Performance Analysis of Relational and Graph Database

Shubhangi S. Marudkar[1], Harsha R. Vyawahare[2]

[1]ME student of (CSE) Sipna College of Engineering and Technology, Amravati, Maharashtra, India
[2]Professor, Dept. Sipna College of Engineering and Technology, Amravati, Maharashtra, India

---***---

Abstract – *Collecting huge amounts of complex information like data and knowledge is very common nowadays. This requires need to store and control complex data. From most recent three decades, the relational databases are being utilized as a part of numerous associations of different natures, for example, Education, Health, Business and in numerous different applications. Relational databases demonstrate huge execution and are intended to deal with organized information with ACID (Atomicity, Consistency, Isolation, Durability) property to oversee information honesty however Relational databases can't process appropriately and oversee extensive measure of information productively. Present requirements are moving towards movement, UI, Internet of things, Browser based IDEs and so on. These innovations require constant reaction and substantial information store. Relational database frameworks recovers and oversees database in a forbidden shape, yet in current situation of dispersed extensive scale database those databases does not perform well. To beat the constraints of relational databases, and to cover the prerequisites of current applications has lead the advancement of new database advances, for example, Nosql databases. Among them The Graph Databases are most well known in the database group in light of the fact that in vogue ventures where a database is required, the extraction of commendable data depends on handling the graph like structure of the information. We are exhibiting an orderly correlation of relational and graph database models, for that we are utilizing MariaDB and Neo4j. Here we will look at the reasonableness of two classes of databases that is Relational database and graph database for putting away and quering datasets. We report consequences of estimations of scalability, query performance, and query expression of, utilizing synthetic datasets.*

Key words: neo4j; Graph database; scalability; query performance; query expression.

1. INTRODUCTION

Now a-days, information is being extended quickly in the business. The idea of information is shifted and enhanced, for example, unstructured, semi-structured and structured data. The issue isn't just how to store and access such enormous measure of information yet in addition need to remove significant learning from such information quickly. The relational model has dominated the computer industry since the 1980s for the most part to store and recovering data. Traditional databases require declarative language, for example, SQL to control organized data. Way back people used database only to store file records like purchase reports and finance records[5]. Relational databases depend on information consistency and can process the information at certain farthest point. To overcome the requirements of current application domain relational databases, associations are required to build their framework limit, for example, RAM, Disk, improved strategies for getting to information etc[1]. numerous associations are depend on unstructured information, for example, emails, blogs, audios, videos, images and such data is generated at very high speed [1]. To cover the requirements of current application areas, has lead the improvement of new advances called NOSQL databases[15]. one of them is graph database. The expansion of enormous and complex graph like information makes a graph database a vital prerequisite. NoSQL databases are horizontal scalable databases while relational databases are vertical scalable databases.

The Neo4j is an open source Java-based graph database[12]. Neo4j models data as graph comprising nodes and edges. Both nodes and edges can be explained with lables. Neo4j gives a brief inquiry dialect, Cypher, that is intended to make it simple to express graph traversal queries, and proficient to execute them[14]. For instance, it is exceptionally easy to express a question to discover the majority of the nodes inside a given number of edges of a given node—an errand that is bulky in SQL. Then again implementing such kind of issues in relational databases includes expansive number of joins which are costly to ascertain.

The relational databases relies upon unbending pattern and make it hard to include new connections between objects while The Graph databases are added substance in nature that implies we can include new sorts of connections, new nodes, and sub graphs to a current structure without disturbing existing queries and application functionality[5]. A graph database gives a large portion of the real parts in DBMS, eg. external interfaces (UI or API), database languages (for information definition, control and questioning), query optimizer, database engine (middle level model), storage engine (low-level model), transaction engine, management and operation features (tuning, reinforcement, recovery,etc.)[19]. As a robust, scalable and high-performance database, Neo4j is suitable for enterprise development. For instance, Facebook scale dataset and Google+ dataset which comprise of billions of edges, a great many refresh rates every second and require complex framework. graph databases are intended for associated information and are utilized as a part of numerous applications, for example, Facebook, Amazon, LinkedIn and numerous more[5][6][7].

