International Journal of Advances in Engineering Research http://www.ijaer.com

(IJAER) 2016, Vol. No. 12, Issue No. V, November **e-ISSN: 2231-5152, p-ISSN: 2454-1796**

GRAPH AND RELATIONAL DATABASES AS PART OF A HYBRID DATABASE STRATEGY[1]

Subin Paul

Lecturer in Computer Engineering, Govt. Polytechnic College, Koratty

ABSTRACT

Scalability and the management of multiple databases in a single application have recently presented a significant obstacle for the database industry. Clearly, a standard for storing, managing, and querying data is the relational database. Business intelligence and analytical querying are two examples of its applications. However, new requirements have emerged and graph-structured data are becoming increasingly important as a result of the massive growth of distributed databases and social networks. Graph data can be stored more naturally in a graph database. Depending on the data's characteristics and the kinds of queries to be evaluated, relational and graph databases each have advantages and disadvantages. However, by reducing some of the constraints, advantages can be obtained from a combination of the two. To circumvent the limitations of individual systems, therefore, you should suggest a hybrid model design in which the two models are combined into a single system. By analyzing their respective strengths and weaknesses, the hybrid system brings together the advantages of graph databases and relational databases. The goal of this paper is to provide a summary of previous research on the hybrid database model.

Keywords: *distributed database; graph database; hybrid database; relational database*

INTRODUCTION

For nearly as long as databases have existed, relational databases have been the industry standard. It is unquestionably useful for storing tabular data in a specific schema that does not adequately accommodate the data set's interconnections. As a result, forcing a highly connected data set into a relational database causes significant issues with query performance and return time. SQL also becomes intricate while taking care of enormous information base. The number of joins in a database increases with its size, which ultimately increases query retrieval time. Data is rapidly becoming more and more connected as a result of the recent rise of social networks and modern technological advancements, making relational databases less suitable. The database industry began looking for more effective alternatives as a result. The Nosql development has brought numerous new data set models in data set industry where each model had a few significant highlights that social model doesn't have. This database's motivations include being schema-independent, having a straightforward design, making use of common hardware, making it simple to add more servers (horizontal scalability), being

[1] How to cite the article:
Paul S., Graph and Relational Database as Part of a Hybrid Database Strategy; *International Journal of Advances in Engineering Research*, November 2016, Vol 7, Issue 6, 48-58

highly distributable, having high availability, and being open source. There are four types of Nosql databases: key-value stores, document stores, column stores, and graph stores. When it comes to handling interim unstructured data, key-value databases perform well; diagram data sets handle connections as top of the line residents and segment data sets are better for putting away authentic information for business investigation. Despite competition from other databases, relational databases continue to dominate. Because different databases are designed for different tasks, choosing which database is best suited to a given task is not always easy. When using a Nosql database, ideal relational data storage frequently presents a challenge. Additionally, today's needs call for polyglot persistence. With the needs of today's customers in mind, an increasing number of businesses are employing multiple database systems in which one database exploits advantages and another covers weaknesses. The interest in chart data sets is constantly developing a result of its capacity to examine the information in non-social organization (for example, long range interpersonal communication information). The fundamental focus of a graph database is on the connections between data. However, there is still ample space for relational databases. For data that is tabular and well-structured, a relational model is useful. As a result, our primary focus is on attempting to combine the advantages of relational and graph databases. The current significance of using non-relational data is motivating. The data retrieval query must search the intended data in both databases when designing a hybrid model. The data classifier algorithm is used by the data insertion query to locate a suitable database that accurately models the data. Classification is therefore founded on the task of locating the features that describe the data.

The most widely used method for storing data is probably a relational database. Taking into account the exponential growth in the volume of data generated by the use of social networks and the internet. Due to the fact that these conditions have made data more unstructured, connected, and relationship-rich, relational databases are no longer suitable for storing it. The need for a different database as a result is inevitable. The arising nosql data sets have thought of numerous choices with its attributes, each work in a space where social data set fizzled. Computer communication networks, protein cell interaction, social networks for fraud detection, and the semantic web all use graphs as data representational models. All of these applications necessitate interacting with highly connected data, which ultimately led to the creation of a new class of storage systems known as graph database management systems (GDBMS) under the umbrella of nosql databases. The way relationships are stored in graph and relational databases is the main difference. Relational databases use referential integrity constraints, whereas graph databases use connected edges between nodes to identify them. The second significant difference between the two databases is how they search for related records. When a table is large, a relational database may search the entire required table inefficiently at times. In this case, graph databases prove to be more advantageous because they can easily handle large data sets without requiring costly join operations. A flexible schema that allows for easy schema changes at any stage is another advantage of the graph database. Unstructured data is the most common type of data available today. Additionally, each database focuses on a subset of features that enable it to handle

