

HZ BOOKS
华章教育

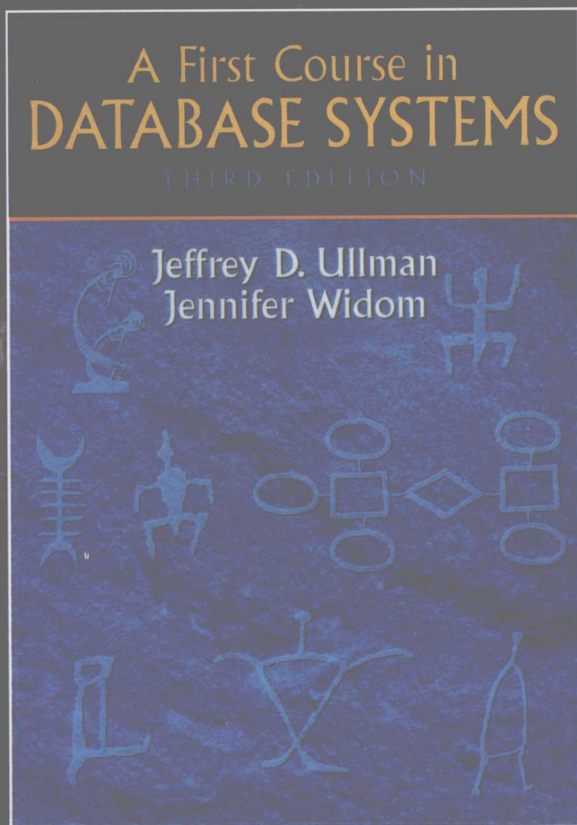
PEARSON
Prentice
Hall

计 算 机 科 学 丛 书

原书第3版

数据库系统基础教程

(美) Jeffrey D. Ullman Jennifer Widom 著 岳丽华 金培权 万寿红 等译
斯坦福大学



A First Course in Database Systems

Third Edition



机械工业出版社
China Machine Press

原书第3版

计 算 机 科 学 丛 书

数据库系统基础教程

(美) Jeffrey D. Ullman Jennifer Widom 著 岳丽华 金培权 万寿红 等译

A First Course in
DATABASE SYSTEMS
THIRD EDITION

A First Course in Database Systems

Third Edition



机械工业出版社
China Machine Press

本书由斯坦福大学知名计算机科学家Jeffrey Ullman和Jennifer Widom合作编写。本书首先介绍流行的关系数据库和对象关系数据库内容，涉及关系数据模型、E/R模型、UML模型以及对象模型等高级数据模型。然后介绍了有关半结构化数据组织管理中比较流行的XML等内容，既包括了数据组织模型的内容，也给出了相关编程语言，如XPath、XQuery、XSLT等。

本书举例丰富翔实，既可作为大学本科、研究生计算机及相关专业数据库课程的教科书，也可作为数据库领域技术人员的参考书。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *A First Course in Database Systems, Third Edition* (ISBN 978-0-13-600637-4) by Jeffrey D. Ullman and Jennifer Widom, Copyright © 2008, 2002, 1997.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2008-1786

图书在版编目（CIP）数据

数据库系统基础教程（原书第3版）/（美）厄尔曼（Ullman, J. D.）等著；岳丽华等译.—北京：机械工业出版社，2009.8

（计算机科学丛书）

书名原文：A First Course in Database Systems, Third Edition

ISBN 978-7-111-26828-4

I. 数… II. ①厄… ②岳… III. 数据库系统—教材 IV. TP311.13

中国版本图书馆CIP数据核字（2009）第057724号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：迟振春

三河市明辉印装有限公司印刷

2009年8月第1版第1次印刷

184mm×260mm·21.75印张

标准书号：ISBN 978-7-111-26828-4

定价：45.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章分社较早意识到“出版要为教育服务”。自1998年开始，华章分社就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章分社欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



译者序

数据库已是当今信息社会须臾不可脱离的重要工具，数据库的教学也就成为计算机科学与技术专业的一门必修课程。

Jeffrey D. Ullman教授是斯坦福大学计算机系的资深教授，自1980年编写了其第一本数据库教材《数据库系统原理》以来，已出版过多本数据库系统方面的教材。该书是他在斯坦福大学计算机系对大学生教授的第一门数据库课程（CS145）中使用的教材，Ullman教授在第2版出版4年后，对其作了更新又出版了第3版。与第2版相比，第3版不仅重新组织了章节从而使这本书的系统性更强，而且内容作了大幅度增加，包括了有关索引的介绍和目前XML数据库技术发展的新内容。

数据库技术发展到现在，其一个很大的变化是，数据库不仅要管理结构化的数据，而且要管理更多的半结构化的数据。本书正是从这个观点出发，将内容分成两大部分：首先仍然是流行的关系数据库和对象关系数据库内容，介绍了关系数据模型、E/R模型、UML模型以及对象模型等高级数据模型。然后介绍了有关半结构化数据组织管理中比较流行的XML等内容，既包括了数据组织模型的内容，也给出了相关编程语言，如XPath、XQuery、XSLT等。

