

archive.today
webpage capture

Saved from <http://www.drdobbs.com/database/sets-data-models-and-d>
no other s

Redirected from <http://drdobbs.com/blogs/database/228700616>
no other s

All snapshots from host drdobbs.com
from host www.drdobbs.com

- Linked from en.wikipedia.org » [Database](#)
- en.wikipedia.org » [MICRO Information Management System](#)
- en.wikipedia.org » [MICRO Relational Database Management System](#)
- en.wikipedia.org » [User:Ibon.coria/sandbox](#)
- en.wikipedia.org » [User:Rebone System/sandbox](#)
- es.wikipedia.org » [Database - Rebone System](#)

Webpage

Screenshot

Welcome Guest | [Log In](#) | [Register](#) | [Benefits](#)

Dr.Dobb's

THE WORLD OF SOFTWARE DEVELOPMENT

Search: Site Source Code

Home

Articles

News

Blogs

Source Code

Dobb's on DVD

Dobb's TV

W

Cloud

Mobile

Parallel

.NET

JVM Languages

C/C++

Tools

Jolt Awards



Ken North

Dr. Dobb's Bloggers

```
Response response = invokeCall(...)
// Extract SessionID from SOAP header
Header header = response.getHeaderEntry();
Vector headerEntries = header.getHeaderEntries();
HeaderElement = headerElement();
```

Tweet

Like

Share

[Permalink](#)

Sets, Data Models and Data Independence

March 10, 2010

[Post a Comment](#)

David Childs authored pioneering research about implementing set operations that enabled a programmer to approach the data storage and retrieval problem from a logical model, rather than a physical model of the data.

A thorough examination of databases and data management will include many flavors of data and information models, including conceptual, logical, physical, mathematical, and application models.

Database technology is constantly evolving, with new approaches and refinements to existing platforms. The choice of a data access solution depends in part on the underlying data model; whether a data store operates with sets, graphs or other types of data. Data management technology has undergone evolutionary development since the 1950s. The modern database management system (DBMS) represents mature, but not static, technology. Besides the emergence of new approaches to data persistence, there are continued refinements to mature DBMS platforms.

Today's data stores implement a variety of data models, including graphs, sets, collections, arrays, cubes and other variants, including the hierarchical data model, relational model, network data model (CODASYL), trees, nested sets, adjacency lists and object stores. The index sequential access method (ISAM), key-value data stores and record management systems have also been implemented in various forms for decades.

The concept of a data store that supported set operations such as union, intersection, domain and range emerged in the 1960s, based of course on Georg Cantor's set theory published in the 19th century. In 1968, D.L. Childs, then at the University of Michigan, wrote seminal papers about set-theoretic data structures that provided data independence, meaning an application did not have to know the physical structure of the data. During that era of first-generation databases (see [CODASYL 1968 Survey of Data Base Systems](#)), data access typically required the use of pointers and descriptions of physical data structures.

Childs authored pioneering research about implementing set operations that enabled a programmer to approach the data storage and retrieval problem from a logical view, rather than a physical view of the data. His March 1968 paper, "[Description of A Set-Theoretic Data Structure](#)", explained that programmers can query data using set-theoretic expressions instead of navigating through fixed structures.

"A set-theoretic data structure (STDS) is virtually a 'floating' or pointer-free structure allowing quicker access, less storage, and greater flexibility than fixed or rigid structures that rely heavily on internal pointers or hash-coding, such as 'associative or relational structures,' 'list structures,' 'ring structures,' etc. An STDS relies on set-theoretic operations to do the work usually allocated to internal pointers. A question in an STDS will be a set-theoretic expression. Each set in an STDS is completely independent of every other set, allowing modification of any set without perturbation of the rest of the structure; while fixed structures resist creation, destruction, or changes in data. An STDS is essentially a meta-structure, allowing a question to 'dictate' the structure or data-flow. A question establishes which sets are to be accessed and which operations are to be performed within and between these sets. In an STDS there are as many 'structures' as there are combinations of set-theoretic operations; and the addition, deletion, or change of data has no effect on set-theoretic operations, hence no effect on the 'dictated structures.' Thus in a floating structure like an STDS the question directs the structure, instead of being subservient to it."

In August 1968, Childs published "[Feasibility of a Set-Theoretic Data Structure. A General Structure Based on a Reconstituted Set-Theoretic Definition for Relations](#)".

"This paper is motivated by an assumption that many problems dealing with arbitrarily related data can be expedited on a digital computer by a storage structure which allows rapid execution of operations within and between sets of datum names. In order for such a structure to be feasible, two problems must be considered: (1) the structure should be general enough that the sets involved may be unrestricted, thus allowing sets of sets of sets...; sets of ordered pairs, ordered triples...; sets of variable length n-tuples, n-tuples of arbitrary sets; etc.; (2) the set-operations should be general in nature, allowing any of the usual set theory operations between sets as described above, with the assurance that these operations will be executed rapidly. A sufficient

condition for the latter is the existence of a well-ordering relation on the union of the participating sets. These problems are resolved in this paper with the introduction of the concept of a 'complex' which has an additional feature of allowing a natural extension of properties of binary relations to properties of general relations."