2. LITURATURE SURVEY

A few late research endeavors have concentrated on preprocessing graph databases with the objective of execution query times. graph databases oversee maps of interconnections between objects. With the ascent of Social Media, Graph databases are progressively being utilized in light of the fact that they are a characteristic fit for demonstrating things like interconnected web joins, proposals, labels, and companion and contact connections. Relational databases simply aren't ready to store this sort of data effortlessly in tables and lines of information. Center standard relational databases are the ideas of substances and connections. In graph databases, the relating ideas are nodes and edges. A node holds the property and information for a protest. Graph which demonstrates the connections kept up in graph databases take after the outlines utilized as a part of question arranged programming, and in light of the closeness, graph databases effortlessly can display the connections between the articles utilized when programming an application.

Emil Eifrem, CEO of Graph Database organization Neo Technology has ran a few tests that he ran which contrasted the speed of relational with graph databases. He made a "friends of friends" query and found that when the question of connections went three levels profound that the graph database beat the relational one by a factor of 150, and when the query profundity was expanded to four the graph database bested the relational one by a factor of 1000[6].

Angles, R.; et al displays an overview of prior work (pre-NOSQL) in graph databases. i.e preceding 2002, particularly topographical, spatial and semi structured database models. More seasoned information models concentrated intensely on semi structured and XML information in a relational database. The creators incorporated the idea of a "graph database model" and think about recommendations accessible right now. Angles, R.; performs correlation and execution examination of various graph database models thinks about current graph databases focusing on their information display includes, that is information structures, query offices, and uprightness limitations. Creator demonstrates that most graph database models give an inborn help to various graph structures, query offices as APIs (the greater part of the models) and question dialects (a couple of them), and essential ideas of respectability constraints[6].

Philip Howard, investigator at Bloor Research announced that Graph databases are basic when the level of partition [ie, I know x who knows y who is identified with z who used to live in an indistinguishable house from w etc.]between substances turns out to be excessively incredible, making it impossible to deal with utilizing ordinary innovation. Prophet or DB2, for instance, can sensibly deal with up to three degrees of detachment however not the six or seven. Howard remarked on the restrictions of graph databases, saying that "The real impediment is that while these are in fact NoSQL databases, by and by they can't be executed over an ease bunch (at any rate not a present) but rather need to keep running on a solitary machine, the reason being that execution corrupts quickly over a system. Another potential disadvantage is that it is possible that you need to compose your own questions utilizing Java or whatever — which implies utilizing costly software engineers — or you utilize SparcQL or one of the other inquiry dialects that have been created to help graph databases, however this implies taking in another skill[6]."

Jouili, S.;et al experimentally thinks about graph databases ie shows Graph Database Benchmark, to analyze four graph databases: Neo4j, DEX, Titan (BerkeleyDB and Cassandra) and OrientDB (neighborhood) on various kinds of workloads, each time distinguishing which database was the best and the less adjusted. In light of measure, the database that got the best outcomes with traversal workloads is unquestionably Neo4j: it beats the various competitors, in any case the workload 8or the parameters used[6].

Healthcare Systems in United States was generating more information and they required new innovation to deal with information investigation viably. Information driven approach is utilized to deal with information examination in healthcare system by utilizing two free assignments, information administration and information administrations. Here, information administration implies putting away the information with insignificant excess structure and blunder free. Information administrations portray different examination questions, for example, join, look and factual inquiries. The issue showed up because of the hole between information administration and information benefits in relational databases. To overcome this issue, they displayed a way to deal with change over third normal form (3NF) of relational databases in equal graph of Graph database. A graph database does not require making more tables and recreating them not at all like relational databases. For instance, Neo4j is reasonable in OLTP (online exchange preparing) condition. Pregel is utilized where high dormancy and high through put have high need. The analyses have demonstrated that Graph database performed superior to relational database (MySQL) in the heterogeneous condition of medicinal services frameworks of United States in OLTP[20].