该版本仍然保留了本教材的主要特点，举例丰富翔实，便于教师教学和自学者学习。书中在每一节后都给出了大量的练习题，并且标注了习题的难易程度，既便于教学安排，又便于学生循序渐进地掌握教学内容。另外，在Jeffrey D. Ullman教授的主页（<http://infolab.stanford.edu/~ullman/fcdb.html>）上还有关于该课程实验的内容，这对于本课程的实验教学有很大帮助。

本书由岳丽华负责翻译审校了第1~7章，金培权负责翻译审校了第8~10章，万寿红负责翻译审校了第11~12章。另外，参加翻译工作的还有刘沾沾、向小岩、田明辉、赵旭剑、秦富童、卢科、孙逸雪、陈艳等。

限于水平，译文中难免有错误与不足之处，欢迎读者批评指正。

译者

2009年5月

前 言

在斯坦福大学，因为实行的是一年四学期制，所以数据库引论课被分为两门课程。第一门课程是CS145，该课程只要求学生学会使用数据库系统，而不要求知道DBMS实现的内容。CS145是CS245的预修课，CS245介绍DBMS实现。学生若想进一步学习数据库方面的课程，可以学习CS345（此课是理论课）、CS346（此课是DBMS实现实验课）以及CS347（此课介绍事务处理及分布式数据库）课程。

从1997年开始，我们已经出版了两本配套教材。《数据库系统基础教程》是为CS145课程编写的。《数据库系统实现》是为CS245课程以及部分CS346课程编写的。由于很多学校实行学期制，或者是将这两门数据库引论课组合成一门引论课，因此，我们感到有必要将上述两本书合成一本《数据库系统全书》。

然而，更多的学生是要学会如何使用数据库系统，而不是如何实现数据库系统，所以我们继续将《数据库系统全书》的前半部分作为《数据库系统基础教程》出版。在第3版中，介绍了很多新内容，并且对编写思路有所调整。当前，数据库系统有两个重要模型：关系模型和半结构模型（XML）。因此，我们决定将面向对象数据库从原来的单独一章改为设计和对象关系系统章节中的内容。

第3版结构

在简短的第1章介绍之后，第2~4章中讨论关系模型。第4章讨论高级模型，除了E/R模型之外，还讨论UML（统一建模语言）。第4章中还包括ODL的简单介绍，主要是将它用作关系数据库模式的设计语言。在本书的Web站点上有更多有关ODL和OQL的介绍。

本版更新了函数依赖和多值依赖的内容，并作为第3章的主题。这里，假定函数依赖在其右部有一组属性集。另外还给出了一些算法，包括“chase”，该算法允许对依赖进行操作。第3章对3NF作了进一步讨论，包括3NF综合算法，以明确3NF和BCNF之间的区别是什么。

第5章除了讨论上一版的关系代数内容外，还增加了上一版第10章中的Datalog部分内容。有关Datalog中的递归内容，或者放入网站，或者放入了本版第10章中有关SQL的递归中讨论。

第6~10章讨论SQL程序设计的有关内容，是由上一版第6、7、8章及部分第10章内容重新组织而成的。有关索引和视图的内容单独组织为第8章，并且讨论了一些重要的新课题，包括物化视图和索引的自动选择等。

第9章基于上一版的第8章（嵌入式SQL），并新增加了有关三层体系结构一节。另外，还扩展了对JDBC的讨论，并加入了新的PHP内容。

第10章收集了一些有关SQL的高级课题。除涵盖上一版第8章中有关授权的内容和第10章中有关SQL的递归的内容外，大部分内容是有关嵌套关系模型（上一版的第4章）和SQL的对象关系特征（上一版的第9章）的。

第11章和第12章讨论XML以及基于XML的系统。除了包括上一版第4章最后的部分内容外，其他内容都是新的。第11章讨论建模，包括DTD以及XML模式。第12章讨论程序设计，包括XPath、XQuery和XSLT等。

如何使用本书

本书的内容很适合一学期（半年）有关数据库建模和程序设计的课程。如果只有四分之一学年的时间，那么需要省略某些内容。我们认为第2~7章是核心内容。虽然我们认为每一个学生都应该从第9章的一节中学会如何在宿主语言中嵌入SQL语句，但是剩余的5章可以按照意愿进行选择。

若如同我们在CS145课程中所做的那样，你想给学生一个真实的数据库应用设计和实现课程作业，则应该对本书的讲解顺序做某些调整，较早开始对SQL的介绍。虽然学生在做数据库设计时需要规范化知识，但可以推迟有关函数依赖的介绍。

预备知识

我们曾经将此书作为本科生和一年级研究生所修课程的教材。正常情况下，该课程是二年级课程，在此之前已学习过：（1）数据结构、算法、离散数学，（2）软件系统、软件工程和程序设计语言等。最重要的是学生至少要对如下内容有基本的理解：代数表达式和代数定律，逻辑，基本的数据结构，面向对象程序设计概念和程序设计环境。可是，我们认为最好修完标准的计算机专业三年级课程后再使用本书作教材。

