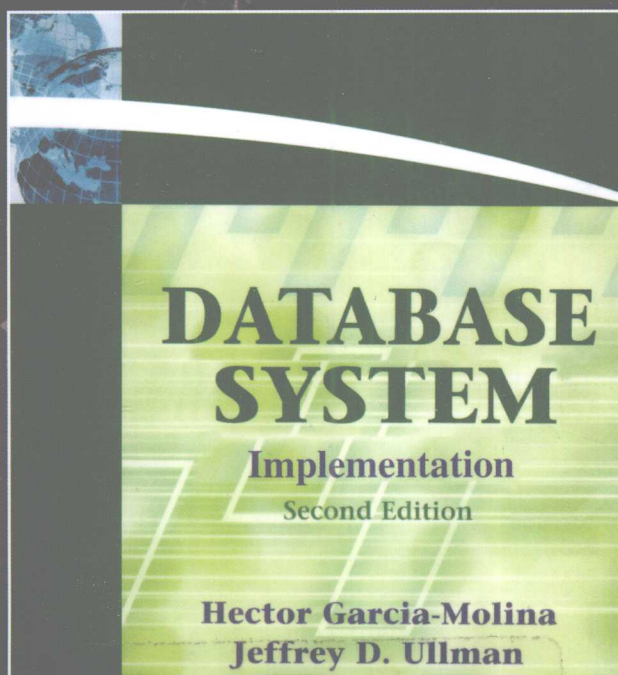


# 数据库系统实现

(美) Hector Garcia-Molina Jeffrey D. Ullman Jennifer Widom 著 杨冬青 吴愈青 包小源 唐世渭 等译  
斯坦福大学



Database System Implementation  
Second Edition

计 算 机 科 学 丛 书

第2版

# 数据库系统实现

(美) Hector Garcia-Molina Jeffrey D. Ullman Jennifer Widom 著 杨冬青 吴愈青 包小源 唐世渭 等译  
斯坦福大学

## DATABASE SYSTEM

Implementation  
Second Edition

Hector Garcia-Molina  
Jeffrey D. Ullman  
Jennifer Widom

Database System Implementation  
Second Edition



机械工业出版社  
China Machine Press

本书是斯坦福大学计算机专业数据库系列课程第二门课的教科书。书中对数据库系统实现原理进行了深入阐述,并具体讨论了数据库管理系统的三个主要成分——存储管理器、查询处理器和事务管理器的实现技术。此外,第2版充分反映了数据管理技术的新进展,对内容进行了扩充,除了在第1版中原有的“信息集成”一章(第10章)中加入了新的内容外,还增加了两个全新的章:“数据挖掘”(第11章)和“数据库系统与互联网”(第12章)。

本书适合作为高等院校计算机专业研究生的教材或本科生的教学参考书,也适合作为从事相关研究或开发工作的专业技术人员的高级参考资料。

Simplified Chinese edition copyright © 2010 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Database System Implementation, Second Edition* (ISBN 978-0-13-135428-9) by Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom, Copyright © 2009, 2002.

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。

封底无防伪标均为盗版

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:01-2009-1342

图书在版编目(CIP)数据

数据库系统实现 第2版/(美)加西亚-莫利纳(Garcia-Molina, H.)等著;杨冬青等译. —北京:机械工业出版社, 2010.5

(计算机科学丛书)

书名原文: Database System Implementation, Second Edition

ISBN 978-7-111-30287-2

I. 数… II. ①加… ②杨… III. 数据库系统 IV. TP311.13

中国版本图书馆 CIP 数据核字 (2010) 第 057560 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

北京京师印务有限公司印刷

2010 年 5 月第 2 版第 1 次印刷

184mm × 260mm · 25 印张

标准书号: ISBN 978-7-111-30287-2

定价: 59.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

## 出版者的话

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,不仅擘划了研究的范畴,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公司较早意识到“出版要为教育服务”。自1998年开始,我们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, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessey, 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。欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

华章网站: [www.hzbook.com](http://www.hzbook.com)

电子邮件: [hzj@hzbook.com](mailto:hzj@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



华章教育

华章科技图书出版中心

# 译者序

随着计算机硬件、软件技术的飞速发展和计算机系统在各行各业的广泛应用，数据已经成为各种机构的宝贵资源，数据库系统对于当今科研部门、政府机关、企事业单位等来说都是至关重要的。而数据库系统中的核心软件是数据库管理系统（DBMS）。DBMS 用于高效地创建和存储大量的数据，并对数据进行有效的管理、处理和维护，是数据库专家和技术人员数十年研究开发的结果，是当前最复杂的系统软件之一。要深入掌握数据库系统的原理和技术，进而从事数据库管理软件和工具的开发，必须学习和研究数据库管理系统实现技术。要深入了解数据库系统的内部结构，以开发出高效的数据库应用系统，也需要学习和研究数据库管理系统实现技术。

