GRAPH DATABASE AND RELATIONAL DATABASE IN DETECTING MONEY LAUNDERING CASES: A LITERATURE REVIEW SYSTEM

DHIKA NUR AISYAH[1], FAUZAN RISQULLAH[2], EDWIN TUNGGORO[3], INDRAJANI SUTEDJA[4]

[1,2,3,4]Information Systems Department. BINUS Undergraduate Program, School of Information Systems, Bina Nusantara University, Jakarta, Indonesia 11480

E-mail: [1]dhika.aisyah@binus.ac.id, [2]fauzan.risqullah@binus.ac.id, [3]edwin.tunggoro@binus.ac.id, [4]indrajani@binus.ac.id

ABSTRACT

Money laundering is the illegal process of making money from criminal activities, such as drug trafficking or terrorist financing, manipulated so that the money appears to come from a legal source. Due to the secretive nature of money laundering, it is difficult to accurately estimate the total money that goes through the money laundering cycle. However, it is estimated that the amount of money laundering in 1 year globally reaches 2-5% of global GDP, or 800 billion - 2 trillion US dollars. The money laundering detection system is currently built using a relational database system, where the system has limitations in terms of performance so that tracking financial transactions in money laundering cases is very difficult. The limitations of this relational database system can be mitigated with the capabilities of the graph database. This is due to the nature of the graph database itself, which uses graphs that represent entities as nodes and relationships between entities as edges. The main advantage of a graph database is that the relations are stored together with data that can be obtained with just one query. Therefore, the purpose of this paper is to determine the performance of graph databases when compared to relational databases in detecting money laundering cases. The method used is by using a System Literature Review of 27 relevant papers. The final result obtained is a graph database capable of tracking the flow of funds in money laundering cases found on the transaction network. The conclusion obtained is that the graph database performance in running queries is better than relational databases, especially processing data that has many relationships.

Keywords: *Graph Database, Money Laundering, Relational Database*

1. INTRODUCTION

Money laundering is the illegal process of making money from criminal activities, such as drug trafficking or terrorist financing, manipulated to make it appear that it came from a legal source. The United Nations (United Nations) [1], estimates that the amount of money laundering in 1 year globally is 2-5% of Global GDP, or $800 billion-$2 trillion US dollars. However, the estimated value is not entirely accurate, this is due to the secretive nature of money laundering itself, making it difficult to estimate the total amount of money that goes through the money laundering cycle. In Indonesia alone there are 102 decisions in money laundering cases that have permanent legal force during 2020, of the 102 decisions 33% of money laundering cases occurred in the DKI Jakarta area with a total of 34 decisions, the Maluku area as many as 9% with a total of 9 decisions, and North Sumatra as much as 8% with a total of 8 decisions.

Based on the results of the report from the Research Team of the Center for Financial Transaction Reports and Analysis (PPATK), the decisions on money laundering are dominated by three predicate crimes, namely narcotics crime, corruption crime, and fraud crime. As for the number, namely narcotics crimes as many as 32 decisions, corruption crimes as many as 22 decisions, and fraud crimes as many as 21 decisions. Meanwhile, money laundering crimes related to corruption dominated the estimated value of losses, which amounted to Rp. 17,350,625,741,999 (93% of the total estimated value of losses in money laundering cases in 2020). Meanwhile, those related to criminal acts of fraud amounted to Rp. 896,163,098,576 (5% of the total estimated value of losses in money laundering cases in 2020) and narcotics crimes of around Rp. 154,887,560,445 (1% of the total estimated value of losses in money laundering cases in year 2020) [2].

At first the money laundering action was carried out physically, where the action was carried out by hiding the source of illegal money which was then manipulated so that the money was seen as legally obtained, this method was of course limited by the ability of the perpetrator to manipulate the physical world. However, currently money laundering has a tendency to use electronic means, thereby reducing the possibility of detecting the illegal money. Plus, the problem of the massive volume of data and the number of transactions in the current banking industry which seems to provide a safe and appropriate place to hide the source of one's funds [4]. Therefore, money laundering actors can easily carry out this motive with people around them.

