Graph Databases: A Survey

Rohit kumar Kaliyar
Computer Science & Engineering
Shiv Nadar University
Greater Noida, India
rk880@snu.edu.in

***Abstract*— In the era of big data, data analytics, business intelligence database management plays a vital role from technical business management and research point of view. Over many decades, database management has been a topic of active research. There are different type of database management system have been proposed over a period of time but Relational Database Management System (RDBMS) is the one which has been most popularly used in academic research as well as industrial setup[1]. In recent years, graph databases regained interest among the researchers for certain obvious reasons. One of the most important reasons for such an interest in a graph database is because of the inherent property of graphs as a graph structure. Graphs are present everywhere in the data structure, which represents the strong connectivity within the data. Most of the graph database models are defined in which data-structure for schema and instances are modeled as graph or generalization of a graph. In such graph database models, data manipulations are expressed by graph-oriented operations and type constructors [9]. Now days, most of the real world applications can be modeled as a graph and one of the best real world examples is social or biological network.**

This paper gives an overview of the different type of graph databases, applications, and comparison between their models based on some properties.

Keywords—Graph database, RDBMS, big data analytics, type constructors

I. INTRODUCTION

In recent years, there has been developed a large number of systems for handling graph like data. In the current era, the importance of graph like data is very high in networks like; social, biological, and other networks. Graphs have been used for a very long time to model different types of domains. In biology, graphs are used to model genetic regulations and in social network graphs are used for modelling relationships between users. On the basis of work done in recent years we can classify graph data management system into two categories – graph databases and distributed graph processing frameworks. The main aim of distributed graph processing is to provide the solutions in case of mining massive graphs that is not possible on single machine due to some resource constraints. On the other hand graph databases implement property graph data model [1]. In property graph data model, graph structure's elements can have some user defined attributes. With the rise of big data there has been a huge demand to design data models and tools. These data models should be capable of handling a variety of data structures. Analysis of graph properties is deeply studied by data-mining community [25]. To enable multi-graph management multi-graph databases are designed by the database community.

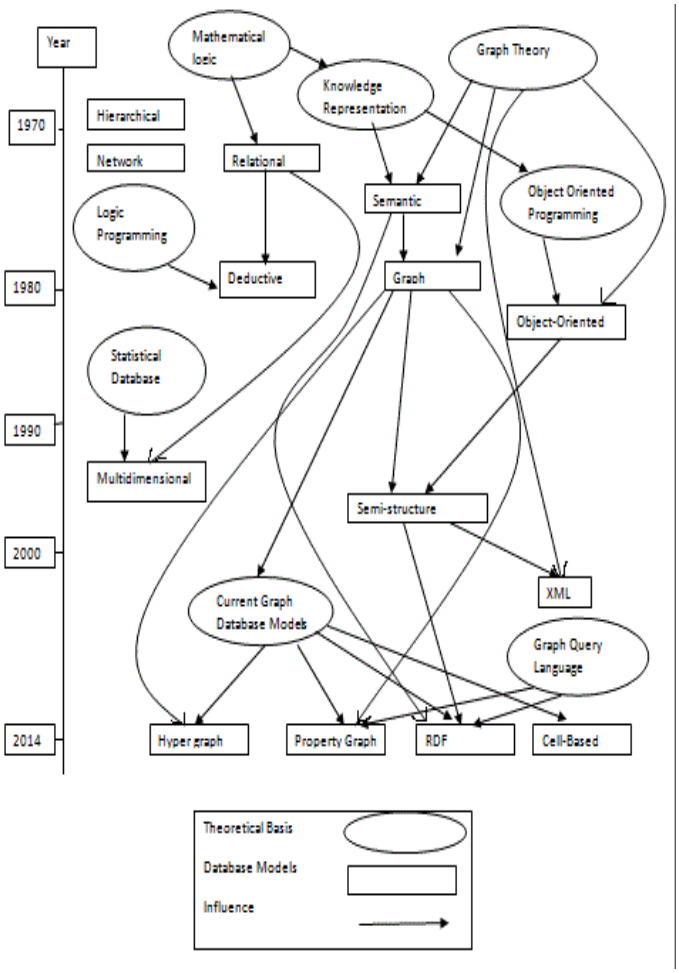

Fig. 1. Evolution of database models

In recent years, most popularity gained by relationship-oriented graph databases. In IT communities the term "data model" is widely used. A database model [8], [18] is a collection of conceptual tools to model representation of

entities and relationships. The term "database model" was first introduced in 1976. A database model has three main components: a set of data structure type, rules, and set of operations. A graph database model is very important for understand and manage the graph data.

