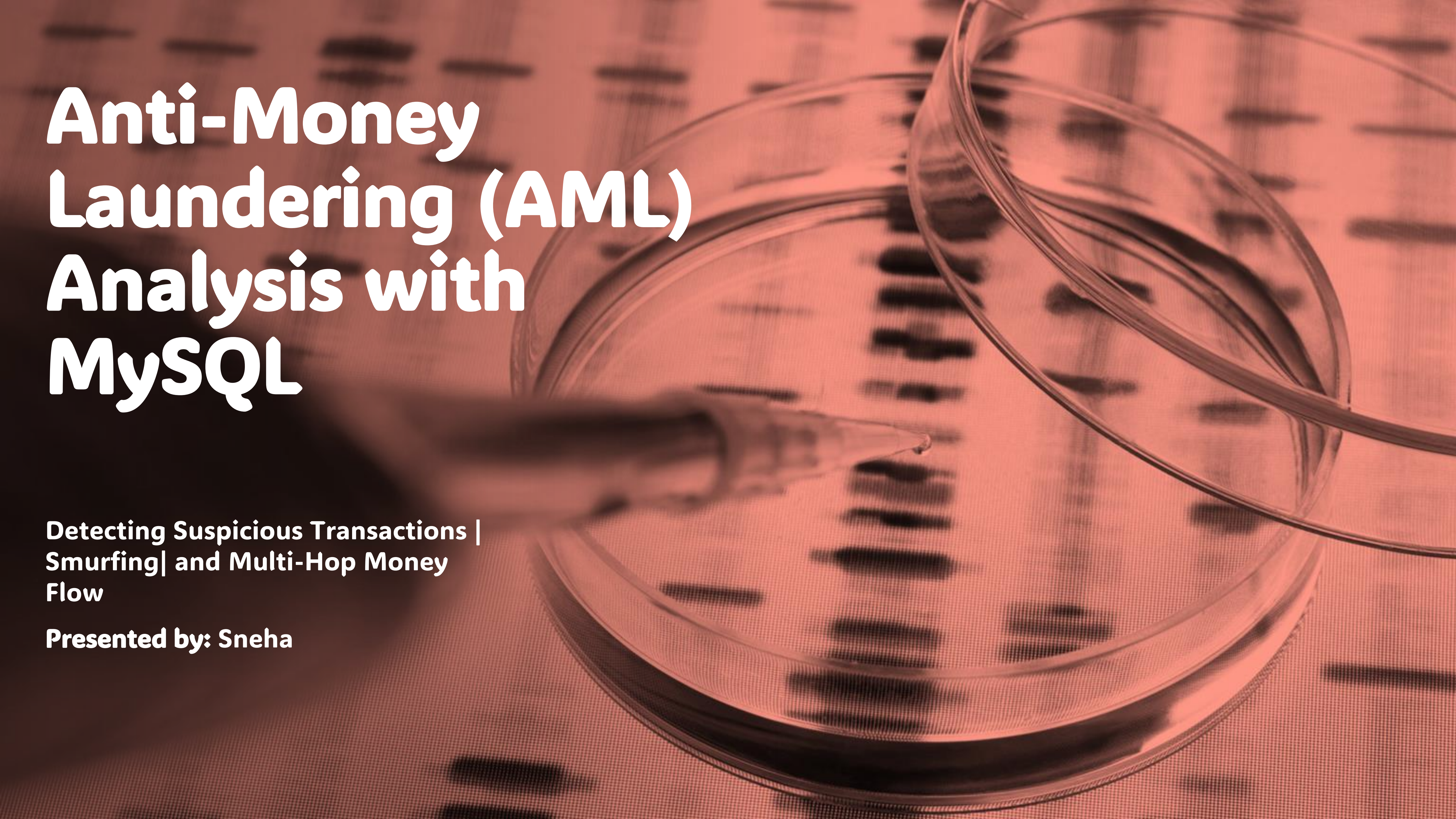
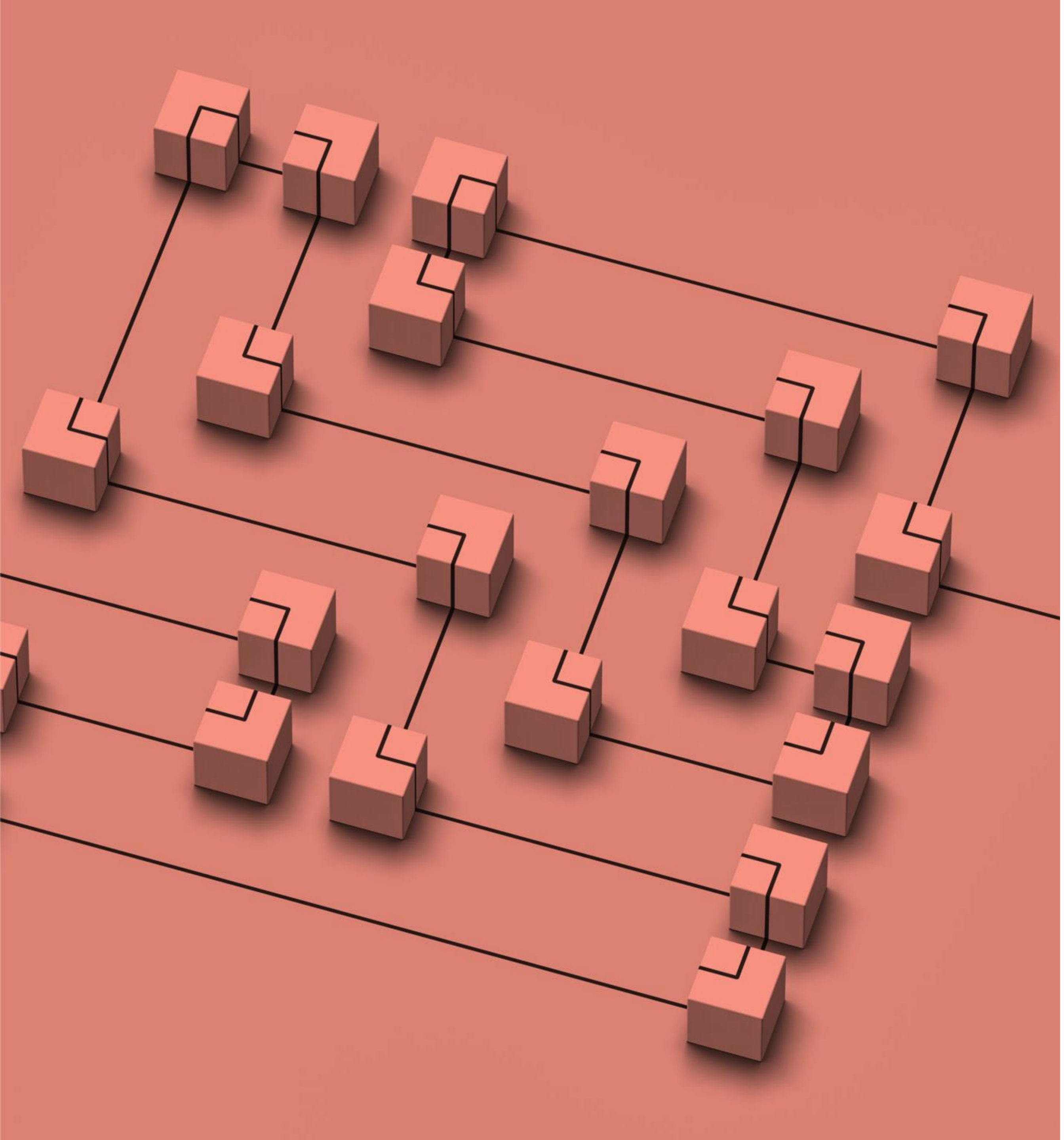