In general, cases of money laundering or money laundering are carried out by borrowing many names from the bank. It could even be that all the accounts of the perpetrators were used to 'launder' each other's money. The perpetrators move fictitious deposits from one account to another, then create a kind of fake depository to hide transactions between accounts. The suspicious transaction detection system must have a high level of accuracy with near-real time. However, with the problem of the existing transaction volume, if it is wrong to apply this system so that the balance between accuracy and speed cannot be achieved, it is certain that the system will be vulnerable to false positive [3][4].

The money laundering detection system is currently built using a relational database, so tracking financial transactions in money laundering cases is very difficult [3]. This is because in the case of money laundering, there is one characteristic where funds are transferred to many parties before reaching their final destination. If searched using a relational database system, it will take up a lot of resources on the system and will take a very long time. Problems also arise because the current suspicious transaction detection system still relies on a rule-base system where the system can be avoided by perpetrators if they know the rules used in the system [3][6][7][8].

The limitations of the relational database can be mitigated by the capabilities of the graph database. This is due to the nature of the graph database itself, which uses graphs that represent entities as nodes and relationships between entities as edges [3][5][9]. The main advantage of the graph database is that the relation is stored together with the data so that it can be retrieved with only one query [3]. Another advantage is its ability to process large interrelated data so that it can find hidden patterns and relationships between entities more efficiently than relational databases, its agile and flexible structure, the representation of relationships between entities is explicit, queries produce real-time and the speed depends on the number of relationships between entities [5][10].

This research will show how graph database, especially *TigerGraph*, when compared to relational database**.** This is due to the limited previous research in discussing the ability of graph databases using *TigerGraph* tools. The majority of previous studies discussed the capabilities of graph databases using *Neo4j* tools.

The scope of this research is to examine the reliability of the graph database by testing its performance based on query execution time to retrieve a large number of rows. Followed by the other test carried out by executing queries to retrieve data by combining (joining) two and three tables. The next test is to execute queries related to the detection of anomalous transactions which are seen based on the frequency of transactions at one time and based on the amount of the nominal of transactions that exceed the threshold in such a short span of time. The tool used for testing the graph database is *TigerGraph*, while the tool used for testing the relational database is *MySQL*.

The motivation of this study is to conduct research on the differences between the performance of *TigerGraph* and relational databases. The method used to achieve this goal is to use a systematic literature review (SLR), which summarizes previous studies that discuss graph databases and relational databases. The research question from this study is whether *TigerGraph* has the ability to outperform relational databases, the same as other graph database tools? The purpose of this study is to find out how the graph database performs when compared to relational databases in detecting money laundering cases. The outcome measure of this study is the time needed to run the queries that have been prepared for various use cases. The less time it takes to run the query, the better. Because the Anti Money Laundering System must be able to work in synchronous time so that it can run optimally [27].

The contribution of this research to the academic community is to show that there are other graph database tools besides Neo4j that have similar capabilities**.** Contribution to the banking industry is to provide alternative new database tools that banks can use for various things, such as detecting money laundering cases, knowing user behavior, or knowing current trends. This is because the data that need to be processed by the bank usually has a large amount.

2. THEORETICAL FOUNDATION

2.1. Money Laundering

Money laundering is the processing of proceeds of crime to disguise their illegal origin [1]. Money laundering has three cycles, namely entering illegal money into the legal financial system, transferring money that previously entered the legal financial system from one bank to another, then making illegal funds almost legal by being invested or converted into goods. 11]. Money laundering has a number of effects, such as deformation in consumption, creating artificial price growth, affecting a country's economic growth rate, corruption and bribery, increasing crime rates, making the political world unstable [11].

