Available at: www.sabauni.net
Journal home page: www.sjitn.net

S.J.I.T.N
Saba Journal of Information Technology and Networking

Graph Database Model and Query Language for Graph Database

Dr. Muneer A. S.Hazaa, IbraheemAljamal
Thamar University, Faculty of Computer and information Technology

Article info

Article history :
Accepted :April 18, 2013

Keywords:
Graph Databases
Graph Databases Model
Query language

Abstract

Graph database has become a popular area of research in recent years, because theScientific datasets with structures,layers, hierarchy, geometry and arbitrary relations can't be accurately modeled using such traditional databases. Graphs become increasingly important in modeling complicated structures, such as circuits, images, chemical compounds, protein structures, biological networks, social networks, the Web, workflows, and XML documents.In this paper, we provide a Graph Database Model for Query languagethat have been proposed.

* Corresponding author: Dr. Muneer A. S.Hazaa
Tel. +967 734773066; Fax: +967 1 467919
E-mail address: Muneer_hazaa@yahoo.com

Introduction

In recent time, there have been few works regarding graph database's architecture, storage, performance, scalability etc.Graphsare widely used for representing data As a general data structure, they have become increasingly important in modeling sophisticated structures and their interactions, with broad applications including chemical informatics, bioinformatics, computer vision, video indexing, text retrieval, and Web analysis[10].The traditional relational database model (RDBMS) has been proved to be ineffective while handling such associated data with problems regarding horizontal scalability and dynamicdata handling[2].The remainder of the paper is organized as follows: The next section briefly describes the graph database is grounded on the concept of graph theory. Another section discusses the must graph database model to manage and efficiently store data. The paper then presents we give a brief overview of some of the graph query languages. the last section concludes the paper.

Graph Database

Graphs are suitable for capturing arbitrary relations between the various elements. A graph database uses graph structures with nodes, edges, and properties to represent and store data. Nodes represent entities such as people, businesses, accounts, or any other item you might want to keep track of, properties are pertinent information that relate to nodes, edges are the lines that connect nodes to nodes or nodes to properties and they represent the relationship between the two. Most of the important information is really stored in the edges. Figure (1) shows the data instance and how we represent it in graph instance.

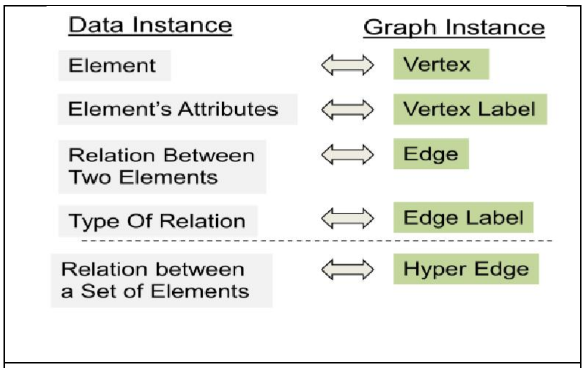

Figure 1: represent data instance in graph instance

General graph databases that can store any graph are distinct from specialized graph databases such as triple stores and network databases. They are based on graph theory[1][2][3].

Application Graph database:

- social networks
- information networks
- technological networks
- biological networks.

we will describe each category via an example.

1- social networks:

In social networks [13], nodes are people or groups, while links show relationships or flows among nodes. Some examples are friendships, business relationships, sexual contact patterns, research networks (collaboration, coauthorship),communication records (mail, telephone calls, email), computer networks [14], and national security [15]. There is growing activity in the area of social network analysis [Brandes 2005], and also in visualization and data processing techniques for these networks.

2- information networks:

Information networks model relations representing information flow, such as citations among academic papers [16], World Wide Web (hypertext, hypermedia) [17], peer-to-peer networks [18], relations among word classes in a thesaurus, and preference networks.

3- technological networks:

—In technological networks, the spatial and geographical aspects of the structure aredominant. Some examples are the Internet (as a computer network), electric powergrids, airline routes, telephone networks, delivery network (post office), and GeographicInformation Systems (GIS) are today covering a big part of this area (roads, railways, pedestrian traffic, rivers) [19].