a particular kind of data. Due to the fact that numerous applications require multiple data storage options, the term "polyglot persistence" is frequently used in today's world. As a result, businesses are developing applications that make use of multiple database systems in order to meet customer requirements. In this case, each database is selected in such a way that one database addresses strengths and another addresses weaknesses. The development of a dual database system that makes use of a relational database (MySQL) and a graph database (Neo4j) in conjunction with migration is the primary focus of this paper. A partial migration approach is utilized because graph databases are intended for modeling relationships. Additionally, join-sensitive and non-join-sensitive queries are used to conduct a comparative analysis of the migrated graph database.

DATA MODELS

A. Relational Model

The relational model, developed by E. F. Codd in the 1970s, provides a mathematically flexible approach to data organization, storage, and utilization. The relational model and some of its terms are depicted in Figure 1.

A collection of tables with distinct names make up a relational database. A table is a collection of relationships between a set of values that are represented by a row (or tuple). Figure 1 depicts the column headers A1, A2,--,An, which represent the attributes of the table. This close correspondence between the table and the mathematical concept of relation gave rise to the term "relational model." A data set outline is legitimate plan of data set though data set occasion is preview of information in data set at a given moment in time. Each tuple must be uniquely identified in order to be distinguished from other tuples by its attributes. This is accomplished by characterizing keys. By defining foreign keys, additional constraints on relationships can be imposed. Extra other honesty imperatives can likewise be determined. In addition, users must use a query language to request data from the database.

B. Graph Model

A graph is made up of two components in mathematical terms: a hub (likewise called vertex) and an edge. Each edge represents a connection or relationship between two nodes, and each node represents an information entity (such as a person, place, thing, category, or other piece of data). The graph data model considers relationships to be first-class members. It refers to any method of storing data in which connected components are linked without the use of an index. Dereferencing a physical pointer makes it possible to access an entity's neighbors. It is a database with Create, Read, Update, and Delete (CRUD) methods that show a graph data model like RDF triples (subject-predicate-object) and hypergraphs (where a relationship can connect any number of nodes). The property graph model's structure is depicted in Figure 2, and an example is shown in Figure 3. The property graph model is the one that is used the most, and the one we propose makes use of it. Neo4j, titan, hypergraphdb,

flockdb, dex, infinite graph, and other well-known graph database models are available. A particular database can be picked out based on the requirements of the user.

C. Hybrid Model

Performance is one of graph databases' main advantages over relational databases and other NoSQL stores. In terms of indexing, computing power, storage, and querying, graph databases typically have thousands of times more power than conventional databases. Rather than social data sets, where the inquiry execution on information relations diminishes as the dataset develops, the presentation of diagram data sets remains moderately steady. Given the advantages and disadvantages of both models, it makes practical sense to use both databases today. As a result, the idea focuses on developing a hybrid database system for big data storage and management. An integrated data system that is made up of collections of two or more independent datasets and/or databases is referred to as a hybrid database approach or multi database system. A hybrid strategy is depicted in Figure 4. There are numerous issues that need to be addressed, including the requirement that developers learn multiple technologies and query languages. A number of studies on hybrid databases based on graph databases and relational databases are included in this literature review.

RELATIONAL AND GRAPH DATABASE ELEMENTS

This section briefly expresses the introductory details of relational and graph database which are used in the proposed approach.

A. Graph Database Model

Data with a lot of relationships is best handled by graph databases. Relationships play the most important role here, and the primary reason to use graph databases over relational databases is probably because graph databases deal with relationships more naturally. Additionally, these relationships are used to obtain the majority of graph database output. Neo4j, a property graph model, is the graph model used in the proposed approach. It is a marked, property chart data set created by Neo innovation. Nodes are the fundamental data entity in graph databases, and edges represent relationships between nodes. A start node, an end node, a type, and a direction make up an edge. The graph data model is shown in Figure 4.