In general, the money laundering process is divided into three stages, this division aims to make it easier to compare, contrast and analyze different methods [23]. The following are the three stages of the money laundering process:

a. Placement
The placement stage is when the actors collect their money somewhere. The collection of this money is usually in the form of cash, transfers, or virtual currency [24].

b. Layering
Layering is the stage where money will be sent to various places continuously from one party to another with the aim of eliminating traces of the source of the money [23].

c. Integration
Money whose origins cannot be traced can be integrated and reused for transactions on the surface [23]. In this phase the money has been returned to the main actors or to the closest people and used to purchase assets or invest [24].

2.2. Database Management System

A database is a system used to manage data on a computer system. Databases can be used to collect data so that collection, storage, maintenance, processing, and security can be carried out effectively and efficiently [25]. A database management system can be defined as software that can be used to make it easier to centralize data, manage data efficiently, and provide data access for application programs. The following functions are owned by the database management system:

a. Maintain data integrity
The database management system can assist in reducing and eliminating data redundancy. In addition, the database management system can also maximize data consistency. That way, when the database management system displays data, the data will match the original data.

b. Data storage management
The database management system has the main function as a data repository. Even database management systems can store data in various types such as videos and images. The database management system has procedures in the storage process and ensures that the data stored matches the data entered.

c. Data dictionary
The database management system can function to manage the elements in the database and how these elements are linked to other data. When a system requires data in the database, the database management system will provide convenience via SQL to access and search for that data. That way, users only need to master SQL to be able to access and search for data in the database.

d. Data transformation and presentation
The database management system can function to perform transformations in data presentation. The database management system can change any data entered in the database into a predetermined structure and format. Thus, the database management system can distinguish between logical data formats and their physical form.

e. Data security
The database management system has an important role in maintaining the security of the data in the database. The database management system can assist in managing user access rights to the database. In addition, the database management system can also manage what users can do with the database.

f. Grant access to multiple users
Database management system can allow multiple users to interact in a database. This function can provide efficiency because there will be many users who can access one database simultaneously. In addition, this function can also assign certain users according to their roles and functions.

g. Provides data backup and recovery procedures
The database management system can allow the data in the database to be backed up. Apart from being backed up, the database management system can also restore data that was previously backed up. This function will be very helpful when something happens to the database such as damage or natural disasters.

h. Provide language and programming access
The database management system provides SQL for manipulating and creating database schemas. Manipulating and creating schemas in databases is also known as Data Manipulation Language (DML) and Data Definition Language (DDL). With this function, users can enter, retrieve, modify, and delete data in the database using the interface in the database management system.

i. Provide an interface for communication between databases
The database management system provides an interface for communicating between one database and another. This function can make it easier for users to access multiple databases. In addition, the database management system can also facilitate communication between databases and other tools such as browsers.

j. Transaction management
The database management system has a mechanism for managing data transactions. This function is intended to ensure the consistency of data in the database. For example, when a database administrator is accessing and deleting data, at the same time if a user accesses that data, the user's access will be delayed until the data is deleted

2.3. Graph Database

Graph database is a data store through the arrangement of nodes, edges, and relationships so as to provide index-free adjacency [12]. Graph databases can well express relationships among individual data elements by creating nodes for each data element and connecting nodes through edges to establish relationships between them. In the graph database it is also possible to query data in real time [13]. Graph databases can reduce storage costs for storing large-scale graphs, although they may require more space when the size of the graph is small [14]. Despite having many advantages, graph databases are not always superior in graph queries. Therefore, the application of graph databases also needs to consider the purpose and context [15]. The following is an example of using graph database [26]:

a. Page rank
Google uses the graph database concept to calculate the order of pages that appear in search results. Graphs are used to connect web pages around the world as nodes and hyperlink each other as edges. The number of edges per graph is defined as the weight on the edges. Thus, page rank is determined based on the weight of a node compared to the weight of other nodes.