4- biological networks.

Biological networks represent biological information whose volume, management and analysis has become an issue due to the automation of the process of data gathering. A good example is the area of genomics, where networks occur in gene regulation, metabolic pathways, chemical structure, map order and homology relationships among species [20]. There are other kinds of biological networks, such as food webs, neural networks, and soon. The reader can consult database proposals for genomics [21], an overview of models for biochemical pathways [22], a tutorial on Graph Data Management for Biology [23], and a model for Chemistry [24].

Graph database model

From database view , Database modelsis a collection of conceptual tools usedto model real-world entities and the relationships among them [3].

Graph database models appeared with the objective of modeling information whose logical structure is a graph. This models can be characterized as those where data structures for the schema and instances are modeled as graphs (nodes , edges , labels and direction) or generalizations (nested graphs ,hyper nodes) of them, and data manipulation is expressed by graph-oriented operations(Graph query language) and type constructors[3][4][8]. The architecture is diagrammed in Figure (2) [7].

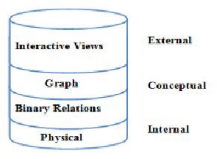

Figure 2: Graph DBMS Architecture

Graph structures are visible to the user. They allow a natural way of handling data appearing in applications (e.g. hypertext or geographic databases). Graphs have an important advantage: they can keep all the information about an entity in a single node and show related information by arcs connected to it. Graph objects (like paths, neighborhoods) may have first order citizenship; a user can define some part of the database explicitly as a graph structure , allowing encapsulation and context definition[3].

Rabi Chandra Shah[2], proposed a graph database model his model (Figure 3) consists of **three** basic layers:

- Presentation Layer
- Application Layer
- Storage Layer.

1- Presentation Layer

takes in user input and request information from the Application Layer. The information or page is then retrieved back to this layer after processing. Presentation Layer is the only layer that the use can interact with. Requests are sent to the Application Layer for retrieval or processing of graph data.First, Graph Database Engine translates request or data into a form understandable by the Graph Database.The translated form is then used for various operations (View Engine, Store Engine) within the Application Layer.

2- **The Application Layer**

directly interacts with the Storage Layer for storing manipulated data inside file.The primary tasks of the Application Layer are: (Translate rawdata,Process the data for operation, Send data to Storage Layer for storing in files, Implementation of Graph Operations, Implementation of Algorithms, Generate data and return to the Presentation Layer).

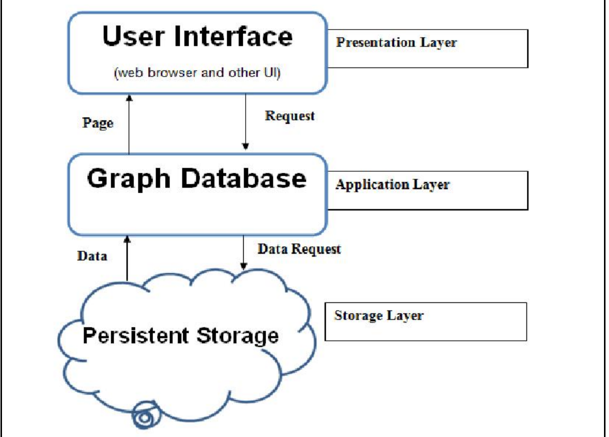

Figure 3:Level 1 Layer Diagram of the Rabi Chandra Shah proposed model

3- **Storage layer**
is responsible for managing the storage of graph data. The request or data from the Application Layer are handed to the Storage Layer with some prior storage instructions. This layer then interprets the instructions and accordingly, creates, appends or deletes raw data files. The primary tasks of the storage layer are: (Create, delete and edit data and files, Manage data, Encryption of data, Maintain the configuration file).

A. Data Structures

The representation of entities and relations are identified as being fundamental to graph database models. An entity or object represents something that exists as a single and complete unit. A relation is a property or predicate that establishes a connection between two or more entities. All graph db-models have as their formal foundation variations on the basic mathematical definition of a graph, directed or undirected graphs, labeled or unlabeled edges and nodes,e.g. hyper graphs, hyper nodes. On top of this basic layer, models present diverse features influenced by the semantic or object oriented approaches[2][3].

