Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 5 (2023) 1088 - 1097 ISSN Media Electronic: 2580-0760

Modeling Metadata and Data from Censuses and Surveys with Graph Databases

Alya Faradila[1], Lutfi Rahmatuti Maghfiroh[2]
[1,2]Department of Statistical Computing, Politeknik Statistika STIS, Jakarta, Indonesia
[1]221910944@stis.ac.id, [2]lutfirm@stis.ac.id

Abstract

Relational database users are switching to non-relational databases because non-relational databases are better able to handle dynamic data storage. One of the institutions that require dynamic data storage is Statistics Indonesia (BPS). Currently, data storage for census and survey activities at BPS is done using a relational database, even though there are metadata changes in each activity. Accommodating metadata changes in each activity requires one database, which creates problems when retrieving some raw data. There is an opportunity for convenience if the data collected is stored in a non-relational database, one of which is a graph database. This research discusses the modeling of metadata and data from censuses and surveys at BPS using a graph database. Followed by implementation on Neo4j DBMS and comparing the proposed model with the relational model on Microsoft SQL Server DBMS. Then a comparison of the features and characteristics of each DBMS is done, and finally, performance testing is done with Apache JMeter. Modeling has been able to handle dynamic data structure changes, but Neo4j's performance is still lagging behind Microsoft SQL Server.

Keywords: graph database; database relational; database modelling

1. Introduction

Along with the development of technology, the need for databases is increasingly diverse. Users who used to use relational-based databases with the standard query language, SQL, are now shifting to non-relational databases such as Facebook, Google, Amazon, and others[1]. The shift is not without reason, database users need a database that can handle their flexible data structure [2] and dynamic data [3] needs which will experience limitations in relational databases.

Relational databases are based on the ACID model, namely Atomicity which guarantees the completeness of transactions, Consistency which guarantees stability in a predefined schema, Isolation which guarantees the independence of transactions executed at the same time, and Durability which ensures that stored transactions do not change state even when they fail [2], [4]. Relational databases with a consistent schema cause the storage schema to be defined at the beginning, which in the event of a need to change the storage schema requires the schema design stage to return This will be a problem if schema changes occur regularly. Therefore, relational databases are not reliable in handling dynamic data structure needs. In contrast to non-relational databases do not require data completeness and do not require a stable structure. So, if there are additional types of data that need to be stored in the future, there is no need to think about overhauling the schema in the existing database.

One of the institutions that have dynamic data storage needs is Statistics Indonesia (known as BPS). BPS is a non-ministerial institution that plays a role in providing data for the government and society in Indonesia[5]. Some of the ways to obtain these data are by conducting censuses and surveys, either conducted by BPS itself, or other government institutions. Some censuses and surveys conducted by BPS are routine and some are conducted only when there are certain data needs. One of the censuses conducted routinely is population census (known as SP) which is conducted every 10 years to collect population resident data in Indonesia and one of the surveys conducted routinely is the National Employment Survey (known as SAKERNAS) which is conducted twice a year, namely in February and August [6], [7].

Currently, the results of SP and SAKERNAS data collection are stored using a relational database on Microsoft SQL Server. Besides BPS, many companies in Indonesia also use relational database models such as

Accepted: 14-07-2023 | Received in revised: 28-09-2023 | Published: 30-09-2023

Telkom, Pertamina, and PLN [8]. The storage begins with the creation of a schema and structure that can store the results of the data collection. However, there is a problem that occurs when this data storage activity is carried out, namely the problem of variable metadata that continues to change both in terms of values, as well as concepts and definitions. As seen in

Table 1, Of the 3 census activities conducted, there are questions that continue to appear, one of which is the question regarding the religion of the respondent. The activity is the same activity and is carried out in different periods, but the value of the variable continues to change. Likewise, the concepts and definitions of the variables in each activity were carried out.

Table 1. Comparison of Answer Items on Religion Questions

Difference	SP2000	SP2010	SP2020
Is the item being asked?	Yes	Yes	Yes
Value	1. Islam 2.Catholic 3.Protestant 4. Hindu 5.Buddhism 6. Other	1. Islam 2. Catholic 3. Protestant 4. Hindu 5. Buddhism 6.Khonghucu 7. Others	1. Islam 2. Christian 3. Catholic 4. Hindu 5.Buddhism 6.Khonghucu 7. Believers 8. Others

Table 2. Comparison Table of Answer Items on Questions on Relationship with Head of Household and Head of Family

Difference	SP2020	SAK2015	SAK2020
Is the item being asked	Yes Family	Yes, household	Yes, household
Value	1. Head of the family 2. Husband 3. Wife 4. Child 5. Daughter in-law 6. Grandchildren 7. Parents 8. Parents In-laws 9. Other family 10. Maid 11. Other	1. Head of household 2. Wife/ Husband 3. Children 4. Daughter in-law 5. Grandchildren 6. Parents/In-law 7. Other family 8. Domestic Helper 9. Other	1. Head of household 2. Wife/ Husband 3. Child/ children 4. Step/ adopted child 5. Daughter in-law 6. Grandchildren 7. Parents/ in-laws 8. Other family 9. Domestic helper 10. Driver/ gardener 11. Others (people who are not related to the head of the household)