习题

本书几乎在每一节都包括大量的习题，我们用感叹号对难题做了标记，对最难的习题用双感叹号做了标记。

网上支持

本书的网址是：

<http://infolab.stanford.edu/~ullman/fcdb.html>

该网站包括勘误表及支持材料。这里还有我们每次教授CS145课程的笔记，包括相关的作业、课程实验及考卷等。另外，我们还第3版中没有出现的第2版的材料放在网站上。

第2版和第3版的对照

下表列出了第2版章节与第3版章节的对照。

2版	3版	2版	3版	2版	3版	2版	3版	2版	3版
1.1	1.1	1.2	1.2	1.3	1.3	2.1	4.1	2.2	4.2
2.3	4.3	2.4	4.4	3.1	2.2	3.2	4.5	3.3	4.6
3.4	3.1	3.5	3.2	3.6	3.3~3.5	3.7	3.6~3.7	4.1	网上
4.2	4.9	4.3	4.9	4.4	4.10	4.5	10.3	4.6	11.1
4.7	11.2	5.1	2.2	5.2	2.4	5.3	5.1	5.4	5.2
5.5	2.5	6.1	6.1	6.2	6.2	6.3	6.3	6.4	6.4
6.5	6.5	6.6	2.3	6.7	8.1~8.2	7.1	7.1	7.2	7.2
7.3	7.3	7.4	7.4~7.5	8.1	9.3	8.2	9.4	8.3	9.2
8.4	9.5	8.5	9.6	8.6	6.6	8.7	10.1	9.1	网上
9.2	网上	9.3	网上	9.4	10.4	9.5	10.5	10.1	5.3
10.2	5.4	10.3	网上	10.4	10.2				

致谢

我们感谢与Donald Kossmann的讨论，特别是有关XML及其相关的程序设计系统内容的讨论。我们还要感谢Bobbie Cochrane 帮助我们理解上一版中有关触发器的语义。

有很多人曾帮助过我们，他们或是提供了本书及早期版本内容的最初材料，或是提供了本书或其他网上材料的勘误表等。对所有帮助过我们的人表示感谢，他们是：

Marc Abromowitz, Joseph H. Adamski, Brad Adelberg, Gleb Ashimov, Donald Aingworth, Teresa Almeida, Brian Babcock, Bruce Backer, Yunfan Bao, Jonathan Becker, Margaret Bentiez, Eberhard Bertsch, Larry Bonham, Phillip Bonnet, David Brokaw, Ed Burns, Alex Butler, Karen Butler, Mike Carey, Christopher Chan, Sudarshan Chawathe.

Per Christensen, Ed Chang, Surajit Chaudhuri, Ken Chen, Rada Chirkova, Nitin Chopra, Lewis Church, Jr., Bobbie Cochrane, Michael Cole, Alissa Cooper, Arturo Crespo, Linda DeMichiel, Matthew F. Dennis, Tom Dienstbier, Pearl D'Souza, Oliver Duschka, Xavier Faz, Greg Fichtenholtz, Bart Fisher, Simon Frettløeh, Jarl Friis.

John Fry, Chiping Fu, Tracy Fujieda, Prasanna Ganesan, Suzanne Garcia, Mark Gjol, Manish Godara, Seth Goldberg, Jeff Goldblat, Meredith Goldsmith, Luis Gravano, Gerard Guillemette, Himanshu Gupta, Petri Gynther, Jon Heggland, Rafael Hernandez, Masanori Higashihara, Antti Hjelt, Ben Holtzman, Steve Huntsberry.

Sajid Hussain, Leonard Jacobson, Thulasiraman Jeyaraman, Dwight Joe, Brian Jorgensen, Mathew P. Johnson, Sameh Kamel, Seth Katz, Pedram Keyani, Victor Kimeli, Ed Knorr, Yeong-Ping Koh, David Koller, Gyorgy Kovacs, Phillip Koza, Brian Kulman, Bill Labiosa, Sang Ho Lee, Younghan Lee, Miguel Licona.

Olivier Lobry, Chao-Jun Lu, Waynn Lue, John Manz, Arun Marathe, Philip Minami, Le-Wei Mo, Fabian Modoux, Peter Mork, Mark Mortensen, Ramprakash Narayanaswami, Hankyung Na, Mor Naaman, Mayur Naik, Marie Nilsson, Torbjorn Norbye, Chang-Min Oh, Mehul Patel, Soren Peen, Jian Pei.

Xiaobo Peng, Bert Porter, Limbek Reka, Prahash Ramanan, Nisheeth Ranjian, Suzanne Rivoire, Ken Ross, Tim Roughgarten, Mema Roussopoulos, Richard Scherl, Loren Shevitz, June Yoshiko Sison, Man Cho A. So, Elizabeth Stinson, Qi Su, Ed Swierk, Catherine Tornabene, Anders Uhl, Jonathan Ullman, Mayank Upadhyay.