Anti-Money Laundering (AML) Analysis with MySQL

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.

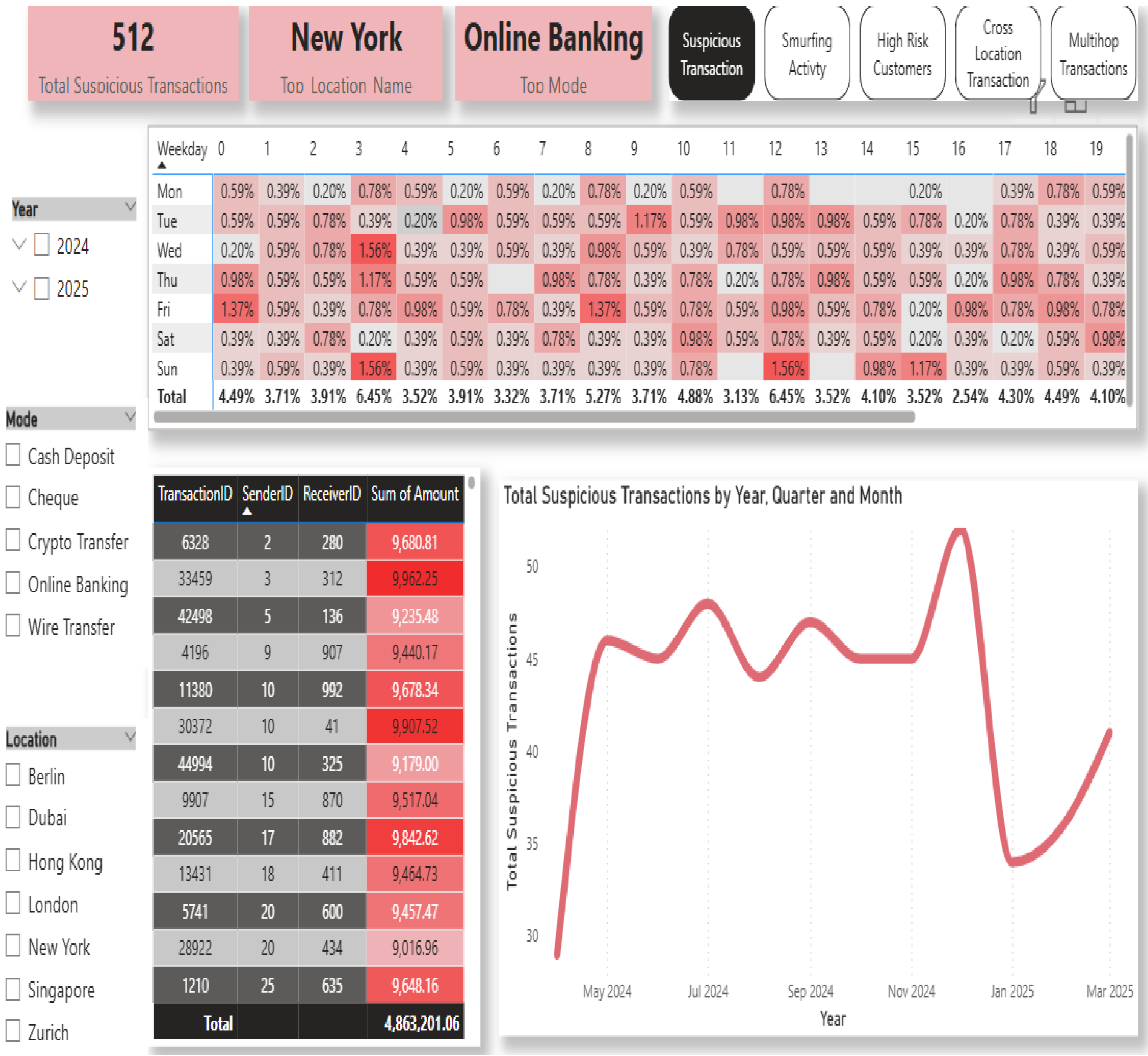


- **Database Schema**

- **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 **2,500,000**.
  - **Smurfing Pattern:** More than **3 transactions**, total **>150,000** in 30 days