As seen in Table 2 there are questions that are asked in each activity, but they have different definitional concepts between periods. For example, SP conducted in 2020 (SP2020) uses a defined concept that leads to families, while SP conducted in 2000 (SP2000), SP conducted in 2010 (SP2010), SAKERNAS conducted in 2015 (SAKERNAS2015), and SAKERNAS conducted in 2020 (SAKERNAS2020) lead to the concept of households. From these changes, there is a need to create a separate database so that it can accommodate changes in the value of each activity carried out. With a separate database, the need to change the data structure can be overcome, but there will be difficulties when retrieving data. Such as the need to request raw census or survey data from several periods. The resulting data must be translated using different metadata for each period of activity. The data retrieval also requires the separation of queries and the resulting data will be fragmented.

From the difficulty of data retrieval, there is an opportunity for convenience if the data that has been collected is stored in a database that is able to handle dynamic structural changes. Non-relational databases can be an alternative to handle dynamic data storage structures [9]. There have been other studies that tried to study survey metadata modeling with document-oriented NoSQL [10], [11]. In the first study, the proposed modeling was carried out using document oriented but no comparison of execution time was made. In the next study, modeling was done with document oriented and a comparison of execution time between the relational model and the proposed model was done. By using a non-relational database, there is no need to formulate the data structure from scratch so that the data collected can be accommodated in one database which will facilitate the data retrieval process. To implement this alternative, this research will compare data storage using a relational database and a non-relational database using data from several censuses and surveys.

Unlike relational databases, non-relational databases have four types of storage methods, namely, key-value, document-oriented, column database, and graph database [12]. This research only focuses on comparing relational database and graph database. Compared to other NoSQL storage methods, a graph database is considered more stable in performance [13]. A graph database is a database storage method using the principles of graphs. Thus, the database will consist of nodes, relations, and properties [14]. Graph databases are considered easier to implement structural changes and have faster query execution when compared to relational databases [15].

To perform a comparison of relational databases and graph databases, a Database Management System (DBMS) is needed, either a DBMS that supports relational data structures or a DBMS that supports graph databases. There are several DBMS that can be used in the construction of this graph-database model, which is commonly called Graph DBMS. Some of which are TinkerPop, DEX, InfiniteGraph, Neo4j, OrientDB, and Titan. In previous research about

DOI: https://doi.org/10.29207/resti.v7i5.5273

comparison of several Graph DBMS, it was concluded that Neo4j and DEX have superior performance compared to other Graph DBMS, but Neo4 tends to have constant superior performance [16]. In addition, Neo4j also has good performance and simplicity in operation [17]. In addition, Neo4j also outperforms MySQL and MariaDB when running complex queries [18]. Thus research will use Neo4j as the graph database. Neo4j is an open-source project consisting of enterprise and development projects that have different characteristics and features from relational DBMS [19].

In this paper, we present some experiment results on graph database modelling which are expected to be able to handle dynamic data structure changes using some census and survey data conducted by BPS. We also include a comparison of the characteristics and features of Neo4j DBMS and Microsoft SQL Server and also performance testing results of graph database model implementation on Neo4j and relational database on Microsoft SQL Server using Apache JMeter.

2. Research Methods

As seen in Figure 1, the research uses an experimental method in making the model, which starts from the idea obtained after going through the problem identification, literature study, and data collection stages, then proceeds with model building and implementation, the last stage is evaluating the results of the experiments carried out. [20], [21].

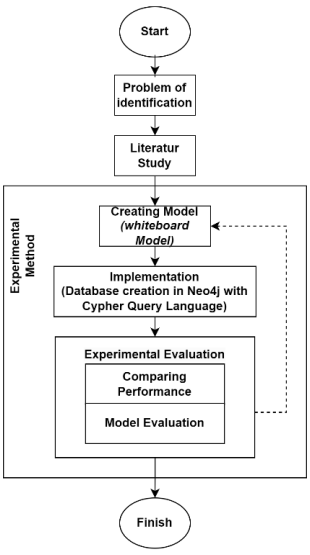

Figure 1. The research process uses Experimental Method

This research begins with identifying problems and finding solutions through literature studies. After the literature is collected, the next stage is to carry out experimental methods. The first stage is to create a model in the form of a whiteboard model using the help of the draw.io application, the second stage is to carry out an implementation in the form of translating image models on Graph DBMS Neo4j using Cypher Query Language. After successful implementation, the next stage is to evaluate the experience by comparing performance and evaluate the model. This experience will be done iteratively until we get the best model.