Since the emergence of the database in the last few years, it has been a great ongoing debate about database models among researchers. As showed in Figure 1[8], there is a lot of diversity in the existing database models and there are many factors which influence their development. Figure 1 shows evaluation of graph database models. In this rectangle denote the database models, circles denote the theoretical models, and arrows denote the influences. On the left hand side time-line in years is indicated.

Each database model is based on some theoretical principal and these principals play an important role in the development of the models. It is clearly shown in the figure that before the relational database models; the main focus was on the actual file system. In 1976 the network model [20] was introduced where the information was in the forms of records.

Relational database model was presented by Codd [18], [19]. The core idea behind introducing was to make a separation between logical and physical level. It is based on the concept of sets and relations. Since increase popularity gained by relational database models in some business applications, Peckham and maryanski [8] introduced objects and their relations in a natural and clear manner. The object-oriented model was capable to capture domain semantics. Kim introduced object-oriented models [21] in 1990 but these had been previously appeared when most of the researchers were concerned with advancing systems for new applications.

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 models.

We can define a Graph database model such that in which the instances are modelled as graphs. These models were very famous in early 1990 with object-oriented models but their influence is died due to the emergence of other graph database models. These new graph database models were geographical, XML, and semi-structured. In the current era we need the information with graph-like data so this field has regained popularity. In this paper, the work done in the field graph database-modelling, query languages, and some features of graph databases is defined based on data structures. These models are introduced to overcome the limitation imposed by traditional database models to capture the inherent graph structure like data appeared in the applications.

Reno angels and Gutierrz [8] introduce a survey of different graph database models. In this survey the information about the evaluation of database models. This analysis provides us a historical data and very in-depth knowledge of database models and also some knowledge about graph databases.

Semi-structured database models [22] were adopted in 1997 by Buneman. These models are intended to model the data with a flexible data structure like web pages or documents. XML (extensible Mark-up Language) model is adopted in 1998 by Bray. The core focuses of this model is based on a tree like structure. In 2004, a family of models for representing ontology's on the web [23] was introduced by McGuinness. Reno Angels [24] introduce the graph database models about consist of current graph databases and their support for query languages.

In recent developments in the area of graph databases, four main graph databases are introduced based on these models named RDF, cell based, property graph and hypergarph. In RDF model, the principal focus is to handle the semi-structured data. In cell based models, typically handle the data which is a graph like structures.

In 2007 Neo4j [3] is introduced, based on property graph model. Infinite graph is entered in 2009. Titan is adopted in 2012, which is based on property graph model. Recently introduced in graph databases models consist their own graph query language to carry out various operations.

The aim of this survey is to provide the current graph databases, their models, and also presents a comparison based on some properties.

II. GRAPH

In recent few years the way Internet and mobile communication has been used for different and varied needs and applications by a common user, academicians, researchers have been started rethinking as how to store the huge data which is being generated every day, every hour and every minute. This rethinking for the storage and retrieval of data and information brought back the concepts of graph and graph models.

Graphs are used to model complicated structures. Graph is a collection of nodes,edges, and the relationships between them. In graph nodes are called entities and there are many ways in which these entities are co-related in different type of applications. The connection between these entities is called as relationship. In graph data term "Attributes" related with entities and relationships are called labels. In a graph like structure data is stored into nodes and these nodes have some properties. In graphs, relationships consist of properties and connect one node to the other node.

III. GRAPH DATABASE

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 model and it also uses index free adjacency [2]. 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 is referred to as a micro index for other nodes and cheaper than using global indexes. It means query time is independent for total size of the graph and simply directly proportional to the length of the graph searched. It simply means that the connected nodes in the database always point to each other. Graph database produce results very fast in terms of query time and also stores large amount of data.

Graph databases [3] 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 [1]. These are designed for transaction integrity and operational availability. 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. These models are simple in nature, but more expensive as compared to relational databases and other NoSQL databases. 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.