3. EVALUATION PARAMETERS FOR RELATIONAL AND GRAPH DATABASE

The assessment amongst MariaDB and Neo4j depends on the subjective and objective parameters. These parameters are the columns to choose which database ought to be embraced for usage. The parameters are as per the following.

A. Level Of Support/Maturity

Maturity can be characterized by how well the framework is tested. If a system has been tested, more number of times, it means it is more stable and more bugs have been found out. Relational databases are more stable and mature than graph databases.

B. Security

MariaDB has an incredible multi client bolster. However Neo4j does not have any worked in components for overseeing security limitations and various clients. It presumes a confided in condition. Despite the fact that there is Access Control List security components yet even Access Control List administration is dealt with at application layer. Then again, there is broad help for ACL based security in MariaDB.

C. Flexibility

Although relational databases are more mature and secure as compared to graph databases, but its schema is fixed, which makes it difficult to extend these databases and less suitable to manage ad-hoc schemas that evolve over time.

Experimental setup

Relational database were initially planned in unthinkable structure. Demonstrating connections are not exactly reasonable in Relational databases. It is so since they utilize outside keys to relate one data with another. In relational databases, tables should have been joined utilizing outside keys to interface data from various tables. On the other hand we need to join numerous tables for queries, it may not be ideal to manage.

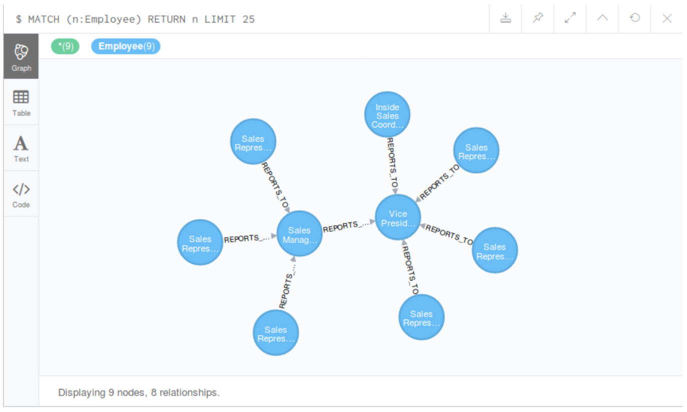
Employees	
Emp_id	Emp_name
E1	Rex
E2	Harry
E3	Bob
E4	John

Table-1: Employee table

Relationships	
Emp_id	Relation_id
E1	R3
E1	R4
E2	R3
E3	R2
E3	R4
E4	R3

Table-2: Relationship table

Here, employees and ralationship are showed up in two tables and the relationship table addresses the relation between employees. An id has been assigned to every employee in Table 1. An emp_id in the specialists' table is referenced as either emp_id or a relation_id, both the emp_id and relation_id portions in relationship table insinuate ids in the Employees' table. For example, Bob has emp_id 'E3' is an relation with Rex and Harry who has emp_id 'E1' and 'E2' exclusively. In the employees' table, emp_id is the primary key and both emp_id and relation_id are used as composite keys in the relationship table. A composite key needs both of the id's to be a stand-out blend of characteristics. For example, emp_id 'E1', relation_id 'R3' and emp_id 'E1', relation_id 'R4' are not same composite regard. In the occasion that planned associations are ought to have been shown, the composite regard, for instance, emp_id 'E3' and realtion_id 'R4' are absolutely one of a kind in connection to the composite motivating force than emp_id 'E4' in the relational databases. Regardless, for our situation, however the relations are undirected, so emp_id 'E3' and relation_id' R4' are the same as emp_id 'E4' and relation_id "R3". By and by, expect we have to make essential request, for instance, to find what Bob's associations are in the relational model. In any case, the inquiry looks specialists' table, finds which emp_id is doled out to Bob, takes that emp_id to the relational table, finds all relation_ids that are doled out to Bob's emp_id, and thereafter those relation_ids are returned back to the delegates' table to find the emp_name identified with those relation_ids. This kind of a request is up 'til now possible with relational model, yet it is computationally expensive. Strangely, for a comparative kind of a model, if we use a graph database, for instance, Neo4j then Neo4j gives us better result to demonstrate the data for above relational model. In Neo4j, the Model is the going with:



```
$ MATCH (n:Employee) RETURN n LIMIT 25
```

```
Displaying 9 nodes, 8 relationships.
```