Weekday	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Mon	0.59%	0.39%	0.20%	0.78%	0.59%	0.20%	0.59%	0.20%	0.78%	0.20%	0.59%		0.78%			0.20%		0.39%	0.78%	0.59%
Tue	0.59%	0.59%	0.78%	0.39%	0.20%	0.98%	0.59%	0.59%	0.59%	1.17%	0.59%	0.98%	0.98%	0.98%	0.59%	0.78%	0.20%	0.78%	0.39%	0.39%
Wed	0.20%	0.59%	0.78%	1.56%	0.39%	0.39%	0.59%	0.39%	0.98%	0.59%	0.39%	0.78%	0.59%	0.59%	0.59%	0.39%	0.39%	0.78%	0.39%	0.59%
Thu	0.98%	0.59%	0.59%	1.17%	0.59%	0.59%		0.98%	0.78%	0.39%	0.78%	0.20%	0.78%	0.98%	0.59%	0.59%	0.20%	0.98%	0.78%	0.39%
Fri	1.37%	0.59%	0.39%	0.78%	0.98%	0.59%	0.78%	0.39%	1.37%	0.59%	0.78%	0.59%	0.98%	0.59%	0.78%	0.20%	0.98%	0.78%	0.98%	0.78%
Sat	0.39%	0.39%	0.78%	0.20%	0.39%	0.59%	0.39%	0.78%	0.39%	0.39%	0.98%	0.59%	0.78%	0.39%	0.59%	0.20%	0.39%	0.20%	0.59%	0.98%
Sun	0.39%	0.59%	0.39%	1.56%	0.39%	0.59%	0.39%	0.39%	0.39%	0.39%	0.78%		1.56%		0.98%	1.17%	0.39%	0.39%	0.59%	0.39%
Total	4.49%	3.71%	3.91%	6.45%	3.52%	3.91%	3.32%	3.71%	5.27%	3.71%	4.88%	3.13%	6.45%	3.52%	4.10%	3.52%	2.54%	4.30%	4.49%	4.10%

TransactionID	SenderID	ReceiverID	Sum of Amount
6328	2	280	9,680.81
33459	3	312	9,962.25
42498	5	136	9,235.48
4196	9	907	9,440.17
11380	10	992	9,678.34
30372	10	41	9,907.52
44994	10	325	9,179.00
9907	15	870	9,517.04
20565	17	882	9,842.62
13431	18	411	9,464.73
5741	20	600	9,457.47
28922	20	434	9,016.96
1210	25	635	9,648.16
Total			4,863,201.06

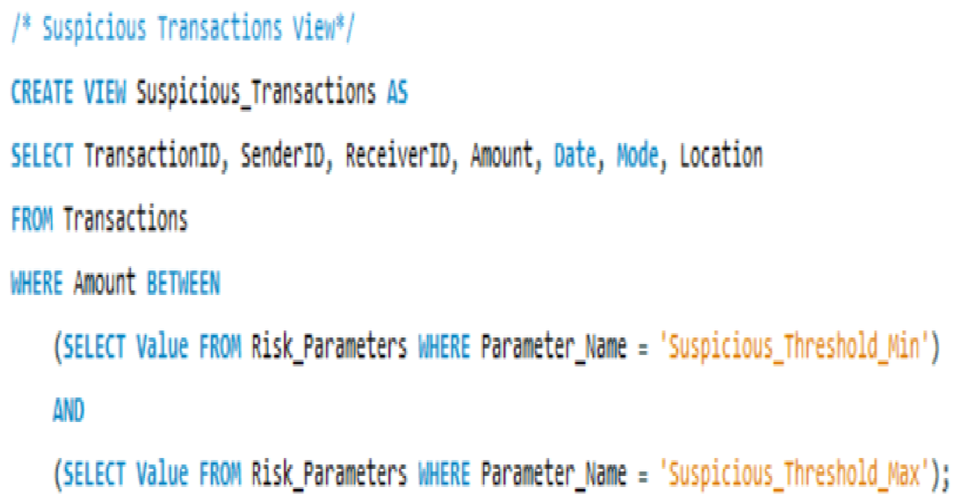
Suspicious Transactions View

•Query Logic:

