doi: 10.3325/cmj.2009.50.198

Colin Ritchie

Database Principles and Design

3rd edition. London: Cengage Learning Business Press; 2008. 388 pages; ISBN 78-1-84480-540-2; price: US $84.95

Goran Kardum
gkardum@ffst.hr

Field: Information technologies.

Audience: Students of computing or information technology, medical students.

Purpose: This book presents a broad overview of database principles and design.

Content: The book is divided into 11 chapters and 2 hands-on chapters: Introduction and Background; The Relational Data Model; Conceptual Database Design; Physical Database Design; Interfacing with the Database; Transactions, Integrity and Security; Network and Distributed Systems; Post-relational Databases; Web Databases; and XML and Databases.

The hands-on chapters serve as a support for students' laboratory work. The first hands-on section is an extensive tutorial on learning Structural Query Language (SQL), the basic programming language for databases. The second hands-on section gives a simple instruction on how to create a basic database with tables, forms, queries, and reports in Microsoft Access.

The main purpose of the first 4 chapters is to describe the theoretical background of the essential principles of modern databases. In the first chapter, author describes the nature of data models, providing a historical overview, from earlier hierarchical and network database models to the modern relational and post-relational object database models. The 1st chapter illustrates how a relational database is composed of a set of tables. With these elementary ideas in place, the 2nd chapter provides more formal principles of functioning of the relational model, as well as mathematical theory of sets and relations and other relational concepts and terminology like primary key, functional dependency, foreign keys, nulls, and referential integrity. After reading the chapter, the reader should be able to explain through the database examples the concepts of relation algebra and its operations like RESTRICT, PROJECT, JOIN, UNION, PRODUCT, and DIFFERENCE. The 3rd chapter consists of 3 parts: The Entity-Relationship (ER) Model, Converting an ER Model Into a Relational Database Model, and Normalization of the Relational Model. Normalization is a formal procedure that can contribute to the design of relational tables. This procedure explores potential functional dependencies and allows us to verify that the designed tables do not contain any anomalies. Each part of this chapter successfully depicts the relationship between entities and ends with a review questions and a set of exercises. While the first 3 chapters describe how design relational model was developed, using tools such as ER modeling and normalization, the 4th chapter deals with physical database design, ie, converting the conceptual design into an actual working system using a chosen database product. This chapter describes the common features of a database management system, the software that represents the hardware of the database through database engine, query processor, schema manager, and form and report managers.

The hands-on section A, Learning SQL, follows after chapter 4 and provides an introductory description of the principal features of the SQL languages through a series of tutorials and practical exercises. SQL is a declarative command language that enables us to perform a wide range of operations on relational tables in a relational database. These operations are divided into Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL) commands. The tutorials were prepared using the Oracle 10g SQL*Plus interpreter (www.oracle.com), but they can be used equally with other systems like MySQL (www.mysql.org) and Microsoft Access.

The 5th chapter covers the general topic of the programming user interface to the database. This chapter briefly explains two main areas: access to the database by different forms of programming and user-interface techniques, such as forms and reports. SQL was shown to be very important and significant tool in database development. This chapter explains the benefits of various standard database access interfaces: ODBC, JDBC, OLEDB, ADO, as well as the benefits of the fourth generation type programming system, such as Oracle PL/SQL, with coding procedure examples.

Chapter 6 explains the concept of a transaction in database access and problems that can arise in a multi-user system, where multiple transactions run concurrently and potentially access the same data. Transaction refers to a group of changes and queries to the database. This chapter also covers commit and rollback transaction commands, problems of concurrency, serialization of transactions, locking, and deadlocks.

Chapter 7 refers to integrity and security. The author describes the database integrity through validation techniques. Privileges and permissions as security facilities in Microsoft Access are given as an example.

Chapter 8 reviews some of the important theoretical and practical aspects of using databases as distributed database systems in the network environment. There are 2 main topics in this chapter: client-server systems and distributed databases.

Hands-on section B is divided into 3 sections: Getting Started in Access, More About Access, and Advanced Access. The first section guides readers through the process of creating a basic database with tables, forms, queries, and reports in Microsoft Access. Theoretical knowledge of database principles and design from the previous chapters is assumed. The practical work includes sample databases that implement case studies described in the previous chapters.

Chapter 9 describes a number of more recent database technologies that offer alternatives to the relational model and provide compromise between relational and object-oriented databases. The main topic areas of this chapter are assessment of the relational database, object-relational databases, object-oriented databases, and object-relational mapping. In this chapter, the author examines the advantages and disadvantages of both relational and object-based databases. The concept of an object-relational or extended relational database is offered as a possible solution to the object vs relational conflict. The basic purpose and advantage of object-relational databases is enhancement of the relational model by some relaxation of relational principles. The author describes two examples of object-relational systems – Oracle's extensions to SQL with object-relational features of standard SQL 1999 and Intersystems Caché, as a novel system with a sophisticated graphical interface and a wide range or storage and access functionality. Object database is typically much better integrated with application programming systems such as Java, C++, and C#, that could be too complex for beginners in database development.

Chapter 10 deals with the techniques for developing databases in World Wide Web applications. In this chapter, the author describes a number of technologies that are commonly used in web database communication as cgi-bin, Active Server Pages (ASP) and ASP.net, Java Server Pages (JSP), and Personal Home Page (PHP). This chapter is primarily intended for readers who have some knowledge and experience in programming.

The last chapter of the book discusses how the worlds of databases and relatively new developmental area of Extensible Markup Language (XML) intersect and interact. This chapter summarizes the available XML support tools with understanding the features available in current XML-enabled and Native XML databases. This chapter covers the most significant developments like SQL/XML standard, XQuery, and Native XML databases.

Highlights. The book is well structured and easy to read with a series of tutorials and practical exercises.

Limitations. The book offers an excellent insight into the theory and practice of database development but it is too advanced for general medical reader. It is primarily intended for readers who already have some knowledge and experience with databases and programming.