International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(04), December 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354

A Discourse on the Advent of Graph Database

Manan Doshi[1], Vruddhi Mehta[2], Yash Shah[3], Sakshi More[4]

[1,2,3,4]*Computer Engineering, Shri Bhagubhai Mafatlal Polytechnic (India)*

ABSTRACT

This is the era of artificial intelligence, big data analytics and internet of things. Computers have superseded the ancient method of processing and storing the written information. In order to deal with such enormous data created and destroyed every minute, databases were specially organized for rapid search, storage, retrieval, modification, and deletion of data in conjunction with various data processing operations. Relational database is based on the concept of sets and relations. It defines normal forms of tables that minimize storage requirements. Graph databases overcome the inefficiency of relational databases at handling heterogeneous data and maintaining the nodal relationships. Graph databases are like next generation of relational databases. It allows you to query and view your data from any imaginable point of interest. In this paper, we are presenting the concept of graph database, comparison of relational database with graph database and implementation of graph with Neo4j.

Keywords - data, data base, graph, Neo4j

I. INTRODUCTION

We live in a century of computers where data is at its core. There is a vast amount of data created every now and then; and this data demands to be stored for further use. For this purpose the term database comes into picture. A database is a collection of huge data that is stored in a specific structured format, so that it can be easily accessed, managed and updated by the computer. Traditional databases are organized by *fields*, *records*, and *files*. A field is a single piece of information; a record is one complete set of fields; and a file is a collection of records.

Relational database was the considered as the most efficient, back then. Relational database is the representation of data in structured and decent tables and thus laid an emphasis on the relationship between the data within. However data complexity, presumably slower searching speed, inefficiency at handling heterogeneous data posed a problem for relational database. Likely in other fields as the development kicked in, there was an advent of graph database. Graph database is the next generation of relational database. Graph database has facilitated us with the benefits of connected data. It is the best for dealing with complex, semi-structure, and densely connected data. It is very fast in terms of queries and gives a response in milliseconds and overcomes the limitations of relational database.

Graph database implements the property of graph data models and fulfills the complex linking of data. There aroused the need of graph database. We have included one of the implementations of graph database using Cypher on Neo4j platform. The remaining paper is organized as follows: Section II describes the Relational databases. Section III explains Graph data bases. Section IV discusses the comparison of relational and graph

databases. Section V describes neo4j technology. Section VI explains Cypher. Section VII illustrates the implementation of graph database using neo4j. Section VIII concludes the paper.

II. RELATIONAL DATABASE

A relational data base is a collection of dataitems organized as a set of formally-described tables from which data can be accessed or reassembled in many different ways without having to re-organize the database tables. The relational database was invented by E.F. Coddat IBM in 1970.

The standard user and application program interface to a relational database is the *structured query language* (SQL). SQL statements are used both for interactive queries, for information from a relational database and for gathering data for reports. In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend.

A relational database is a set of tables containing data fitted into predefined categories. Each table contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. [1]. Relational databases are powerful since they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table. [2]

The main idea behind these object oriented models is to represent data as collections of objects. Graph database models made their presence alongside with the object-oriented modelsIn2004, a family of models for representing technologies on the web was introduced by McGuiness. Reo Angels introduced the graph database models about consist of current graph databases and their support for query languages. In 2007 neo4j is introduced, based on property graph model. Infinite graph entered in 2009. Titan is adopted in 2012 which is based on property of graph model. Recently introduced in graph databases' models consist of their own query language to carry out various operations. The aim of this survey is to provide databases, their models and also presents a comparison based on some properties. [3].