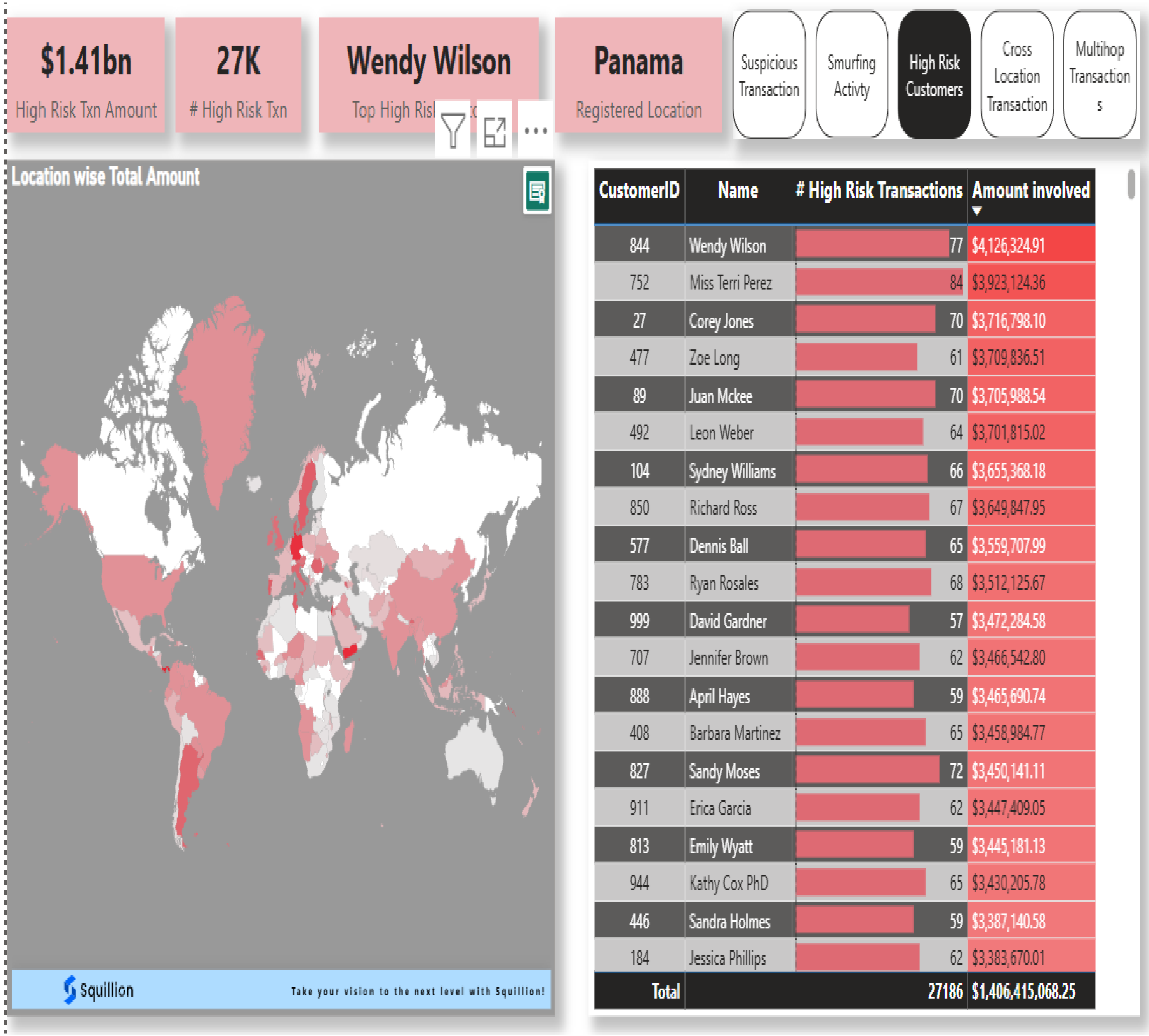
- **Extract transactions where amount is in suspicious range.**
- **Uses Risk_Parameters table for dynamic thresholding.**

