Outcome Analysis of Relational and Graph Databases

[1]Shubhangi S. Marudkar, [2]Harsha R. Vyawahare
[1]ME Student, [2]Professor at Sipna C.O.E.T.
Department of Computer Science and Engineering, Sipna college of Engineering, Amravati,India

***Abstract*—Getting together enormous measures of complex data like information and learning is exceptionally regular these days. This requires the need to represent, store and manupulate complex data. From most recent three decades, the relational databases are being utilized as a part of numerous associations of different natures, for example, Instruction, Wellbeing, Business and in numerous different applications. Conventional databases indicate colossal execution and are intended to deal with organized information with ACID (Atomicity, Consistency, Isolation, Durability) property to oversee information uprightness but relational databases can't process appropriately and oversee substantial measure of information proficiently. Presently a day's advancements are moving towards movement, UI, Web of things, Program based IDEs and so on. These advancements require constant reaction and vast information store. A conventional database framework recovers and oversees database in a forbidden shape, yet in current situation of conveyed huge scale database those databases does not perform well. To conquer the impediments of customary databases, and to cover the necessities of current applications has lead the improvement of new database advances, for example, graph databases. We are showing an orderly examination of relational and graph database models, for that we are utilizing MySQL and Neo4j. Here we will think about the reasonableness of two classes of databases that is Relational database and graph database for putting away and questioning datasets. We will report aftereffects of estimations of scalability, query performance, and ease of query expression utilizing synthetic datasets.**

***IndexTerms*— neo4j; Graph database; scalability; query performance; query expression.**

I. Introduction (Heading 1)

Now-a-days, data is being expanded rapidly in the industry. The nature of data is varied and diversified such as unstructured, semi-structured and structured data[1]. The issue is not only how to store and access such big amount of data but also need to extract meaningful knowledge from such data rapidly. The relational model has dominated the computer industry since the 1980s mainly for storing and retrieving data[7]. Traditional databases require declarative language such as SQL to manipulate structured data[1].Way back people used database just for storing tabular data like purchase reports and finance records[5]. Relational databases are based on data consistency and can process the data at certain limit. To manage large datasets using relational databases, organizations are required to increase their system capacity such as RAM, Disk, optimized methods of accessing data etc[1]. many organizations are rely on unstructured data such as emails, blogs, audios, videos, images and such data is generated at very high speed[1]. To cover the requirements of current application domains, has lead the development of new technologies called NOSQL databases[15]. one of them is graph database. The increase of massive and complex graph-like data makes a graph database a crucial requirement. 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 information as graphs including nodes and edges. The two nodes and edges can be commented on with names. Neo4j provides a succinct query language, Cypher, that is designed to make it easy to express graph traversal queries, and efficient to execute them[14]. For instance, it is exceptionally easy to express a question to discover the greater part of the nodes inside a given number of edges of a given node—an errand that is bulky in SQL. Then again Implementing such sort of issues in relational databases includes extensive number of joins which are costly to ascertain.

The relational databases depends on rigid schema and make it difficult to add new relationships between objects while The Graph databases are additive in nature that means we can add new kinds of relationships, new nodes, and sub graphs to an existing structure without disturbing existing queries and application functionality[5]. A graph database provides most of the major components in DBMS, eg. external interfaces (user interface or API), database languages (for data definition, manipulation and querying), query optimizer, database engine (middle-level model), storage engine (low-level model), transaction engine, management and operation features (tuning, backup, recovery,etc.)[19]. Neo4j is reasonable for full venture organization. For instance, Facebook and Google+ datasets which comprise of billions of edges, a large number of refresh rates every second and require complex stockpiling 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].

II. RELATED WORK/LITERATURE SURVEY

The relational database management system was introduced in the 1970s[7]. And soon it became the primary data storage structure in academic and commercial pursuits. Graph database researches was popular in the early 1990s, but died out for a series of reasons including the surge of XML research and hypertext[7][17]. With the rise of the Internet the data began to increase day by day, because of that the software developers have been investigating storage alternatives to relational databases[17]. NoSQL is a term for some of those new systems. BigTable, CouchDB, Cassandra, Project Voldemort, and Dynamo are all NoSQL projects, as they are all high-volume data stores that reject the object-relational and relational models. Atomicity, consistency, isolation, and durability (ACID) are a set of governing principles of the relational model. They guarantee database reliability. NoSQL rejects ACID[7]. The graph model was used to represent huge amounts of data more than it had in the past[16]. Traditional data stores were capable of handling graph data. Yet, they were neither designed to do so nor efficient at it. There was clear desire for data store tailored to the needs of graph data. A few late research endeavors have concentrated on preprocessing graph databases with the objective of enhancing question times[4].