Anatoly Varakin, Vassilis Vassalos, Krishna Venuturimilli, Vikram Vijayaraghavan, Terje Viken, Qiang Wang, Mike Wiacek, Kristian Widjaja, Janet Wu, Sundar Yamunachari, Takeshi Yokukawa, Bing Yu, Min-Sig Yun, Torben Zahle, Sandy Zhang.

J. D. U.

J. W.

2007年7月于斯坦福

目 录

出版者的话

译者序

前言

第1章 数据库系统世界	1
1.1 数据库系统的发展	1
1.1.1 早期的数据库管理系统	1
1.1.2 关系数据库系统	2
1.1.3 越来越小的系统	2
1.1.4 越来越大的系统	2
1.1.5 信息集成	3
1.2 数据库管理系统概述	3
1.2.1 数据定义语言命令	3
1.2.2 查询处理概述	4
1.2.3 存储器和缓冲区管理器	5
1.2.4 事务处理	5
1.2.5 查询处理器	6
1.3 本书概述	6
1.4 参考文献	7

第一部分 关系数据库模型

第2章 关系数据模型	9
2.1 数据模型概述	9
2.1.1 什么是数据模型	9
2.1.2 一些重要的数据模型	9
2.1.3 关系模型简介	10
2.1.4 半结构化模型简介	10
2.1.5 其他数据模型	11
2.1.6 几种建模方法的比较	11
2.2 关系模型基础	12
2.2.1 属性	12
2.2.2 模式	12
2.2.3 元组	12
2.2.4 域	12
2.2.5 关系的等价描述	13
2.2.6 关系实例	13
2.2.7 关系上的键	13

2.2.8 数据库模式示例	14
2.2.9 习题	15
2.3 在SQL中定义关系模式	16
2.3.1 SQL中的关系	16
2.3.2 数据类型	16
2.3.3 简单的表定义	17
2.3.4 修改关系模式	18
2.3.5 默认值	18
2.3.6 键的声明	19
2.3.7 习题	20
2.4 代数查询语言	21
2.4.1 为什么需要一种专门的查询语言	21
2.4.2 什么是代数	21
2.4.3 关系代数概述	21
2.4.4 关系上的集合操作	21
2.4.5 投影	23
2.4.6 选择	23
2.4.7 笛卡儿积	24
2.4.8 自然连接	24
2.4.9 θ 连接	25
2.4.10 组合操作构成查询	26
2.4.11 命名和重命名	27
2.4.12 操作之间的联系	27
2.4.13 代数表达式的线性符号	28
2.4.14 习题	29
2.5 关系上的约束	32
2.5.1 作为约束语言的关系代数	32
2.5.2 引用完整性约束	33
2.5.3 键约束	33
2.5.4 其他约束举例	34
2.5.5 习题	35
2.6 小结	35
2.7 参考文献	36
第3章 关系数据库设计理论	37
3.1 函数依赖	37
3.1.1 函数依赖的定义	37
3.1.2 关系的键	38

3.1.3 超键	39	3.7.4 投影MVD	69
3.1.4 习题	40	3.7.5 习题	70
3.2 函数依赖的规则	40	3.8 小结	70
3.2.1 函数依赖的推导	40	3.9 参考文献	71
3.2.2 分解/结合规则	40	第4章 高级数据库模型	73
3.2.3 平凡函数依赖	41	4.1 E/R模型	73
3.2.4 计算属性的闭包	42	4.1.1 实体集	73
3.2.5 闭包算法为何有效	43	4.1.2 属性	74
3.2.6 传递规则	44	4.1.3 联系	74
3.2.7 函数依赖的闭包集合	45	4.1.4 实体-联系图	74
3.2.8 投影函数依赖	45	4.1.5 E/R图实例	75
3.2.9 习题	47	4.1.6 二元E/R联系的多样性	75
3.3 关系数据库模式设计	48	4.1.7 多路联系	76
3.3.1 异常	48	4.1.8 联系中的角色	77
3.3.2 分解关系	49	4.1.9 联系的属性	78
3.3.3 Boyce-Codd范式	50	4.1.10 多路联系到二元联系的转换	79
3.3.4 分解为BCNF	50	4.1.11 E/R模型中的子类	80
3.3.5 习题	52	4.1.12 习题	81
3.4 分解的优劣	53	4.2 设计原则	82
3.4.1 从分解中恢复信息	53	4.2.1 忠实性	82
3.4.2 无损连接的chase检验	55	4.2.2 避免冗余	83
3.4.3 为什么chase检验有效	56	4.2.3 简单性	83
3.4.4 依赖的保持	57	4.2.4 选择正确的联系	83
3.4.5 习题	58	4.2.5 选择正确的元素种类	84
3.5 第三范式	58	4.2.6 习题	86
3.5.1 第三范式的定义	59	4.3 E/R模型中的约束	87
3.5.2 3NF模式综合算法	59	4.3.1 E/R模型中的键	87
3.5.3 为什么3NF综合算法有效	60	4.3.2 E/R模型中键的表示	87
3.5.4 习题	60	4.3.3 引用完整性	88
3.6 多值依赖	61	4.3.4 度约束	89
3.6.1 属性独立及随之产生的冗余	61	4.3.5 习题	89
3.6.2 多值依赖的定义	61	4.4 弱实体集	89
3.6.3 多值依赖的推导	62	4.4.1 弱实体集的来源	89
3.6.4 第四范式	64	4.4.2 弱实体集的要求	90
3.6.5 分解为第四范式	64	4.4.3 弱实体集的符号	91
3.6.6 范式间的联系	65	4.4.4 习题	91
3.6.7 习题	65	4.5 从E/R图到关系设计	92
3.7 MVD的发现算法	66	4.5.1 实体集到关系的转化	92
3.7.1 闭包和chase	66	4.5.2 E/R联系到关系的转化	92
3.7.2 将chase扩展到MVD	67	4.5.3 关系组合	94
3.7.3 chase为何对MVD有效	69	4.5.4 处理弱实体集	95

