大学计算机教育丛书（影印版）

A FIRST COURSE IN DATABASE SYSTEMS

JEFFREY D.ULLMAN
JENNIFER WIDOM

数据库系统基础教程

清华大学出版社 · PRENTICE HALL

A First Course in Database Systems

数据库系统基础教程

Jeffrey D. Ullman

and

Jennifer Widom

Department of Computer Science

Stanford University

清华大学出版社

Prentice-Hall International, Inc.

(京) 新登字 158 号

图书在版编目(CIP)数据

数据库系统基础教程: 英文/厄尔曼(Ullman, J. D.), 威多姆(Widom, J.)著. - 影印版. - 北京: 清华大学出版社, 1998.2
(大学计算机教育丛书)
ISBN 7-302-02829-X

Ⅰ. 数… Ⅱ. ①厄… ②威… Ⅲ. 数据库系统-英文 Ⅳ. TP311.13

中国版本图书馆 CIP 数据核字(98)第 01676 号

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)
因特网地址: www.tup.tsinghua.edu.cn
印刷者: 清华大学印刷厂
发行者: 新华书店总店北京科技发行所
开 本: 850×1168 1/32 印张: 15.25
版 次: 1998 年 4 月第 1 版 1998 年 4 月第 1 次印刷
书 号: ISBN 7-302-02829-X/TP·1486
印 数: 0001~5000
定 价: 23.00 元

出版前言

我们的大学生、研究生毕业后，面临的将是一个国际化的信息时代。他们将需要随时查阅大量的外文资料；会有更多的机会参加国际性学术交流活动；接待外国学者；走上国际会议的讲坛。作为科技工作者，他们不仅应有与国外同行进行口头和书面交流的能力，更为重要的是，他们必须具备极强的查阅外文资料获取信息的能力。有鉴于此，在国家教委所颁布的“大学英语教学大纲”中有一条规定：专业阅读应作为必修课程开设。同时，在大纲中还规定了这门课程的学时和教学要求。有些高校除开设“专业阅读”课之外，还在某些专业课拟进行英语授课。但教、学双方都苦于没有一定数量的合适的英文原版教材作为教学参考书。为满足这方面的需要，我们挑选了 7 本计算机科学方面最新版本的教材，进行影印出版。首批影印出版的 6 本书受到广大读者的热情欢迎，我们深受鼓舞，今后还将陆续推出新书。希望读者继续给予大力支持。Prentice Hall 公司和清华大学出版社这次合作将国际先进水平的教材引入我国高等学校，为师生们提供了教学用书，相信会对高校教材改革产生积极的影响。

清华大学出版社

Prentice Hall 公司

1997.11

Preface

This book developed from course notes used in Stanford's CS145 introduction to databases. CS145 is the first in a sequence of five courses.[1] As taught originally by Arthur Keller, it evolved into a course that stresses the aspects of database systems of most use to the majority of computer science students: database design and programming. The course also involves an extensive, running project, in which the students design and then implement a substantial database application. Assignments related to this project, other homework assignments, exams, and other course materials are available from this book's home page; see the section on "Support on the World Wide Web."

Use of the Book

This book is suitable for a one-semester course. In a one-quarter course, such as CS145, we have had to omit or skim some of the material. We'll leave it to the instructor what can best be elided, but obvious possibilities include material on Datalog, advanced aspects of SQL programming, and detailed coverage of SQL3.

If you have a running project as part of the course, then it is important that SQL instruction be moved earlier than it appears in the book. Some of the things that are suitable for postponement include the material on Datalog, SQL3 sections in Chapters 5 and 6, and some of the theory in Chapter 3 (but students need normalization and possibly multivalued dependencies if they are to do a good relational design before starting SQL programming).

Prerequisites

We have used the book at the "mezzanine" level, in a course taken both by advanced undergraduates and beginning graduate students. The formal prerequisites for the course are Sophomore-level treatments of: (1) Data structures, algorithms, and discrete math, and (2) Software systems, software engineering,

[1] The four that follow are: database system principles, a project course in database system implementation, transactions and distributed databases, and database theory.

and programming languages. Of this material, it is important that students have at least a rudimentary understanding of such topics as: algebraic expressions and laws, logic, basic data structures such as search trees, object-oriented programming concepts, and programming environments. However, we believe that adequate background is surely acquired by the end of the Junior year in a typical computer science program.

Exercises

The book contains extensive exercises, with some for almost every section. We indicate harder exercises or parts of exercises with an exclamation point. The hardest exercises have a double exclamation point.