The idea learning framework in light of substructure disclosure is Graph Based Induction (GBI) (Yoshida, Motoda, 1995). It employs coloured digraph as the representation framework where colours attached to the nodes represent the attributes of the facts. GBI examines each connected pair of nodes, and merges the frequent typical ones. The final merged substructures are labelled as concepts[2].

Domingues-Sal et al.evaluated the execution of three graph databases (Neo4j, HypergraphDB and DEX) and a RDF Database (Jena). The tests, that incorporated the assessment of a few average graph operations over various graph sizes, demonstrated that DEX and Neo4j were the most efficient implementations[4].

B.T. Messmer et al. presents a decision tree approach for indexing models for isomorphism and subgraph Isomorphism. This strategy creates replies in polynomial time, at the cost of a record which is exponential in measure as for database size[4].

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

Angles, R.; et al presents a survey of earlier work (pre-NOSQL) in graph databases. i.e. prior to 2002,particularly geographical, spatial and semi structured database models. Older data models focused heavily on semi structured and XML data in a traditional database. The authors synthesized the notion of a “graph database model” and compare proposals available at the moment. Angles, R.; performs comparison & performance analysis of different graph database models compares current graph databases concentrating on their data model features, that is data structures, query facilities, and integrity constraints. Author shows that most graph database models provide an innate support for different graph structures, query facilities in the form of APIs (most of the models) and query languages (a few of them), and basic notions of integrity constraints[6].

Philip Howard, examiner at Bloor Research detailed that Graph databases are basic when the level of division [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 traditional innovation. Prophet or DB2, for instance, can sensibly deal with up to three degrees of detachment yet not the six or seven. Howard remarked on the constraints of graph databases, saying that "The real impediment is that while these are actually NoSQL databases, practically speaking they can't be executed over a minimal effort group (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 downside is that possibly 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 exactly thinks about graph databases ie shows Graph Database Benchmark, to look at four graph databases: Neo4j, DEX, Titan (BerkeleyDB and Cassandra) and OrientDB (neighborhood) on various sorts of workloads, each time recognizing which database was the best and the less adjusted. In light of measure, the database that got the best outcomes with traversal workloads is certainly Neo4j: it outflanks the various competitors, in any case the workload or the parameters used[6].

General frameworks for creating engineered records for benchmarking and testing record linkage approaches are portrayed in (Christen and Vatsalan, 2013; Ioannou, Rassadko, and Velegrakis, 2013; Talburt, Zhou, and Shivaiah, 2009). The data generation tool used in this work is specialised for genealogical population structures, and the global configuration parameters that are supported reflect this[13].

Healthcare system in United States was generating more data and they required new technology to handle data analytics effectively. Data driven approach is used to handle data analytics in healthcare systems by using two independent tasks, data management and data services. Here, data management means storing the data with minimal redundant structure and error free. Data services describe various analytics queries such as join, search and statistical queries. The problem appeared due to the gap between data management and data services in relational databases. To overcome this problem, they presented an approach to convert third normal form (3NF) of relational databases in equivalent graph of Graph database. A graph database does not require creating more tables and replicating them unlike relational databases. For example, Neo4j is suitable in OLTP (online transaction processing) environment. Pregel is used where high latency and high through put have high priority. The experiments have shown that Graph database performed better than relational database (MySQL) in the heterogeneous environment of healthcare systems of United States in OLTP[20].

III. Evaluation parameters for relational and graph database

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

Level Of Support/Maturity

Maturity can be characterized by how well the framework is tested. In the event that a framework has been tested, more number of times, it implies it is more steady and more bugs have been discovered. Relational databases are more stable and mature than graph databases.

Security

MySQL has an great multi user support. However Neo4j does not have any built in mechanisms for overseeing security limitations and multiple users. It presumes a trusted environment. In spite of the fact that there is Access Control List security instruments yet even Access Control List administration is taken care of at application layer. Then again, there is extensive support for ACL based security in MySQL.

Flexibility

Although graph databases are more mature and secure when contrasted with graph databases, yet its schema is fixed, which makes it hard to extend these databases and less reasonable to manage ad-hoc schemas that evolve over time.

Ease of Programming

The ease of programming is task dependent.Graph traversals are are extremely straightforward in Neo4j on the grounds that it contains its own particular APIs. MySQL graph traversals are significantly more confounded and can involve looping or recursing through the graph possibly executing multiple expensive joins.

IV. Experimental setup

1. Apple is a fruit and Carrot is a vegetable.
2. Apple and Carrot can be eat together.
3. Fruit and Vegetable are the subclass of Food.

In Relational databases this data can be expressed as appeared in the following three tables.