4.5.5	习题	96
4.6	子类结构到关系的转化	97
4.6.1	E/R方式转化	97
4.6.2	面向对象方法	98
4.6.3	使用空值组合关系	99
4.6.4	各种方法的比较	99
4.6.5	习题	100
4.7	统一建模语言	101
4.7.1	UML类	101
4.7.2	UML类的键	101
4.7.3	关联	101
4.7.4	自关联	102
4.7.5	关联类	103
4.7.6	UML中的子类	103
4.7.7	聚集与组合	104
4.7.8	习题	105
4.8	UML图到关系的转化	105
4.8.1	UML到关系的基础知识	105
4.8.2	从UML子类到关系	106
4.8.3	从聚集与组合到关系	106
4.8.4	UML与弱实体集的类比	107
4.8.5	习题	108
4.9	对象定义语言	108
4.9.1	类声明	108
4.9.2	ODL中的属性	108
4.9.3	ODL中的联系	109
4.9.4	反向联系	109
4.9.5	联系的多重性	110
4.9.6	ODL中的类型	110
4.9.7	ODL中的子类	112
4.9.8	在ODL中声明键	113
4.9.9	习题	113
4.10	从ODL设计到关系设计	114
4.10.1	从ODL类到关系	114
4.10.2	类中的复杂属性	114
4.10.3	值集合类型属性的表示	115
4.10.4	其他类型构建器的表示	116
4.10.5	ODL中联系的表示	117
4.10.6	习题	117
4.11	小结	118
4.12	参考文献	119

第二部分 关系数据库程序设计

第5章	代数和逻辑查询语言	121
5.1	包上的关系操作	121
5.1.1	为什么采用包	121
5.1.2	包的并、交、差	122
5.1.3	包上的投影操作	123
5.1.4	包上的选择操作	123
5.1.5	包的笛卡儿积	124
5.1.6	包的连接	124
5.1.7	习题	125
5.2	关系代数的扩展操作符	125
5.2.1	消除重复	126
5.2.2	聚集操作符	126
5.2.3	分组	127
5.2.4	分组操作符	127
5.2.5	扩展的投影操作符	128
5.2.6	排序操作符	129
5.2.7	外连接	129
5.2.8	习题	130
5.3	关系逻辑	131
5.3.1	谓词和原子	131
5.3.2	算术原子	131
5.3.3	Datalog规则和查询	132
5.3.4	Datalog规则的意义	133
5.3.5	扩展谓词和内涵谓词	134
5.3.6	Datalog规则应用于包	135
5.3.7	习题	136
5.4	关系代数与Datalog	136
5.4.1	布尔操作	136
5.4.2	投影	137
5.4.3	选择	137
5.4.4	积	138
5.4.5	连接	139
5.4.6	用Datalog模拟多重操作	139
5.4.7	Datalog与关系代数的比较	140
5.4.8	习题	141
5.5	小结	141
5.6	参考文献	142
第6章	数据库语言SQL	143
6.1	SQL中的简单查询	143
6.1.1	SQL中的投影	145

