Database Experts' Series

Database Design

The Semantic Modeling Approach

NAPHTALI RISHE

Database Design

The Semantic Modeling Approach

Naphtali Rishe
School of Computer Science
Florida International University
Miami, Florida

McGraw-Hill, Inc.
New York St. Louis San Francisco Auckland Bogotá
Caracas Lisbon London Madrid Mexico Milan
Montreal New Delhi Paris San Juan São Paulo
Singapore Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Rishe, Naphtali.
Database design : the semantic modeling approach / Rishe, Naphtali.
p. cm. — (The Database experts' series)
Includes bibliographical references and index.
ISBN 0-07-052955-8
1. Data base design. 2. Relational data bases. I. Title.
II. Series
QA76.9.D26R57 1992
005.75'6—dc20 91-31803
CIP

1 2 3 4 5 6 7 8 9 0 DOC/DOC 9 7 6 5 4 3 2 1

ISBN 0-07-052955-8

The sponsoring editor for this book was Neil Levine, the editing supervisor was David E. Fogarty, and the production supervisor was Donald Schmidt.

Printed and bound by R. R. Donnelley & Sons Company.

Database Design

The Semantic Modeling Approach

To Noa Rishe

Preface

This book is intended for present and future designers of database applications, software engineers, systems analysts, and programmers. It focuses on the fundamental knowledge needed by designers of database applications and on methodologies of structured design. With the exception of an optional chapter on implementational aspects, the book does not go into the system's internals, which are irrelevant to the application designers. The current database technology isolates its users from its internals. Therefore, in-depth understanding of internals will be important only to that small category of system designers who develop new database management systems. In contrast, most software engineers will develop or maintain database applications at one time or another.

The semantic approach

This book presents the field of database design from the perspective of semantic modeling. The focus on semantic modeling serves three purposes:

- The semantic and object-oriented data models are now occupying a significant part of the frontier of the database technology and are expected to become predominant in tomorrow's databases, replacing the current relational database technology. (Although somewhat different in their approach to database modeling, the semantic and object-oriented models are quite similar. Their differences are described in Chapter 10.)
- The semantic modeling approach is used in this book to unify the ideas and terminology of the various database models. Instead of separately introducing the relational database model with its concepts, terminology, and languages, then the network database model with its concepts and terminology, and so on, this book

unifies all of the database models into one framework. Most of the concepts and languages are presented in terms of a unifying semantic model. The other models are technically treated as subsets of the semantic model; therefore, the concepts and languages automatically apply to them.

- Most importantly, semantic modeling is presented as a tool of database design in the relational and other database models. Thus, the top-down relational database methodology presented in this book proceeds as follows. First, the user's application is analyzed and specified semantically. This produces a concise, flexible, user-oriented specification of the application's database, unconstrained by computer-oriented concerns. In the second stage, this specification is converted into a relational database schema, with its integrity constraints, data manipulation programs, etc. The semantic description remains a high-level documentation of the database.

Contents

Chapter 1 introduces the fundamental aspects of databases. These aspects are described in terms of a semantic database model, the Semantic Binary Model (SBM). (In later chapters, other database models, such as the relational, network, and hierarchical models, are defined technically as subsets of the semantic model of Chapter 1.) This chapter defines and discusses the concepts of a database, a database management system (DBMS), a database schema, modeling real-world information, categorization of real-world objects, relations between objects, graphic representation of database schemas, integrity constraints, quality of database schemas, sub-schemas, userviews, database languages, services of DBMS, and multimedia databases.

Chapter 2 presents two fundamental database languages, from which most database languages can be derived with some adjustment of syntax. The first language is a fourth-generation data manipulation language. It is shown as a structured extension of Pascal. The second language is a nonprocedural language called Database Predicate Calculus. Chapter 2 defines these languages in terms of the Semantic Binary Model. Later chapters show the use of these languages in other database models.

Chapter 3 defines the Relational Data Model and presents a top-down methodology for the design of relational databases.

