

Named Sets as an Efficient Tool for Modeling Data Relationships in Database Models [†]

Mark Burgin ^{1,*} and H. Paul Zellweger ²¹ Department of Computer Science, University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA² ArborWay Labs, Rochester, MN 55901, USA; pz@arborwaylabs.com

* Correspondence: markburg@cs.ucla.edu

[†] Presented at the Conference on Theoretical and Foundational Problems in Information Studies, IS4SI Summit 2021, Online, 12–19 September 2021.

Abstract: An important problem for databases is the unification of utilized data structures and amplification of operation tools. Here, after a brief overview and analysis of database models, we demonstrate that all considered data can be reduced to systems of named sets allowing representation of the described database models as special cases of the named-set database model, which provides efficient operations for data mining, information extraction, and database management.

Keywords: database; data; information; data structure; database model; graph; system; network; relation

1. Introduction

Data are representations and containers (carriers) of information. Data and their relationships form data structures. That is why modeling data relationships in the context of data and knowledge structures is critically important for organization and optimization of information processes in databases and beyond.

Data have different structures and both data processing in general and data mining in particular depends on these structures. For instance, the well-known computer scientist and mathematician Yuri Gurevich concluded his lecture [1] on the advancement of theoretical computer science with the statement that to be useful for database technology, computational models have to work with structures and not with strings of symbols. The most popular data structures include Boolean values, characters, integers, fixed-precision number values, floating-point number values, arrays, records, lists, streams, sets, multisets, stacks, queues, and graphs, just to mention the most important of them. Here, in addition to these conventional data structures, we consider named sets and chains of named sets as the fundamental data structures for modeling data relationships in database models.

2. Database Models

As Angles and Gutierrez write, from a database point of view, the conceptual tools that make up a database model should at least address data structuring, description, maintenance, and a way to retrieve or query the data [2]. These principles imply that a database model consists of three components: a system of utilized data structure types with their logical and operational organization; a system of operators and inference rules; and a system of integrity rules [3]. The logical structure of a database includes the relationships and constraints that determine how data can be stored and accessed. As a rule, database models mainly pay attention to utilized data structures, which are represented by a database model. Let us consider the main database models used for storing and preserving information.

The *hierarchical database model* is the oldest being developed by IBM for information management system (IMS) [4]. In it, data are organized in the tree structure.

The *network database model* represents data with records and sets. Records contain fields, which may be organized hierarchically, while sets define one-to-many relationships



Citation: Burgin, M.; Zellweger, H.P. Named Sets as an Efficient Tool for Modeling Data Relationships in Database Models. *Proceedings* **2022**, *81*, 51. <https://doi.org/10.3390/proceedings2022081051>

Academic Editor: Rao Mikkilineni

Published: 16 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

between records. This model is an expansion of the hierarchical model allowing many-to-many relationships in a tree-like structure.

The *flat(or table) database model* consists of a single, two-dimensional array of data elements, where all members of a given column are assumed to be similar values and all members of a row are assumed to be related to one another having the same type. The flat model predates the relational model.

The *relational database model* was introduced by Codd [5] and highlights the concept of abstraction levels by introducing the idea of separation between physical and logical levels. It is based on the notions of sets and relations. Due to its ease of use, it gained wide popularity among business applications.

The *multivalued database model* is an extension of the relational model. In it, a field/attribute can have several values at the same time.

The *semantic database model* represents objects and their relations in a natural and clear way, providing users with tools to correctly reflect the desired domain semantics. The *entity-relationship model* is an example of semantic database models [6].

The *resource space database model* (RSM) is a non-relational data model based on multi-dimensional classification [7].

The *object-oriented database model* is based on the object-oriented paradigm representing data as a collection of objects that are organized into classes and assigned complex values [7].

The *graph database model* represents objects and their relations in the form of a graph overcoming the limitations imposed by traditional database models with respect to capturing the inherent graph structure of data appearing in applications such as hypertext or geographic information systems, where the interconnectivity of data is an important aspect [2].

The *semistructured database model* exemplifies data with a flexible structure, for example, documents or Web pages. Semistructured data are neither raw nor strictly typed, as in the conventional database systems [8].

The *XML (eXtensible Markup Language) database model* focuses on information with tree-like structure [9].

The *named-set database model* represents information in the form of systems of named sets such as named set chains [10].