Table 1. Object Table

ID	OBJECT NAME	OBJECT TYPE
01	Apple	Fruit
02	Carrot	Vegetable

Table 2. Relationship Table

ID1	ID2
01	02
02	01

Table 3. Subclass Table

TYPE1	TYPE2
Fruit	Food
Vegetable	Food

The similar information can be represented in graph database as shown in following Figure1:

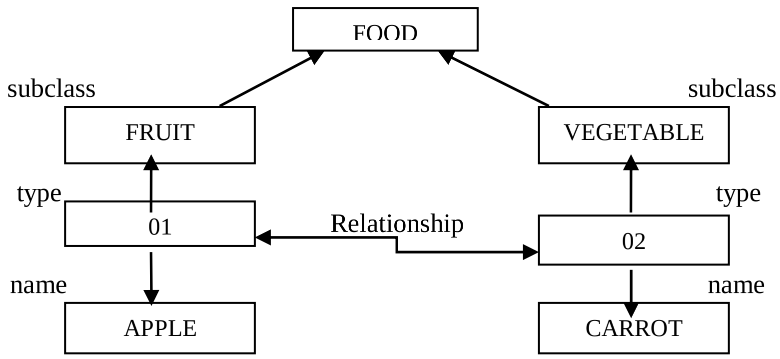


Fig1. Representation in graph database

Now if we want to add some new information for example:

- Apple and carrot both are food.

In Relational databse to store this information we need to restructure the entire relational database schema as shown in the following tables (table4, table5, table6 and table7).

Table 4.Object Table

ID	OBJECT NAME	OBJECT TYPE
01	Apple	Fruit
02	Carrot	Vegetable

Table 5. Relationship Table

ID1	ID2
01	02
02	01

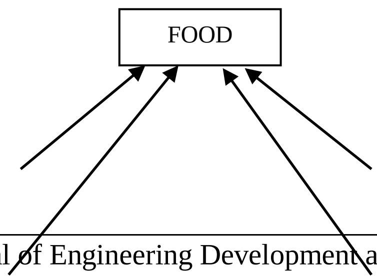
Table 6. Subclass Table

TYPE1	TYPE2
Fruit	Food
Vegetable	Food

Table7 . Type Table

ID	TYPE
01	Fruit
02	Vegetable
01	Food
02	Food

However in graph databases, there is no need to restructure the entire schema every time a new relationship is added; only a few edges and nodes are added to the graph[5]. For example the same information can be easily added in the graph databases without any restructuring of the original graph shown in figure2.



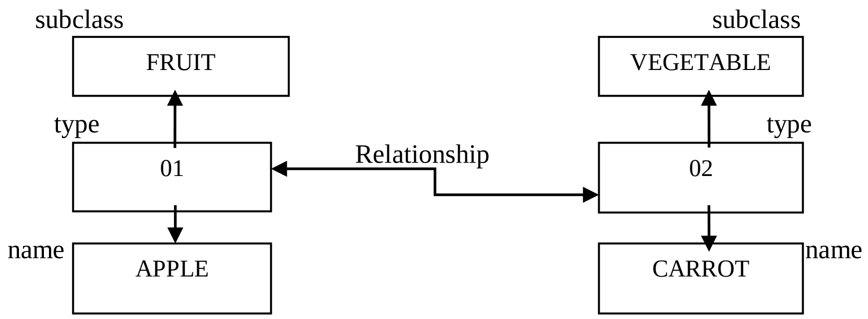


Fig 2. Modified Representation in Graph Database.

Henceforth, Neo4j has an effortlessly changeable composition while Relational databases are less mutable[15]. From above case we can state that relational model works best when there are relatively small and static number of relations between objects. At the point when the informational indexes wind up plainly bigger, they require costly join operations since they look through the majority of the information to discover the information that meets the inquiry criteria. The bigger the informational index, the more it takes to discover matches. On the other hand, a graph database does not scan the whole graph to discover the nodes that meet the search criteria. It looks only at records that are directly connected to other records, increasing the number of nodes which does not increased the retrieval time **.**

V. Analysis of Problem

. As we have found in the above area, storage of data is very important in all the fields, Relational databases have been the power-stallion of programming applications since many years and proceed with so this day. Relational databases, references to different rows and tables are indicated by referring to their primary key attributes by means of foreign key columns. This is enforceable with imperatives, yet just when the reference is never discretionary. Joins are figured at inquiry time by coordinating primary and foreign keys of the many columns of the to-be-joined tables. These operations are compute and memory-intensive and have an exponential cost. In this section we will see the issue happened when both the techniques will did. In graph database The complete examination of the relative suitability of various graph databases for storing genealogical structures. That work does not address the subjective assessment of capacity expenses and query execution. This Project work is an endeavor to defeat the issues, for example, qualitative evolution of storage cost and query execution performance.