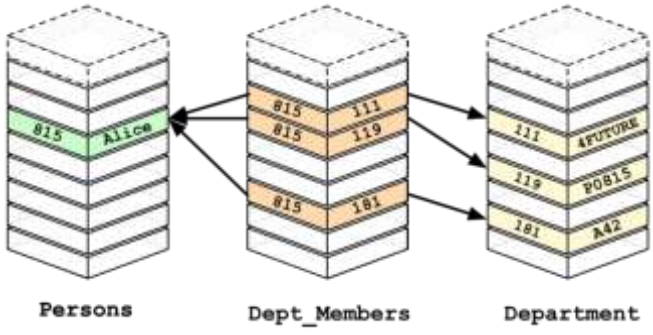


Fig. 1 Relational database[a]

III. GRAPH DATABASE

Formally, a graph is just a collection of vertices and edges {V,E}—or, in less intimidating language, a set of nodes and the relationships that connect them. Graphs represent entities as nodes and the ways in which those

entities relate to the world as relationships. This general-purpose, expressive structure allows us to model all kinds of scenarios, from the construction of a space rocket, to a system of roads, and from the supply-chain or provenance of foodstuff, to medical history for populations, and beyond.

In the contemporary era technology is rapidly changing and we are facilitating the benefits of connected data. Graph database is the best for dealing with complex, semi-structure and densely connected data. It is very fast in terms of queries and gives a response in milliseconds. Graph databases are highly useful in enterprise level like: - communication, healthcare, retail, financial, social network on-line business solution, on-line media etc. Graph database system follows CRUD (create, read, update, delete) methods that are used in a graph data mode land it also uses index free adjacency. Index free adjacency is important in order to high performance traversal. If any graph database utilizes this then every node maintains direct reference to the adjacent nodes It means query time is independent for total size of the graph and simply directly proportional to the length of the graph searched. Graph database produce results very fast in terms of query time and also stores large amount of data. Graph databases do not keep data into tables. There is a single data structure in a graph database – the graph and there is no join operation so every vertex or edge is directly connected to other vertex. Graph stores the data into nodes which have a few relationships. Graph databases follow property graph model. Graph databases are under construction for the purpose of transactions OLTP systems. Currently known graph databases come under NoSQL databases. An efficient graph database model is necessary for better management of graphs. Graph databases provide such models which are more closely to the user's problem. In the current era, graph database regained its popularity due to handle graph like structure in current applications and these are called the future of database management systems. [3]

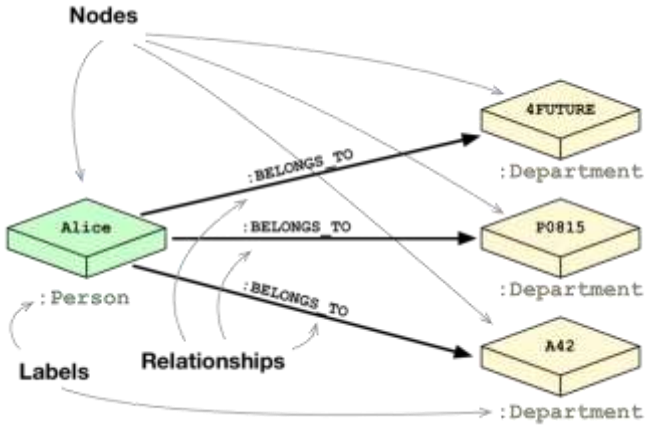


Fig 2 Graph Database[a]

A.ALLEGRO GRAPH DATABASES

AllegroGraph is oriented to meet the Semantic Web standards (i.e., RDF/S, SPARQL andOWL). It provides special features for GeoTemporal Reasoning and Social Network Analysis.[4]

B. DEX

Dex provides a Java library for management of persistent and temporary graphs. Its implementation, based on bitmaps and other secondary structures, is oriented to ensure a good performance in the management of very

large graphs.[4]

C. HYPER GRAPH DATABASE

Hyper Graph DB is a database that implements the hyper graph data model where the notion of edge is extended to connect more than two nodes. This model allows a natural representation of higher-order relations, and is particularly useful for modeling data of areas like knowledge representation, artificial intelligence and bio-informatics.[4]

D. INFINITE GRAPH

Infinite Graph aims the efficient traversal of relations across massive and distributed data stores. Its focus of attention is to extend business, social and government intelligence with graph analysis [4].

E.NEO4J

Neo4j is based on a network oriented model where relations are first class objects. It implements an object-oriented API, a native disk-based storage manager for graphs and a framework for graph traversals [4].