b. Data management
Cisco, which is one of the largest networking companies in the world, has adopted a hierarchical management system. The hierarchical management system used by Cisco is centered on the graph utility of the Neo4j database. This gives Cisco very fast access to data when compared to Oracle RAC. Cisco also applies this concept to the product hierarchy to serve users in real time.

c. Social interconnect
Social media sites such as Facebook, Twitter and LinkedIn store connections between users in the form of graph databases as relationships. Recommendations are important for users on social media sites. Relationships in graph databases can be managed very well and accessed in real time compared to relational databases.

d. Network management
The world's telcos are turning to graph databases to model their networks of highly interconnected packets, subscribers, and groups. Graphs can help in analyzing networks and data centers. In addition, graphs can also help reduce network interference cases because they can process data quickly.

e. Security and access management
The capabilities possessed by the graph database can also be utilized in security and access management. An example of a company that utilizes graph databases in security and access management is

Adobe. Adobe uses a graph database structure to link authentication details and also provide access to content for its administrators as well as its users.

f. Bioinformatics
Era7 is an example of a company specializing in DNA sequencing, i.e., storing information about proteins, enzymes, etc. This can be done with the help of Bio4j. Bio4j is a bioinformatics graph database system. Bio4j can store information about genes, proteins, and other complex related information. Bio4j implemented by Era7 has all the features owned by Neo4j.

2.4. Relational Database

The concept of a relational database is that all data can be defined or represented as a series of relationships with or to other data. [16]. Relational databases can make it easy to summarize data, add, update and can delete data and store data constantly. However, conventional relational database systems and platforms are inefficient, with slow response times and lack of scalability, relational databases have difficulty retrieving unstructured data and the relational database schema cannot be modified over time [17].

3. RESEARCH METHOD

Basically, advances in knowledge must be based on pre-existing knowledge. In order to understand the breadth and depth of pre-existing frameworks and to identify gaps to explore, we can use a literature review system. By summarizing, analyzing, and synthesizing a group of related literature, we can test certain hypotheses and/or develop new theories. We can also evaluate the validity and quality of existing work against criteria for revealing weaknesses, inconsistencies, and contradictions [18].

3.1. Literature Search and Evaluation

a. Inclusion criterion
Various studies on graph databases were included, including studies on comparisons between graph databases and relational databases. The included studies come from various disciplines such as information systems, social, artificial intelligence, and health. The studies entered are within the last five years since their publication.

b. Literature identification
Google Scholar, which is usually used for research from various disciplines, can be used to search for literature. When searching for literature with Google Scholar, the keywords used are:

- Graph database OR relational database.
- Graph database AND money laundering.
- *TigerGraph* AND money laundering.
- Relational database AND money laundering.
- Graph database AND efficiency

From a search using these keywords, various papers from various publishers were found, such as Springer, IEEE, Applied Science, Elsevier, Emerald Publishing Limited, and others.

c. Screening for inclusion
In screening the various papers that have been found, the abstract of the paper is read first to find out whether the paper presents topics that are relevant to this research. After being read and analyzed, as many as 27 papers relevant to this research were found. This paper will be used in this research.

d. Quality and eligibility assessment
To conduct a Quality and eligibility assessment, the paper is read in its entirety. The papers used are published papers from publishers who have a good reputation and feasibility. In addition, the paper used is a maximum of five years after publication to ensure the relevance of the paper to today's era.

e. Iteration
The search for papers that will be used as literature is carried out repeatedly. This repeated search is to ensure that papers related to this topic have been included in the paper that will be used for the literature review. More than 50 papers have been read which are then re-sorted so that only those papers are truly in accordance with the research conducted.

3.2. Data Extraction and Analysis
A literature study was conducted on X papers, of which only X papers were in accordance with this study after reading the abstracts from the collection of papers. In the end, 27 papers were found that were suitable and could be used as a reference for this research. From this collection of papers, various information was obtained on the topic of Graph Database. From these topics, it can be used to formulate research problems, search for literature, compare quality, analyze and filter all data. And to compare it using *TigerGraph* software, *MySQL*.