B. Integrity Constraints

Integrity constraints are general statements and rules, which define the set of consistent database states or changes of state or both. In the case of integrity constraints database models, In the case of integrity constraints database models are schema instance consistency, identity and referential integrity constraints, functional and inclusion dependencies[3].

Graph Query Language

A query language is a collection of operators or inferencing rules that can be applied to any valid instances of the data structure types of the model, with the objective of manipulating and querying data in those structures in any combinations desired. Among graph db-models, there is substantial work focused on query languages, the problem of querying graphs, the visual presentation of results, and graphical query languages ,due to the volume of the research done. Query languages for graph databases started to be investigated some 25 years ago. With much current data, such as linked data on the Web and social network data, being graph structured, there has been a recent resurgence in interest in graph query languages[3][4].

In order to define the GDB we need to specify: the Data Definition Language (DDL) that showing how to represent a graph, the Query Language (more generally, Data Manipulation Language - DML) and the Informal Semantics of the DDL and DML languages[9].

The Logical Database Model presents a logic language very much in the spirit of relational tuple calculus, which uses fixed sort variables and atomic formulas to represent queries over a schema using the power of full first order languages. The result of a query consists of those objects over avalid instance that satisfy the query formula. In addition the model presents an alternative algebraic query language proven to be equivalent to the logical one. The query languages G, G+, and GraphLog integrate a family of related graphical languages defined over a general simple graph model.G and GraphLog query languages are used in data model that of a labeled, directed graph[3][4][9].

The graphical query language G is based on regular expressions that allow simple formulation of recursive queries.Figure 3 shows an example of a graph G. Node labels in G denote names of authors, literary prizes and locations. Edge labels denote the hasWon relationship between authors and prizes (abbreviated w), the born in relationship between authors and places (abbreviated b), the lives in relationship between authors and places(abbreviated I), and the located in relationship between places. In G, a query is a set of pairs of graphs, each pair comprising a pattern graph and a summary graph. This pair of graphs essentially represents a CRPQ, with a set of such pairs being interpreted as disjunction.

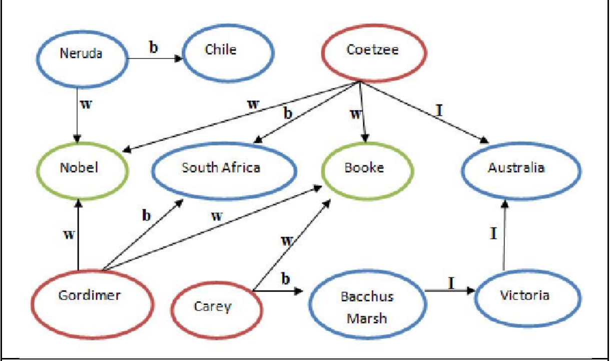

Figure 3: A graph of authors, prizes they have won, and places where they were born.

Gevolved into a more powerful language called G+ [Cruz et al. 1989], in which a query graph remains as the basic building block. A simple query in G+ has two elements, a query graph that specifies the class of patterns to search, and a summarygraph, which represents how to restructure the answer obtained by the query graph.

A SPARQL query language is extend to G+ that is the official W3C standard for querying and extracting information from RDF graphs. It is based on a powerful graph matching facility that allows the binding of variables to components in the input RDF graph.G-SPARQL proposed for querying attributed graphs. The language expresses types of querieswhich of large interest for applications which model theirdata as large graphs such as: pattern matching, reachability and shortest path queries[3][11].

GraphLog emplaced the summary graph with a distinguished edge, as shown in Figure(4). GraphLog also added edge inversion, negation and aggregation functions, while defining a semantics different from that of G. The semantics of G was defined in terms of matching simple paths in the graph being queried ,whereas the meaning of a GraphLog query was given by the meaning of the stratified Datalog program to which it was translated[4].