•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');
```



CustomerID	Name	# High Risk Transactions	Amount involved
844	Wendy Wilson	77	$4,126,324.91
752	Miss Terri Perez	84	$3,923,124.36
27	Corey Jones	70	$3,716,798.10
477	Zoe Long	61	$3,709,836.51
89	Juan Mckee	70	$3,705,988.54
492	Leon Weber	64	$3,701,815.02
104	Sydney Williams	66	$3,655,368.18
850	Richard Ross	67	$3,649,847.95
577	Dennis Ball	65	$3,559,707.99
783	Ryan Rosales	68	$3,512,125.67
999	David Gardner	57	$3,472,284.58
707	Jennifer Brown	62	$3,466,542.80
888	April Hayes	59	$3,465,690.74
408	Barbara Martinez	65	$3,458,984.77
827	Sandy Moses	72	$3,450,141.11
911	Erica Garcia	62	$3,447,409.05
813	Emily Wyatt	59	$3,445,181.13
944	Kathy Cox PhD	65	$3,430,205.78
446	Sandra Holmes	59	$3,387,140.58
184	Jessica Phillips	62	$3,383,670.01
Total		27186	$1,406,415,068.25

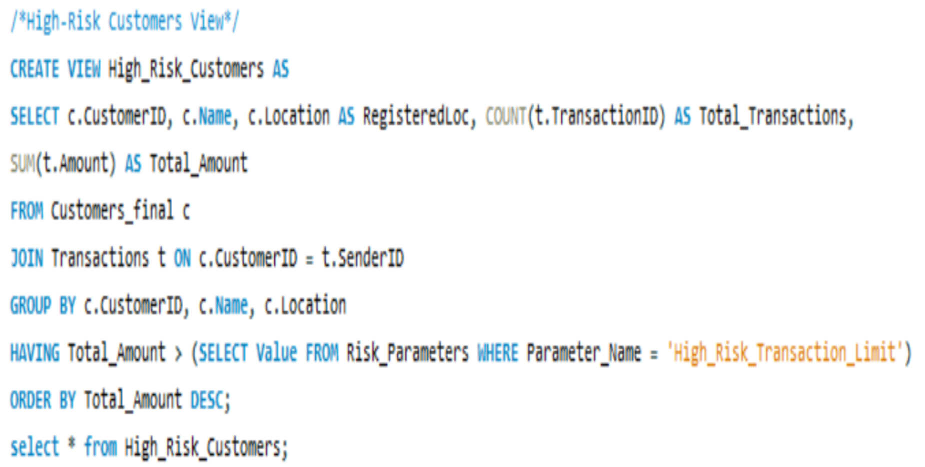
High-Risk Customers Analysis

•Query Logic:

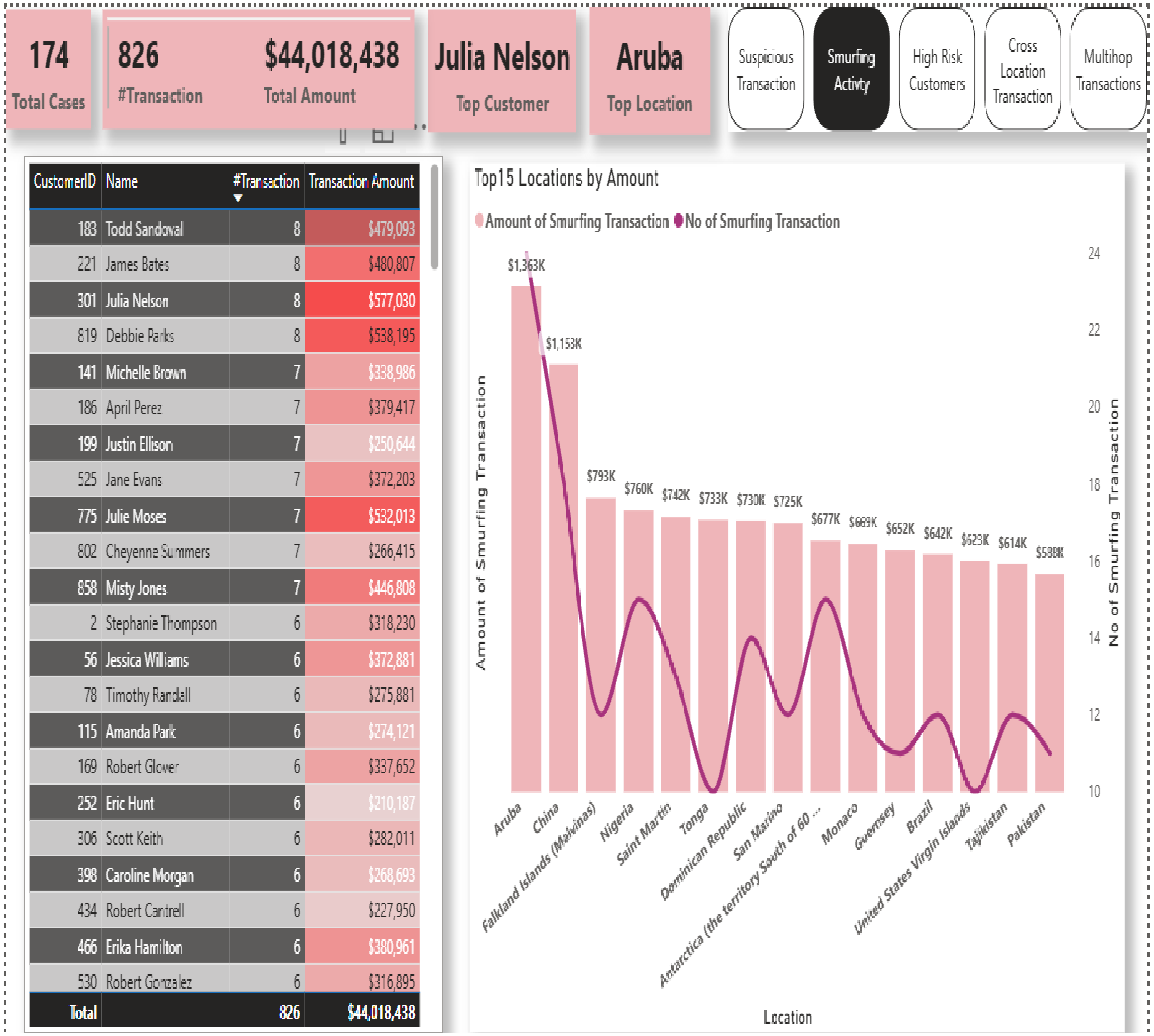
- **Aggregates transaction amounts per customer.**
- **Flags customers exceeding 100,000 in transactions.**

•Use Case:

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