Fig-1. : Graph Model for Employee and Relationship

4. ANALYSIS OF PROBLEM

As we have found in the above area, storage of information is essential in every one of the fields, Relational databases have been the power-stallion of programming applications since numerous years and proceed with so this day. Relational databases, references to different lines and tables. This is enforceable with requirements, however just when the reference is never discretionary. Joins are processed at question time by coordinating composite and foreign keys of the numerous lines of the to-be-joined tables. These activities are register and memory-serious and have an exponential cost. In this segment we will see the issue happened when both the techniques will completed. In graph database The far reaching examination of the relative reasonableness of various graph databases for putting away genealogical structures. That work does not address the subjective parameters of capacity expenses and query execution. This Project work is an endeavor to conquer the issues, for example, subjective parameters of capacity cost and query execution.

5. IMPLEMENTATION DETAILS

The assessment amongst MariaDB and Neo4j depends on an arrangement of predefined queries. To execute relational databases, MariaDB version 10.2.12 was used. The database was queried utilizing Python scripting language. Graph databases are used Neo4j version 1.6. The database is queried with Cypher Query Language. Query were intended to break down the execution contrast between a relational database and a graph database.

Schema for relational database included the following tables:

1) Employee: Employee_id, Employee_name

2) Customer: Customer_id, Customer_name

3)Order: Order_id , Order_name

4)Supplier: Supplier_id, Supplier_name Schema for graph databases is represented in Following figure.

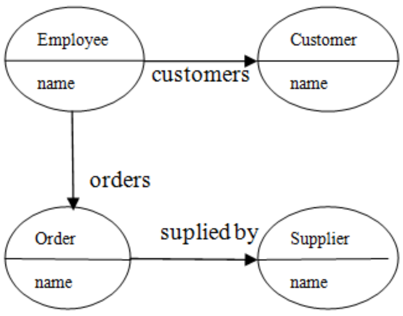


Fig-2: Graph Database Schema

The six queries defined were:

q1: Find who sold the products.

q2: Find who purchase the products.

q3: Find suppliers of the products.

q4: Find the product Categories.

q5: Merge Orders with Different Products.

q6: Find Employees who reports to the Manager.

6. EXPERIMENTAL RESULTS

In this project we looked at the relational and graph database based on retrieval time complexity, schema load time and throughput.

Retrieval Time Complexity:-

This is the time required to execute the related queries i.e. SQL queries for Relational database and Cypher queries for graph database.

Schema Load Time:-

This is the time required to load the schema queries.

Throughput:-

The throughput that measures the number of queries completed in an interval of time. In this project we took 30 sec time for each database to examine how many sql and cypher queries execute in the given time.

Execution times were collected after executing the queries and noted in milliseconds. The results have been tabulated in following tables. It can be easily observed from the values retrieved that the retrieval times of graph databases is less than relational databases. This is because relational databases search all of the data to find the data that meets the search criteria. The larger the data set, the longer it takes to find matches, so when number of users get increased the retrieval time gets increased manifold. On the other hand graph database looks only at records that are directly connected to other records, it does not scan the entire graph to find the nodes that meet the search criteria. So, increasing number of nodes from one hundred to five hundred does not increase the retrieval time much as can be visualized from the chart 1and 2.