B. Relational Database Model

The Relational Database, which was developed in the 1970s and stores data in two-dimensional tables, organizes data into tables. Rows (tuples, records) and columns (attributes, fields) make up a two-dimensional structure called a table. Data from a relational database can be retrieved using SQL. It doesn't support scalability and can't handle a lot of data with many connections because it needs too

many join operations. Example: RDBMS (MySql, which is free). Ri(Ai) denotes the relational schema in a relational database, where Ri denotes the relation and Ai denotes its set of attributes. Entity properties are referred to as attributes or columns. The relational schema R1(A1)....Rn(An) is the set of tuples that make up the Relational Database R. A table is another name for relationship. The primary key is a key that is used to uniquely identify each tuple in the table. Foreign keys can be used to refer to this primary key from another database table [1-3]. Figure 5shows social information model.

LITERATURE REVIEW

Ebb and flow examination can be isolated into three stages. Step one covers the aspect of relational databases; step two focuses on the limitations of relational databases and the emergence of graph databases as an alternative to relational databases. The final step is a hybrid of the two. Numerous researchers inquire about relational databases' superiority, matured nature, market hold, popularity, robustness, and numerous other success stories. The social data set model previously came to the power during the 1970s with Codd's Social Model of Information [1]. SQL is a well-established querying technology that is used by millions of developers. In his chapter, Johannes Zollmann wrote about the RDBMS's ACID properties. He also mentions that ACID properties provide solid consistency guarantees [2]. The second step discusses its inability to adapt to the web 3.0 scenario of today. The performance of relational database applications is decreasing despite advancements in computing, faster processors, and fast networks. The striking limitations of a relational database include a longer query time, numerous joins and self-joins, a rigid schema, structural limitations, and high costs for setup and maintenance. Another disadvantage of relational databases is the rise in information complexity. This everything is occurring a result of a general development not just in the volume and speed of information, yet in its assortment, intricacy, and interconnectedness. The data of today can be described as semi-structured, densely connected, and having a high degree of data model volatility. Data relationships are also expanding at a faster rate as data volume, velocity, and variety rise. With a consistent structure and a predetermined schema, relational databases were made for tabular data. Relational databases, despite their name, do not store relationships between data elements; They are not suitable for the highly connected data of today [3, 4].

Relational databases have been suggested as an alternative by numerous researchers. A review of the literature suggests graph db as a suitable alternative [3] [4]. The idea of a graph database is introduced by Adrian Silvescu et al. [5]. Jaroslav Pokorny explains a lot of important graph database concepts in [6]. He offered two perspectives, focusing on both the bright and dark sides of graph DB. Lack of maturity, limitations on functionality, and significant analytics requirements are the primary shortcomings highlighted. Additionally, it discusses the limitations of pattern matching queries, the necessity of appropriate benchmarking, and other limitations encountered during design. As a result, researchers began planning a migration from their existing relational database to a graph database taking into account the advantages and disadvantages of both database approaches. However, that also presented difficulties. Therefore, selecting a solution that is compatible with both databases would be

ideal. Additionally, polyglot persistence is a requirement in today's market due to the increasing importance of catering to the requirements of customers. In these systems, databases are selected in such a way that one database exploits advantages and another covers weaknesses. Blessing E. James and P. O. Asagba developed a different strategy for the storage and management of big data called a hybrid database system. The most widely used relational and NoSQL (non-relational) database servers, MySQL and MongoDB, comprise the hybrid system. The author has loaded the data in hybrid, SQL, and MongoDB modes, respectively. Big data management and storage can be improved with their hybrid model [6]. Between the traditional relational database management system (RDBMS) and the in-memory graph store, Christopher J. O. Little's Grapht offers an intermediate query processing layer. It allows queries similar to those in SQL but with the power of a graph handler, and it describes a graph in a relational environment. The model design can handle hybrid queries and is a build-in-memory store. At the point when client hits inquiry, question processor isolates them into column driven sub questions for social data set straightforwardly, and diagram driven sub inquiries for the chart overseer. Additionally, gSQL, a hybrid query language, is presented [7]. Luis Ferreira attempted to overcome any issues among SQL and NoSQL by building a layer between the SQL code and mediator, and the real information base under it. This lets SQL queries be run on top of a NoSQL system, but performance may suffer as a result. A similar strategy was used by [8] to combine relational and graph databases with SQL and NoSQL. FishBase, a relational database used by researchers, fishery managers, biologists, and enthusiasts, is the platform on which the work is implemented. The relational FishBase was converted into graphical FishBase to meet current requirements. However, there was no direct way to update or add new data to the graph model. Regraph was the name of the implemented version, which mapped data from a relational database to a graph database and provided a hybrid architecture that kept both databases connected, synchronized, and in their native representations. Similar to how Rune Ettrup and Lisbeth Nielsen presented the distributed database concept Bridge-DB, which targets multiple data sources. It uses its own query language, BQL, but it is able to perform all CRUD operations. The multi-database system that Bridge-DB developed makes use of a middleware layer between heterogeneous databases and is connected to Neo4J and PostgreSQL. The author has implemented a cost-based optimizer that combines dynamic and black box cost models to determine which database should be queried or whether the query should be enumerated to run on multiple databases. Final post-processing of the results to satisfy the query is carried out by the optimizer [9]. The results of running a set of queries on each relational database, graph database, and optimizer are shown in Table 1. Martin Grund and coworkers;, have proposed a new architecture for an enterprise application-specific database system that combines the benefits of relational and graph data processing in a single in-memory database engine by allowing semantic and graph data to be directly included in the same storage engine. The use of In-Memory technology is made possible by its ability to combine various types of storage in a single storage engine without sacrificing performance. The author builds his architecture on HYRISE, a Main Memory Hybrid Storage Engine that also uses compression and partitioning algorithms. The second aspect is query execution, which includes integrating two distinct storage engine types [10].