IV. CURRENT GRAPH DATABASES

Graph databases support fast traversal and this is the main reason that we don't use a tabular database like HBase and Cassandra to store the data in some web applications. In this section there is a description about some current graph databases, based on current database models. In the current era, there are many graph databases which are benefitting our business especially in IT. Description of these databases is given below:-

A. Neo4j(Neo Technology)

Neo4j is a disk-based transactional graph database and named as "World leading graph database". Its first release date is in 2007. Neo4j also supports other language like Python except Java for graph operations. Neo4j is an open source project [4] available in a GPLv3 Community edition, with Advanced and Enterprise editions available under both the AGPLv3 as well as a commercial license. Neo4j is best graph database for enterprise deployment. 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 nodes and relationship 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 is similar to other graph databases. Neo4j is most popular graph databases today [26].

Neo4j's working is based graph model called the "property graph model [4]". It is a model as showed in Figure 2 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 [1]. It supports full ACID transactions by having in memory transaction logs and lock manager. It does not support sharding.

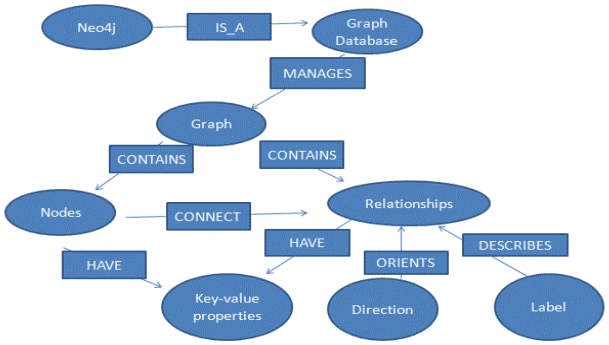

Fig.2. Graph Model

B. DEX

DEX [12] is said to be very efficient and bitmaps-based graph database and written in C++. It was first released in 2008. It makes graph querying possible in different networks like social network analysis and pattern recognition. It is also known as high performance graph database in the case of large graphs and useful for most of the NoSQL applications. Latest version of DEX supports both java and .NET programming. It's portable and requires only a single JAR file for execution. DEX is called the fourth most popular graph database today [25].

DEX consists of good integrity model for temporary graphs. Due to its functionality it gives good results in the applications like IMDB. It is being used in various social networking applications, providing better result with his java API. It can support up to 1 million nodes. As shown in Figure 3[27] that C++ DEX Core is the key and only a single JAR is required. It has two main layers called java API layer and DEX core layer.

We use a graph data model in DEX called the "Labeled and directed attributed multigraph" because edges can be directed or undirected. There exists more than one edge between two edges. DEX uses its bitmap based storage due to its light and independent data structures [12]. Data structure used in DEX is "Link" which is a combination of map and number of bitmaps that is useful for fast conversion between an object identifier and its value.

The main structure behind the map is B+ tree. The values are stored as UTF-8 strings and the element identifiers. These identifiers are 37 bit unsigned integers. These identifiers are compressed in order to reduce the size of the structure. DEX offers partial ACID transaction support because atomicity and isolation cannot be always guaranteed.

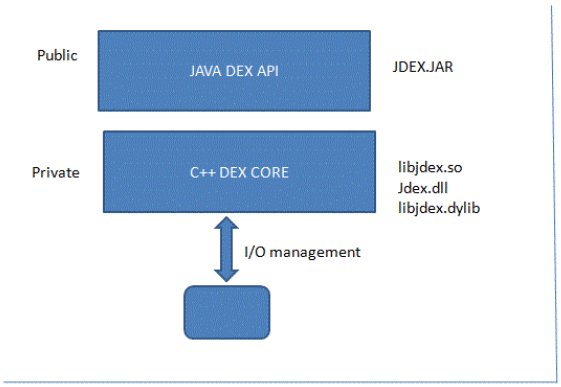

Fig.3. Architecture of DEX

C. *InfiniteGraph (Objectivity)*

InfiniteGraph is produced by an organization called Objectivity. It is a type of company that works to develops database technologies supporting large-scale, object persistence and relationship analytics.

Infinitegraph database is a distributed graph database in java and it is based on a graph like structure. We can call infinite graph as a cloud enabled graph database. It is designed for to handle very high throughput [13], [14], [15]. It is a single graph database distributed across multiple machines. There is a lock server which handles lock requests from database applications.

It is capable to deal with complex relationship requiring multiple hops. It provides graph-wise indexes on multiple key fields and also provides high performance in terms of query.

