT t.TransactionID, t.SenderID, t.ReceiverID, t.Amount, t.Date, t.Mode,
       t.Location AS Txn_Location, c.Location AS Registered_Location
FROM Transactions t
JOIN Customers_final c ON t.SenderID = c.CustomerID
WHERE t.Location <> c.Location
AND t.Amount > 80000
ORDER BY t.Amount DESC;
```



TransactionID	SenderID	ReceiverID	Amount	Date	Mode	Txn_Location	Registered_Location
40807	348	20	99999.40	2024-08-06 08:41:34	Cash Deposit	Zurich	Saudi Arabia
20467	519	578	99997.87	2024-09-17 03:45:21	Wire Transfer	New York	Afghanistan
23252	688	479	99996.95	2024-08-25 00:23:39	Wire Transfer	New York	Lebanon
38511	128	446	99996.64	2024-11-04 05:21:12	Cheque	New York	Guadeloupe
2045	394	496	99995.44	2024-10-08 00:35:20	Crypto Transfer	Hong Kong	Saint Vincent and the Grenadines
46352	704	40	99994.95	2024-03-03 10:03:19	Wire Transfer	Singapore	Macedonia
15891	370	265	99988.87	2024-07-29 22:27:54	Wire Transfer	Singapore	Bolivia

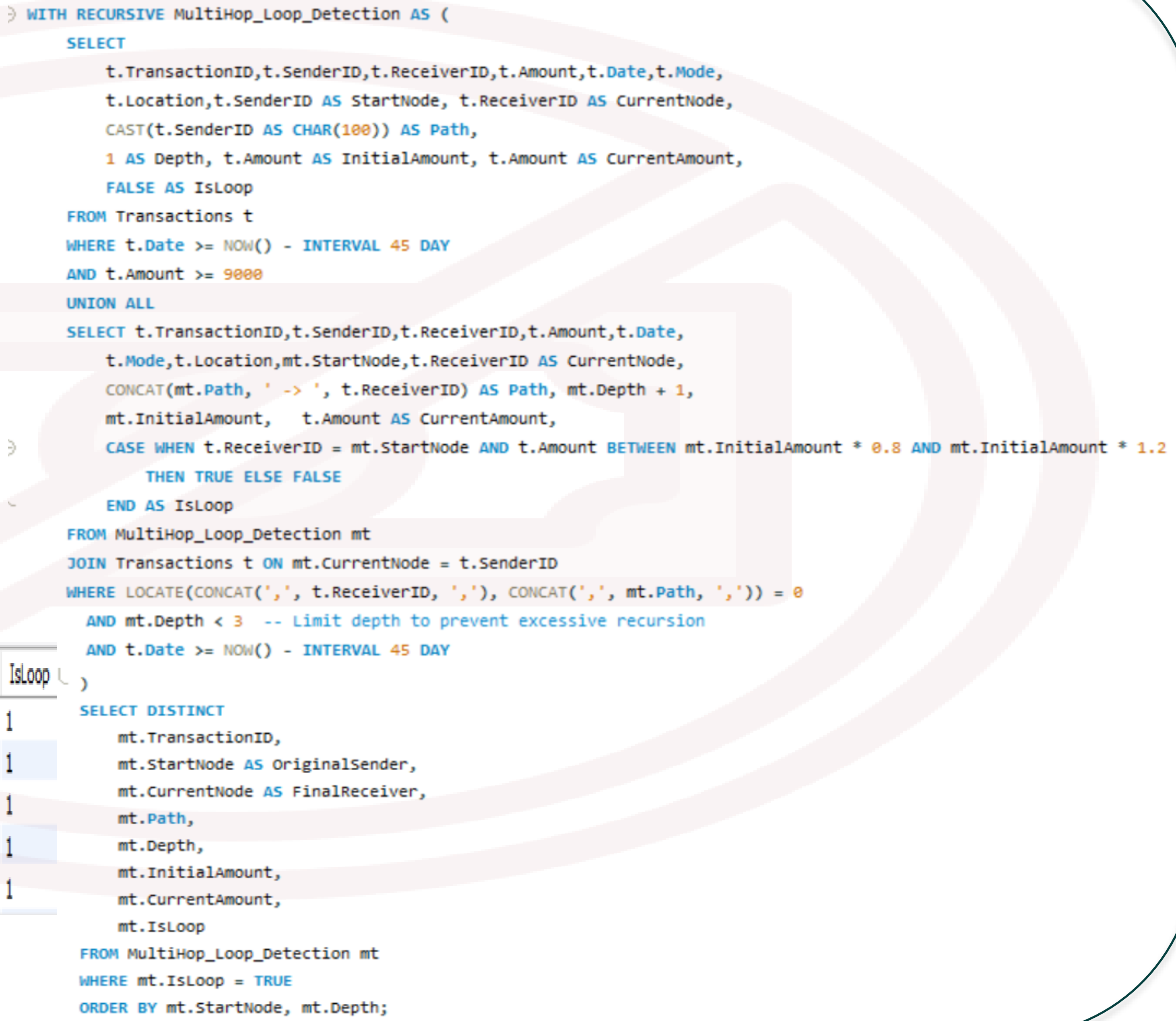
Multi-Hop Transaction Analysis

•Recursive CTE for Multi-Hop Transactions

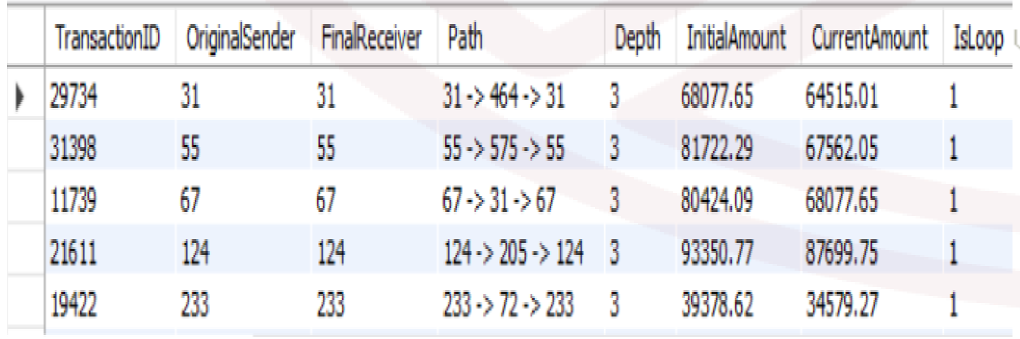
- **Tracks money flow across multiple transactions.**
- **Detects loops where money returns to the original sender within ±20% of the initial amount.**

•Use Case:

- **Identifies layering techniques used to obscure money trails**



```
WITH RECURSIVE MultiHop_Loop_Detection AS (
    SELECT
        t.TransactionID,t.SenderID,t.ReceiverID,t.Amount,t.Date,t.Mode,
        t.Location,t.SenderID AS StartNode, t.ReceiverID AS CurrentNode,
        CAST(t.SenderID AS CHAR(100)) AS Path,
        1 AS Depth, t.Amount AS InitialAmount, t.Amount AS CurrentAmount,
        FALSE AS IsLoop
    FROM Transactions t
    WHERE t.Date >= NOW() - INTERVAL 45 DAY
    AND t.Amount >= 9000
    UNION ALL
    SELECT t.TransactionID,t.SenderID,t.ReceiverID,t.Amount,t.Date,
        t.Mode,t.Location,mt.StartNode,t.ReceiverID AS CurrentNode,
        CONCAT(mt.Path, ' -> ', t.ReceiverID) AS Path, mt.Depth + 1,
        mt.InitialAmount,   t.Amount AS CurrentAmount,
        CASE WHEN t.ReceiverID = mt.StartNode AND t.Amount BETWEEN mt.InitialAmount * 0.8 AND mt.InitialAmount * 1.2
            THEN TRUE ELSE FALSE
        END AS IsLoop
    FROM MultiHop_Loop_Detection mt