IV. COMPARISON OF RDBMS AND GRAPH DATABASE

In relational database a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

Graph models are introduced to overcome the limitation imposed by traditional database models to capture the inherent graph structure like data appeared in the applications.

Parameter	RDBMS	Graph Databases
Data Storage	Storage is fixed, pre-defined tables with rows and columns with connected data cripples query efficiency	Graph storage structure with index-free adjacency results in faster processing.
Data Modeling	Translated from a logical model to a physical one. Any changes require weeks of downtime for implementation	Flexible data model with no mismatch between logical and physical model. Data types and sources can be added or changed at any time
Query Performance	Data processing suffers with the number and depth of JOINs	Performs, regardless of the number or depth of relationships
Query Language	SQL	NOSQL
Transaction Support	ACID transaction support required for consistent and reliable data.	Retains ACID transactions for fully consistent and reliable data
Processing at Scale	Complex data relationships are not harvested at scale.	Graph model inherently scales for pattern-based queries.. Massive scale up possibilities
Data Center Efficiency	Expensive in terms of purchase, energy use and management time.	Performance improving as complexity and scale grow. Efficient use of hardware.

Table of Comparison [5]

V. NEO4J

Neo4j is a highly scalable native graph database that leverages data relationships as first class entities, helping enterprises build intelligent applications to meet today's evolving data challenges. "World's leading graph database" was first released in 2007. Neo4j also supports other languages like python except java for graph operations. Neo4j is an open source project. It scales to billions of nodes and relationships in a network. Neo4j manages all the operations that modify data in a transaction. In Neo4j both nods and relationships can contain properties. Neo4j is a graph database that manages graphs and is optimized for graph structure instead of tables. It is more expressive type of graph database as compared to other graph databases. Neo4j is most popular graph databases today. Neo4j's working is based graph model called the "property graph model". It is a model that abstains some mathematical bits of graph theory for easy understanding and design. The property graph consists of nodes that are connected by relationships. Every relationship consist of two key features together a name and direction. These two provide semantic context for the nodes connected by the relationship. There are many ways to query in Neo4j, for example native traverser API or cypher query language. It supports full ACID (properties mean that once a transaction is complete, its data is consistent and stable on disk, which may involve multiple distinct memory locations.) transactions by having in memory transaction logs and lock manager. It does not support sharing. [3]

Neo4j extends its services to many potential clients. eBay Tackles E-Commerce Delivery Service Routing with Neo4j. Global 2000 Adidas uses neo4j technology for real time recommendation. Other uses include network impact analysis, route finding, logistics, fraud detection and authorization and access control. [5]

VI. CYPHER

Cypher is a declarative query language that allows for expressive and efficient querying and updating of graph data. Cypher is a relatively simple and very powerful language. Complex database queries can easily be expressed through Cypher, allowing you to focus on your domain instead of getting lost in database access. Cypher's construct uses English prose and iconography to make queries more self-explanatory.

Being a declarative language, Cypher focuses on the clarity of expressing what to retrieve from a graph, not on how to retrieve it. This is in contrast to imperative, programmatic APIs for database access. This approach makes query optimization an implementation detail instead of a burden on the user, removing the requirement to update all traversals just because the physical database structure has changed.

The Cypher language has been made available to anyone to implement and use as open Cypher, allowing any database vendor, researcher or other interested party to reap the benefits of developing a first class graph query language. [6]

VII. CYPHER

The data model in our case study includes nodes with six different labels (each with their own properties), and different types of relationships (one of which has its own property). The underlying structure of the database is visualized in the image below. In brief, the graph is made up of various State nodes that are related to each other by means of transport ie air, rail or road. We also include travel time as properties of relationships. This case

study will use the Cypher query language to introduce and explain the Cypher concepts for building graph databases.