3. Named Sets in Database Modeling and Data Representation

Here we show that all of these database models can be treated as special cases of the named-set database model since all utilized data structures are either named sets or systems of named sets.

So, the question is as follows: Why are named sets really essential, and what is so specific about them?

First, it is proved that any mathematical structure is a named set or is built of named sets and thus, the named set is the most fundamental structure in all mathematics [11]. For instance, functions, relations, variables, graphs, multigraphs, and morphisms (arrows) in categories are special cases of named sets. Ordinary sets are also specific named sets, namely they are singlenamed sets since all elements in a set with the name, say Q , have the common name “an element of the set Q ” [11].

Second, we see that named sets are vital for representation of data, knowledge, and information as well as all cognitive processes and communication [12]. Taking any book on databases, we see many examples of named sets (cf., for example, [13]).

Third, it is proved that the named set (also called fundamental triad) is the most basic structure in nature [12]. As a consequence, named sets have become ubiquitous in modeling natural systems.

Let us consider the basic definition.

Definition 1. (a) A basic named set, also called a basic fundamental triad, is a triad $\mathbf{X} = (X, f, N)$ with the following visual (graphic) representation:

$$\begin{array}{c} f \\ X \rightarrow N \end{array}$$

(b) A bidirectional named set, also called a bidirectional fundamental triad, is a triad $\mathbf{X} = (X, f, Z)$ with the following visual (graphic) representation:

$$\begin{array}{c} f \\ X \leftrightarrow N \end{array}$$

In this triad $\mathbf{X} = (X, f, N)$, the components X and N are two objects and f (in case a) is a correspondence (e.g., a binary relation) from X to N and (in case b) is a correspondence (e.g., a binary relation) between X and Z , which goes in two directions. With respect to $\mathbf{X}$, the object X is called the *support* of $\mathbf{X}$ and denoted $S(\mathbf{X})$, the object N is called the *component of names* (reflector) or *set of names* of $\mathbf{X}$ and denoted $N(\mathbf{X})$, and the object f is called the *namingcorrespondence* (reflection) of $\mathbf{X}$ and denoted $r(\mathbf{X})$. It means that $\mathbf{X} = (S(\mathbf{X}), r(\mathbf{X}), N(\mathbf{X}))$. Note that in $\mathbf{X}$, components X and N are not automatically sets, while f is not necessarily a mapping or a function even if X and N are sets. For instance, X and N are sets of words and f is an algorithm.

The standard example is a basic named set (basic fundamental triad), in which X consists of people, N consists of their names, and f is the correspondence between people and their names.

Let us analyze the considered database models in the context of named sets.

Hierarchical data are organized as tree structures, which are chains of named sets starting with the root of the tree and ending with its leaves [14]. Consequently, the hierarchical database model is a special case of the named-set database model.

Any network in general and a network of records, in particular, have the structure of a graph. A graph consists of vertices (nodes) and edges connecting some vertices. If V is the set of all vertices and E is the set of all edges in graph G , then this graph is a named set (V, E, V) . Consequently, the network database model is a special case of the named-set database model. Note that as records contain fields and fields are named sets with values as their names, any network of records is a nested named set [15].

Any two-dimensional arrays in general and two-dimensional arrays of data elements, in particular, are nested named sets [15]. Consequently, the flat database model is a special case of the named-set database model.

Relations are special cases of set-theoretical named sets [11]. Consequently, the relational and multivalued database models are special cases of the named-set database model.

Objects (entities) with relations form a named set [11]. Consequently, the semantic database model is a special case of the named-set database model.

Any classification is a set-theoretical named set [11]. Consequently, the resource space database model is a special case of the named-set database model.

In the object-oriented approach, data are formed as objects and each object has a name, set of attributes, and behaviors. Thus, the support of an object consists of its name which is connected to its attributes and behaviors. This is the structure of a named set. In addition, any set of objects is the named set, which is the union of the named sets of individual objects [11]. Consequently, the object-oriented database model is a special case of the named-set database model.

As it was demonstrated that any graph is a named set, the graph database model is a special case of the named-set database model.

Finally, as any structure is built from named sets, the semistructured and XML database models are special cases of the named-set database model. For XML data, this was demonstrated in [16].

4. Conclusions

An important peculiarity of utilization of named sets in databases is that algorithms in general, and software systems in particular, for operation with data are also specific named sets and systems of named sets. Namely, they are algorithmic named sets and their systems (i.e., such named sets in which the relation f is an algorithm or a program) [11,17].