Some of the exercises or parts are marked with a star. For these exercises, we shall endeavor to maintain solutions accessible through the book's web page. These solutions are publicly available and should be used for self-testing. Note that in a few cases, one exercise B asks for modification or adaptation of your solution to another exercise A. If certain parts of A have solutions, then you should expect the corresponding parts of B to have solutions as well.

Support on the World Wide Web

The book's home page is

```
http://www-db.stanford.edu/~ullman/fcdb.html
```

Here are solutions to starred exercises, errata as we learn of them, and backup materials. We hope to make available the notes for each offering of CS145 as we teach it, including homeworks, solutions, and project assignments.

Acknowledgements

Special thanks go to Bobbie Cochrane and Linda DeMichiel, for their help with aspects of the SQL3 standard. Many others helped us debug the manuscript as well, including: Donald Aingworth, Jonathan Becker, Larry Bonham, Christopher Chan, Oliver Duschka, Greg Fichtenholtz, Bart Fisher, Meredith Goldsmith, Steve Huntsberry, Leonard Jacobson, Thulasiraman Jeyaraman, Dwight Joe, Seth Katz, Brian Kulman, Le-Wei Mo, Mark Mortensen, Ramprakash Narayanaswami, Torbjorn Norbye, Mehul Patel, Catherine Tornabene, Jonathan Ullman, Mayank Upadhyay, Vassilis Vassalos, Qiang Wang, Sundar Yamunachari, and Takeshi Yokukawa. Remaining errors are ours, of course.

J. D. U.
J. W.

Table of Contents

Chapter 1

The Worlds of Database Systems

In this book the reader will learn the effective use of database management systems, including the design of databases and the programming of operations on databases. This chapter serves to introduce a number of important database concepts. After a brief history of the subject, we learn what makes database systems different from other software genres. This chapter also provides background concerning the implementation of the database management systems that support databases and their use. An understanding of what goes on "behind the scenes" is important if we are to have an appreciation of why databases are designed as they are or why there are limits on the way operations can be performed on databases. Finally, we review some ideas, such as object-oriented programming, with which the reader may be familiar but that are essential in the chapters to follow.

1.1 The Evolution of Database Systems

What is a database? In essence a database is nothing more than a collection of information that exists over a long period of time, often many years. In common parlance, the term *database* refers to a collection of data that is managed by a *database management system*, also called a *DBMS*, or just *database system*. A DBMS is expected to:

1. Allow users to create new databases and specify their *schema* (logical structure of the data), using a specialized language called a *data-definition language.*

2. Give users the ability to *query* the data (a "query" is database lingo for a question about the data) and modify the data, using an appropriate language, often called a *query language* or *data-manipulation language.*

3. Support the storage of very large amounts of data — gigabytes or more — over a long period of time, keeping it secure from accident or unauthorized use and allowing efficient access to the data for queries and database modifications.

4. Control access to data from many users at once, without allowing the actions of one user to affect other users and without allowing simultaneous accesses to corrupt the data accidentally.

1.1.1 Early Database Management Systems

The first commercial database management systems appeared in the late 1960's. These evolved from file systems, which provide some of item (3) above; file systems store data over a long period of time, and they allow the storage of large amounts of data. However, file systems do not generally guarantee that data cannot be lost if it is not backed up, and they don't support efficient access to data items whose location in a particular file is not known.

Further, file systems do not directly support item (2), a query language for the data in files. Their support for (1) — a schema for the data — is limited to the creation of directory structures for files. Finally, file systems do not satisfy (4). When they allow concurrent access to files by several users or processes, a file system generally will not prevent situations such as two users modifying the same file at about the same time, so the changes made by one user fail to appear in the file.

The first important applications of DBMS's were ones where data was composed of many small items, and many queries or modifications were made. Here are some of these applications.

Airline Reservations Systems

Here, the items of data include:

1. Reservations by a single customer on a single flight, including such information as assigned seat or meal preference.

2. Information about flights — the airports they fly from and to, their departure and arrival times, or the aircraft flown, for example.

3. Information about ticket prices, requirements, and availability.

Typical queries ask for flights leaving about a certain time from one given city to another, what seats are available, and at what prices. Typical data modifications include the booking of a flight for a customer, assigning a seat, or indicating a meal preference. Many agents will be accessing parts of the data at any given time. The DBMS must allow such concurrent accesses, prevent problems such as two agents assigning the same seat simultaneously, and protect against loss of records if the system suddenly fails.