Hector Garcia-Molina、Jeffrey D. Ullman 和 Jennifer Widom 是斯坦福大学著名的计算机科学家，多年来他们在数据库系统领域中做了大量的开创性工作，由他们撰写的《数据库系统实现》一书是关于数据库系统实现方面内容最为全面的著述之一。我们于 2000 年将《数据库系统实现》的第 1 版译成中文，国内许多大学采用它作为研究生数据库课程的教材或主要教学参考书，收到了良好的效果。

现在我们又翻译了《数据库系统实现》第 2 版。第 2 版保持了第 1 版的总体风格，首先对数据库系统实现原理进行了深入阐述，并具体讨论了数据库管理系统的三个主要成分——存储管理器、查询处理器和事务管理器的实现技术。与第 1 版相比，第 2 版对于数据存储和索引结构的阐述进行了适当的压缩，分别将原来的两章合并为一章；另外，增加了一章“并行与分布式数据库”（第 9 章），其中包括了第 1 版中分散在查询处理和事务管理的相关章节中的内容，并增加了有关分布式查询执行的一些新内容，例如，map-reduce 并行架构、P2P 数据库以及分布式散列的实现等。同时，第 2 版充分反映了数据管理技术的新进展，对内容进行了扩充，除了在第 1 版中原有的“信息集成”一章（第 10 章）中加入了新的内容外，还增加了两个全新的章：“数据挖掘”（第 11 章）和“数据库系统与互联网”（第 12 章）。“数据挖掘”一章中包含了关联规则与频繁项集挖掘技术，从一个非常大的数据库或 Web 页面集合中发现“相似”的项的“最小散列”和“局部敏感散列”等关键技术，以及高维空间中大规模数据的聚簇问题等。“数据库系统与互联网”一章中重点阐述了与互联网相关的两个方面的数据库技术：Web 搜索引擎及其 PageRank 算法，流数据模型以及管理数据流形式的大量数据所需的技术。

我们认为这本书既适合作为高等院校计算机专业研究生的教材或本科生的教学参考书，又适合作为从事相关研究或开发工作的专业技术人员的高级参考资料。

杨冬青全面组织了本书的翻译，吴愈青、包小源、唐世渭在本书的翻译和审校中做了大量的工作。除此之外，参加翻译的还有闫秋玲、郑丽丽、蔡慧慧、马煜、张棋、陈巍、郭思祺、夏海峰、翁学天、郭少松、李树节。

限于译者水平，译文中难免有疏漏和错误，欢迎批评指正。

译者  
于北京大学

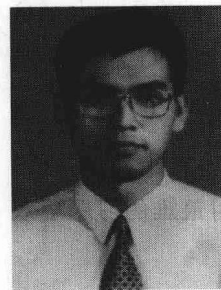
## 译者简介



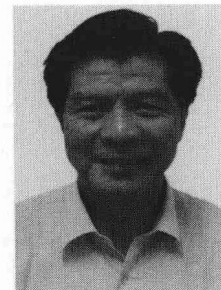
**杨冬青** 1969年毕业于北京大学数学力学系数学专业，现任北京大学信息科学技术学院教授，博士生导师，计算机科学技术系主任，中国计算机学会数据库专委会委员。多年来承担并完成973、863、国家科技攻关、国家自然科学基金等多项国家重点科研项目，曾获国家科技进步二等奖、三等奖和多项省部级奖励，在国内外杂志及会议上发表论文百余篇，著译作十余部。目前主要研究方向为数据库系统实现技术、Web环境下的信息集成与共享、数据仓库和数据挖掘等。



**吴愈青** 分别于1995年和1998年在北京大学计算机系获得学士学位和硕士学位；2004年于EECS Department, University of Michigan获得博士学位。现就职于美国Indiana University，任Assistant Professor。主要研究方向为数据库系统及实现，包括数据库查询语言、查询优化、索引技术等，及其在半结构化数据上的应用与实现。在国际会议及期刊上发表论文三十余篇。



**包小源** 博士，副教授。兰州大学计算数学专业硕士、北京大学计算机应用专业博士。主要研究方向为数据库实现技术、XML数据管理、对等计算、服务计算等。



**唐世渭** 1964年毕业于北京大学数学力学系计算数学专业，毕业后留校任教至今，现为北京大学信息科学技术学院教授，博士生导师，中国计算机学会数据库专委会委员，中国软件行业协会数据库及应用软件分会理事长。多年来承担并完成973、863、国家科技攻关、国家自然科学基金等多项国家重点科研项目，曾获国家科技进步二等奖、三等奖各1项，省部级科技进步奖多项，在国内外杂志及会议上发表论文百余篇，著译作多部。目前主要研究方向为数据库系统、数据仓库和数据挖掘、Web环境下的信息集成与共享、典型应用领域的信息系统等。

# 出版前言

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 课程（此课介绍事务处理及分布式数据库）。

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》<sup>①</sup>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。本书就是《数据库系统实现》的最新版——第 2 版。