6.1.2	SQL中的选择	146
6.1.3	字符串比较	147
6.1.4	SQL中的模式匹配	147
6.1.5	日期和时间	148
6.1.6	空值和涉及空值的比较	149
6.1.7	布尔值UNKNOWN	149
6.1.8	输出排序	151
6.1.9	习题	151
6.2	多关系查询	152
6.2.1	SQL中的积和连接	153
6.2.2	消除属性歧义	153
6.2.3	元组变量	154
6.2.4	多关系查询的解释	155
6.2.5	查询的并、交、差	156
6.2.6	习题	157
6.3	子查询	158
6.3.1	产生标量值的子查询	159
6.3.2	关系的条件表达式	160
6.3.3	元组的条件表达式	160
6.3.4	关联子查询	161
6.3.5	FROM子句中的子查询	162
6.3.6	SQL的连接表达式	163
6.3.7	自然连接	164
6.3.8	外连接	164
6.3.9	习题	165
6.4	全关系操作	167
6.4.1	消除重复	167
6.4.2	并、交、差中的重复	168
6.4.3	SQL中的分组和聚集	168
6.4.4	聚集操作符	168
6.4.5	分组	169
6.4.6	分组、聚集和空值	170
6.4.7	HAVING子句	171
6.4.8	习题	171
6.5	数据库更新	172
6.5.1	插入	172
6.5.2	删除	174
6.5.3	修改	174
6.5.4	习题	175
6.6	SQL中的事务	175
6.6.1	可串行化	175

6.6.2	原子性	177
6.6.3	事务	177
6.6.4	只读事务	178
6.6.5	读脏数据	179
6.6.6	其他隔离层次	180
6.6.7	习题	181
6.7	小结	182
6.8	参考文献	183
第7章	约束与触发器	184
7.1	键和外键	184
7.1.1	外键约束声明	184
7.1.2	维护引用完整性	185
7.1.3	延迟约束检查	186
7.1.4	习题	188
7.2	属性和元组上的约束	188
7.2.1	非空值约束	189
7.2.2	基于属性的CHECK约束	189
7.2.3	基于元组的CHECK约束	190
7.2.4	基于元组和基于属性的约束的比较	191
7.2.5	习题	191
7.3	修改约束	192
7.3.1	给约束命名	192
7.3.2	修改表上的约束	193
7.3.3	习题	193
7.4	断言	194
7.4.1	创建断言	194
7.4.2	使用断言	194
7.4.3	习题	195
7.5	触发器	196
7.5.1	SQL中的触发器	196
7.5.2	触发器设计的选项	197
7.5.3	习题	199
7.6	小结	200
7.7	参考文献	201
第8章	视图与索引	202
8.1	虚拟视图	202
8.1.1	视图定义	202
8.1.2	视图查询	203
8.1.3	属性重命名	203
8.1.4	习题	204
8.2	视图更新	204

8.2.1	视图删除	204	9.3.7	游标更新	231
8.2.2	可更新视图	204	9.3.8	避免并发修改	231
8.2.3	视图中的替换触发器	206	9.3.9	动态SQL	232
8.2.4	习题	207	9.3.10	习题	233
8.3	SQL中的索引	207	9.4	存储过程	234
8.3.1	建立索引的动机	208	9.4.1	创建PSM函数和过程	234
8.3.2	索引的声明	208	9.4.2	PSM中的简单语句格式	234
8.3.3	习题	209	9.4.3	分支语句	235
8.4	索引的选择	209	9.4.4	PSM中的查询	236
8.4.1	简单代价模型	209	9.4.5	PSM中的循环	237
8.4.2	一些有用的索引	209	9.4.6	for循环	238
8.4.3	计算最佳索引	211	9.4.7	PSM中的异常处理	239
8.4.4	索引的自动选择	213	9.4.8	使用PSM函数和过程	241
8.4.5	习题	213	9.4.9	习题	241
8.5	物化视图	214	9.5	使用调用层接口	242
8.5.1	物化视图的维护	214	9.5.1	SQL/CLI简介	242
8.5.2	物化视图的定期维护	215	9.5.2	进程语句	244
8.5.3	利用物化视图重写查询	216	9.5.3	从查询结果中取数据	244
8.5.4	物化视图的自动创建	217	9.5.4	向查询传递参数	246
8.5.5	习题	218	9.5.5	习题	247
8.6	小结	218	9.6	JDBC	247
8.7	参考文献	219	9.6.1	JDBC简介	247
第9章	服务器环境下的SQL	220	9.6.2	JDBC中的创建语句	248
9.1	三层体系结构	220	9.6.3	JDBC中的游标操作	249
9.1.1	Web服务器层	220	9.6.4	参数传递	249
9.1.2	应用层	221	9.6.5	习题	250
9.1.3	数据库层	222	9.7	PHP	250
9.2	SQL环境	222	9.7.1	PHP基础	250
9.2.1	环境	222	9.7.2	数组	251
9.2.2	模式	223	9.7.3	PEAR DB库	251
9.2.3	目录	223	9.7.4	使用DB创建数据库连接	251
9.2.4	SQL环境中的客户和服务	224	9.7.5	执行SQL语句	251
9.2.5	连接	224	9.7.6	PHP中的游标操作	252
9.2.6	会话	225	9.7.7	PHP中的动态SQL	252
9.2.7	模块	225	9.7.8	习题	253
9.3	SQL/宿主语言接口	225	9.8	小结	253
9.3.1	阻抗不匹配问题	226	9.9	参考文献	254
9.3.2	SQL与宿主语言连接	227	第10章	关系数据库的新课题	255
9.3.3	DECLARE节	227	10.1	SQL中的安全机制和用户认证	255
9.3.4	使用共享变量	227	10.1.1	权限	255
9.3.5	单元组选择语句	228	10.1.2	创建权限	256
9.3.6	游标	229	10.1.3	检查权限的过程	257