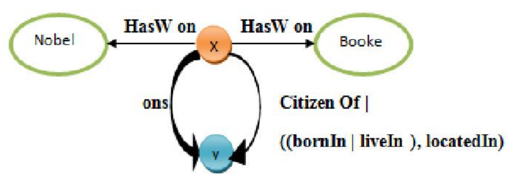

Figure 4: Query to find places related to authors who have won both the Nobel and Booker prizes.

Renzo Angles, Pablo Barcel, and Gonzalo Rıos [12] proposepropositionaldynamic logic (PDL) as a yardstick query language for graph database engines, based on the fact that it can express many relevant properties with very low computational cost. they present an implementation of the language that shows its potential applicability for querying massive graph databases by building on existing graph database support.

Other early graph query languages include GRAM [4] and GraphDB. The data models of both require the presence of a graph schema. Both provide regular expressions defined over alternating sequences of node and edge types. GraphDB includes object-oriented features such as classes for nodes and edges, as well as paths. The intended area of application for GRAM was Hypertext, while that for GraphDB was spatial networks such as transportation systems. As a result, GraphDB provides built-in operators such as one for shortest path. GOOD is another graph query language based on an object-oriented model [3][2].

GOOD's querying mechanism is via graph transformations: node addition/deletion and edge addition/deletion. Also provided is an abstraction mechanism to group objects by means of their properties, as well as methods for defining sequences of operations. GOOD gave rise to a number of successor languages, such as G-Log [2][3][4][11] and the update language GUL. A number of query languages were developed to query graphs represented in the Object Exchange Model (OEM) or one of its variants/derivatives. OEM was developed to model semi-structured data which had no predefined schema and could be heterogeneous[4].

In common with many graph query languages, Lorel uses a syntax based on OQL, allowing regular expressions over edge labels. A distinctive feature of Lorel is the availability of path variables.Say we wish to formulate a Lorel query Q equivalent to the ECRPQ shown in (3). In Lore, each nodehas an oid rather than a label as in Figure (3), so authors' names, for example, would be represented by separate atomic objects connected to author nodes by an edge labeled name, say. We also assume that Winners is a named entry point, with edges labeled author to author nodes. Then Q can be written as:

```
select X, Yfrom Winners.author A,
Winners.author X
A.#@P.Y , X.#@Q.Y
where A.name = 'Coetzee'
and path-of (P) = path-of (Q)
```

where # denotes a path of any length, so is equivalentto the regular expression Σ∗ used earlier. @Pbinds the path (of oids and labels) to variable P,and path-of returns a sequence of edge labels.

Query Language Functionality

The following subsections will consider functionality in terms of the following broad categories: subgraph matching, finding nodes connected by paths, comparing and returning paths, aggregation, node creation, and approximate matching and ranking a number of query languages for graphs have been proposed over the past few decades. In addition, many languages offer operations such as union (disjunction),composition and negation of queries, butwe will not cover these separately[4].

Conclusions

We have provided of data models and Query language languages for graph databases. Having this goal in mind, we design Graph Databases whose main building blocks are graphs representing uniformly data, knowledge, models and queries. The preference for graph structures came from the recent tendency of using graphical representations for data with the purpose of taking into account natural structure that data usually presents, as opposed to working with flat representations.

The requirements of social network modeling and analysis provide further opportunities to extend the capabilities of graph languages. For example, many aspects of social network analysis rely on some probabilistic interpretation of graphs, so query languages need to be adapted and studied accordingly. Work in the area of expressive query languages for probabilisticdatabases has recently been initiated
.

References

[1] Dan Dong &Feili Liu," Survey On Graph Databases ", International Conference On Computer Networks And Communication Systems, 2012.

[2] Rabi Chandra Shah ,"Graph Database Model For Querying, Searching And Updating", International Conference On Software And Computer Applications, 2012.

[3] Renzo Angles And Claudio Gutierrez ,"Survey Of Graph Database Models", Computer Science Department, Universidad De Chile, 2005.

[4] Peter T. Wood, "Query Languages For Graph Databases", Department Of Computer Science And Information Systems, Birkbeck, University Of London, March 2012.