2.1 Data Collection Methods

This study uses dummy data generated from the census and survey questionnaire tables obtained from BPS. There are five types of questionnaires used, namely: population census (SP) conducted in 2000 (SP2000), in 2010 (SP2010), and in (SP2020) and also National Employment Survey (SAKERNAS) conducted in February 2015 (SAKERNAS 2015) and in February 2020 (SAKERNAS2020). These questionnaires were chosen because they have almost the same database scheme, namely household data collection. So modeling will focus on metadata changes that occur between activities and between periods. Some supporting data is also used, one of which is regional code data obtained from https://sig.bps.go.id/bridging-kode. In this study, each type of census and survey has 21500 rows of data for a total of 107500 and several additional rows for supporting metadata.

2.2 Modeling Method

On the official Neo4j website page, it states that there are 2 stages of creating a graph database model, namely the creation of a whiteboard model and implementation [22]. In the whiteboard model stage, the data that has been collected is grouped so that nodes, relations, and properties are formed. Each node has a name called a label, labels can also be interpreted as a substitute for table names in relational databases. The relation in the graph database serves as a link between nodes, so the join table in the relational database is no longer needed. Properties in a graph database can be found in both the nodes and the relations. These properties have functions that can indicate the uniqueness of a node, the characteristics of a node or relation, and other additional information on both nodes and relations. This modeling is done by drawing the nodes and relationships using an online diagramming application, diagrams.net.

From the whiteboard model that has been made, further modeling can be done by translating the model into cypher language. After translating the data is entered using the .csv format which is then processed by neo4j so that the data can be stored.

2.3 Evaluation Method

Performance comparison testing is done by running several reads, writes, and queries on both DBMS with the help of the Apache JMeter application with the testing scheme as shown in Figure 2 [23]. The read

DOI: https://doi.org/10.29207/resti.v7i5.5273

query uses simple and not simple queries. Whereas the write and delete queries use simple queries.

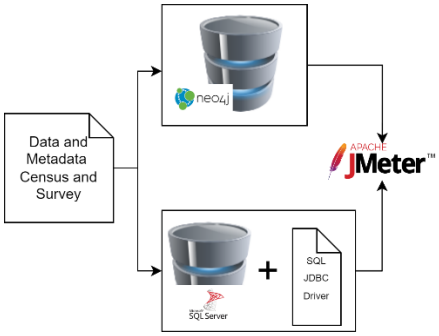

Figure 2. Performance Testing Scheme

Simple queries are queries that involve only one table, while unsimple queries are queries that involve more than one table [24]. After the results from the JMeter application appear, the data is saved and visualized in the form of graphs.

The list of tasks performed to test the performance of the query execution are: Query 1: Display respondent id and age; Query 2: Displays a list of the number of respondents' last education based on the village of residence; Query 3: Displays 1 respondent with a specific id with marital status, latest education, employment status, relationship with the head of household, and religion; Query 4: Displays respondent id, household id, village name, sub-district name, district name, province name, hub desc; Query 5: Displays respondent id, gender, relationship to head of household, respondents with at least high school education; Query 6: Displays the respondent id, household id, village name, sub-district name, district name, province name, relationship to the household head, gender, and religion practiced.

3. Results and Discussions

3.1 Census and survey data and metadata modeling with graph database

From the data that has been obtained and observed with the help of data layouts and questionnaires from BPS, a graph database model is formed as in Figure 3.

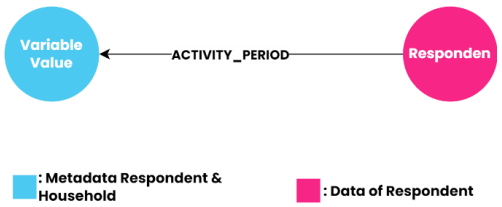

Figure 3. Census and survey data and metadata modeling with graph database iteration 1

The figure is the result of modeling iteration 1 variable values containing nodes that provide answer values that can be selected by respondents. The label name is the variable that is searched for in the activity. After inputting data from 1 activity, there are shortcomings in this model, namely information related to the activities carried out is not included, because information related to activities is included in the relationship, so that each input of relationship data must enter activity information one by one.

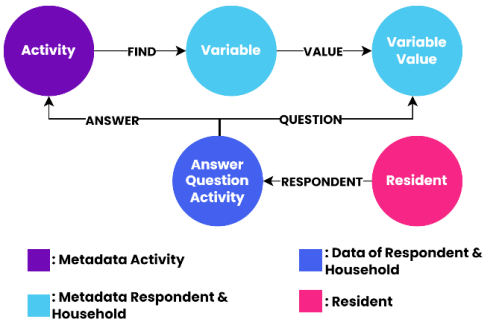

Figure 4. Census and survey data and metadata modeling with graph database iteration 2

In iteration 2, can be seen in Figure 4 that it has been able to handle raw data retrieval directly as expected. Activity metadata contains data related to the activities being carried out, respondent, and household metadata consists of two types of nodes. The variable node contains the definition of the variable, while the VariableValue node is a list of answers that can be selected by respondents from the questions asked. The AnswerActivity node is the respondent data from the enumeration result that is linked to the Activity and ValueVariable. The population node is data on residents who have become respondents so that they have Respondent's Answer data. The population node includes information that cannot be replaced such as place of birth, date of birth, and gender. However, because the analysis that is often carried out also often involves non-replaceable data such as gender and age, it is necessary to streamline the data model by storing this information in the respondent data only.

