Journal of Database Management
Volume 30 • Issue 1 • January-March 2019

Designing Graph Databases With GRAPHED

Gustavo Cordeiro Galvão Van Erven, Ministry of Transparency and Office of the Comptroller General, Brasília, Brazil
Rommel Novaes Carvalho, Ministry of Transparency and Office of the Comptroller General, Brasília, Brazil
Waldeyr Mendes Cordeiro da Silva, Federal Institute of Goiás (IFG), Formosa, Brazil
Sergio Lifschitz, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
Harley Vera-Olivera, Universidade de Brasília, Brasília, Brazil
Maristela Holanda, University of Brasília (UnB), Brasília, Brazil

ABSTRACT

In recent years, graph database systems have become very popular and been deployed mainly in situations where the relationship between data is significant, such as in social networks. Although they do not require a particular schema design, a data model contributes to their consistency. Designing diagrams is an approach to satisfying this demand for a conceptual data model. While researchers and companies have been developing concepts and notations for graph database modeling, their notations focus on their specific implementations. In this article, the authors propose a diagram to address this lack of a generic and comprehensive notation for graph databases modeling, named GRAPHED (Graph Description Diagram for Graph Databases). The authors verified the effectiveness and compatibility of GRAPHED in two case studies: fraud identification, and a biological network model.

KEYWORDS

Data Modeling, Database Design, Diagrams, Graph Description

INTRODUCTION

Data modeling is an essential part of the development of quality information systems. There are sufficient reasons such as leverage, consistency and data quality why data modeling should deserve more attention than other components of a system (Bera and Poels, 2017; Sabegh and Recker, 2017). Data modeling of relational databases is well established and standardized. The academic and business community widely accept the entity-relationship model. However, new ways of storing data arose with the emergence of social networks, mobile devices, web technologies and the Internet of Things called NoSQL databases (Davoudian, Chen, and Liu 2018). These new technologies of storing data like document-oriented, key-value, column and graphs have properties that allow better performance when storing a large amount of data (Gil and Song, 2016; Batini, Rula, Scannapieco, and Viscusi, 2015).

DOI: 10.4018/JDM.2019010103

The importance of information about the relationships among objects is increasing in applications. Those solutions are used for problems in several areas such as business intelligence (Maté et al., 2015), chemical and biological networks (Silva, 2018), relationship structures, connected circuits in engineering, friends, and partners in social media or even the search for and identification of criminal cells are just some of the examples of the importance of relationships among objects (Srinivasa, 2011). Graphs often represent such data sets and their relationships.

Graphs can be defined generally as consisting of a set of vertices (or nodes) connected by edges defining some relationship between them (Deo, 1994). Graph algorithms provide solutions to several problems such as finding friends of friends in a social network, best paths between places in a city on a map, or people with similar interests in e-commerce (Batra and Tyagi, 2012). Thus, applications that address these kinds of problems can be enhanced using persistent graphs.

Using a graph as the primary structure in databases is not a new idea (Angles and Gutierrez, 2008). Moreover, the number of initiatives that use graph-based structures in databases has increased (Ruflin, Burkhart, and Rizzotti, 2011; Angles, Prat-Pérez, Dominguez-Sal, and Larriba-Pey, 2013). Several groups and companies have developed and improved database systems that can store a graph in its native form. Such initiatives include structures and query mechanisms that simplify the work of processing highly connected data (Kaur and Rani, 2013; Angles, 2012; Hecht and Jablonski, 2011; Vicknair et al., 2010).

However, although there have been advances in graph database technology, a notation to represent the conceptual graph model continues to present a challenge. There is no approach for data model widely accepted by the academic and business community in the graph databases. Several specific proposals of notations for diagram designs can be found in specialized literature (Angles and Gutierrez, 2008; Gyssens, Paredaens, and van Gucht, 1990; Amann and Scholl, 1992; Mainguenaud and Simatic, 1992; Kuper and Vardi, 1993; Hidders and Paredaens, 1994; Levene and Loizou, 1995; Levene and Poulovassilis, 1991; Hidders, 2003). Almost all of them bring schemas and instances to facilitate comprehension; however, they are focused on their particular implementations.

In this context, this paper proposes GRAPHED - Graph Description Diagram for Graph Databases (Van Erven, Holanda, and Carvalho, 2017), designed to be an independent notation for conceptual graph data modeling. GRAPHED formalizes representations of objects contained in graphs to provide a diagram representing the data model for graph databases. Within its basic elements, it supports the idea of a label for the domain of the relationship in the schema, and values in instances, besides notation for weight and types. It also covers hyperedges and hypernodes by respectively generalizing the simple edge and using a restructured concept from the hypernode.

Related Works

Several studies have focused on graph data models (Angles and Gutierrez, 2008) and the research on this topic waned considerably with the work on NoSQL databases (Angles, 2012). Certain papers proposed an organization and notation for their data models. Prime examples are GOOD (Gyssens, Paredaens, and van Gucht, 1990), Gram (Amann and Scholl, 1992), Simatic-XT (Mainguenaud and Simatic, 1992), LDM (Kuper and Vardi, 1993), GOAL (Hidders and Paredaens, 1994), Hypernode (Levene and Loizou, 1995), GROOVY (Levene and Poulovassilis, 1991) and GDM (Hidders, 2003).

The LDM (Kuper and Vardi, 1993) uses a labeled directed multigraph (graphs that can have several edges between the same pair of vertices) to describe the schema. All the internal nodes have a symbol that can represent iterations of domain sets as the names of different people, which can define a cartesian product of the set of names. The nodes are the basic type and represent attributes. However, the model lacks a data type description for the attributes.

Gram (Amann and Scholl, 1992) is a simple model for data organized as graphs, which uses the entity's name as the node label. The edges are arrows that link the entities, and their labels indicate the domain of the relationship in the schema. The instances can use the values of the domain for vertices (e.g., Paris for City), as well as edges.

Related Content

Elitist and Ensemble Strategies for Cascade Generalization
Huimin Zhao, Atish P. Sinhaand Sudha Ram (2006). *Journal of Database Management (pp. 92-107).*
www.irma-international.org/article/elitist-ensemble-strategies-cascade-generalization/3359

Research Review: A Systematic Literature Review on the Quality of UML Models
Marcela Genero, Ana M. Fernández-Saez, H. James Nelson, Geert Poelsand Mario Piattini (2011). *Journal of Database Management (pp. 46-70).*
www.irma-international.org/article/research-review-systematic-literature-review/55133

Self-Tuning Database Management Systems
Camilo Porto Nunes, Cláudio de Souza Baptistaand Marcus Costa Sampaio (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends (pp. 753-761).*
www.irma-international.org/chapter/self-tuning-database-management-systems/20761

Mutual Recognition Mechanism Based on DVCS Oracle in the Blockchain Platform: DVCS Oracle in the Global Supply Chain
Vladimir Nikolaevich Kustovand Ekaterina Sergeevna Selanteva (2022). *Utilizing Blockchain Technologies in Manufacturing and Logistics Management (pp. 81-103).*
www.irma-international.org/chapter/mutual-recognition-mechanism-based-on-dvcs-oracle-in-the-blockchain-platform/297159

Using Ontology and Rule-Based Reasoning for Conceptual Data Models Synonyms Detection: A Case Study
Ljubica Kaziand Zoltan Kazi (2019). *Journal of Database Management (pp. 1-21).*
www.irma-international.org/article/using-ontology-and-rule-based-reasoning-for-conceptual-data-models-synonyms-detection/230292