THE HYBRID DATABASE APPROACH

A database design model that can be used with a variety of databases, such as graphical databases and traditional relational tables with rows and columns. It can also natively store graphs for the storage and management of big data. As a result, the developed system will have the power of two systems and will overcome the limitations of the individual systems because this approach of integrating graph and relational data was ignored because both data types store data in different ways, resulting in different access patterns to disk that were not compatible. It aims to improve big data storage and management for simple retrieval without database migration. The resultant framework essentially utilizes a classifier that performs information bifurcation while putting away. The classifier examines the data's nature. Relational databases are used for data with a lot of structure, while graph databases are used for less structured data. Figure 5 depicts a possible proposed strategy. As a result, the proposed approach combines the advantages of relational and NoSQL databases in a single database system rather than sacrificing them.

CONCLUSION

As a result, a strategy for putting together a hybrid database system that works with two different heterogeneous databases is proposed. This system would be able to break down and run queries on multiple databases in order to speed up response times by allowing the databases to work together. In order to achieve results, the following phase would attempt to put the aforementioned idea into action. An extensive review of various hybrid database approaches is conducted in this paper.

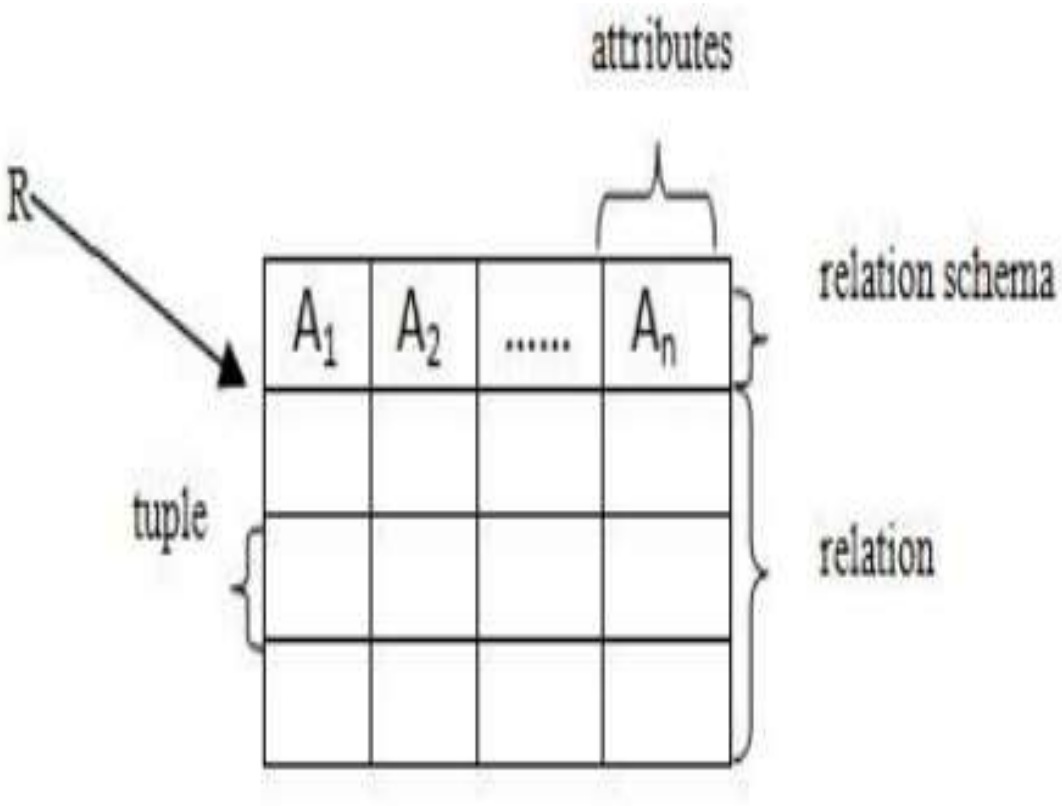

Fig. 1. The Relational Model

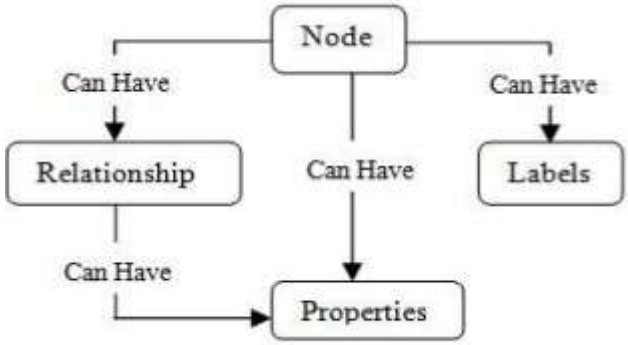

Fig. 2. Building blocks for property graph model

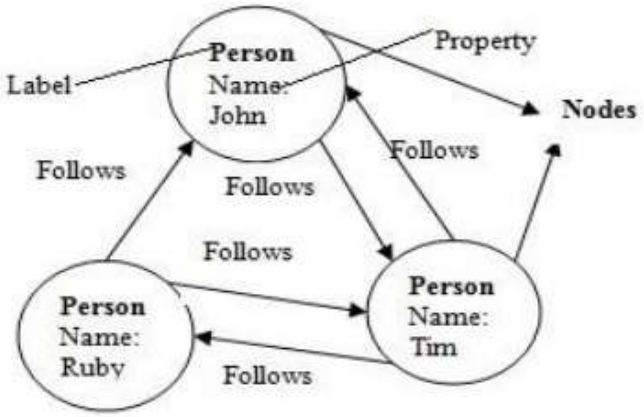

Fig. 3. A property graph model

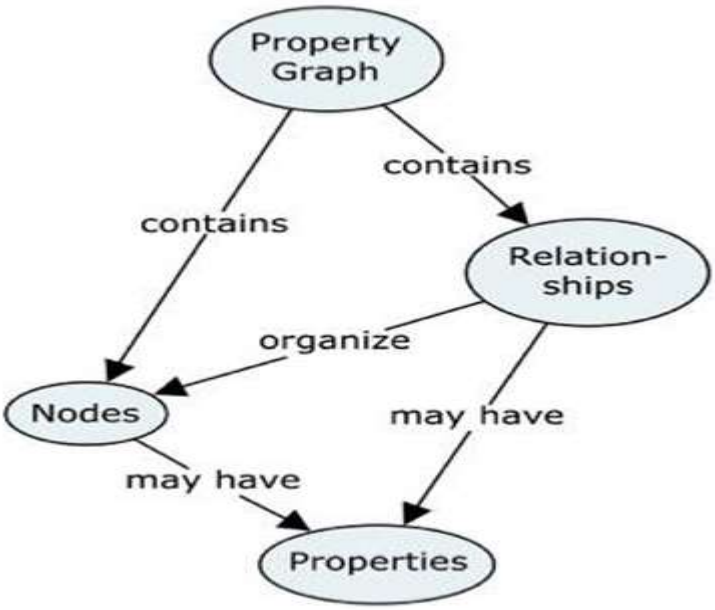

Fig. 4. The Graph Model

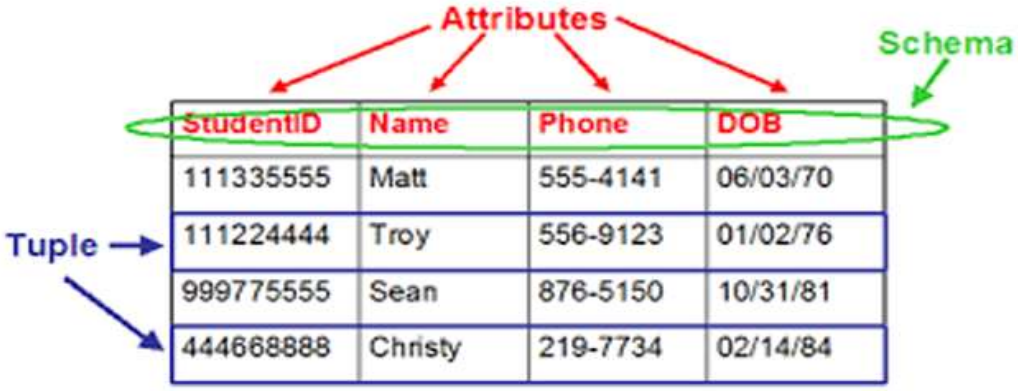

Fig. 5. The Relational Model

Table I. Response Times In Ms For Merged Dataset

Query	Neo4j Cached	Neo4j Non cached	Postg-reSQL cached	Postg-reSQL Non cached	Optim-izer Single MQ	Optimi-zer Multiple MQ
1	19	1654	247	324	263	24
2	29	1557	213	287	244	36
3	26	1685	212	290	144	34
4	15	1908	701	1150	346	25
5	32	1668	474	681	289	41

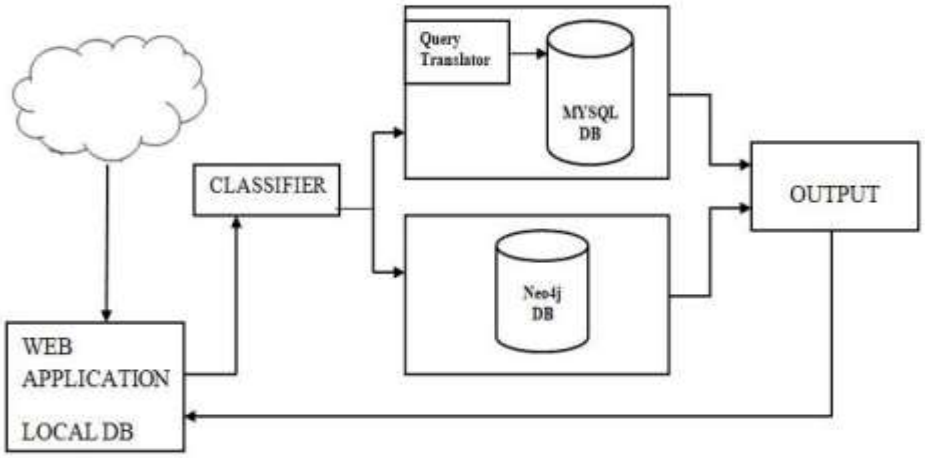

Fig. 6. Proposed Model (Functional Block Diagram)

REFERENCES

[1]. Sakr, S., Elnikety, S. and He, Y., (2014). Hybrid query execution engine for large attributed graphs. *Information Systems*, 41, pp.45-73.

[2]. Kanade, A., Gopal, A. and Kanade, S., (2014, February). A study of normalization and embedding in MongoDB. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 416-421). IEEE.

[3]. Difallah, D.E., Pavlo, A., Curino, C. and Cudre-Mauroux, P., (2013). Oltp-bench: An extensible testbed for benchmarking relational databases. *Proceedings of the VLDB Endowment*, 7(4), pp.277-288.

[4]. Chen, A., Liu, L. and Shang, J., (2012, March). A hybrid strategy to construct scientific instrument ontology from relational database model. In *2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring* (pp. 25-33). IEEE.

[5]. Sakr, S., Elnikety, S. and He, Y., (2012, October). G-SPARQL: a hybrid engine for querying large attributed graphs. *In Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 335-344).

[6]. Sil, A., Cronin, E., Nie, P., Yang, Y., Popescu, A.M. and Yates, A., (2012, July). Linking named entities to any database. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 116-127).

[7]. Jayathilake, D., Sooriaarachchi, C., Gunawardena, T., Kulasuriya, B. and Dayaratne, T., (2012, September). A study into the capabilities of NoSQL databases in handling a highly heterogeneous tree. In *2012 IEEE 6th International Conference on Information and Automation for Sustainability* (pp. 106-111). IEEE.

[8]. Choi, H., Son, J., Yang, H., Ryu, H., Lim, B., Kim, S. and Chung, Y.D., (2013, April). Tajo: A distributed data warehouse system on large clusters. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)* (pp. 1320-1323). IEEE.

[9]. Schink, H., (2013, September). Sql-schema-comparer: Support of multi-language refactoring with relational databases. In *2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)* (pp. 173-178). IEEE.