    JOIN Transactions t ON mt.CurrentNode = t.SenderID
    WHERE LOCATE(CONCAT(',', t.ReceiverID, ','), CONCAT(',', mt.Path, ',')) = 0
     AND mt.Depth < 3  -- Limit depth to prevent excessive recursion
     AND t.Date >= NOW() - INTERVAL 45 DAY
)
 SELECT DISTINCT
     mt.TransactionID,
     mt.StartNode AS OriginalSender,
     mt.CurrentNode AS FinalReceiver,
     mt.Path,
     mt.Depth,
     mt.InitialAmount,
     mt.CurrentAmount,
     mt.IsLoop
 FROM MultiHop_Loop_Detection mt
 WHERE mt.IsLoop = TRUE
 ORDER BY mt.StartNode, mt.Depth;
```



TransactionID	OriginalSender	FinalReceiver	Path	Depth	InitialAmount	CurrentAmount	IsLoop
29734	31	31	31 -> 464 -> 31	3	68077.65	64515.01	1
31398	55	55	55 -> 575 -> 55	3	81722.29	67562.05	1
11739	67	67	67 -> 31 -> 67	3	80424.09	68077.65	1
21611	124	124	124 -> 205 -> 124	3	93350.77	87699.75	1
19422	233	233	233 -> 72 -> 233	3	39378.62	34579.27	1

Key Insights from Analysis

- •Identified patterns of structuring (Smurfing).
- •Detected high-risk customers engaging in large transactions.
- •Mapped complex money movement networks via Multi-Hop Analysis.
- •Highlighted unusual geographic transaction flows.

Conclusion

- •AML analysis is essential for fraud detection.
- •MySQL enables structured and efficient risk monitoring.
- •By leveraging advanced SQL techniques, financial institutions can proactively identify and mitigate money laundering risks.
- •Continuous improvement in AML frameworks ensures better regulatory compliance and security.

Analyst Role Description

•**Role:** AML Data Analyst

•**Key Responsibilities:**

- **Data Processing & Analysis:** Extract, clean, and analyze transaction data for AML insights.
- **Risk Assessment:** Identify suspicious patterns, high-risk customers, and cross-border transactions.
- **SQL Querying:** Develop optimized queries and views for AML rule implementation.
- **Report Generation:** Provide actionable insights and reports for financial risk teams.
- **Regulatory Compliance Support:** Ensure data aligns with AML laws and guidelines.

•**Skills Required:**

- Strong SQL and MySQL knowledge
- Experience in financial data analysis
- Understanding of AML regulations
- Proficiency in data visualization tools (optional)

THANK YOU