Chapter 4 describes relational database languages. Sections 1 and 2 show examples of how the languages of Chapter 2 (the fourth-generation and the logic-based languages) apply to the relational databases. A case study in Section 1 discusses the principles of writing a transaction-processing program for an application. The optional Section 3 defines the Relational Algebra. Section 4 describes SQL, a popular commercial language related to the logic-based language of Section 2. The expressive power and the equivalence of relational languages are discussed in Section 5.

Chapter 5 begins with a case study of the design of an actual database application. Section 2 summarizes the flow of database design. Section 3 compares the methodology of this book to the older methodology of normalization.

Chapter 6 defines the Network (CODASYL) data model and adapts the top-down database design methodology to network databases. Section 3 of this chapter discusses network database languages: application of the generic fourth-generation and logic-based languages and a special navigational language for the Network Model.

Chapter 7 defines the hierarchical data model and adapts the top-down database design methodology to hierarchical databases. Section 3 of this chapter discusses hierarchical database languages: application of the generic fourth-generation and logic-based languages.

Chapter 8 compares the semantic, relational, network, and hierarchical data models with respect to application programming efforts, data independence, and other factors.

Chapter 9 discusses aspects of DBMS implementation. Section 1 describes an efficient algorithm for the implementation of semantic databases. Section 2 addresses questions of transaction handling, including the enforcement of integrity constraints, backup and recovery, and concurrency control. Section 3 addresses issues of data definition languages and data dictionaries.

Chapter 10 addresses object-oriented databases. This chapter discusses the similarities and the minor difference between the semantic and object-oriented databases and augments the Semantic Binary Model with object-oriented features related to modeling database behavior.

Chapter 11 discusses several fifth-generation languages. Sections 1 through 3 address issues of expressive power of logic-based database languages and discuss Prolog-like languages and a logic-based language which attains computational completeness. Section 4 discusses user-friendly interfaces, using the Query-By-Example language as an example.

Chapter 12 is the bibliography. Section 1 gives annotated references to papers on issues of semantic modeling addressed in this book. Section 2 is a listing of recent books on databases.

Chapter 13 contains solutions of problems.

Use of this book as a glossary and a reference handbook

The reader who wants to obtain the definition of a database term can look it up in the index, which provides a pointer to the page on which the term is defined. On that page the user will find the term set in bold face (for easy locating), normally followed by an example. When a term has different uses or aspects in several database models, the index contains several references marked according to their use.

Prerequisites

The reader is expected to be familiar with the fundamentals of the art of programming. Knowledge of structured programming is desirable, preferably in Pascal or a similar language. No knowledge of file organization or data structures is required, except for the optional Chapter 9.

Structure of the book

The book is composed primarily of explanations of concepts and examples. The examples are offset and boxed so that the experienced reader or browser can easily skip them. The examples constitute a continuous case study of an application, for which databases are designed in different models, application programs are written in different languages, etc.

Most sections are followed by problems. Many of the problems are solved in the last chapter of this book. Page-number pointers direct the reader from the problems to their solutions. If after reading a chapter the reader fails to solve a problem marked "Advanced" or "Optional," it does not mean a lack of understanding of the chapter but probably means that the reader has a lack of mathematical knowledge or experience, which is not prerequisite to the reading of this book.

The sections marked with an asterisk (*) contain optional advanced material and may be skipped. Optional advanced material within the regular sections is given in the footnotes.

Acknowledgments

I am grateful to the Florida International University for providing facilities and an environment conducive to writing this book, to the Florida High Technology and Industry Council for their support, and to the following persons for their valuable comments and advice: Michael Alexopoulos, Vanessa Allen, David Barton, David Buker, David Fogarty, Jim Fegen, Randy Goetz, Scott Graham, Ehud Gudes, Carlos Ibarra, Alok Jain, Robert Johnson, Ranjana Kizakkevariath, Neil Levine, Steve Luis, Thomas McElwee, Jai Mokherje, Shamkant Navathe, Jainendra Navlakha, N. Prabhakaran, Timothy Riley, Dan Simovici, Martin Solomon, Wei Sun, Doron Tal, N. Vijaykumar, and Nancy Young.