Table 1: Number Studies in Selected Sources

Journal of Theoretical and Applied Information Technology
28th February 2023. Vol.101. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

No.	Source	Studies Found	Candidate Studies	Selected Studies
1	IEEE	75	10	3
2	Elsevier	34	5	1
3	Emerald	27	4	1
4	Springer	62	9	5
5	Other publishers	782	25	17
	Total	980	53	27

4. RESULT

Based on 27 analyzed papers, 15 papers discussed graph databases by using *Neo4j* tools. Most of the papers that use *Neo4j* tools conclude that the time required to execute queries is reduced and has better performance when compared to relational databases. There are only 8% of papers that discuss the advantages of graph databases using *TigerGraph* tools. Even though *TigerGraph* can consistently outperform *Neo4j* on most queries execution performance [19]. Meanwhile, based on the findings of papers published in 2022 which also related to this research, namely the performance comparison between graph database and relational database. There is no research that specifically compares the performance of graph databases and relational databases in executing queries related to money laundering cases by detecting anomalous transactions based on the frequency and nominal value of financial transactions. Even so, the test was carried out using a larger dataset with more diverse entities and attributes. So that the test results will be closer to the actual condition of the existing database on the system that is already running.

This study will show how *TigerGraph's* performance is compared to relational databases, especially in the detection of anomalous transactions and the flow of funds in money laundering cases. This test uses the *TigerGraph* tool as one of the tools in the Graph database, while the relational database uses the *MySQL* tool**.** The platform used for *TigerGraph* is on the cloud platform with 2 cores CPU and 8GB memory (RAM). As for the platform used for *MySQL* is on-premise using the Linux operating system with the same specifications, 2 cores CPU and 8GB memory (RAM). The details of the dataset used in the tests in this paper are as shown in table 4.1.

Table 2: Details of Dataset Used

No.	Entitas	Size (MB)	Rows
1	User	75	999999
2	Transaction	34	999999
3	Payment Instrument	27	999999
	Total	136	2999997

The performance comparison test process between graph database and relational database management system (RDBMS) is carried out on three (3) use cases. The data used is dummy data consisting of one million user records, one million device data, one million payment instrument data and one million transaction data. For the first use case related to data retrieval, this test is divided into four (4) queries, namely, retrieving 1000 rows of data, 10000 data lines, 100000 data lines and 1000000 data lines. While the second use case is retrieving data from several tables that are joined, this use case is divided into two (2) queries, namely, retrieving data from two tables that are joined and retrieving data from three tables that are joined.

The third use case is a case related to money laundering. Referring to the results of research conducted by the research team of the Financial Transaction Reports and Analysis Center (PPATK) in 2020 regarding the typology of money laundering, it is explained about the types of red flags for suspicious financial transactions. One example of the red flag described is breaking up transactions into several accounts in the near future, and another example is making several transactions with large nominal or not in accordance with the customer profile in the near future [2]. Therefore, for testing this third use case, a query will be made to detect anomalies based on transaction frequency and detect anomalies based on nominal transactions.

Table 3: Data Retrieval Use Case Query Runtime Result

Database	1000 Rows	10000 Rows	100000 Rows	1000000 Rows
TigerGraph	529 ms	672 ms	1467 ms	32060 ms
MySQL	15 ms	16 ms	109 ms	1078 ms

Data is much faster than *TigerGraph* if the retrieval process is carried out only from one table without being joined with other tables and not added with conditions. And for the second use case related to retrieving data from joined tables, the first stage of

Journal of Theoretical and Applied Information Technology
28th February 2023. Vol.101. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

testing for this case is a query where the user id input parameter is given to find the payment instrument used by the user. The second query is to find the payment instrument used by the user who makes the transaction.