Queries	q1	q2	q3	q4	q5	q6
MariaDB	20.16	21.68	40.09	69.11	73.89	80.11
Neo4j	9.00	11.46	22.10	32.49	49.22	50.01

Table-3: Retrieval times of queries by neo4j and MariaDB (100 objects).

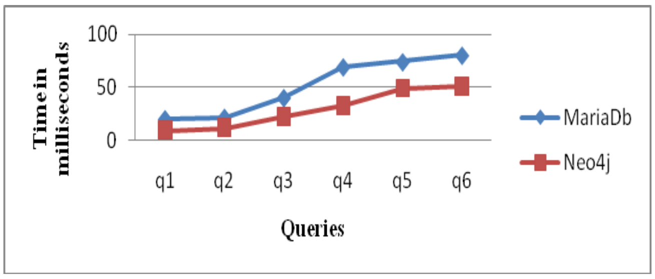


Chart-1: Retrieval times of queries by neo4j and MariaDB (100 objects).

Queries	q1	q2	q3	q4	q5	q6
MariaDB	185.49	282.48	331.89	486.54	539.78	620.10
Neo4j	18.95	26.11	29.83	34.19	52.55	57.48

Table 4: Retrieval times of queries by neo4j and MariaDB (500 objects).

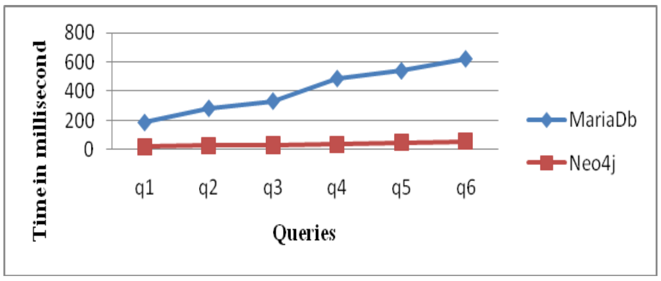


Chart-2: Retrieval times of queries by neo4j and MariaDB(500 objects)

Table 4. represent the Retrieval time complexity results in milliseconds and Chart 3. shows the graphical representation of the retrieval time complexity, from the graph we can say that time taken by cypher queries is less than that of the sql queries.

Queries	q1	q2	q3	q4	q5	q6
MariaDB	0.0058	0.0037	0.0079	0.0065	0.007	0.006
Neo4j	0.0045	0.0029	0.0056	0.0057	0.0062	0.0062

Table-4: Retrieval time complexity queries by neo4j and MariaDB.

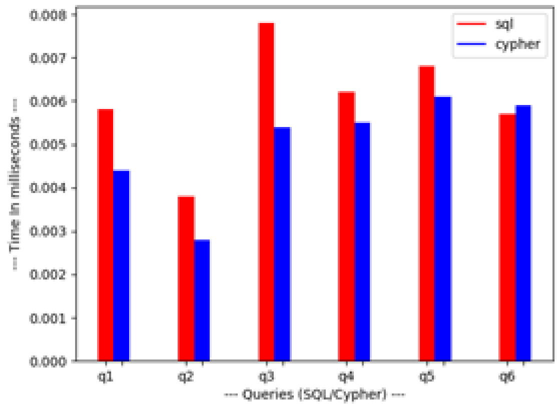


Chart-3: Graph for retrieval time complexity

Table 5. represent the schema load time taken by sql and cypher queries in milliseconds and chart 4. shows the graphical representation of it, from the graph we can say that time taken to load the schema by cypher queries is less than that of the sql queries.