```sql
/*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;
```



174 Total Cases

826 #Transaction

$44,018,438 Total Amount

Julia Nelson Top Customer

Aruba Top Location

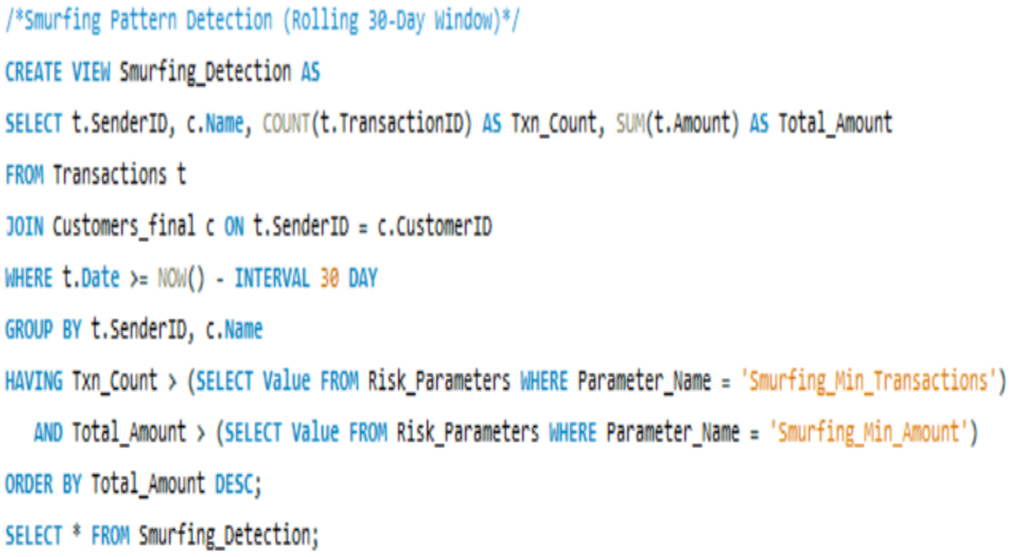
CustomerID	Name	#Transaction	Transaction Amount
183	Todd Sandoval	8	$479,093
221	James Bates	8	$480,807
301	Julia Nelson	8	$577,030
819	Debbie Parks	8	$538,195
141	Michelle Brown	7	$338,986
186	April Perez	7	$379,417
199	Justin Ellison	7	$250,644
525	Jane Evans	7	$372,203
775	Julie Moses	7	$532,013
802	Cheyenne Summers	7	$266,415
858	Misty Jones	7	$446,808
2	Stephanie Thompson	6	$318,230
56	Jessica Williams	6	$372,881
78	Timothy Randall	6	$275,881
115	Amanda Park	6	$274,121
169	Robert Glover	6	$337,652
252	Eric Hunt	6	$210,187
306	Scott Keith	6	$282,011
398	Caroline Morgan	6	$268,693
434	Robert Cantrell	6	$227,950
466	Erika Hamilton	6	$380,961
530	Robert Gonzalez	6	$316,895
Total		826	$44,018,438

Top15 Locations by Amount

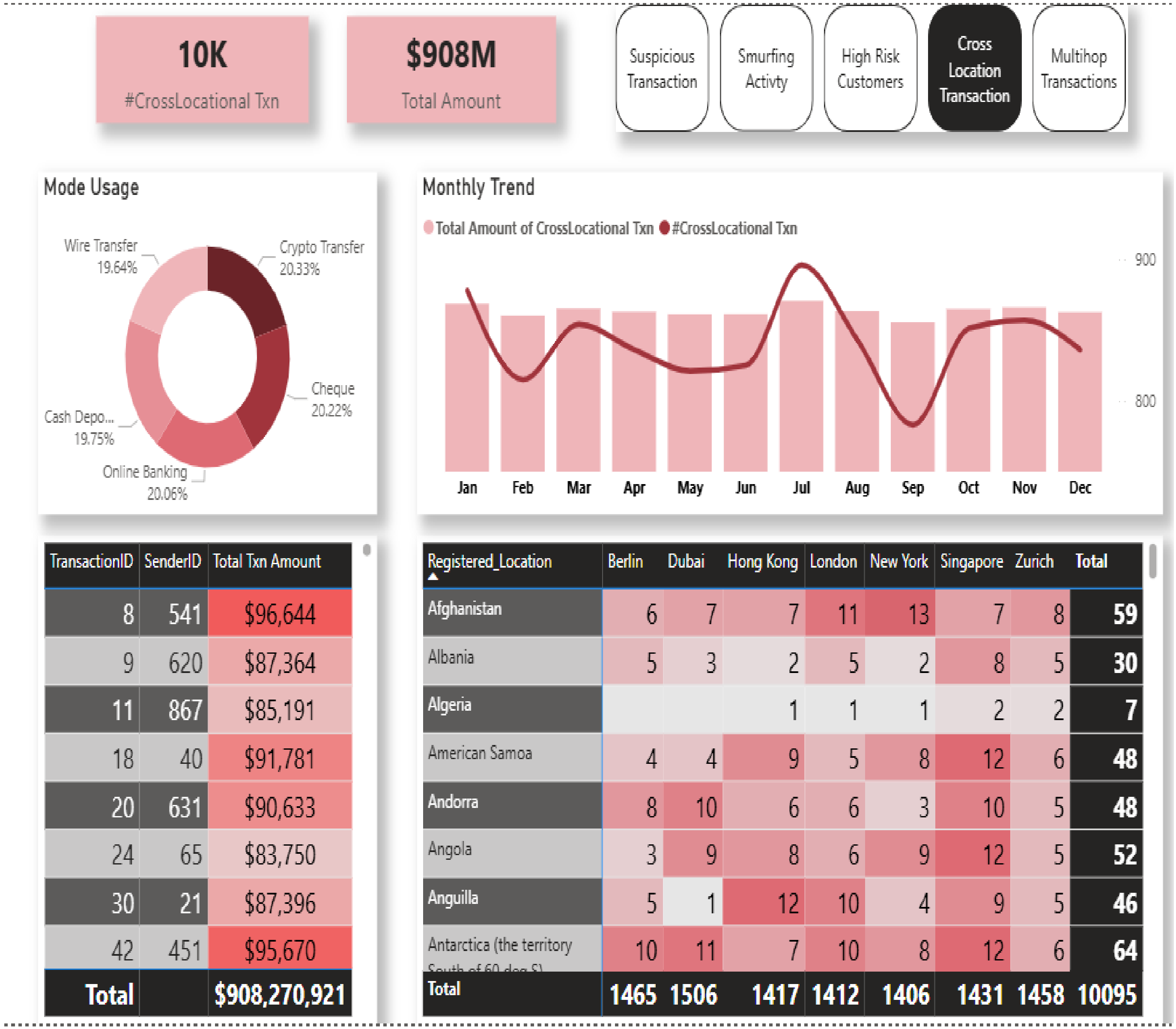
● Amount of Smurfing Transaction ● No of Smurfing Transaction
Amount of Smurfing Transaction
No of Smurfing Transaction
24
22
20
18
16
14
12
10
$1,363K
$1,153K
$793K
$760K
$742K
$733K
$730K
$725K
$677K
$669K
$652K
$642K
$623K
$614K
$588K
Aruba
China
Falkland Islands (Malvinas)
Nigeria
Saint Martin
Tonga
Dominican Republic
San Marino
Antarctica (the territory South of 60 ...
Monaco
Guernsey
Brazil
United States Virgin Islands
Tajikistan
Pakistan
Location

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**