Table 4: Join Tables Condition Use Case Query Runtime Result

Database	Join 2 Tables w/ Condition	Join 3 Tables w/ Condition
TigerGraph	8 ms	5 ms
MySQL	4297 ms	211248 ms

The third use case is related to red flag transactions related to money laundering. The first query will detect anomalies based on frequency by looking for users who make transactions that exceed the threshold for the number of transactions for a certain time window. This query is related to the red flag described previously, where one of the modus operandi of money laundering actors is to transfer money to multiple accounts in a short period of time [2]. The next query is related to the amount transferred by the user, another modus operandi performed by the user is to transfer money with a significant amount that does not match the user's profile. As for the test for this second query, it will look for outgoing transactions that are anomalous when compared to the spending profile of a user.

Table 5: Anomaly by Frequency Query Runtime Result

Database	Query "anomaly_by_frequency"
TigerGraph	512 ms
MySQL	12341 ms

In this first query test, we search for users whose transactions exceed the predetermined threshold value (9 transactions) within two months (1 March - April 30, 2017). The threshold value and time window selected in this query only aim to have at least one row returned. In fact, the threshold value can be changed according to the needs and the time window value can be made narrower to one month, one week or even one day.

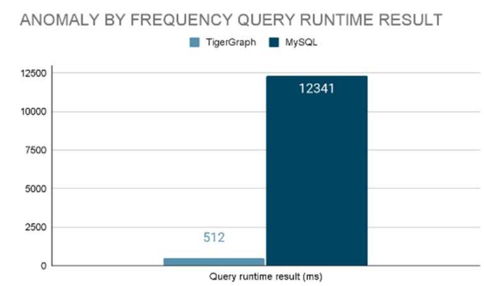

Figure 1: Anomaly by Frequency Query Runtime Result Graph

As seen in figure 1 which describes the test for anomaly checking query based on frequency, the performance time required for *TigerGraph* to run the query is 512 ms, while for *MySQL* to run a similar query it takes 12341ms. The speculations related to the length of time for query execution in *MySQL* are because this query requires several tables join and union. Whereas in *TigerGraph*, this query takes much faster time because the relations between entities have been defined previously, so in the graph database query there is no need to join between entities anymore.

Table 6: Anomaly by Amount Query Runtime Result

Database	Query "anomaly_by_amount"
TigerGraph	4320 ms
MySQL	12265 ms

In this second query test, checks are made on outgoing transactions for a four-month time window (1 January - 30 April 2017) which do not match the spending profile of the user who made the transaction. transaction. This query looks for outliers based on 2.75 standard deviations, which means all transactions that fall outside 2.75 standard deviations will be declared as anomalies. If implemented in real terms, the value of the standard deviation can be adjusted as needed. However, if the standard deviation is lowered, the possibility of false positive anomalous transaction reports will increase. And vice versa if the standard deviation is increased then the possibility of false negative will increase. The query execution time on *TigerGraph* took 4320 ms while on *MySQL* it took 12265 ms. The significant difference in query execution time is assumed to be because there are several tables that are joined to the query in the relational database system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

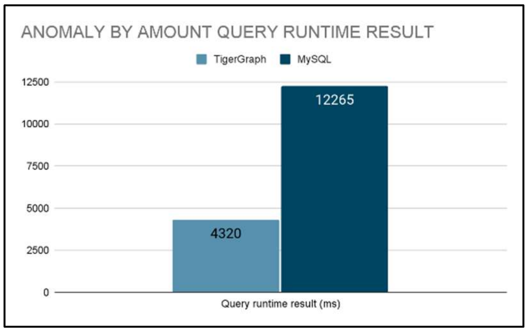

Figure 2: Anomaly by Amount Query Runtime Result Graph

The data represented in the form of a graph structure can provide additional insight [20]. With a graph structure, hidden relationships can be found between entities and semantic information [21]. Therefore, we will present an additional use case that aims to show the flow of funds in money laundering cases. For this test, it is only carried out on the graph database system and queries related to circle detection have been provided by *TigerGraph* as one of the best practice money laundering prevention systems. In this query, a user code input will be given who is an alleged money launderer, from which it is found that the flow of funds originating from the inputted user is transferred to several users until it returns to the original perpetrator, a technique commonly used in money laundering cases.