第 2 版保持了第 1 版的总体风格，但对于数据存储和索引结构的阐述进行了适当的压缩，分别将原来的两章合并为一章；另外，增加了一章“并行与分布式数据库”（第 9 章），其中包括了第 1 版中分散在查询处理和事务管理的相关章节中的内容，并增加了有关分布式查询执行的一些新内容。同时，第 2 版充分反映了数据管理技术的新进展，对内容进行了扩充，除了在第 1 版中原有的“信息集成”一章（第 10 章）中加入了新的内容外，还增加了两个全新的章：“数据挖掘”（第 11 章）和“数据库系统与互联网”（第 12 章）。

## 预备知识

学生一般不会在大学的第一学年选修数据库系统实现课程，所以本书读者应具有广泛的计算机科学背景知识。我们假定读者已经学过数据库语言，特别是 SQL。读者最好了解关系代数，并且熟悉基本的数据结构。同样，关于文件系统和操作系统的知识也是很有用的。

## 习题

本书几乎在每一节都包括大量的练习，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。

---

① 本书中文版（ISBN 978-7-111-26828-4）和影印版（ISBN 978-7-111-24733-3）已由机械工业出版社出版。——编辑注

# 目 录

|       |  |
|-------|--|
| 出版者的话 |  |
| 译者序   |  |
| 译者简介  |  |
| 出版前言  |  |

|                  |   |
|------------------|---|
| 第 1 章 DBMS 系统概述  | 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     | 3 |
| 1.2.3 主存和缓冲区管理器  | 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 数据库模型和语言回顾   | 7 |
| 1.4.1 关系模型回顾     | 7 |
| 1.4.2 SQL 回顾     | 7 |
| 1.5 参考文献         | 9 |

## 第一部分 数据库系统实现

|                   |    |
|-------------------|----|
| 第 2 章 辅助存储管理      | 11 |
| 2.1 存储器层次         | 11 |
| 2.1.1 存储器层次       | 11 |
| 2.1.2 在存储器层次间传送数据 | 12 |
| 2.1.3 易失和非易失存储器   | 13 |
| 2.1.4 虚拟存储器       | 13 |
| 2.1.5 习题          | 13 |
| 2.2 磁盘            | 14 |
| 2.2.1 磁盘结构        | 14 |
| 2.2.2 磁盘控制器       | 15 |
| 2.2.3 磁盘存取特性      | 15 |
| 2.2.4 习题          | 16 |
| 2.3 加速对辅助存储器的访问   | 17 |

|                     |    |
|---------------------|----|
| 2.3.1 计算的 I/O 模型    | 17 |
| 2.3.2 按柱面组织数据       | 18 |
| 2.3.3 使用多个磁盘        | 18 |
| 2.3.4 磁盘镜像          | 19 |
| 2.3.5 磁盘调度和电梯算法     | 19 |
| 2.3.6 预取和大规模缓冲      | 20 |
| 2.3.7 习题            | 20 |
| 2.4 磁盘故障            | 21 |
| 2.4.1 间断性故障         | 21 |
| 2.4.2 校验和           | 22 |
| 2.4.3 稳定存储          | 22 |
| 2.4.4 稳定存储的错误处理能力   | 23 |
| 2.4.5 从磁盘崩溃中恢复      | 23 |
| 2.4.6 作为冗余技术的镜像     | 23 |
| 2.4.7 奇偶块           | 24 |
| 2.4.8 一种改进: RAID 5  | 26 |
| 2.4.9 多个盘崩溃时的处理     | 26 |
| 2.4.10 习题           | 28 |
| 2.5 组织磁盘上的数据        | 29 |
| 2.5.1 定长记录          | 29 |
| 2.5.2 定长记录在块中的放置    | 30 |
| 2.5.3 习题            | 31 |
| 2.6 块和记录地址的表示       | 31 |
| 2.6.1 客户机-服务器系统中的地址 | 31 |
| 2.6.2 逻辑地址和结构地址     | 32 |
| 2.6.3 指针混写          | 33 |
| 2.6.4 块返回磁盘         | 35 |
| 2.6.5 被钉住的记录和块      | 36 |
| 2.6.6 习题            | 36 |
| 2.7 变长数据和记录         | 37 |
| 2.7.1 具有变长字段的记录     | 37 |
| 2.7.2 具有重复字段的记录     | 38 |
| 2.7.3 可变格式的记录       | 39 |
| 2.7.4 不能装入一个块中的记录   | 40 |
| 2.7.5 BLOB          | 40 |
| 2.7.6 列存储           | 41 |
| 2.7.7 习题            | 41 |

