Money Laundering Detection based on Neo4j

Cheng Liang , email: a812789400@gmail.com

Информационные системы и технологии
Информационные системы и сетевые технологии

Аннотация. *Money laundering is a criminal activity, usually related to illegal and criminal activities such as drug trafficking, corruption, and bribery. Money launderers obtain a huge amount of illegal funds through criminal activities. This part of the illegal funds cannot be directly deposited in the bank. To avoid detection by relevant departments, money laundering the means of the people are endless. This work is to analyze common money laundering cases and make corresponding tests for different money laundering scenarios. The detection system is based on the Neo4j database.*

Ключевые слова: *Neo4j, money laundering, docker, detection system.*

Abstract

With the rapid development of the information age and the financial industry, money laundering methods are also changing with each passing day. Due to the large amount of financial data, multiple types of data, deep data depth, and complex relationships between various data, it is impossible to be timely and effective in large amounts of data. The discovery of money laundering. The anti-money laundering detection system is an identification service, which is a risk identification of suspicious funds such as large amounts of funds but few but many transfers of funds. The core idea is to use a large amount of data collection, analysis, and processing, establish various models, prevent risks in advance, and take corresponding measures based on this.

With the rapid growth of data volume, data analysis is becoming more and more complex, mode changes are becoming more frequent and query time is longer and longer. For various reasons, traditional relational databases are quite difficult to deal with. The monitoring of anti-money laundering suspicious funds does face problems such as a large amount of data analysis, low timeliness, rapid transfer of suspicious funds, and concealed transaction paths that are difficult to track [1].

Based on the above background, this article chooses to introduce and use a graphic database to design an anti-money laundering detection system. Graph database is a type of NoSQL database, which uses graph theory to

store the relationship information between entities. The graph database is a non-relational database, which uses graph theory to store the relational information between entities.

The most common example is the relationship between people in social networks. The effect of relational database for storing "relational" data is not good. Its query is complicated, slow, and exceeds expectations. The unique design of graph database just makes up for this defect [2].

1. Money laundering

Money laundering refers to the act of concealing, concealing, and transforming illegal income obtained from crimes or other illegal activities to legalize it in form. The crime of money laundering refers to the perpetrator's violation of the relevant provisions of the criminal law of our country. Crimes constituted by committing the above-mentioned money laundering behaviors [3].

The increasing digitization of financial transactions provides an easier way for money launderers to launder money. The inherent nature and complexity of financial money laundering make it more difficult for researchers to track and ultimately prevent fraud in the financial sector. Existing audit system organizations fail to identify money laundering activities more effectively, such as fragmented money laundering in this part, money launderers use their friends, relatives, etc. to create many accounts, and use these fragmented accounts to make multiple transfers. This has led to an increase in the demand for research in the field of anti-money laundering testing.

2. Project Goal

The goal of this work is to establish a bank card money laundering test. The detection system constructs corresponding detection scenarios by analyzing common money laundering patterns in society. This includes creating corresponding data, analyzing business logic to build code, and finally finding out the account information of money launderers or money laundering gangs.

3. Neo4j Database

Neo4j is a NoSQL database written in Java and Scala, and it is also a high-performance graph database. Graph database is an important branch of non-relational database, used to store large-scale structured or unstructured data. In a graph database, each object can be represented by a vertex, and the relationship between objects and objects is represented by edges [4]. Both vertices and edges can set attributes, and each object and relationship can have one or more attributes. Vertices and edges can build a relational network

graph [5]. Neo4j is excellent in developing documents, transaction concepts, actual project application maturity, open source, etc. Neo4j supports structured or unstructured data types and has efficient query and retrieval performance.

Neo4J will be used as the database of this system.

4. Typical Scenario

In December 2019, the police arrested a criminal who was engaged in money laundering activities and later found out that he had obtained 200,000 illegal cash through drug trafficking. His money laundering plan is typical, and this is how he works:

He illegally obtained 10 bank cards with no information connection through purchase, stealing, etc.

He divided 200,000 illegal cash into 10 shares of 200,00.