10.1.4 授权	257	第三部分 半结构化数据的建模和程序设计
10.1.5 授权图	258	第11章 半结构化数据模型
10.1.6 收权	259	11.1 半结构化数据
10.1.7 习题	262	11.1.1 为何需要半结构化数据模型
10.2 SQL中的递归	262	11.1.2 半结构化数据表示
10.2.1 在SQL中定义递归关系	262	11.1.3 信息集成与半结构化数据
10.2.2 有问题的递归SQL表达式	264	11.1.4 习题
10.2.3 习题	266	11.2 XML
10.3 对象关系模型	267	11.2.1 语义标签
10.3.1 从关系到对象关系	267	11.2.2 有模式和无模式的XML
10.3.2 嵌套关系	267	11.2.3 格式规范的XML
10.3.3 引用	268	11.2.4 属性
10.3.4 面向对象与对象关系的比较	269	11.2.5 连接元素的属性
10.3.5 习题	270	11.2.6 命名空间
10.4 SQL中的用户定义类型	270	11.2.7 XML和数据库
10.4.1 在SQL中定义类型	270	11.2.8 习题
10.4.2 用户定义类型中的方法声明	271	11.3 文档类型定义
10.4.3 方法定义	272	11.3.1 DTD的格式
10.4.4 用UDT声明关系	272	11.3.2 使用DTD
10.4.5 引用	272	11.3.3 属性列表
10.4.6 为表生成对象标识符	273	11.3.4 标识符和引用
10.4.7 习题	274	11.3.5 习题
10.5 对象关系数据上的操作	274	11.4 XML模式
10.5.1 引用的跟随	274	11.4.1 XML模式的格式
10.5.2 访问UDT类型的元组分量	275	11.4.2 元素
10.5.3 生成器和转换器函数	276	11.4.3 复杂类型
10.5.4 UDT上联系的排序	277	11.4.4 属性
10.5.5 习题	278	11.4.5 受限的简单类型
10.6 联机分析处理	279	11.4.6 XML模式中的键
10.6.1 OLAP和数据仓库	279	11.4.7 XML模式中的外键
10.6.2 OLAP应用	279	11.4.8 习题
10.6.3 OLAP数据的多维视图	280	11.5 小结
10.6.4 星型模式	280	11.6 参考文献
10.6.5 切片和切块	281	第12章 XML程序设计语言
10.6.6 习题	283	12.1 XPath
10.7 数据立方体	283	12.1.1 XPath数据模型
10.7.1 立方体算子	284	12.1.2 文档节点
10.7.2 SQL中的立方体算子	285	12.1.3 路径表达式
10.7.3 习题	286	12.1.4 相对路径表达式
10.8 小结	286	12.1.5 路径表达式中的属性
10.9 参考文献	287	12.1.6 轴

12.1.7 表达式的上下文	313	12.2.9 XQuery表达式中的分支	324
12.1.8 通配符	313	12.2.10 查询结果排序	324
12.1.9 路径表达式中的条件	314	12.2.11 习题	325
12.1.10 习题	315	12.3 扩展样式表语言	326
12.2 XQuery	316	12.3.1 XSLT基础	326
12.2.1 XQuery基础	317	12.3.2 模板	326
12.2.2 FLWR表达式	318	12.3.3 从XML数据中获取值	327
12.2.3 通过变量的值置换变量	320	12.3.4 模板的递归应用	328
12.2.4 XQuery中的连接	321	12.3.5 XSLT中的迭代	329
12.2.5 XQuery比较操作符	322	12.3.6 XSLT中的条件	330
12.2.6 消除重复	323	12.3.7 习题	331
12.2.7 XQuery中的量词	323	12.4 小结	332
12.2.8 聚集	324	12.5 参考文献	332

第1章 数据库系统世界

在当今的生活中数据库已是每一项业务的基础。无论何时访问一个提供信息的Web站点——不论这个站点是著名的Google、Yahoo!、Amazon.com还是成千上万较小的站点——都有一个数据库为用户的信息访问提供服务。企业也将其所有重要的记录存放在数据库中进行维护。数据库同样也应用在很多科学研究的核心中。天文学家、人类基因研究者、探索蛋白质医药性质的生物学家，以及其他很多科学活动中获取的数据也是用数据库表示的。

数据库的能力来自于已发展了数十年的知识和技术，这些知识和技术蕴藏在名为数据库管理系统 (database management system) 的软件中。该软件也叫做DBMS，或更通俗地称为“数据库系统”。DBMS是一个能有效建立和维护大量数据的强大工具，并且能安全地长期保存这些数据。数据库系统是最复杂的软件系统之一。本书中，读者将学习如何设计数据库，如何用各种程序语言和DBMS一起编写应用程序，以及如何设计DBMS本身。