|                               |    |                                   |     |
|-------------------------------|----|-----------------------------------|-----|
| 2.8 记录的修改 .....               | 42 | 3.5.1 网格文件 .....                  | 74  |
| 2.8.1 插入 .....                | 42 | 3.5.2 网格文件的查找 .....               | 74  |
| 2.8.2 删除 .....                | 43 | 3.5.3 网格文件的插入 .....               | 75  |
| 2.8.3 修改 .....                | 44 | 3.5.4 网格文件的性能 .....               | 76  |
| 2.8.4 习题 .....                | 44 | 3.5.5 分段散列函数 .....                | 77  |
| 2.9 小结 .....                  | 44 | 3.5.6 网格文件和分段散列的比较 .....          | 78  |
| 2.10 参考文献 .....               | 45 | 3.5.7 习题 .....                    | 79  |
| 第3章 索引结构 .....                | 47 | 3.6 多维数据的树结构 .....                | 79  |
| 3.1 索引结构基础 .....              | 47 | 3.6.1 多键索引 .....                  | 80  |
| 3.1.1 顺序文件 .....              | 48 | 3.6.2 多键索引的性能 .....               | 80  |
| 3.1.2 稠密索引 .....              | 48 | 3.6.3 <i>kd</i> -树 .....          | 81  |
| 3.1.3 稀疏索引 .....              | 49 | 3.6.4 <i>kd</i> -树的操作 .....       | 81  |
| 3.1.4 多级索引 .....              | 49 | 3.6.5 使 <i>kd</i> -树适合辅助存储器 ..... | 83  |
| 3.1.5 辅助索引 .....              | 49 | 3.6.6 四叉树 .....                   | 83  |
| 3.1.6 辅助索引的运用 .....           | 50 | 3.6.7 <i>R</i> -树 .....           | 84  |
| 3.1.7 辅助索引中的间接 .....          | 51 | 3.6.8 <i>R</i> -树的操作 .....        | 85  |
| 3.1.8 文档检索和倒排索引 .....         | 52 | 3.6.9 习题 .....                    | 86  |
| 3.1.9 习题 .....                | 55 | 3.7 位图索引 .....                    | 87  |
| 3.2 <i>B</i> -树 .....         | 55 | 3.7.1 位图索引的动机 .....               | 88  |
| 3.2.1 <i>B</i> -树的结构 .....    | 56 | 3.7.2 压缩位图 .....                  | 89  |
| 3.2.2 <i>B</i> -树的应用 .....    | 57 | 3.7.3 分段长度编码位向量的操作 .....          | 90  |
| 3.2.3 <i>B</i> -树的查找 .....    | 59 | 3.7.4 位图索引的管理 .....               | 91  |
| 3.2.4 范围查询 .....              | 59 | 3.7.5 习题 .....                    | 91  |
| 3.2.5 <i>B</i> -树的插入 .....    | 60 | 3.8 小结 .....                      | 92  |
| 3.2.6 <i>B</i> -树的删除 .....    | 62 | 3.9 参考文献 .....                    | 93  |
| 3.2.7 <i>B</i> -树的效率 .....    | 64 | 第4章 查询执行 .....                    | 96  |
| 3.2.8 习题 .....                | 64 | 4.1 物理查询计划操作符介绍 .....             | 97  |
| 3.3 散列表 .....                 | 65 | 4.1.1 扫描表 .....                   | 97  |
| 3.3.1 辅存散列表 .....             | 65 | 4.1.2 扫描表时的排序 .....               | 97  |
| 3.3.2 散列表的插入 .....            | 66 | 4.1.3 物理操作符计算模型 .....             | 98  |
| 3.3.3 散列表的删除 .....            | 66 | 4.1.4 衡量代价的参数 .....               | 98  |
| 3.3.4 散列表索引的效率 .....          | 67 | 4.1.5 扫描操作符的 I/O 代价 .....         | 99  |
| 3.3.5 可扩展散列表 .....            | 67 | 4.1.6 实现物理操作符的迭代器 .....           | 99  |
| 3.3.6 可扩展散列表的插入 .....         | 68 | 4.2 一趟算法 .....                    | 101 |
| 3.3.7 线性散列表 .....             | 69 | 4.2.1 一次单个元组操作的<br>一趟算法 .....     | 102 |
| 3.3.8 线性散列表的插入 .....          | 70 | 4.2.2 整个关系的一元操作的<br>一趟算法 .....    | 102 |
| 3.3.9 习题 .....                | 71 | 4.2.3 二元操作的一趟算法 .....             | 104 |
| 3.4 多维索引 .....                | 72 | 4.2.4 习题 .....                    | 106 |
| 3.4.1 多维索引的应用 .....           | 72 | 4.3 嵌套循环连接 .....                  | 106 |
| 3.4.2 利用传统索引执行范围查询 .....      | 73 | 4.3.1 基于元组的嵌套循环连接 .....           | 106 |
| 3.4.3 利用传统索引执行<br>最近邻查询 ..... | 73 | 4.3.2 基于元组的嵌套循环连接的<br>迭代器 .....   | 107 |
| 3.4.4 多维索引结构综述 .....          | 74 |                                   |     |
| 3.5 多维数据的散列结构 .....           | 74 |                                   |     |