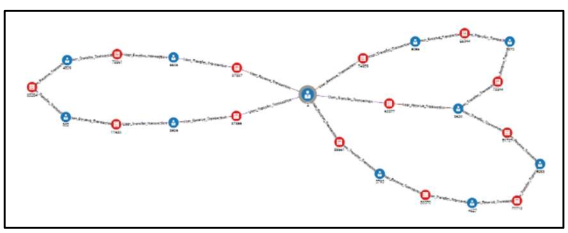

Figure 3: Circle detection query result on TigerGraph

This circle detection query can detect the flow of funds in money laundering cases. This query receives input in the form of a user who is suspected of laundering money and sees the funds that have come out of that user. If there is a revolving fund scheme so that the funds originating from the user who are inputted return to the same user after being transferred several times, then the actors and all transaction details in this case can be clearly identified.

5. LIMITATION

The limitations in this paper are related to the process of testing several use cases that use a small threshold value and a small-time window. This is done because of the limitations that exist in the hardware where the database system is installed. Problems arise when the time window is enlarged, often the query fails to run and displays errors related to excessive memory usage. In addition, limitations come in the dataset that is owned, the use of a small threshold is done because if the threshold value is enlarged then no rows are returned at all.

6. CONCLUSION

With the increasing development of technology, the data generated will also increase. Increasing data will make conventional database performance heavier, and it will take a lot of time to process the data. This study compares the performance of conventional databases such as relational databases with graph databases, to find out how the performance difference between the two databases is.

Based on the test results of the three use cases, it can be concluded that *MySQL* can outperform *TigerGraph* in the case of retrieving data without using table joins. However, if there is one or more joins between tables, such as in the second case regarding data retrieval by joining table with conditions and the third case regarding finding financial transaction anomalies, then *TigerGraph* is superior to *MySQL*. *MySQL* will be suitable if the data stored does not have many relationships such as minimarket employee data or restaurant sales data. While the graph database will be suitable for use if the data stored is very large and has complex relationships such as banking data that has many transactions and integrations with other systems or application user data that is routinely performed by user behaviour analysis.

Because in this paper we test the performance between relational database and graph database, for future research, we recommend experimenting with different graph database tools. And as for the test, future research can perform the test using the use cases that already exist in this paper. Future research will be aimed to find out which graph database tool is best at detecting financial transaction anomalies, especially in the case of money laundering.

REFERENCE

[1] United Nations Office on Drugs and Crime. Money Laundering [Internet]. [cited: 2022 August 07]. Available from:

https://www.unodc.org/unodc/en/moneylaundering/overview.html.

[2] Directorate of Inspection and Research, Deputy for Eradication, Center for Financial Transaction Reports and Analysis. (2021). Typology of Money Laundering Based on Court Decisions on Cases of Money Laundering.

[3] El-Banna, MM, Khafagy, MH, & El Kadi, HM (2020). *Smurf Detector: a Detection technique of criminal entities involved in Money Laundering. 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE).*

[4] Labib, Nevine & Rizka, Mohamed & Shokry, Amr. (2020). *Survey of Machine Learning Approaches of Anti-money Laundering Techniques to Counter Terrorism Finance*.

[5] Lysenko, A., Roznovăţ, IA, Saqi, M., Mazein, A., Rawlings, CJ, & Auffray, C. (2016). Representing and querying disease networks using graph databases. *BioData mining*, *9*(1), 1-19.