An important advantage of the named-set database model is not only structural unification but also operational affluence. Indeed, manipulation with data demands utilization of various operations and, in the case of using named sets for data representation, a variety of operations such as mappings of different kinds, union, intersection, difference, renaming, naming, interpreting, and reinterpreting, and their properties are provided by the theory of named sets [11].

Operating with data in the named-set database model involves structural recursion capturing the system's repeating patterns. As a result, an important direction for future research is exploration of structural recursion in the context of named sets and its application to the problems of data search, as well as to database development and management.

Nesting is an important phenomenon in many areas in general and the database technology in particular. Nested structures are efficiently modeled by nested named sets [15]. Thus, one more interesting direction for future research is to study nested named sets and operations in their domain with the goal of their employment in database operation and control.

Author Contributions: M.B. and H.P.Z. contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gurevich, Y. The Church-Turing Thesis: Story and Recent Progress. 2009. Available online: <https://youtu.be/7XfA5EhH7Bc> (accessed on 21 January 2022).
2. Angles, R.; Gutierrez, C. Survey of Graph Database Models, ACM Computing Surveys. *ACM Comput. Surv.* **2008**, *40*, 1. [CrossRef]
3. Codd, E.F. Data Models in Database Management. In Proceedings of the 1980 Workshop on Data Abstraction, Databases, and Conceptual Modeling, Pingree Park, CO, USA, 23–26 June 1980; ACM Press: New York, NY, USA, 1980; pp. 112–114.
4. Tsichritzis, D.C.; Lochovsky, F.H. Hierarchical data-base management: A survey. *ACM Comput. Surv.* **1976**, *8*, 105–123. [CrossRef]
5. Codd, E.F. A relational model of data for large shared data banks. *Commun. ACM* **1970**, *13*, 377–387. [CrossRef]
6. Zhuge, H. The Web Resource Space Model. In *Web Information Systems Engineering and Internet Technologies Book Series*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4, p. 2.
7. Kim, W. Object-oriented databases: Definition and research directions. *IEEE Trans. Knowl. Data Eng.* **1990**, *2*, 327–341. [CrossRef]
8. Buneman, P. Semistructured Data. In Proceedings of the 16th Symposium on Principles of Database Systems (PODS), Tucson, AZ, USA, 12–14 May 1997; ACM Press: New York, NY, USA, 1997; pp. 117–121.
9. Bray, T.; Paoli, J.; Sperberg-McQueen, C.M. Extensible Markup Language (XML) 1.0, W3C Recommendation 10. 1998. Available online: <http://www.w3.org/TR/1998/REC-xml-19980210> (accessed on 11 November 2021).
10. Burgin, M. Structural Organization of Temporal Databases. In Proceedings of the International Conference on Software Engineering and Data Engineering (SEDE-2008), Las Vegas, NV, USA, 14–17 July 2008; ISCA: Los Angeles, CA, USA, 2008; pp. 68–73.
11. Burgin, M. *Theory of Named Sets*; Mathematics Research Developments; Nova Science: New York, NY, USA, 2011.
12. Burgin, M. *Theory of Knowledge*; World Scientific: New York, NY, USA; London, UK; Singapore, 2016.
13. Date, C.J. *An Introduction to Database Systems*; Addison Wesley: Boston, MA, USA; San Francisco, CA, USA; New York, NY, USA, 2004.
14. Burgin, M.; Zellweger, P. A Unified Approach to Data Representation. In Proceedings of the 2005 International Conference on Foundations of Computer Science, Las Vegas, NV, USA, 27–30 June 2005; CSREA Press: Las Vegas, NV, USA, 2005; pp. 3–9.
15. Burgin, M.; Zellweger, P. Nested Named Sets in Information Retrieval. In *Advances in Data Science and Information Engineering*; Springer: New York, NY, USA, 2021; pp. 451–467.

16. Nocedal, A.S.; Gerrikagoitia Arrien, J.K.; Burgin, M. A mathematical model for managing XML data. *Int. J. Metadata Semant. Ontol. (IJMSO)* **2011**, *6*, 56–73. [[CrossRef](#)]
17. Zellweger, H.P. Tree Visualizations in Structured Data Recursively Defined by the Aleph Data Relation. In Proceedings of the 20th International Conference Information Visualisation, Lisbon, Portugal, 19–22 July 2016; Volume 4, pp. 21–26.