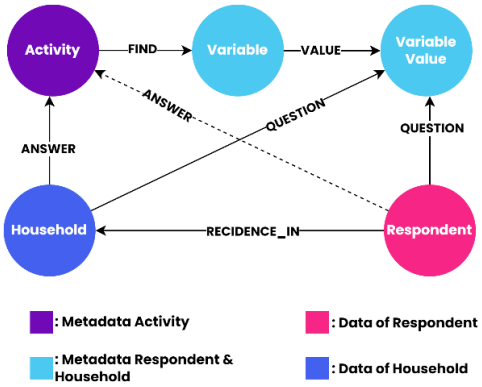

Figure 5. Census and survey data and metadata modeling with graph database iteration 3

Figure 5 is the result of iteration 3 modeling that has been done. The modeling is the result of analyzing the initial needs of the data and metadata stored in the census and survey activities. The result of the modeling

DOI: https://doi.org/10.29207/resti.v7i5.5273

is several nodes with activity labels, respondent and household metadata, household data, and respondent data that are connected using several types of relationships

In this activity metadata, it includes information from the activities carried out. This activity metadata becomes several nodes with the name of the activity label. This node has attributes in it, namely id, name, year, category, date_implementation, and explanation, and other attributes can be added if needed to store information related to activity metadata.

The respondent and household metadata in this modeling includes information related to the answers owned by each respondent and household. This metadata consists of two types of labels, namely *Variable* labels and *VariableValue* labels. The *Variable* label is a variable that is searched for in each survey or census activity. The Variable label has id, name, and description attributes. The name attribute contains the name of the variable being sought, while the description is the concept and definition of the variable.

The *Variable* label consists of many nodes with various labels, according to the variable name in the node with the *Variable* label. For example, in the node with the *Variable* label with id:1, Name: Religion, and Description: The religion practiced by the respondent at the time of the activity. Then in the ValueVariable node there will be a node labeled Religion, with id and description attributes, the id attribute contains a unique number and the description contains an explanation of the id such as Islam, Protestantism, Catholicism, Hinduism, Buddhism, or Confucianism. The answer chosen by each respondent will be addressed directly to the VariableValue node through a relation.

Data on respondents who participated in census or survey activities are stored in this node. In this research, respondent data becomes a node labeled as RespondentSP2000, RespondentSP2010, Respondent SP2020, RespondentSAK2015, and RespondentSAK2020. Each of these nodes has attributes that are only owned by that respondent and tend to be different for each respondent. Namely the attributes id, date_birth, month_birth, year_birth, and age.

Household data is data that contains household information if an activity also collects information related to households. In this research, the nodes in the household data are labeled with the names RutaSP2000, RutaSP2010, RutaSP2020, RutaSAK2015, and RutaSAK2020. In this node, there are also id and jart (what kind of household) attributes or the number of household members.

There are several types of relationships that connect nodes with each other, namely: The SEARCH relation is a relation that connects the node with the Activity label to the node with the Variable label. So that with this relation it can be known what variables are searched for in an activity.

The VALUE relation is a relation that connects the node with the Variable label and the *VariableValue* node. For example, the last education variable will be connected to the node labeled *LastEducation* (last education of respondent Last). So that it can be known what answer options a variable has by using this relation.

The *RECIDENCE_IN* relation is a relation that connects the respondent node and the household node. The *ANSWER* relation is a relation that connects the household node with the activity so that it can be known which households or respondents are participants in an activity. For activities that do not include household information, the respondent node will be directly connected to the activity node through the ANSWER relation. The *QUESTION* relation is a relation that connects the respondent or household with the answer chosen. This question relation can vary according to the question asked. For example, if the question is about the respondent's last education, then the relation is named LAST_EDUCATION.

3.2 Characteristic Analysis

Table 3 shows the results of the comparison of features and characteristics of Microsoft SQL Server and Neo4j sourced from literature reviews obtained from [25], [26] and several other supporting references.

Table 3. Features and Characteristics Comparison Table

Things to compare	Microsoft SQL Server (Relational DBMS)	Neo4j (Graph DBMS)
Owner	Microsoft	Neo Technology
Licenses	Proprietary	GPL v3
Cost	Free and paid	Free and paid
Text Query Language	Structured Query Language	Cypher Query Language
Data model	Relational Database	Graph database
Storage structure	Table	Graph
Index	Index dependency	Index free
Schema	Statics schema	Schema less
Privacy	Not customizable	can be customized
Scalability	File stream based OLTP engine according to user requirements	graph of things, graph of the transaction, dan graph of activity and/or behavior

The owner of Neo4j is Neo Technology which has a gpl3 license so that other parties can participate in developing this DBMS [27]. Microsoft SQL Server is maintained by Microsoft and has a proprietary license that can only be developed by the owner of the application [28]. Cost of DBMS provide free and paid applications. For the standard enterprise version, neo4j's price starts at 65USD/month while Microsoft SQL Server's price starts at 5435USD/year [29], [30].