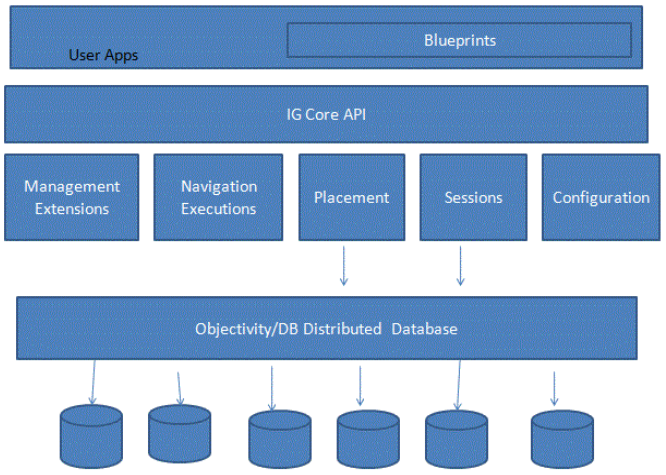

Fig.4. Architecture of InfiniteGraph

It natively compatible with the concepts of vertices and edges and also provides fast traversal result using their API. It is also called flexible graph visualization tool and perform navigation queries. The REST interface is required for interactive access to a database from a browser.

D. *Infogrid*

Infogrid [8] is called a web graph database with many additional software components, whose functions are framed for web applications. It is developed in java. It is used in standalone graph database as compare to all other infogrid projects.

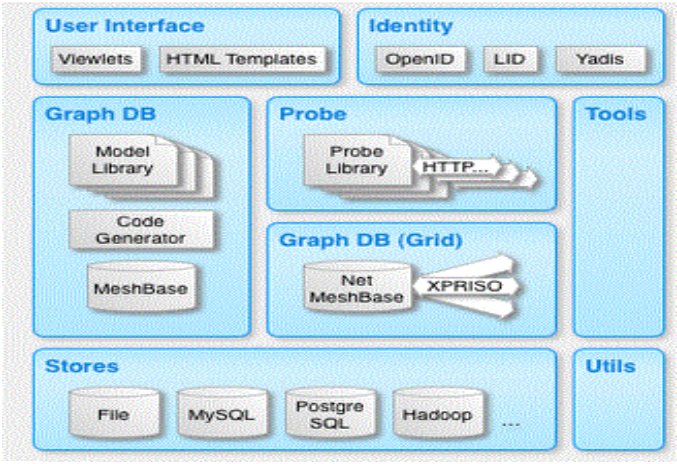

Fig.5. Architecture of InfoGrid

It is clear from Figure 5 that GraphDB is not the major part in infogrid framework. InfoGrid [8] consist some applications in OpenID project. The main weakness of InfoGrid is that its new application which is written in java is incomplete. It gives an abstract interface to store technology like SQL. Project also implements a library of MeshWorld examples.

E. *HyperGraphDB*

It is an open-source database basically supports hypegraphs. Hypergraph [5] is different from the normal graph because in this edge is points to the other edges. In various fields, it is used in the modelling of the graph data. It supports online querying with an API written in java.

It is based on the HyperGraphDB model. It is a universal data model highly complex and large scale knowledge application. It has graph-oriented storage and customizable indexing. In this graph database, a hyperedge is easy to convert into tuple. It is a distributed and graph oriented database. In this V (nodes) + E (edges) = A (atoms). It is invented by Dr. Ben Goertzel for an AGI system.

The main benefit using hypergraphDB is its high order logic. In these models of relations are more compact and naturally represented. It is used in many fields like computational biology and relational algebra. In this graph database many terms are used like atom, value, type, targetset (set of atom that an atom points to), arity (size of target set), link (an atom with arity>0), and incidence set (the set of atom pointing to an atom). HypergraphDB is used in artificial

intelligence and also in the field of bioinformatics. It shows more generality than any other graph database.

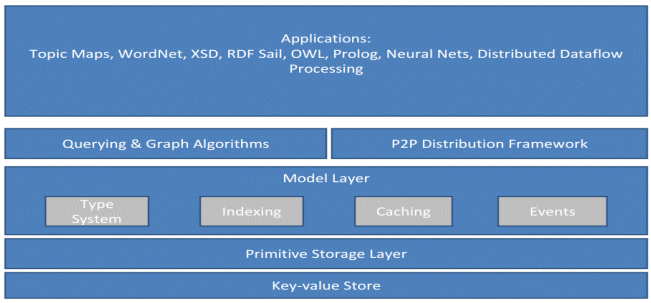

Fig.6. Architecture of HyperGraphDB

It shows more generality than any other graph database. In this architecture there are various layers like applications, model layer, primitive storage layer and a key value store. HypergraphDB is used in artificial intelligence and also in the field of bioinformatics.