Queries	q1	q2	q3	q4	q5	q6
MariaDB	0.010	0.010	0.010	0.010	0.010	0.040
Neo4j	0.009	0.008	0.006	0.005	0.004	0.0039

Table5: Query Results In Milliseconds

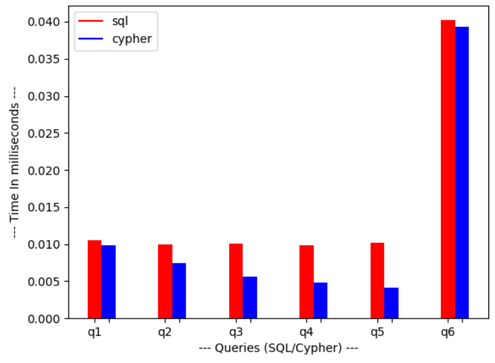


Chart-4: Graph for schema load time

Table 6. represent the throughput for sql and cypher in 30 seconds and chart 5. shows the graphical representation of it, from the graph we can say that the number of queries executed by cypher in given time interval is more than that of the sql queries.

Queries	q1	q2	q3	q4	q5	q6
MariaDB	32	33	42	45	40	38
Neo4j	38	39	45	46	48	43

Table-6: Throughput in 30 secounds.

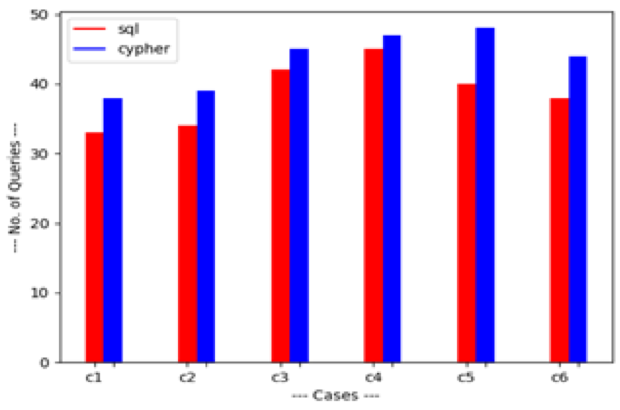


Chart-5: Throughput in 30 seconds.

7. CONCLUSION

From the above discussion we can say that the graph databases are more scalable and flexible than relational databases as new relationships can be added to graph databases without the need to restructure the schema again. In this project we examine both the databases using different parameters i.e. retrieval time complexity, schema load time and throughput from that we conclude that the time required for execution of the queries is less in neo4j than the MariaDB, the schema load time is less in neo4j than in MariaDB and and for throughput neo4j executes more number of queries in given time than MariaDB. Also we examine the retrieval queries for 100 objects and 500 objects respectively for both the databases and we come to the conclusion that increasing number of nodes from one hundred to five hundred does not increase the retrieval time much.

8. FUTURE SCOPE

Future research could include building up user friendly web application for altering the data of graph database and building up an IFC trade interface for sub-models or blending models. The future work will include also developing special stored procedures which allow accessing the Java API of Neo4j directly and running the import and data retrieval queries much faster compared with solely using of Cypher commands and followed with a benchmark to compare the performance with other existing query approaches.

REFERENCES

[1] Wisal Khan,Ejaz ahmed,Waseem Shahzad, “Predictive Performance Comparison Analysis of Relational & NoSQL Graph Databases,”National University of Computerand Emerging Sciences,Islamabad, Pakistan.

[2] Steve Ataky Tsham Mpinda1, Lucas Cesar Ferreira1, Marcela Xavier Ribeiro1, Marilde Terezinha Prado Santos1, “EVALUATION OF GRAPH DATABASES PERFORMANCE THROUGH INDEXING TECHNIQUES”, 1Department of Computer Science – Federal University of São Carlos (UFSCar) São Carlos – SP – Brazil.

[3] Sotirios Beis, Symeon Papadopoulos, and Yiannis Kompatsiaris, “Benchmarking graph databases on the problem of community detection”, Information Technologies Institute, CERTH, 57001, Thermi, Greece.