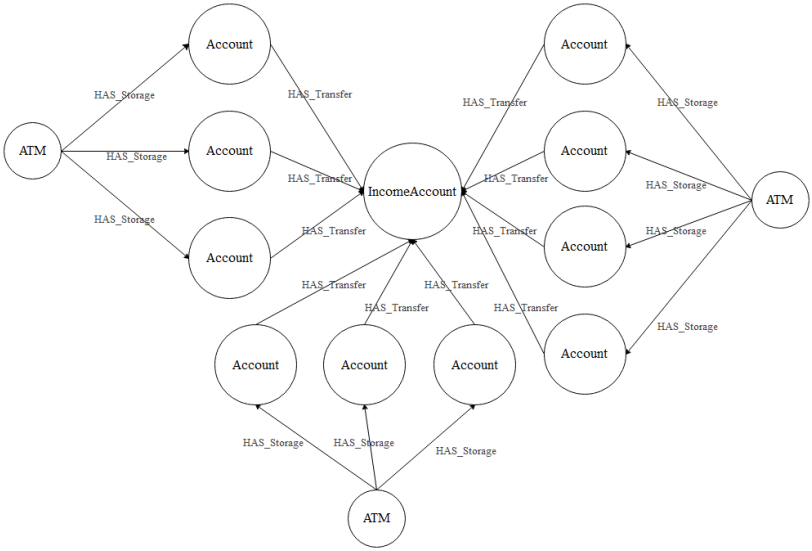
He deposited the money on 10 bank cards on three different ATM machines.

Finally, he transfers all the money to a bank card.

The graph database can help find money launderers faster. By representing the transaction as a graph, we can find the beneficiaries of money laundering in the case.

The interaction between ATM machines and bank cards can be represented by graphs.

The graph data model below represents what the data looks like as a graph.