|                            |     |                            |     |
|----------------------------|-----|----------------------------|-----|
| 4.3.3 基于块的嵌套循环连接算法         | 107 | 4.8.5 习题                   | 129 |
| 4.3.4 嵌套循环连接的分析            | 108 | 4.9 小结                     | 129 |
| 4.3.5 迄今为止的算法的总结           | 109 | 4.10 参考文献                  | 130 |
| 4.3.6 习题                   | 109 | 第5章 查询编译器                  | 132 |
| 4.4 基于排序的两趟算法              | 109 | 5.1 语法分析和预处理               | 132 |
| 4.4.1 两阶段多路归并排序            | 110 | 5.1.1 语法分析与语法分析树           | 132 |
| 4.4.2 利用排序去除重复             | 111 | 5.1.2 SQL的一个简单子集的语法        | 133 |
| 4.4.3 利用排序进行分组和聚集          | 111 | 5.1.3 预处理器                 | 135 |
| 4.4.4 基于排序的并算法             | 111 | 5.1.4 预处理涉及视图的查询           | 135 |
| 4.4.5 基于排序的交和差算法           | 112 | 5.1.5 习题                   | 137 |
| 4.4.6 基于排序的一个简单的<br>连接算法   | 112 | 5.2 用于改进查询计划的代数定律          | 137 |
| 4.4.7 简单的排序连接的分析           | 113 | 5.2.1 交换律与结合律              | 137 |
| 4.4.8 一种更有效的基于排序的<br>连接    | 113 | 5.2.2 涉及选择的定律              | 138 |
| 4.4.9 基于排序的算法的总结           | 114 | 5.2.3 下推选择                 | 140 |
| 4.4.10 习题                  | 114 | 5.2.4 涉及投影的定律              | 141 |
| 4.5 基于散列的两趟算法              | 115 | 5.2.5 有关连接与积的定律            | 142 |
| 4.5.1 通过散列划分关系             | 115 | 5.2.6 有关消除重复的定律            | 142 |
| 4.5.2 基于散列的消除重复算法          | 115 | 5.2.7 涉及分组与聚集的定律           | 143 |
| 4.5.3 基于散列的分组和聚集算法         | 116 | 5.2.8 习题                   | 144 |
| 4.5.4 基于散列的并、交、差算法         | 116 | 5.3 从语法分析树到逻辑查询计划          | 145 |
| 4.5.5 散列连接算法               | 116 | 5.3.1 转换成关系代数              | 145 |
| 4.5.6 节省一些磁盘 I/O           | 117 | 5.3.2 从条件中去除子查询            | 146 |
| 4.5.7 基于散列的算法的总结           | 118 | 5.3.3 逻辑查询计划的改进            | 149 |
| 4.5.8 习题                   | 119 | 5.3.4 可结合/可分配的运算符的<br>分组   | 150 |
| 4.6 基于索引的算法                | 119 | 5.3.5 习题                   | 151 |
| 4.6.1 聚簇和非聚簇索引             | 119 | 5.4 运算代价的估计                | 151 |
| 4.6.2 基于索引的选择              | 120 | 5.4.1 中间关系大小的估计            | 151 |
| 4.6.3 使用索引的连接              | 121 | 5.4.2 投影运算大小的估计            | 152 |
| 4.6.4 使用有序索引的连接            | 122 | 5.4.3 选择运算大小的估计            | 152 |
| 4.6.5 习题                   | 123 | 5.4.4 连接运算大小的估计            | 154 |
| 4.7 缓冲区管理                  | 124 | 5.4.5 多连接属性的自然连接           | 155 |
| 4.7.1 缓冲区管理结构              | 124 | 5.4.6 多个关系的连接              | 156 |
| 4.7.2 缓冲区管理策略              | 124 | 5.4.7 其他运算大小的估计            | 157 |
| 4.7.3 物理操作符选择和缓冲区<br>管理的关系 | 126 | 5.4.8 习题                   | 157 |
| 4.7.4 习题                   | 126 | 5.5 基于代价的计划选择介绍            | 158 |
| 4.8 使用超过两趟的算法              | 127 | 5.5.1 大小参数估计值的获取           | 158 |
| 4.8.1 基于排序的多趟算法            | 127 | 5.5.2 统计量的计算               | 160 |
| 4.8.2 基于排序的多趟算法的<br>性能     | 127 | 5.5.3 减少逻辑查询计划代价的<br>启发式估计 | 161 |
| 4.8.3 基于散列的多趟算法            | 128 | 5.5.4 枚举物理计划的方法            | 162 |
| 4.8.4 基于散列的多趟算法的性能         | 128 | 5.5.5 习题                   | 164 |
|                            |     | 5.6 连接顺序的选择                | 165 |
|                            |     | 5.6.1 连接的左右参数的意义           | 165 |