```
/*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;
```



10K
#CrossLocational Txn

$908M
Total Amount

TransactionID	SenderID	Total Txn Amount
8	541	$96,644
9	620	$87,364
11	867	$85,191
18	40	$91,781
20	631	$90,633
24	65	$83,750
30	21	$87,396
42	451	$95,670
Total		$908,270,921

Registered_Location	Berlin	Dubai	Hong Kong	London	New York	Singapore	Zurich	Total
Afghanistan	6	7	7	11	13	7	8	59
Albania	5	3	2	5	2	8	5	30
Algeria			1	1	1	2	2	7
American Samoa	4	4	9	5	8	12	6	48
Andorra	8	10	6	6	3	10	5	48
Angola	3	9	8	6	9	12	5	52
Anguilla	5	1	12	10	4	9	5	46
Antarctica (the territory South of 60 deg S)	10	11	7	10	8	12	6	64
Total	1465	1506	1417	1412	1406	1431	1458	10095

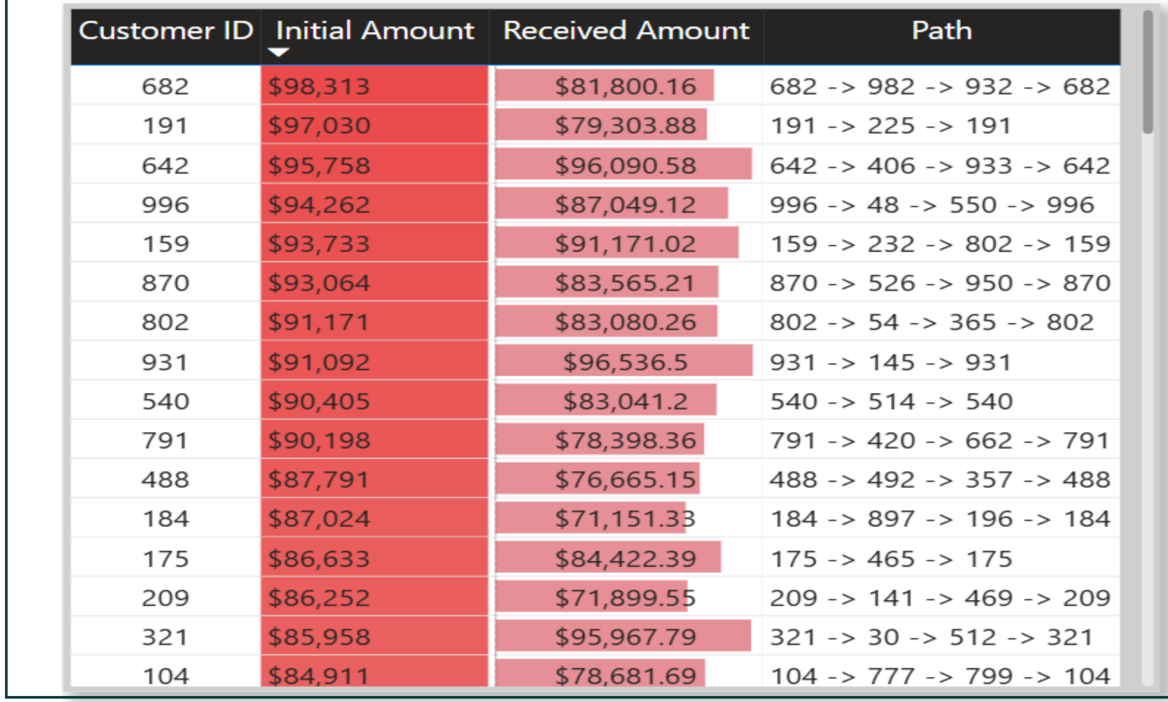
Cross-Location Transaction Analysis

- **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**

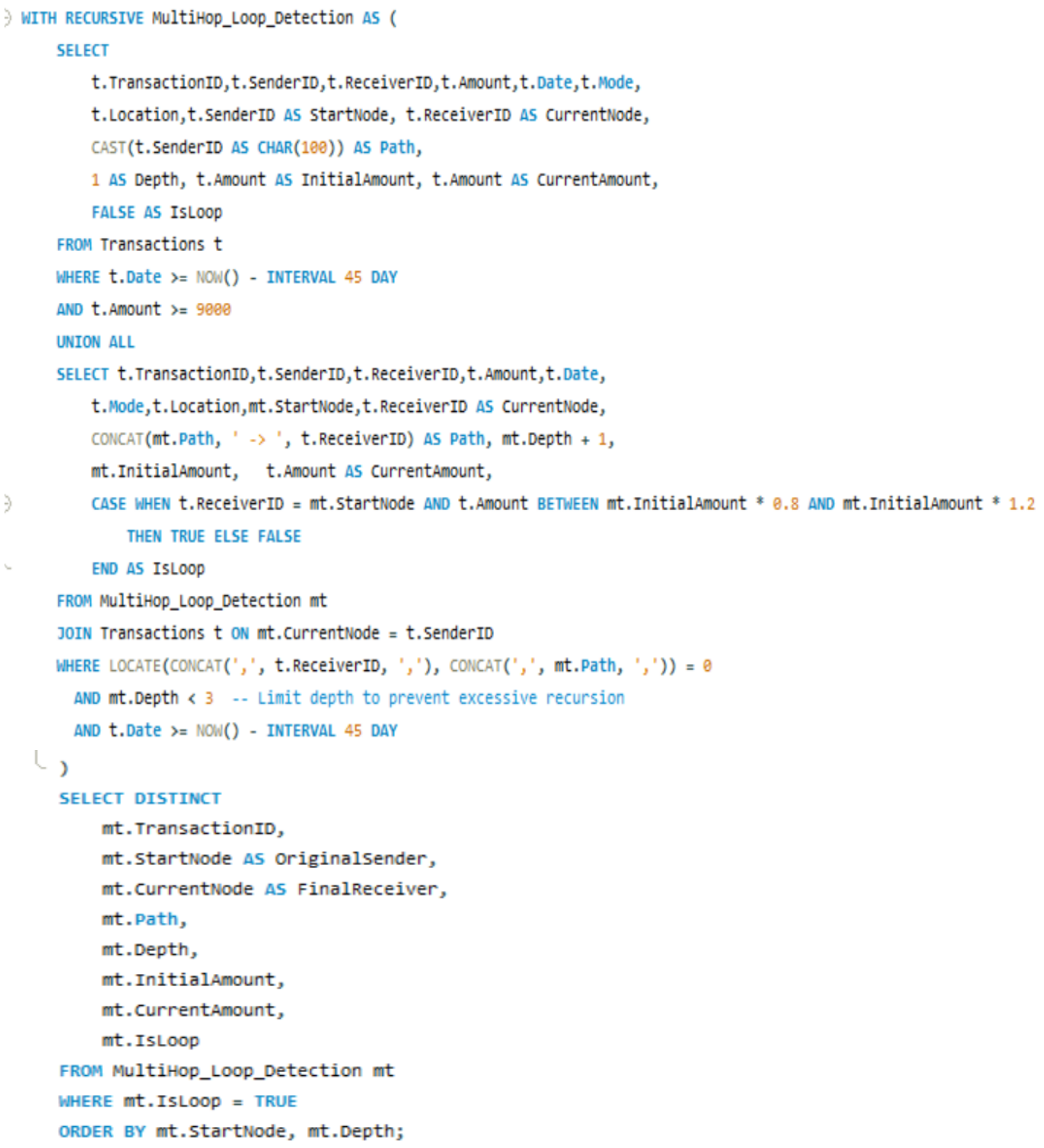
```
/*Cross-Location Transaction Analysis (Geospatial Risk Insight)*/
CREATE VIEW Cross_Location_Transactions AS
SELECT t.TransactionID, t.SenderID, t.ReceiverID, t.Amount, t.Date, t.Mode,
       t.Location AS Txn_Location, c.Location AS Registered_Location
FROM Transactions t
JOIN Customers_final c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