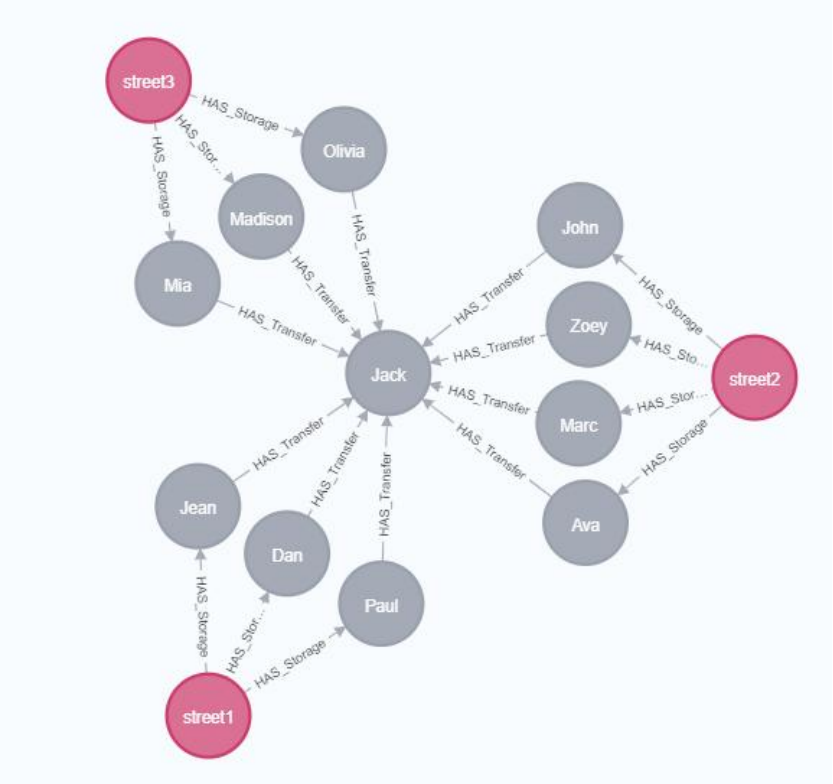
5. Sample Data Set

Query 1

```
// Create accounts
CREATE (Acc1:Account {id:'1', name:'Paul', gender:'man', age:'50'})
CREATE (Acc2:Account {id:'2', name:'Jean', gender:'man', age:'48'})
CREATE (Acc3:Account {id:'3', name:'Dan', gender:'man', age:'23'})
CREATE (Acc4:Account {id:'4', name:'Marc', gender:'man', age:'30'})
CREATE (Acc5:Account {id:'5', name:'John', gender:'man', age:'31'})
CREATE (Acc6:Account {id:'6', name:'Zoey', gender:'woman', age:'52'})
CREATE (Acc7:Account {id:'7', name:'Ava', gender:'woman', age:'23'})
CREATE (Acc8:Account {id:'8', name:'Olivia', gender:'woman', age:'58'})
CREATE (Acc9:Account {id:'9', name:'Mia', gender:'woman', age:'51'})
CREATE (Acc10:Account {id:'10', name:'Madison', gender:'woman', age:'37'})
// Create ATMS
CREATE (Atm1:ATM {id:'1', address:'street1'})
CREATE (Atm2:ATM {id:'2', address:'street2'})
CREATE (Atm3:ATM {id:'3', address:'street3'})
// Create Money laundering income account
```

```
CREATE (InAcc:Account {id:'11', name:'Jack', gender:'man', age:'24'})
// Create storage history
CREATE (Atm1)-[:HAS_Storage {amount:'20000', time:'12/17/2021'}]->(Acc1)
CREATE (Atm1)-[:HAS_Storage {amount:'20000', time:'12/17/2021'}]->(Acc2)
CREATE (Atm1)-[:HAS_Storage {amount:'20000', time:'12/17/2021'}]->(Acc3)
CREATE (Atm2)-[:HAS_Storage {amount:'20000', time:'12/18/2021'}]->(Acc4)
CREATE (Atm2)-[:HAS_Storage {amount:'20000', time:'12/18/2021'}]->(Acc5)
CREATE (Atm2)-[:HAS_Storage {amount:'20000', time:'12/18/2021'}]->(Acc6)
CREATE (Atm2)-[:HAS_Storage {amount:'20000', time:'12/19/2021'}]->(Acc7)
CREATE (Atm3)-[:HAS_Storage {amount:'20000', time:'12/19/2021'}]->(Acc8)
CREATE (Atm3)-[:HAS_Storage {amount:'20000', time:'12/19/2021'}]->(Acc9)
CREATE (Atm3)-[:HAS_Storage {amount:'20000', time:'12/19/2021'}]->(Acc10)
// Create transfer history
CREATE (Acc1)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc2)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc3)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc4)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc5)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc6)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc7)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc8)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
CREATE (Acc9)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)
```

CREATE (Acc10)-[:HAS_Transfer {amount:'20000', time:'12/20/2021'}]->(InAcc)



6. Query

//Storage information

MATCH (ATM)-[r:HAS_Storage]->(Accout)

RETURN ATM.address AS `ATM_Street`, Accout.name AS `Accout_Name`, r.amount AS Amount ORDER BY r.time

	ATM_Street	Accout_ Name	Amount
1	"street1"	"Dan"	"20000"
2	"street1"	"Jean"	"20000"
3	"street1"	"Paul"	"20000"
4	"street2"	"Zoey"	"20000"
5	"street2"	"John"	"20000"
6	"street2"	"Marc"	"20000"
7	"street2"	"Ava"	"20000"
8	"street3"	"Madison"	"20000"
9	"street3"	"Mia"	"20000"
10	"street3"	"Olivia"	"20000"

```
//Query transfer information
MATCH (n:Account)-[r:HAS_Transfer]->(m:Account)
RETURN m.name AS `Suspects`, r.time AS `Transfer_Time`, r.amount AS Amount, n.name AS `Accout_ Name`
ORDER BY `Transfer_Time`
```

	Suspects	Transfer_Time	Amount	Accout_ Name
1	"Jack"	"12/20/2021"	"20000"	"Paul"
2	"Jack"	"12/20/2021"	"20000"	"Jean"
3	"Jack"	"12/20/2021"	"20000"	"Dan"
4	"Jack"	"12/20/2021"	"20000"	"Marc"
5	"Jack"	"12/20/2021"	"20000"	"John"
6	"Jack"	"12/20/2021"	"20000"	"Zoey"
7	"Jack"	"12/20/2021"	"20000"	"Ava"
8	"Jack"	"12/20/2021"	"20000"	"Olivia"
9	"Jack"	"12/20/2021"	"20000"	"Mia"
10	"Jack"	"12/20/2021"	"20000"	"Madison"

Conclusion

Graph analysis helps to detect money laundering activities in the increasing number of cases. As criminals have become more sophisticated in money laundering, so have the tools that help counter them. When financial institutions are equipped with graphical analysis, there is no opportunity for money laundering activities.

The cases analyzed in this article are universal, and there are many ways to launder money. Money laundering detection will reserve interface extension functions to provide corresponding solutions for different money laundering scenarios. For example, gang crimes, online gambling, and other cases. In the future, I will enrich the function and completeness of the system and add more case models.

Bibliography

1. He Jing, Yang Shenggang, Wu Zhiming. "International Experience and Countermeasures of Anti-Money Laundering". In Theory and Practice of Finance and Economics, 2004, No.25(5), pp.119-124

2. CSDN, “Five graph databases worthy of attention”. In CSDN.com, 2012, No.26(8), pp.2-4..

3. Zhang Yanling. "Research on Anti-Money Laundering in the Financial Industry". In international Finance Research, 2002, No.19(9), pp.10-11.

4. Ma Kaihang, Gao Yongming, Wu Zhiquan. “A review of data management technology research in the era of big data”. In Software, 2015, No.426(10), pp.52-55.

5. Gong Chang. “Technical analysis of NoSQL database under big data [J]”. In Information recording materials, 2018, No.019 (006), pp.118-119