[6] Garcia-Bedoya, O., Granados, O., & Cardozo Burgos, J. (2020). *AI against money laundering networks: the Colombian case. Journal of Money Laundering Control, 24(1), 49–62.*

[7] Sarma, D., Alam, W., Saha, I., Alam, MN, Alam, MJ, & Hossain, S. (2020). *Bank Fraud Detection using Community Detection Algorithm. 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA).*

[8] Li, X., Liu, S., Li, Z., Han, X., Shi, C., Hooi, B., Huang, H., & Cheng, X. (2020). FlowScope: Spotting Money Laundering Based on Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 4731-4738.

[9] Deutsch, A., Xu, Y., Wu, M., & Lee, V. (2019). *Tigergraph: A native MPP graph database*. *arXiv preprint arXiv:1901.08248*.

[10] Dancuk, M. (2022, July 11). *What is a graph database? {definition, use cases & benefits}*. Knowledge Base by phoenixNAP. Retrieved October 12, 2022, from https://phoenixnap.com/kb/graph-database

[11] Gjoni, M., Gjoni, AK, & Kora, HB (2015). Money laundering effects.

[12] Paul, S., Mitra, A., & Koner, C. (2019, March). A Review on Graph Database and its representation. In *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)* (pp. 1-5). IEEE.

[13] Kundu, G., Mukherjee, N., & Mondal, S. (2020, May). Building a graph database for storing heterogeneous healthcare data. In *International Conference on Information and Communication Technology for Intelligent Systems* (pp. 193-201). Springer, Singapore.

[14] Chen, Y. (2016). *Comparison of Graph Databases and Relational Databases When Handling Large-Scale Social Data* (Doctoral dissertation, University of Saskatchewan).

[15] Timón-Reina, S., Rincón, M., & Martínez-Tomás, R. (2021). An overview of graph databases and their applications in the biomedical domain. *Databases*, *2021*.

[16] Dilling, TJ (2020). Artificial intelligence research: the utility and design of a relational database system. *Advances in Radiation Oncology*, *5*(6), 1280-1285.

[17] Sahi, B., & AL-Khafaji, N. (2022). NoSQL: Will it be an alternative to a relational database? MySQL vs MongoDB comparison.

[18] Xiao, Y., & Watson, M. (2019). Guidance on conducting a systematic literature review. *Journal of planning education and research*, *39*(1), 93-112.

[19] Rusu, F., & Huang, Z. (2019). In-depth benchmarking of graph database systems with the Linked Data Benchmark Council (LDBC) Social Network Benchmark (SNB). *arXiv preprint arXiv:1907.07405*.

[20] Xirogiannopoulos, K., & Deshpande, A. (2017). *Extracting and Analyzing Hidden Graphs from Relational Databases. Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17.*

[21] Wang, R., Yan, Y., Wang, J., Jia, Y., Zhang, Y., Zhang, W., & Wang, X. (2018). *AceKG. Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18.* doi:10.1145/3269206.3269

[22] Truskowski, W., Klewek, R., & Skublewska-Paszkowska, M. (2020). Comparison of MySQL, MSSQL, PostgreSQL, Oracle databases performance, including virtualization. *Journal of Computer Sciences Institute*, *16*, 279-284.

[23] Ferguson, G. (2018). *Global corruption: Law, theory & practice*. https://dspace.library.uvic.ca/bitstream/handle/1828/9253/Ch.%2004_April2018_web.pdf?sequence=5&isAllowed=y

[24] FATF (2018), Professional Money Laundering, FATF, Paris, France, www.fatf-gafi.org/publications/methodandtrends/documents/professional-money-laundering.html
[25] Meiryani, A.S. (2019). Database Management System. International Journal of Scientific & Technology Research, 8.
[26] ShefaliPatil, G., & Bhatia, A. (2014). Graph databases-an overview. *1Student, ME Computers, Terna College of Engg, Navi Mumbai*, *2*, 657-660.
[27] Eifrem, E. (2019). How graph technology can map patterns to mitigate money-laundering risk. *Computer Fraud & Security*, *2019*(10), 6-8.