VI. Proposed System

From the above Literature Review we can state that both the databases performs well yet if there should be an occurrence of sentenced information neo4j is superior to the relational databases. So the proposed work of this undertaking is to comprise of research and correlation of two databases, for example, Neo4j and MySQL databases. A graph database stores information in a graph, the most nonexclusive of information structures, able to do exquisitely speaking to any sort of information in a very available manner. MySQL has huge market infiltration in the scholarly and logical fields. Moreover MySQL has huge help, both from the makers and from the client group. We will compare them based on subjective parameters such as Maturity/level of support, ease of programming, flexibility and security and objective parameters such as Query evolution time. We will choose queries of varing complexity and of varing type such as pattern matching, reachability and summarization.

Objectives are:-

1. Identification of Dataset.
2. Implementation of Relational database i.e. MySQL.
3. Implementation of Graph Database i.e. Neo4j.
4. Application of Queries to Analyze the performance difference between Relational and Graph database.
5. Comparison of Relational and Graph database.

VII. Conclusion and Future Scope

From above study it is clear that the data processing in graph database is much faster as compare to relational database. Graph databases are much flexible than relational database as new relationships can be easily added to graph databases without restructuring the database schema again. This is the reason why the graph databases are mostly used in social media analytics and big data processing.

In future the data processing can be done using different graph databases such as FlockDB (Used by Twitter), Infinite Graph (Used by Facebook), HyperGraphDB and many more to identify the best application areas by analyzing these different Graph Databases for different objects.

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, IJACSA,Vol **8,** No**.5**,2017

[2].Mahmut I□dea*, Yusuf Kavurucua, Alev MutlubMahmut Idea*, Yusuf Kavurucua, Alev Mutlub, "Graph Representation of Relational Database for Concept Discovery", aTurkish Naval Academy ,Computer Engineering, Istanbul-34940, Türkiye bKocaeli University,Computer Engineering,Kocaeli-41000, Türkiye, Science Direct Procedia - Social and Behavioral Sciences **195 (2015) 1981 – 1989**

[3].Jaroslav pokorny,michal valanta,jiri kovacic, " Integrity constraints in graph databases", Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, 118 00 Prague, Czech Republic bFaculty of Information Technology, Czech Technical University, Thákurova 9, 160 00 Prague, Czech Republic, ScienceDirect Procedia Computer Science **109C(2017)975-981**

[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.

[5].Shalini Batra, Charu Tyagi, "Comparative Analysis of Relational And Graph Databases",IJCSE ISSN: **2231-2307**,Volume-**2**,Issue-**2,**may **2012**.

[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,IJIRCCE, Vol-**3**, Issue **8**, August**2015**.

[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, IJCI,Vol. **5,**No**. 2,** April **2016.**

[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, IJESC, Vol-**6**, Issue No. **5**.

[9].1Reena, 2 Renu, " RELATIONAL DATABASE WITH SQL AND GRAPH DATABASE" ,1,2 Computer Science & Engineering Department, M.D.U, (India.),IJARSE, Vol-**4,**Spacial Issue**(01),**March2015 ISSN-2319-8354(E).

[10].Deepak GC, "A Critical Comparison of NOSQL Databases in the Context of Acid and Base", St Cloud State University, gcde0401@stcloudstate.edu.

[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,IJESC,Vol-**7**,Issue No.**4.**

[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), IOSRJEN, e-ISSN:2250-3021,p-ISSN:**2278-8719** Vol-**3**, Issue **8**(August.**2013**),pp **25-27.**

[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, IJCSIT, Vol-**5**,2014,657-660,ISSN: **0975-9646.**

[15].Mukul Sharma, Pradeep Soni, " Quantitative Analysis and Implementation of Relational and Graph Database Technologies", Department of Computer Science & Engineering SBCET, Jaipur, IJMCSA, ISSN:**2321-2632**, Vol-**2,**Issue No-**5,**September,**2014.**

[16].Divya Kumawat1, Aruna Pavate2, "Correlation of NOSQL & SQL Database", *1(Computer Dept., Atharva College of Engineering, Malad(W) , India)*,IOSR-JCE e-ISSN*:***2278-0661**, p-ISSN:**2278-8727**, Vol **18**,Issue **5**, Ver. V(Sep-Oct**. 2016**), PP **70-74**.

[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, IJARCRCSSE, vol **5,** May **2014,** ISSN**:2277 128x**

[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 **30**th International Conference on, pp. **1219. IEEE.**