F. Trinity

Trinity is a distributed graph system [6] over a memory cloud. Memory cloud is globally addressable in memory key value store over a cluster of machine. It provides fast data access power when we have large datasets. It is a large graph processing machine. It provides fast graph exploration and parallel computing for larger datasets. It also provides high throughput on large graphs which have billion nodes.

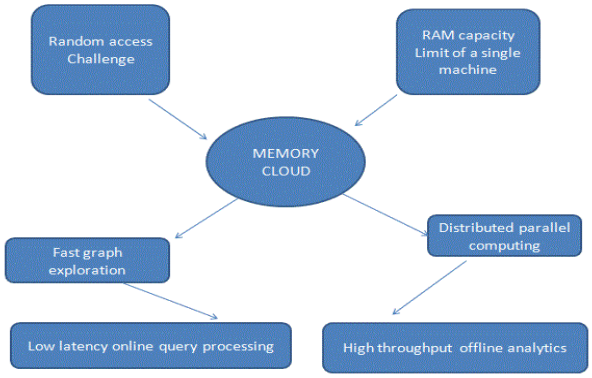

Fig.7. Architecture of Trinity

It is clear from Figure 7 that trinity is a costly system management and development tools. It has various declarative graph modeling utilities and in memory the data is highly compact.

Trinity provides C# API's to the user for various graph operations in various parts. Outside of Microsoft, its package is not open and follows hypergraph as the data model.

G. Titan

Titan [17] was adopted in 2012. It is written in java and an open source project. The main benefit to using titan is its scaling feature. It also provides support to very large graphs and scales with the number of machines in a cluster. It is also highly scalable graph database in terms of concurrent users and size of graph. It provides a batch graph processing with Hadoop framework and also gives answers for complex queries in milliseconds. It consists of three main components:

- Native Blueprints Implementation
- Gremlin Query language
- Rexster Server

It follows property graph model and supports Gremlin: a graph traversal query language. It also offers an optimized disk representation for efficient use of storage and speed of accessing data. Applications can interact with titan with mainly two ways:

- First Method is that calls Java-language API's related to titan which includes its native API implementation.
- TinkerPop stack utilities such as Gremlin query language built atop Blueprints.

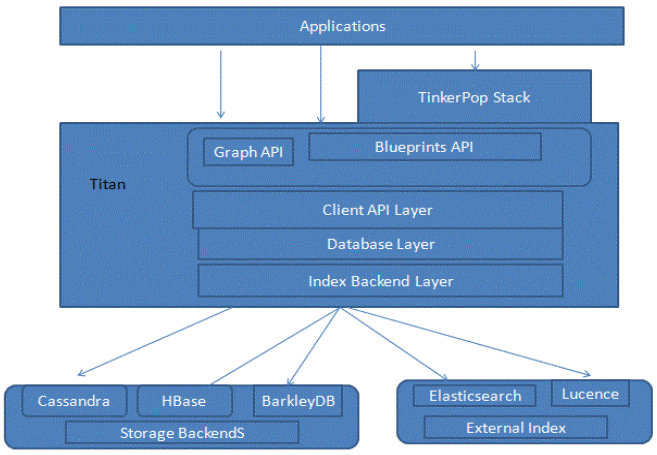

Fig.8. Architecture of Titan

In table I the term "Usability" means similar like capability of graph database and term "Reachability" means to handle large number of vertices in real world graphs to increase query time.

V. CONCLUSIONS

In this paper, our comparison is based on data modelling features of various current graph databases. As showed in table I, most of the graph databases there are different types of data structures, query APIs, available models, and some protocols which they follow. In this paper, a fine comparison is done in terms of graph database query languages also