1.1 数据库系统的发展

数据库是什么？本质上讲，数据库就是信息的集合。该集合可以存在很长时间，通常是很多年。一般来讲，数据库是指由DBMS管理的数据的集合。DBMS需要有如下功能：

1. 允许用户使用特殊的数据定义语言 (data-definition language) 建立新的数据库，并说明它们的模式 (schema) 即数据的逻辑结构。

2. 使用合适的查询语言 (query language) 或数据操作语言 (data-manipulation language)，为用户提供查询 (query, “查询”是数据库关于数据申请的术语) 和更新 (modify) 数据的能力。

3. 支持超大数据量 (吉字节或更多) 数据的长时间存储，并且在数据查询和更新时支持对数据的有效存取。

4. 具有持久性，在面对各种故障、错误或用户错误地使用数据库时，数据库的恢复保证了数据的一致性。

5. 控制多个用户对数据的同时存取，不允许一个用户的操作影响另一个用户 (称作独立性, isolation)，也不允许对数据的不完整操作 (称作原子性, atomicity)。

1.1.1 早期的数据库管理系统

第一个商用数据库管理系统出现在20世纪60年代末。这些系统都是来自于文件系统，它们提供某些上面提到的第(3)项功能：文件系统可以长期地存储数据，并且允许存储大数据量数据。可是，如果数据不做备份，文件系统通常并不保证数据不会丢失。当不知道数据项在某个文件中的存储位置时，文件系统也不提供数据的有效访问。

文件系统不直接支持上述第(2)项功能，即没有对文件的查询语言。对第(1)项功能的支持 (即数据模式的支持) 也只限于文件目录结构的建立。对于功能(4)，文件系统也不能满足，没有备份的数据可能会丢失。最后，文件系统也不满足功能(5)。当有多个用户或进程对文件并发访问时，文件系统不能防止两个用户同时对同一个文件的修改，于是将出现一个用户的修

改被丢失的情形。

DBMS第一批重要的应用是数据由很多小数据项组成，并且完成很多查询或修改。下面是一些例子。

1. 银行系统：维护账目，并且保证系统故障时不会损失金钱。

2. 飞机订票系统：如同银行系统，需要保证数据不会丢失，并且能够接受顾客发出的大量小操作。

3. 企业记录系统：雇员和税收记录、库存、销售记录，大量各种其他类型的信息，这些记录大多是关键数据。

早期的DBMS需要程序员直接面对数据的存储格式。这些数据库系统使用多个不同的数据模型描述数据库中的信息结构，主要有基于树结构的“层次”模型和基于图的“网状”模型。网状模型通过CODASYL（数据系统语言委员会）报告在20世纪60年代末被标准化^①。

早期模型和系统的一个问题是不支持高级查询语言。例如，CODASYL查询语言的语句只允许用户通过数据元素间的指针，从一个数据元素跳到另一个数据元素。因此，即使是写一个非常简单的查询程序，用户也要花费很大的气力。

1.1.2 关系数据库系统

随着1970年Ted Codd著名论文的发表^②，数据库系统有了重大的改变。Codd提出数据库系统应该将数据组织成表的形式呈现给用户。这种形式称作关系（relation）。在关系的后面，可能是一个复杂的数据结构，实现对各种查询问题的快速响应。但是，不同于早期数据库系统的程序员，关系数据库的程序员将不必关心数据的存储结构，查询可以用非常高级的语言表述，因此可以极大地增加数据库程序员的工作效率。本书的大部分内容都与数据库系统的关系模型相关。SQL（结构化查询语言）这一最重要的基于关系模型的查询语言，将广泛地应用在本书中。

到1990年，关系数据库系统已成为标准。但是，数据库领域在继续发展，数据管理的新课题、新方法不断出现。面向对象特征已经渗入关系模型。有些大型数据库已不再使用关系方法组织。后续部分将讨论数据库系统的某些新趋势。

1.1.3 越来越小的系统

最初，DBMS是运行在大型计算机上的既庞大又昂贵的软件系统。因为存储吉字节（gigabyte）数据需要大的计算机系统，所以大容量是必需的。如今，单个磁盘的容量就可达若干吉字节，DBMS运行在个人计算机上已成为可能。因此，关系模型数据库可以在非常小的机器上运行。而且，如同以前的电子表格和字处理系统，关系数据库正在成为计算机应用的普通工具。

另一个重要趋势是文档的应用，这些应用中常常使用XML（扩展的模型语言，eXtensible Modeling Language）。大量小文档的集合可以如同数据库，而对这些文档的查询和操作不同于原来的关系数据库系统。

1.1.4 越来越大的系统

另一方面，吉字节数据量还不够大。企业数据库里可以存储太字节（ 10^{12} 字节）数据。而

^① CODASYL “Data Base Task Group” Apr. 1971报告，ACM, New York.

^② Codd, E. F., “大量共享数据库的关系模型”，COMM. ACM, 13:6, pp. 377-387, 1970.