[5] C.C. Aggarwal And H. Wang (Eds.) ,"A Survey Of Algorithms For Keyword Search On Graph Data", Managing And Mining Graph Data, Advances In Database Systems, Springer Science + Business Media, LLC 2010.

[6] JiaweiHan ,“Data Mining :Concepts And Techniques”, Second Edition, University Of Illinois At Urbana-Champaign, 2006.
[7] Mark Graves, Ellen R. Bergeman, Charles B. Lawrence, ”Graph Database Systems For Genomics”, IEEE Engineering In Medicine And Biology Special Issue On Managing Data For The Human Genome Project.
[8] Renzo Angles, ”A Comparison of Current Graph Database Models”, Universidad de Talca, 2012.
[9] Adrian Silvescu, Doina Caragea, Anna Atramentov,” Graph Databases”.
[10] Graph Mining, Social Network Analysis, And Multi Relational Data Mining.
[11] Sherif Sakr, “G-SPARQL: A Hybrid Engine for Querying Large Attributed Graphs”, Copyright ACM , CIKM’12, October 29–November 2, 2012, Maui, HI, USA.
[12] Renzo Angles, Pablo Barcel , and Gonzalo Rıos,”A Practical Query Language for Graph DBs”, Department of Computer Science, Universidad de Talca,2013.
[13] HANNEMAN, R. A. 2001. Introduction to social network methods. Tech. Rep., Department of Sociology, Uni-versity of California, Riverside.
[14] Wellman, b., s alaff, j., d imitrova, d., garton, l., gulia, m., and haythornthwaite, c. 1996. computer networks as social networks :collaborativeork, telework, and virtual community. ann. rev. sociol. 22, 213–238.
[15] sheth, a., a leman -meza, b., a rpinar,i.b.,halaschek -wiener, c., r amakrishnan, c., b ertram, c., warke ,y., avant , d., a rpinar, f. s., a nyanwu, k., and kochut, k. 2005. semantic association identification and knowledge discovery for national security applications.J. Database Manag. 16, 1 (Jan-March), 33–53.
[16]DE S. P RICE, D. J. 1965. Networks of scientific papers. Science 149, 510–515. DENG,Y. ANDCHANG, S.-K. 1990. A G-Net model for knowledge representation and reasoning. IEEE Trans.Knowl. Data Eng. 2, 3 (Dec), 295–310.
[17] broder, a., kumar, r., m aghoul, f., r aghavan, p., r ajagopalan, s., s tata , r., tomkins, a., and wiener,j.2000. Graph structure in the Web. In Proceedings of the 9th International World Wide Web conferenceon Computer Networks: The International Journal of Computer and Telecommunications Networking.
[18] nejdl , w., siberski,w.,andsintek, m. 2003. Design issues and challenges for rdf- and schema-based
peer-to-peer systems. SIGMOD Record 32, 3, 41–46.
North-Holland Publishing Co., 309–320.
[19] shekhar, s., c oyle , m., goyal, b., l iu, d.-r., andsarkar, s. 1997. Data models in geographic information
systems. Commun. ACM 40, 4, 103–111.
[20] Graveshttp://www.xweave.com/people/mgraves/pubs/
[21] Graves. 1995b;Graves http://www.xweave.com/people/mgraves/pubs/ ; Hammer and Schneider 2004
[22] deville , y., g ilbert ,d., vanhelden,j., andwodak, s. J. 2003. An overview of data models for the analysisof biochemical pathways. In proceedings of the first international workshop on computational methods
in Systems Biology . Springer-Verlag, 174.
[23] JAGADISH ,H.V. ANDOLKEN , F. 2003. Data management for the biosciences: report of the NLM Workshop onData Management for Molecular and Cell Biology. Tech. Rep. LBNL-52767, National Library of Medicine.
[24] benk¨o, g., f lamm,c., and stadler, p. F. 2003. A graph-based toy model of chemistry. J. Chem. Inform.Computer science (jcisd) 43, 1 (jan), 1085–1093.