DOI: https://doi.org/10.29207/resti.v7i5.5273

Neo4j uses cypher query language, by using cypher the user can write queries by describing the graph pattern and the desired relationships. This query has an advantage, if you want to connect many relations at once this query is very easy to use without having to think about the concept of join. Microsoft SQL Server uses a structured query language (SQL), this language is commonly used by various DBMSs, especially those that use relational storage structures. To query multiple tables, joins are required, and SQL joins tend to take a long time to execute.

Neo4j uses a graph data model in its storage. The storage method is done by connecting nodes with relations, each node and relation has a property. Properties in graph databases are generally used to distinguish the contents of each node and relation. Meanwhile, Microsoft SQL Server uses four data model storage options, namely the relational model, graph database, document store, and spatial DBMS.

Microsoft SQL Server uses a storage structure that is likened to a table of tables that have relationships. However, the weakness of this type of storage structure is that it is not possible to know the relationship between rows in the same table. Meanwhile, Neo4j shows the relationship between each node. In each node, there is a label that can be likened to a table name in table form storage. The node can be connected to other nodes even though it has the same label name.

In Neo4j, the search process that connects several tables does not require an index or often with index-free adjacency [31]. By using index-free adjacency, the search process can be done faster because the nodes in Neo4j are connected by relations so that database users can directly point to the relation and the results of the search can be displayed more quickly. This index is optional according to the search needs. Microsoft SQL Server with relational storage uses index dependency to link the relationship of one table to another in searching.

In Microsoft SQL Server, the schema must be created first, whether the data storage used is relational or graph database. At the beginning of the database creation, the model design must be done first and continued with the implementation, which begins with the creation of tables and continues with the design of the relationship between each table. So, if in the future there is a change in the schema, either addition or subtraction, the schema definition stage must be carried out first. In Neo4j, the database storage schema does not need to be created first. Users can start creating it by inputting data first, so that the addition of data can be easily done.

Authorization and authentication on Neo4j both processes are combined with the native auth provider, LDAP auth provider (can be enabled through plugins), single sign-on provider (using the OpenID Connect mechanism) so that it can monitor client behavior centrally, custom-built plugin auth provider (can be requested as needed), Kerberos authentication and single on (network authentication protocol that allows network nodes to show their identity over the network) [32]. And the default native auth provider is set in a system settings file. In this file, we can set how much failure tolerance and time to authenticate. Authorization on Microsoft SQL Server is securable classes (permission on server settings), granular server permission (securable on schema security and setting permissions on the server), and granular database permission (security on the database by defining database users). And the authentication is from Windows or SQL server and occurs at the login [33].

The Neo4j encryption process can be selected using the framework that has been developed by the developer. Encryption uses different methods. For example, using Neo4j Data Encryption with OGM by setting it through the DBMS file system. Whereas Microsoft SQL server uses BitLocker Encryption for drive level, NTFS Encryption for folder level, Transparent Data Encryption for file level, and Backup Encryption File which is also for file level.

Security connection on Neo4j is a configuration that includes Bolt, HTTPS, and HTTP. This connection is more focused on browser connections both locally and on the network. Connection Security on Microsoft SQL Server consists of two features, namely Firewall Protection and Encrypting Data in Transit.

The auditing process on Neo4j is monitoring logs, matrix monitoring, managing queries so that developers can see the process running, managing transactions, managing background jobs, and Monitoring the State of individual databases. Auditing on Microsoft SQL Server is Automated Auditing and Custom Audit. Automated Auditing is called the SQL Server Audit feature which is useful for recording all activities that occur in the database. While Custom Audit is inserted through DDL Triggers and DML Triggers.

There are three types of scalabilities provided by Neo4j. Graph of Things is a database useful for visualizing the treasury data of an object that is not operated continuously and is only used when the user of an application thinks of questions related to the stored data. The graph of transactions is larger in size than the graph of things because this database is required to store all types of transactions made by customers of this database user. The graph of activity and behavior is the highest level of scalability because the database is expected to be able to store activity and metadata related to transactions. In Microsoft SQL Server, increasing scalability is done by optimizing tables in memory. This optimization aims to overcome the database defense from many data requests and perform data recovery on several servers at once. The process is to use a file stream based OLTP engine.

DOI: https://doi.org/10.29207/resti.v7i5.5273

3.3 Performa Comparison

From the experiments that have been carried out, the following are the results of the comparison of execution times that have been carried out on write, delete, and read operations as seen in Figure 6 and 7. The write operation experiment was carried out inputting data from 1000 to 20000 rows.