[4] David W. Williams, Jun Huan, Wei Wang, “Graph Database Indexing Using Structured Graph Decomposition”, 1Department of Computer Science University of North Carolina, Chapel Hill 2Department of Electrical Engineering and Computer Science University of Kansas 1{dwwill, weiwang}@cs.unc.edu, 2jhuan@ittc.ku.edu.

[5] Shalini Batra, Charu Tyagi, “Comparative Analysis of Relational And Graph Databases”.

[6] Harsha R.Vyawahare, Dr P.P.Karde, “An Overview on Graph Database Model”, Assistant Professor, Dept. of CSE, Sipna College of Engineering and Technology, Amravati, Amravati University, India Head, Dept. of CSE, Govt. Polytechnic, Yavatmal, Maharashtra, India.

[7] Mary Femy P.F, Reshma K.R, Surekha Mariam Varghese, “OUTCOME ANALYSIS USING NEO4J GRAPH DATABASE”, Department of Computer Science and Engineering, Mar Athanasius college of engineering, Kothamangalam, Kerala, India.

[8] Priyanka1, AmitPal2, “A Review of NoSQL Databases, Types and Comparison with Relational Database”, Department of Computer Science and Engineering Guru Nanak Dev University Regional Campus, Gurdaspur, India.

[9] 1Reena, 2 Renu, “ RELATIONAL DATABASE WITH SQL AND GRAPH DATABASE” ,1,2 Computer Science & Engineering Department, M.D.U, (India.).

[10] Surajit Medhi 1 , Hemanta K. Baruah2, “RELATIONAL DATABASE AND GRAPH DATABASE: A COMPARATIVE ANALYSIS”, 1Department of Computer Science, Gauhati University, Assam, India 2Bodoland University, Assam, India surajitmdh@gmail.com; hemanta_bh@yahoo.com

[11] Neha Tyagi1, Neelam Singh2, “ Comparative Analysis of Graph Database and a Relational Database”, M.Tech Scholar1, Assistant Professor2 Dept of Computer Science & Engineering GEU Dehradun, India.

[12] Garima Jaiswal, Arun Prakash Agrawal, “ Comparative analysis of Relational and Graph databases”, 1(Amity School of Engineering & Technology Noida, India) 2(Amity School of Engineering & Technology Noida, India).

[13] Graham Kirby, Conrad de Kerckhove, Ilia Shumailov, Jamie Carson, Alan Dearle, Chris Dibben,Lee Williamson, “Comparing Relational and Graph Databases for Pedigree Data Sets”. School of Computer Science University of St Andrews Fife KY16 9SX, ScotlandLongitudinal Studies Centre Scotland Universities of St Andrews & Edinburgh.

[14] ShefaliPatil1 , GauravVaswani2 , Anuradha Bhatia3, “Graph Databases- An Overview”, 1Student, ME Computers, Terna College of Engg, Navi Mumbai 2 Student, , Computer Technology, VESIT, Mumbai 3 Computer Technology Department, VES Polytechnic,Mumbai.

[15] Mukul Sharma, Pradeep Soni, “ Quantitative Analysis and Implementation of Relational and Graph Database Technologies”, Department of Computer Science & Engineering SBCET, Jaipur.

[16] Divya Kumawat1, Aruna Pavate2, “Correlation of NOSQL & SQL Database”, 1(Computer Dept., Atharva College of Engineering, Malad(W) , India).

[17] Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, Dawn Wilkins, “A Comparison of a Graph Database and a Relational Database”, Department of Computer and Information Science University of Mississippi.

[18] Adrian Silvescu, Doina Caragea, Anna Atramentov, “Graph Databases”, Artificial Intelligence Research

Laboratory Department of Computer Science Iowa State University Ames, Iowa 50011.

[19] Pallavi Madan, Anuj Saxena, " Review: Graph Databases", Graphic Era University, Dehradun, India.

[20] Park, Y., M. Shankar, B.-H. Park, and J. Ghosh (2014). Graph databases for large-scale healthcare systems: A framework for efcient data management and data services. In Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on, pp. 1219. IEEE.