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



Customer ID	Initial Amount	Received Amount	Path
682	$98,313	$81,800.16	682 -> 982 -> 932 -> 682
191	$97,030	$79,303.88	191 -> 225 -> 191
642	$95,758	$96,090.58	642 -> 406 -> 933 -> 642
996	$94,262	$87,049.12	996 -> 48 -> 550 -> 996
159	$93,733	$91,171.02	159 -> 232 -> 802 -> 159
870	$93,064	$83,565.21	870 -> 526 -> 950 -> 870
802	$91,171	$83,080.26	802 -> 54 -> 365 -> 802
931	$91,092	$96,536.5	931 -> 145 -> 931
540	$90,405	$83,041.2	540 -> 514 -> 540
791	$90,198	$78,398.36	791 -> 420 -> 662 -> 791
488	$87,791	$76,665.15	488 -> 492 -> 357 -> 488
184	$87,024	$71,151.33	184 -> 897 -> 196 -> 184
175	$86,633	$84,422.39	175 -> 465 -> 175
209	$86,252	$71,899.55	209 -> 141 -> 469 -> 209
321	$85,958	$95,967.79	321 -> 30 -> 512 -> 321
104	$84,911	$78,681.69	104 -> 777 -> 799 -> 104



```sql
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;
```

🧠 Key Findings: AML Risk Analysis Dashboard

Powered by MySQL & Power BI | 45-day transaction window

🔍 1. Suspicious Transaction Detected

•Identified **$9,000–$9,999 transactions** clustering just below reporting thresholds.
•**Top channels:** Online Banking transfers showed elevated frequency of suspicious activity.
•**High-risk zones: New York** is the top location by volume of flagged transactions.

🧮 2. High-Risk Customer Profiles

•Customers with cumulative **outflows exceeding $2500000** flagged for priority review.
•Transaction patterns suggest **potential layering** and structuring tactics.

🧩 3. Smurfing Activity Uncovered

•Several accounts engaged in **>3 micro-transactions** within 30 days totaling over **$150,000**.
•Indicates **possible structuring behavior** to evade single transaction thresholds

.

🔁 4. Multi-Hop Transaction Loops

•Traced funds flowing through **2–3 intermediary accounts**, eventually returning to origin.
•Looping transactions-maintained **value consistency within ±20%**, a classic **layering red flag**.

🌍 5. Cross-Location Transaction Anomalies

•Detected **high-value ($80K+) transfers** originating from customers operating in locations **different from their registration**.

•Suggests **potential proxy usage**, identity misuse, or transactional laundering.

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