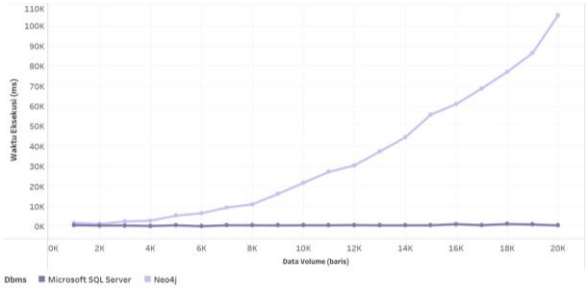

Figure 6. Comparison of write execution time

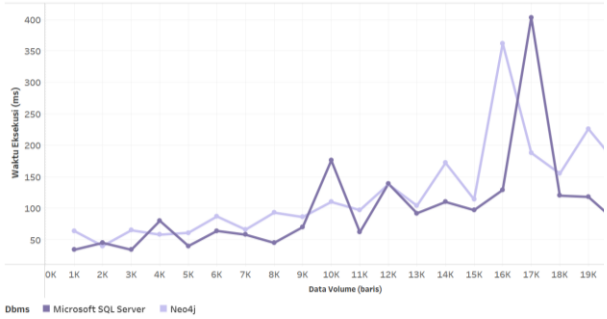

Figure 7. Comparison of delete execution time

In Figure 8, these results that the power of Microsoft SQL Server write operations is far superior to Neo4j. Microsoft SQL Server does not show any significant spike in execution time as the number of rows increases. At the 20000th-row write operation, the speed of Microsoft SQL Server is 200 times faster than Neo4j.

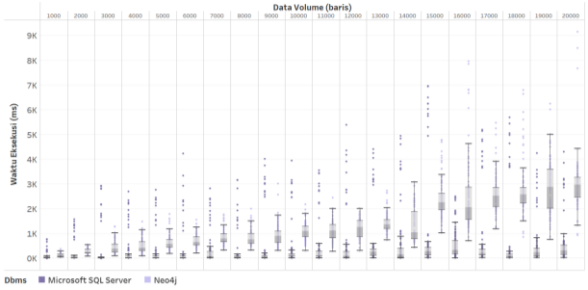

Figure 8. Comparison of query execution time 1

The delete operation is performed by deleting 1000 to 20000 rows of data. In this experiment, a simple delete operation was performed. It can be seen in Figure 9 from the experimental results that the time required by Neo4j and Microsoft SQL Server is not so different. However, in the delete operation, Microsoft SQL Server remains superior compared to Neo4j. The speed of Microsoft SQL Server in executing on average is 1.5 times faster than Neo4j.

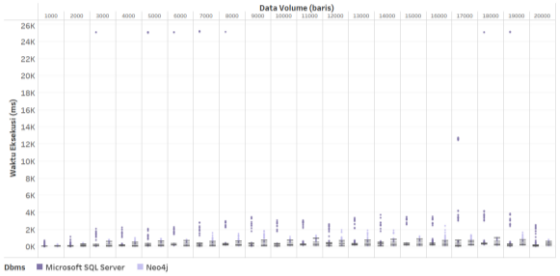

Figure 9. Comparison of query execution time 2

In the read operation test, the resulting query is the most effective query made in this research by knowing how many rows must be traced by each query by using the pipeline feature in each DBMS. The most effective query in this research refers to the fewer rows traced then the query becomes more effective.

The results of the read operation test can be seen in Figure 8 to Figure 13. In simple select queries such as query 1 in Figure 8 Microsoft SQL Server is superior to Neo4j. As for aggregated queries such as query 2 in Neo4j is able to rival Microsoft SQL Server with a constant speed and has no significant spikes such as, but on average it is still superior to Microsoft SQL Server DBMS as we can see in

Figure 9.

Figure 10. Comparison of query execution time 3

For search queries such as query 8 Neo4j is on average still superior to Microsoft SQL Server as shown in Figure 10. While complex select queries involving multiple tables such as queries 4 and 6, the more tables that are linked the superiority of Microsoft SQL Server decreases, as can be seen in Figure 11 and Figure 13 while the performance of Neo4j remains rather constant. When the number of joining tables is increased as in query 6 Neo4j on average shows faster than Microsoft SQL Server. For the results of query 5 as can be seen in Figure 13 Neo4j appears to be far behind Microsoft SQL Server.

From the results of the read operation, other values are obtained, namely throughput, memory, and received. The throughput value is used to compare the number of processes successfully performed by both DBMSs per unit of time.