Naphtali Rishe

School of Computer Science
Florida International University
University Park, Miami, FL 33199, U.S.A.
Telephone: (305) 348-2025, 348-2744
FAX: (305)-348-3549; E-mail: rishen@servax.bitnet, rishen@fiu.edu

Contents

CHAPTER 1

SEMANTIC INTRODUCTION TO DATABASES

This chapter defines fundamental concepts of databases. These concepts are described here in terms of the Semantic Binary Model (SBM) of data. A data model is a convention for the specification of the logical structure of real-world information. The cornerstone of the contemporary theory and technology of databases was the development of the Relational Data Model. The recent development of the new generation of data models — the semantic models — offers a simple, natural, implementation-independent, flexible, and nonredundant specification of information. The word *semantic* means that this convention closely captures the meaning of user's information and provides a concise, high-level description of that information.

SBM is one of several existing semantic models. The various semantic models are roughly equivalent and have common principles, even though they somewhat differ in terminology and in the tools they use. SBM is simpler than most other semantic data models: it has a small set of sufficient tools by which all of the semantic descriptors of the other models can be constructed. After mastering SBM, a systems analyst may wish to explore more complex semantic models.

This chapter defines and discusses the concepts of a database, a database management system (DBMS), a database schema, modeling real-world information, categorization of real-world objects, relations between objects, graphic representation of database schemas, integrity constraints, quality of database schemas, subschemas, userviews, database languages, services of DBMS, and multimedia databases.

1.1. Databases, DBMS, Data Models

General-purpose software system — a software system that can serve a variety of needs of numerous dissimilar enterprises.

> *Example:*
>
> A compiler for a programming language.

Application — a software system serving the special needs of an enterprise or a group of similar enterprises.

> *Example:*
>
> The registration of students in a university.

Application's real world — all the information owned by and subject to computerization in an enterprise *or* all such information which is relevant to a self-contained application within the enterprise.

> *Example:*
>
> The examples of this text constitute a case study. Its application world is the educational activities of a university. The information contains:
>
> - A list of the university's departments (including all the full and short names of each department)
> - Personal data of all the students and their major and minor departments
> - Personal data of all the instructors and their work information (including all the departments in which the instructor works and all the courses which the instructor teaches)
> - The list of courses given in the university catalog
> - The history of courses offered by instructors
> - The history of student enrollment in courses and the final grades received

Database — an updatable storage of information of an application's world *and* managing software that conceals from the user the physical aspects of information storage and information representation. The information stored in a database is accessible at a *logical* level without involving the physical concepts of implementation.

> *Example:*
>
> Neither a user nor a user program will try to seek the names of computer science instructors in track 13 of cylinder 5 of a disk or in "logical" record 225 of file XU17.NAMES.VERSION.12.84. Instead, the user will communicate with the database using some *logical* structure of the application's information.

Normally, a database should cover *all* the information of one application; there should not be two databases for one application.

Database management system, DBMS — a general-purpose software system which can manage databases for a very large class of the possible application worlds.

> *Example:*
>
> A DBMS is able to manage our university database and also completely different databases: an Internal Revenue Service database, an FBI *WANTED* database, a UN database on world geographical data, an Amtrak schedule, etc.

Instantaneous database — all the information represented in a database at a given instant. This includes the historic information which is still kept at that time.

The actual information stored in the database changes from day to day. Most changes are additions of information to the database.

> *Example:*
>
> A new student, a new instructor, new events of course offerings.

Fewer changes are deletions of information.

> *Example:*
>
> Historic information past the archival period;
> a course offering which was canceled before it was given.

Some changes are replacements: updates; correction of wrongly recorded information.

> *Example:*
>
> Update of the address of a student;
>
> correction of the student's birth year (previously wrongly recorded).