|                                   |     |                          |     |
|-----------------------------------|-----|--------------------------|-----|
| 5.6.2 连接树 .....                   | 165 | 6.4.4 习题 .....           | 200 |
| 5.6.3 左深连接树 .....                 | 165 | 6.5 针对介质故障的防护 .....      | 200 |
| 5.6.4 通过动态规划来选择连接<br>顺序和分组 .....  | 168 | 6.5.1 备份 .....           | 201 |
| 5.6.5 带有更具体的代价函数的<br>动态规划 .....   | 170 | 6.5.2 非静止转储 .....        | 201 |
| 5.6.6 选择连接顺序的贪婪算法 .....           | 171 | 6.5.3 使用备份和日志的恢复 .....   | 202 |
| 5.6.7 习题 .....                    | 171 | 6.5.4 习题 .....           | 203 |
| 5.7 物理查询计划选择的完成 .....             | 172 | 6.6 小结 .....             | 203 |
| 5.7.1 选取一个选择方法 .....              | 172 | 6.7 参考文献 .....           | 204 |
| 5.7.2 选取连接方法 .....                | 173 | 第7章 并发控制 .....           | 205 |
| 5.7.3 流水操作与物化 .....               | 174 | 7.1 串行调度和可串行化调度 .....    | 205 |
| 5.7.4 一元流水运算 .....                | 175 | 7.1.1 调度 .....           | 205 |
| 5.7.5 二元运算的流水操作 .....             | 175 | 7.1.2 串行调度 .....         | 205 |
| 5.7.6 物理查询计划的符号 .....             | 176 | 7.1.3 可串行化调度 .....       | 206 |
| 5.7.7 物理运算的排序 .....               | 178 | 7.1.4 事务语义的影响 .....      | 207 |
| 5.7.8 习题 .....                    | 179 | 7.1.5 事务和调度的一种记法 .....   | 207 |
| 5.8 小结 .....                      | 179 | 7.1.6 习题 .....           | 208 |
| 5.9 参考文献 .....                    | 180 | 7.2 冲突可串行化 .....         | 208 |
| 第6章 系统故障对策 .....                  | 182 | 7.2.1 冲突 .....           | 208 |
| 6.1 可恢复操作的问题和模型 .....             | 182 | 7.2.2 优先图及冲突可串行化判断 ..... | 209 |
| 6.1.1 故障模式 .....                  | 182 | 7.2.3 优先图测试发挥作用的原因 ..... | 211 |
| 6.1.2 关于事务的进一步讨论 .....            | 183 | 7.2.4 习题 .....           | 211 |
| 6.1.3 事务的正确执行 .....               | 184 | 7.3 使用锁的可串行化实现 .....     | 213 |
| 6.1.4 事务的原语操作 .....               | 185 | 7.3.1 锁 .....            | 213 |
| 6.1.5 习题 .....                    | 186 | 7.3.2 封锁调度器 .....        | 214 |
| 6.2 undo 日志 .....                 | 187 | 7.3.3 两阶段封锁 .....        | 214 |
| 6.2.1 日志记录 .....                  | 187 | 7.3.4 两阶段封锁发挥作用的原因 ..... | 215 |
| 6.2.2 undo 日志规则 .....             | 188 | 7.3.5 习题 .....           | 216 |
| 6.2.3 使用 undo 日志的恢复 .....         | 189 | 7.4 有多种锁模式的封锁系统 .....    | 217 |
| 6.2.4 检查点 .....                   | 191 | 7.4.1 共享锁与排他锁 .....      | 217 |
| 6.2.5 非静止检查点 .....                | 191 | 7.4.2 相容性矩阵 .....        | 218 |
| 6.2.6 习题 .....                    | 193 | 7.4.3 锁的升级 .....         | 219 |
| 6.3 redo 日志 .....                 | 194 | 7.4.4 更新锁 .....          | 220 |
| 6.3.1 redo 日志规则 .....             | 194 | 7.4.5 增量锁 .....          | 220 |
| 6.3.2 使用 redo 日志的恢复 .....         | 195 | 7.4.6 习题 .....           | 221 |
| 6.3.3 redo 日志的检查点 .....           | 195 | 7.5 封锁调度器的一种体系结构 .....   | 223 |
| 6.3.4 使用带检查点 redo 日志的<br>恢复 ..... | 196 | 7.5.1 插入锁动作的调度器 .....    | 223 |
| 6.3.5 习题 .....                    | 197 | 7.5.2 锁表 .....           | 225 |
| 6.4 undo/redo 日志 .....            | 197 | 7.5.3 习题 .....           | 226 |
| 6.4.1 undo/redo 规则 .....          | 197 | 7.6 数据库元素的层次 .....       | 226 |
| 6.4.2 使用 undo/redo 日志的恢复 .....    | 198 | 7.6.1 多粒度的锁 .....        | 227 |
| 6.4.3 undo/redo 日志的检查点 .....      | 199 | 7.6.2 警示锁 .....          | 227 |
|                                   |     | 7.6.3 幻象与插入的正确处理 .....   | 229 |
|                                   |     | 7.6.4 习题 .....           | 230 |
|                                   |     | 7.7 树协议 .....            | 230 |