The Federal government, including the Defense Advanced Research Projects Agency (DARPA), frequently funded computer science research and development during that era. One such effort was the University of Michigan's Research in the Conversational Use of Computers (CONCOMP) project for which Childs did his work on set-theoretic data structures. During that era, DARPA also funded development of packet-switched network technology and the ARPAnet, the forerunner of today's Internet.

Childs' CONCOMP papers were available only to 'qualified requesters' although Childs presented the August 1968 paper at that year's Congress of the International Federation for Information Processing (IFIP). Those 1968 papers did not receive the broad dissemination of research papers published today via the Internet. Nonetheless Dr. Edgar F. Codd, who'd gotten his PhD at the University of Michigan, cited Childs' paper on set-theoretic data structures in his June 1970 paper about the relational model.

Many persons who had not discovered Childs' papers erroneously believed the foundation of data independence and set-theoretic operations over data had been laid by Codd.

Following Codd's 1970 paper on the relational model, other database researchers published papers that discussed the concept of data independence. In 1971, Chris Date and Paul Hopewell authored "Storage Structures and Physical Data Independence" for the ACM Workshop on Data Definition, Access and Control. The authors wrote about data independence being integral to the relational model:

"Such data independence was explicitly called out as one of the major objectives of the relational model by Ted Codd in 1970 in his famous paper "A Relational Model of Data for Large Shared Data Banks" (Communications of the ACM 13, No. 6, June 1970)."

Dr. Michael Stonebreaker's 1974 paper, "A Functional View of Data Independence", cited Codd's 1970 paper and Date's 1971 paper, but not Childs' papers in 1968. Similarly I've found other publications that credit the notion of data independence or physical data independence to Codd and Date, without referring to Childs' papers.

During the 1990s, the advent of object databases and object-oriented programming frequently surfaced topics related to the relational model and data independence in articles, conference presentations and online discussions. Prominent defenders of relational fidelity included Chris Date, David McGoveran, Hugh Darwen and Fabian Pascal. In debates about the relational model (then and now), data independence and relational algebra are often cited as key factors that differentiate Codd's relational model from less formal approaches. Relational algebra includes the group of set-theoretic operations that provide mathematical underpinnings to the relational model.

With the emergence of the Internet, Childs' papers are now widely-available to researchers. We now know Childs' pioneered concepts of data independence and set-theoretic operations over data. The works of Georg Cantor and D.L. Childs provided groundwork that enabled Dr. Edgar F. Codd to develop the relational model. Several years ago I had an e-mail exchange about this with Don Chamberlin, the co-inventor of SQL who worked with the late Dr. Codd at IBM. He acknowledged Childs' contribution:

i<>"Thanks for the reminder of David Childs' work. As you have observed, modern relational databases owe a lot to Childs and he deserves recognition for this early and pioneering work."

Since his pioneering work on set-theoretic data structures, David Childs has published papers about extended set processing, XML processing and other subjects that will be a topic for the future.

Related

Part 2:

[Part 3: "Information Density, Mathematical Identity, Set Stores and Big Data"](#) David Childs authored pioneering research about implementing set operations that enabled a programmer to approach the data storage and retrieval problem from a logical model, rather than a physical model of the data.

Related Reading

News

Commentary

[Write An iPhone/iPad App In BASIC](#)

[UrtheCast "Live" Earth Video Camera Opens API](#)

[New Software Library For Vision Intelligence In Mobile](#)

[Developers Get "Locked Down" HTML5 Specification From W3C](#)

[More News»](#)

Most Popular

On the Web

[Software Development in the Obama Campaign](#)

[Understanding Core Data on iOS](#)

[Building Scalable Web Architecture and Distributed Systems](#)

[Jolt Awards: The Best Books](#)

[More Popular»](#)

Slideshow

Video

[Developer Reading List](#)

[Developer's Reading List](#)

[2012 Jolt Award Winners for Third-Party Libraries](#)

[Dr. Dobb's Excellence in Programming Awards: Through the Years](#)

[More Slideshows»](#)

More Insights

White Papers

[BYOD Risk and Rewards](#)

[Forrester: The Total Economic Impact Of VMware View](#)

[More >>](#)

Reports

[Strategy: Rebooting DLP](#)

[Research: The Next Wave In APM](#)

[More >>](#)

Webcasts

[How SMBs Can Avoid the Storage Gap Trap](#)

[Agile Development: Three Pillars of Success](#)

[More >>](#)