TABLE I. GRAPH DATABASES COMPARISION

Features	Graph Databases						
	Neo4j	***DEX***	***Infinite Graph***	***Infogrid***	***HyperGraph***	***Trinity***	***Titan***
API	Java	C++,Java	Java	Java	Java	C#	Java
Free?	YES	YES	YES	YES	YES	NO	YES
Property graph	YES	YES	YES	YES	NO	NO	YES
Hypergraph	NO	NO	NO	NO	YES	YES	NO
Portable	YES	YES	YES	YES	YES	YES	YES
Protocol	REST/JSON	-	REST	REST/JSON	REST/JSON	C# language binding	REST
Query language	Cypher	SQL based	Gremlin	Web user interface with html	SQL style	SPARQL	Java
Graph type	Attributed	Attributed	Attributed	Simple	Hypergraphs	Attributed hypergraphs	Simple
Usability	Retrival	Retrival analysis	Retrival	Retrival analysis	Retrival	Retrival	Retrival
Rechability	Fixed length regular Simple path	Fixed length	Fixed length regular Simple path	-	-	Fixed length path	Fixed length

REFERENCES

[1] Robinson, Ian, Jim Webber, and Emil Eifrem. Graph databases. "O'Reilly Media, Inc.", 2013.

[2] Rodriguez, Marko A., and Peter Neubauer. "The graph traversal pattern." arXiv preprint arXiv:1004.1001 (2010).

[3] The neo database. [Online]. Available: http://dist.Neo4j.org/ Neo-technologyintroduction.pdf./Mannual2.1.5or http://neo4j.com/docs/stable/

[4] Buerli, Mike, and Cal Poly San Luis Obispo. "The current state of graph databases." Department of Computer Science, Cal Poly San Luis Obispo, mbuerli@ calpoly. edu (2012).

[5] Iordanov, Borislav. "Hypergraphdb: a generalized graph database." Web-Age Information Management. Springer Berlin Heidelberg, 2010. 25-36.

[6] Shao, Bin, Haixun Wang, and Yatao Li. "Trinity: A distributed graph engine on a memory cloud." Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. ACM, 2013.

[7] Infogrid: http://infogrid.org/trac/

[8] Angles, Renzo, and Claudio Gutierrez. "Survey of graph database models."ACM Computing Surveys (CSUR) 40.1 (2008): 1.

[9] Todd Hoff (June 13, 2009). "Neo4j - a Graph Database that Kicks Buttox". High Scalability. Possibility Outpost. Retrieved February 17, 2010.

[10] Gavin Terrill (June 5, 2008). "Neo4j - an Embedded, Network Database". InfoQ. C4Media Inc. Retrieved February 17, 2010.

[11] Martinez-Bazan, Norbert, Sergio Gómez-Villamor, and Francesc Escale-Claveras. "DEX: A high-performance graph database management system."Data Engineering Workshops (ICDEW), 2011 IEEE 27th International Conference on. IEEE, 2011.

[12] "The Rise of the Cloud Database". Readwrite. May 7, 2013. Retrieved September 8, 2014.

[13] "Georgetown University taps Objectivity for Big Data research". Readwrite. May 1, 2013. Retrieved September 8, 2014.

[14] Levi Gundert (December 11, 2013). "Big Data in Security – Part III: Graph Analytics". Readwrite. Retrieved September 8, 2014.

[15] Trinity: http://research.microsoft.com/en-us/projects/trinity/

[16] Titan: http://s3.thinkaurelius.com/docs/titan/0.5.0/

[17] graphs." Data Warehousing and Knowledge Discovery. Springer Berlin Heidelberg, 2013. 1-12.

[18] Codd, Edgar F. "A relational model of data for large shared data banks."Communications of the ACM 13.6 (1970): 377-387.

[19] Codd, Edgar Frank. "A relational model of data for large shared data banks."Communications of the ACM 26.1 (1983): 64-69.

[20] Taylor, Robert W., and Randall L. Frank. "CODASYL data-base management systems." ACM Computing Surveys (CSUR) 8.1 (1976): 67-103.

[21] Kim, Won. "Object-oriented databases: Definition and research directions."IEEE Transactions on knowledge and Data Engineering 2.3 (1990): 327-341.

[22] Buneman, Peter. "Semistructured data." Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. ACM, 1997.

[23] McGuinness, Deborah L., and Frank Van Harmelen. "OWL web ontology language overview." W3C recommendation 10.10 (2004): 2004.

[24] Angles, Renzo. "A comparison of current graph database models." Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on. IEEE, 2012.

[25] Jiawei, Han, and Micheline Kamber. "Data mining: concepts and techniques."San Francisco, CA, itd: Morgan Kaufmann 5 (2001).

[26] http://db-engines.com/en/ranking/graph+dbms

[27] http://blog.newitfarmer.com/newsql/gizzard/8739/repost-a-survey-on-graph-databases-for-java-developers