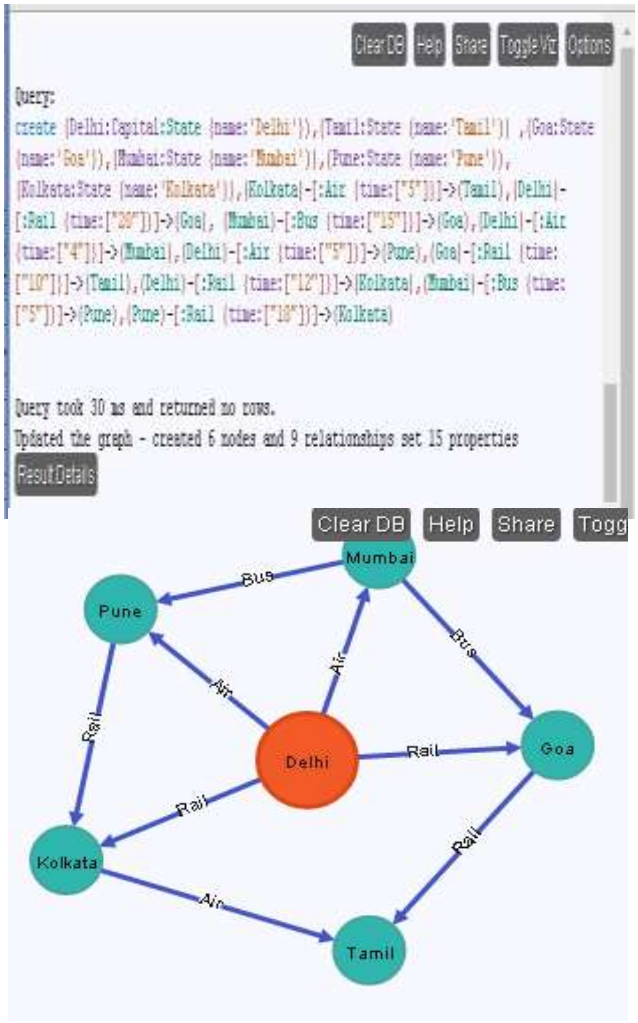


Fig 6 Creating graph database[b]

B.ADDING INFORMATION

We can include additional details about various places by incorporating properties such as population and places of interest. We first find the node using match and then set properties for it.

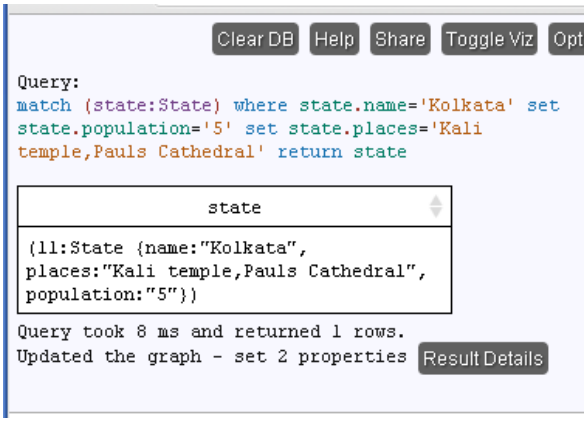
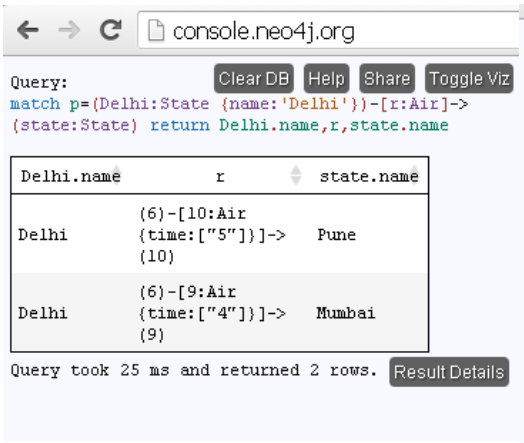


Fig 7 Adding information[b]

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(04), December 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354

C. SEARCHING INFORMATION

We find all places Delhi connects to via Airways and also the travel time by firing the query below. It first finds Delhi and returns all nodes having relation airways with Delhi along with time.