DOI: https://doi.org/10.29207/resti.v7i5.5273

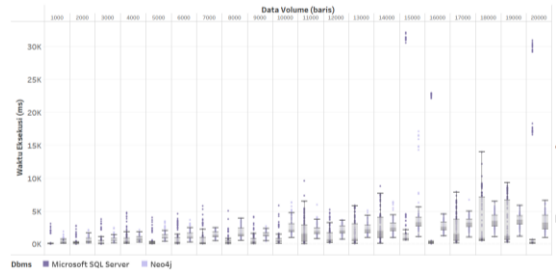

Figure 11. Comparison of query execution time 4

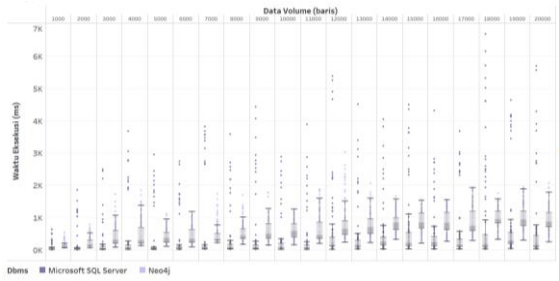

Figure 12. Comparison of query execution time 5

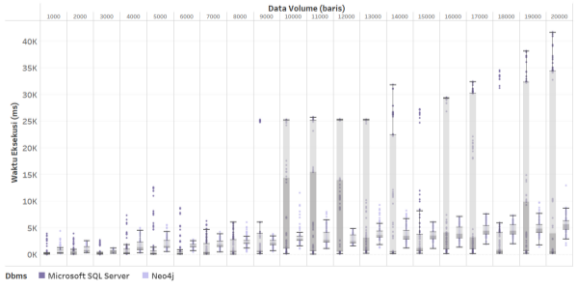

Figure 13. Comparison of query execution time 6

The results show that as the rows of data volume increase, the throughput value will decrease, and Microsoft SQL Server has a higher value, so Microsoft SQL Server has better performance than Neo4j. In addition, the results of the test also show the amount of data memory that is the output of the process of each query, Neo4j has a larger size. The memory number is also influenced by the testing tool that outputs the database graph output with a one-by-one result node, instead of directly displaying a single result. Then the received number is a comparison of the amount of data that was successfully given at one unit of time, in this study using KB/second units, so much higher this value, the better the performance of a DBMS. However, this comparison is also strongly influenced by the amount of data processed, the larger the data, the higher the value of received. This comparison is done on query 1 to query 12. In this test, the received value of Neo4j is higher than Microsoft SQL Server.

4. Conclusion

From the research conducted, the following conclusions can be generated. Modeling of metadata and data in census and survey has been successfully implemented using graph database. From the analysis of the characteristics and features that have been presented, Neo4j has the ability to customize security features by utilizing available packages, lower prices, and easy-to-understand query languages. For Microsoft SQL Server, it already has a stronger built-in security architecture and scalability that can be adjusted according to needs. The write experiment performance of Neo4j is still far behind when compared to Microsoft SQL Server. The delete experiment performance of Neo4j is almost the same as Microsoft SQL Server. The read experiment of Microsoft SQL Server is superior to Neo4j, on simple select queries. On simple or complex aggregate queries, the execution time of both seems not much different, but Microsoft SQL Server is still superior. In complex select queries involving many relationships, Microsoft SQL Server is still superior although there is a large spike in values and Neo4j's performance tends to be consistent without any high time spikes and when involving many tables Neo4j tends to be consistently faster. This experiment is strongly influenced by the testing tool used, Apache JMeter. It can be seen that one of the reasons the memory output from Neo4j is larger is that each line of output from Apache JMeter contains the name of each node. The number received is also strongly influenced by the amount of data being processed so that it cannot be compared directly. While the throughput numbers show that Microsoft SQL Server is superior to Neo4j. So from the experiments that have been carried out, Neo4j has not been able to match the performance of Microsoft SQL Server.

References

[1] Jatin and Shalini Batra, "MONGODB Versus SQL: A Case Study on Electricity Data," in *Emerging Research in Computing, Information, Communication and Applications*, Springer Singapore, 2016, pp. 297–308. doi: 10.1007/978-981-10-0287-8_28.

[2] D. Kunda and H. Phiri, "A Comparative Study of NoSQL and Relational Database," *Zambia Information Communication Technology (ICT) Journal*, vol. 1, no. 1, pp. 1–4, 2017.

[3] E. Khatibi and S. L. Mirtaheri, "A dynamic data dissemination mechanism for Cassandra NoSQL data store," *Journal of Supercomputing*, vol. 75, no. 11, pp. 7479–7496, Nov. 2019, doi: 10.1007/s11227-019-02959-7.

[4] R. Hogan, "A Practical Guide to Database Design," 2018.

[5] Badan Pusat Statistik, "Informasi Umum BPS," *https://ppid.bps.go.id/*, 2020.

[6] BPS, "Metadata Sensus: Keterangan Umum," *https://sensus.bps.go.id/metadata_kegiatan/index/sp2020/keterangan%20umum*, 2020.

[7] BPS, "Istilah," Jun. 11, 2009. https://www.bps.go.id/istilah/index.html?Istilah%5Bberawalan%5D=S&Istilah_page=5 (accessed Jul. 20, 2023).

[8] Binus University, "PENGGUNAAN DATABASE ORACLE DI BEBERAPA PERUSAHAAN INDONESIA," Aug. 08, 2018. https://sis.binus.ac.id/2018/08/08/penggunaan-database-oracle-di-beberapa-perusahaan-indonesia/ (accessed Sep. 09, 2023).