|                               |     |                               |     |
|-------------------------------|-----|-------------------------------|-----|
| 7.7.1 基于树的封锁的动机 .....         | 230 | 8.4 小结 .....                  | 262 |
| 7.7.2 访问树结构数据的规则 .....        | 231 | 8.5 参考文献 .....                | 263 |
| 7.7.3 树协议发挥作用的原因 .....        | 232 | 第9章 并行与分布式数据库 .....           | 265 |
| 7.7.4 习题 .....                | 233 | 9.1 关系的并行算法 .....             | 265 |
| 7.8 使用时间戳的并发控制 .....          | 233 | 9.1.1 并行模型 .....              | 265 |
| 7.8.1 时间戳 .....               | 234 | 9.1.2 一次一个元组的操作的并行 .....      | 267 |
| 7.8.2 事实上不可实现的行为 .....        | 234 | 9.1.3 整个关系的操作的并行算法 .....      | 267 |
| 7.8.3 脏数据的问题 .....            | 235 | 9.1.4 并行算法的性能 .....           | 268 |
| 7.8.4 基于时间戳调度的规则 .....        | 235 | 9.1.5 习题 .....                | 270 |
| 7.8.5 多版本时间戳 .....            | 237 | 9.2 map-reduce 并行架构 .....     | 270 |
| 7.8.6 时间戳与封锁 .....            | 238 | 9.2.1 存储模式 .....              | 270 |
| 7.8.7 习题 .....                | 238 | 9.2.2 映射函数 .....              | 270 |
| 7.9 使用有效性确认的并发控制 .....        | 239 | 9.2.3 归约函数 .....              | 271 |
| 7.9.1 基于有效性确认调度器的<br>结构 ..... | 239 | 9.2.4 习题 .....                | 272 |
| 7.9.2 有效性确认规则 .....           | 239 | 9.3 分布式数据库 .....              | 272 |
| 7.9.3 三种并发控制机制的比较 .....       | 241 | 9.3.1 数据的分布 .....             | 272 |
| 7.9.4 习题 .....                | 242 | 9.3.2 分布式事务 .....             | 273 |
| 7.10 小结 .....                 | 242 | 9.3.3 数据复制 .....              | 273 |
| 7.11 参考文献 .....               | 243 | 9.3.4 习题 .....                | 274 |
| 第8章 再论事务管理 .....              | 245 | 9.4 分布式查询处理 .....             | 274 |
| 8.1 可串行性和可恢复性 .....           | 245 | 9.4.1 分布式连接操作问题 .....         | 274 |
| 8.1.1 脏数据问题 .....             | 245 | 9.4.2 半连接化简 .....             | 274 |
| 8.1.2 级联回滚 .....              | 246 | 9.4.3 多个关系的连接 .....           | 275 |
| 8.1.3 可恢复的调度 .....            | 246 | 9.4.4 非循环超图 .....             | 276 |
| 8.1.4 避免级联回滚的调度 .....         | 247 | 9.4.5 非循环超图的完全化简 .....        | 277 |
| 8.1.5 基于锁对回滚的管理 .....         | 247 | 9.4.6 为什么完全化简算法有效 .....       | 277 |
| 8.1.6 成组提交 .....              | 249 | 9.4.7 习题 .....                | 278 |
| 8.1.7 逻辑日志 .....              | 249 | 9.5 分布式提交 .....               | 278 |
| 8.1.8 从逻辑日志中恢复 .....          | 251 | 9.5.1 支持分布式原子性 .....          | 278 |
| 8.1.9 习题 .....                | 252 | 9.5.2 两阶段提交 .....             | 279 |
| 8.2 死锁 .....                  | 253 | 9.5.3 分布式事务的恢复 .....          | 280 |
| 8.2.1 超时死锁检测 .....            | 253 | 9.5.4 习题 .....                | 281 |
| 8.2.2 等待图 .....               | 253 | 9.6 分布式封锁 .....               | 282 |
| 8.2.3 通过元素排序预防死锁 .....        | 255 | 9.6.1 集中封锁系统 .....            | 282 |
| 8.2.4 通过时间戳检测死锁 .....         | 256 | 9.6.2 分布式封锁算法的代价模型 .....      | 282 |
| 8.2.5 死锁管理方法的比较 .....         | 257 | 9.6.3 封锁多副本的元素 .....          | 283 |
| 8.2.6 习题 .....                | 258 | 9.6.4 主副本封锁 .....             | 283 |
| 8.3 长事务 .....                 | 258 | 9.6.5 局部锁构成的全局锁 .....         | 284 |
| 8.3.1 长事务的问题 .....            | 259 | 9.6.6 习题 .....                | 285 |
| 8.3.2 saga (系列记载) .....       | 260 | 9.7 对等分布式查找 .....             | 285 |
| 8.3.3 补偿事务 .....              | 260 | 9.7.1 对等网络 .....              | 285 |
| 8.3.4 补偿事务发挥作用的原因 .....       | 261 | 9.7.2 分布式散列问题 .....           | 286 |
| 8.3.5 习题 .....                | 262 | 9.7.3 分布式散列的集中式解决<br>方案 ..... | 286 |