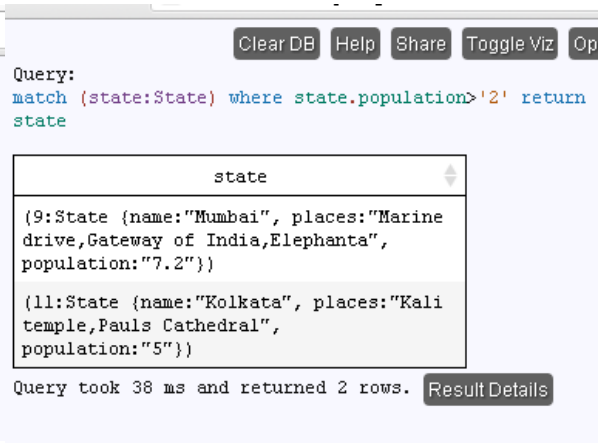
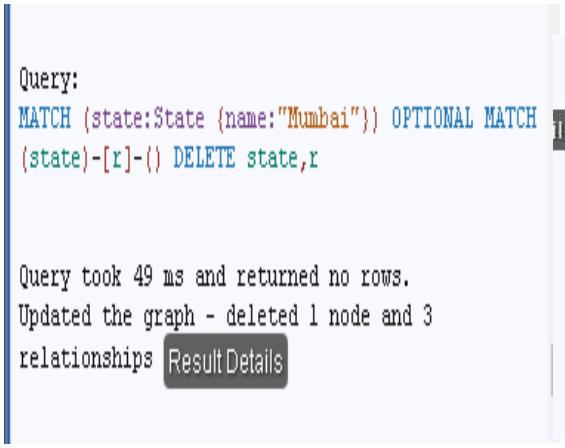


Fig 8 Searching information[b] **Fig 9 Searching of information [b]**

We find all states having population more than 2 crores by firing the query below. The nodes with population, as a property, greater than 2 are returned.

D. DELETING NODES

The statement below deletes a node – Mumbai along with all its relations:



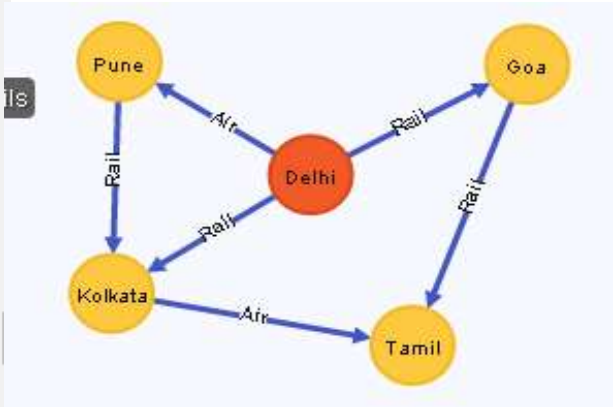


Fig 10 Deleting nodes[b] **Fig 11 Deleted node[b]**

VIII. CONCLUSION

The ability to handle complex semi structured data with less response time has enabled the graph data bases as a backbone for social networking applications. In this paper we presented the concept of graph database and some prominent comparison features of Relational database and Graph databases. At the later stage of paper we have implemented a small Graph database using Cypher on the Neo4j platform. The support of native graph storage, scalable architecture, and ACID compliance to ensure predictability of relationship-based queries is the basis of worldwide acceptance of Neo4j to realize enterprise oriented graph databases.

IJARSE
ISSN (O) 2319 - 8354

REFERENCES

[1] http://searchsqlserver.techtarget.com/definition/relational-database

[2] http://www.webopedia.com/TERM/R/RDBMS.html.

[3] Rohit Kumar Kaliyar,"Graph Databases: a Survey" in International Conference on Computing, Communication & Automation (ICCCA), 2015

[4] Renzo Angles."A comparison of Current Graph Database Modules" in 28th International Conference on Data Engineering Workshops (ICDEW), 2012 IEEE

[5] https://neo4j.com

[6] https://neo4j.com/developer/cypher-query-language/