[9] W. Ali, M. U. Shafique, M. A. Majeed, and A. Raza, "Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics," *Asian Journal of*

DOI: https://doi.org/10.29207/resti.v7i5.5273

Research in Computer Science, pp. 1–10, Oct. 2019, doi: 10.9734/ajrcos/2019/v4i230108.

[10] L. R. Maghfiroh and I. G. B. B. Nugraha, "Survey data and metadata modelling using document-oriented NoSQL," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Apr. 2018. doi: 10.1088/1742-6596/971/1/012030.

[11] L. R. Maghfiroh and I. Santoso, "NoSQL Model Data Warehaouse Metadata Survei Dinamis Studi Kasus: Survei Rumah Tangga," 2019. [Online]. Available: https://silastik.bps.go.id

[12] G. Harrison, *Next generation databases : NoSQL, NewSQL, and Big Data*. 2015.

[13] A. Gupta, S. Tyagi, N. Panwar, and S. Sachdeva, "NoSQL Databases: Critical Analysis and Comparison," *International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 2017.

[14] I. Robinson, J. Webber, and E. Eifrem, "Graph Databases," 2015.

[15] S. Medhi and H. Baruah, "Relational database and graph database: A comparative analysis," *Journal of Process Management. New Technologies*, vol. 5, no. 2, pp. 1–9, 2017, doi: 10.5937/jouproman5-13553.

[16] V. Vojtˇ, V. Kolomičenko, M. Svoboda, and I. Holubová, "Experimental Comparison of Graph Databases *," 2013. [Online]. Available: http://www.sparsity-technologies.com/dex

[17] D. Fernandes and J. Bernardino, "Graph databases comparison: Allegrograph, arangoDB, infinitegraph, Neo4J, and orientDB," in *DATA 2018 - Proceedings of the 7th International Conference on Data Science, Technology and Applications*, SciTePress, 2018, pp. 373–380. doi: 10.5220/0006910203730380.

[18] P. Kotiranta, M. Junkkari, and J. Nummenmaa, "Performance of Graph and Relational Databases in Complex Queries," *Applied Sciences (Switzerland)*, vol. 12, no. 13, Jul. 2022, doi: 10.3390/app12136490.

[19] C. Kemper, *Beginning Neo4j*. 2015.

[20] D. G. Feitelson, "Experimental Computer Science: The Need for a Cultural Change," 2006.

[21] Y. Y. Sahria and D. H. Fudholi, "Pemodelan Pengetahuan Graph Database Untuk Jejaring Penelitian Kesehatan di Indonesia," *Jurnal Media Informatika Budidarma*, vol. 4, no. 3, p. 604, Jul. 2020, doi: 10.30865/mib.v4i3.2183.

[22] neo4j.com, "Graph Data Modeling," Aug. 24, 2018.

[23] E. H. Halili, *Apache JMeter : a practical beginner's guide to automated testing and performance measurement for your websites*. Packt Pub, 2008.

[24] H. Garcia-Molina, J. D. Ullman, and J. Widom, "DATABASE SYSTEMS The Complete Book Second Edition," 2009.

[25] R. McColl, D. Ediger, J. Poovey, D. Campbell, and D. A. Bader, "A performance evaluation of open source graph databases," in *PPAA 2014 - Proceedings of the 2014 Workshop on Parallel Programming for Analytics Applications*, Association for Computing Machinery, 2014, pp. 11–17. doi: 10.1145/2567634.2567638.

[26] K. Sahatqija, J. Ajdari, X. Zenuni, B. Raufi, and F. Ismaili, "Comparison between relational and NOSQL databases," 2018.

[27] Neo4j, "Neo4j Licensing," Jun. 20, 2016. https://neo4j.com/licensing/ (accessed Jul. 20, 2023).

[28] Micrososft, "Microsoft Licensing Resources," *https://www.microsoft.com/id-id/licensing/product-licensing/sql-server*, 2019.

[29] Neo4j, "Neo4j Pricing," *https://neo4j.com/pricing/#graph-database*, 2023.

[30] Microsoft, "SQL Server 2019 -- Pricing," *https://www.microsoft.com/en-us/sql-server/sql-server-2019-pricing*, 2019. https://www.microsoft.com/en-us/sql-server/sql-server-2019-pricing (accessed May 10, 2023).

[31] J. Stegeman, "Native vs. Non-Native Graph Database," May 08, 2023. https://neo4j.com/blog/native-vs-non-native-graph-technology/ (accessed Jun. 06, 2023).

[32] Neo4j, "The Neo4j Operations Manual v5," *https://neo4j.com/docs/operations-manual/current/*, 2023.

[33] Microsoft, "Security for SQL Server Database Engine and Azure SQL Database," *https://learn.microsoft.com/en-us/sql/relational-databases/security/security-center-for-sql-server-database-engine-and-azure-sql-database?view=sql-server-ver15*, 2019.

DOI: https://doi.org/10.29207/resti.v7i5.5273