|                                     |     |                                      |     |
|-------------------------------------|-----|--------------------------------------|-----|
| 9.7.4 带弦的圆 .....                    | 287 | 10.6.4 合取查询的包含 .....                 | 317 |
| 9.7.5 带弦的圆上的链接 .....                | 287 | 10.6.5 为什么包含映射测试有效 .....             | 318 |
| 9.7.6 使用手指表查找 .....                 | 288 | 10.6.6 发现 mediator 查询的<br>解决方法 ..... | 319 |
| 9.7.7 加入新结点 .....                   | 289 | 10.6.7 为什么 LMSS 定理能成立 .....          | 320 |
| 9.7.8 当一个端离开网络 .....                | 291 | 10.6.8 习题 .....                      | 320 |
| 9.7.9 当一个端崩溃了 .....                 | 291 | 10.7 实体解析 .....                      | 320 |
| 9.7.10 习题 .....                     | 291 | 10.7.1 决定是否记录代表一个<br>共同实体 .....      | 321 |
| 9.8 小结 .....                        | 292 | 10.7.2 合并相似记录 .....                  | 322 |
| 9.9 参考文献 .....                      | 293 | 10.7.3 相似性和合并函数的<br>有用性质 .....       | 323 |
| <b>第二部分 现代数据库系统专题</b>               |     | 10.7.4 ICAR 记录的 R-Swoosh<br>算法 ..... | 324 |
| <b>第 10 章 信息集成</b> .....            | 295 | 10.7.5 为什么 R-Swoosh 算法<br>会有效 .....  | 325 |
| 10.1 信息集成介绍 .....                   | 295 | 10.7.6 实体解析的其他方法 .....               | 325 |
| 10.1.1 为什么要进行信息集成 .....             | 295 | 10.7.7 习题 .....                      | 326 |
| 10.1.2 异质性问题 .....                  | 296 | 10.8 小结 .....                        | 327 |
| 10.2 信息集成的方式 .....                  | 298 | 10.9 参考文献 .....                      | 328 |
| 10.2.1 联邦数据库系统 .....                | 298 | <b>第 11 章 数据挖掘</b> .....             | 330 |
| 10.2.2 数据仓库 .....                   | 299 | 11.1 频繁项集挖掘 .....                    | 330 |
| 10.2.3 mediator .....               | 300 | 11.1.1 市场 - 购物篮模型 .....              | 330 |
| 10.2.4 习题 .....                     | 301 | 11.1.2 基本定义 .....                    | 331 |
| 10.3 基于 mediator 的系统中的<br>包装器 ..... | 302 | 11.1.3 关联规则 .....                    | 332 |
| 10.3.1 查询模式的模板 .....                | 302 | 11.1.4 频繁项集的计算模型 .....               | 333 |
| 10.3.2 包装器生成器 .....                 | 303 | 11.1.5 习题 .....                      | 334 |
| 10.3.3 过滤器 .....                    | 304 | 11.2 发现频繁项集的算法 .....                 | 334 |
| 10.3.4 包装器上的其他操作 .....              | 304 | 11.2.1 频繁项集的分布 .....                 | 334 |
| 10.3.5 习题 .....                     | 305 | 11.2.2 寻找频繁项集的朴素算法 .....             | 335 |
| 10.4 基于能力的优化 .....                  | 306 | 11.2.3 A-Priori 算法 .....             | 336 |
| 10.4.1 有限的数据源能力问题 .....             | 306 | 11.2.4 A-Priori 算法的实现 .....          | 337 |
| 10.4.2 描述数据源能力的记号 .....             | 306 | 11.2.5 更好地使用主存 .....                 | 337 |
| 10.4.3 基于能力的查询计划<br>选择 .....        | 307 | 11.2.6 何时使用 PCY 算法 .....             | 338 |
| 10.4.4 加入基于成本的优化 .....              | 308 | 11.2.7 多级算法 .....                    | 339 |
| 10.4.5 习题 .....                     | 308 | 11.2.8 习题 .....                      | 340 |
| 10.5 优化 mediator 查询 .....           | 309 | 11.3 发现近似的商品 .....                   | 341 |
| 10.5.1 简化的修饰符记号 .....               | 309 | 11.3.1 相似度的 Jaccard 度量 .....         | 341 |
| 10.5.2 获得子目标的回答 .....               | 310 | 11.3.2 Jaccard 相似度的应用 .....          | 341 |
| 10.5.3 Chain 算法 .....               | 310 | 11.3.3 最小散列 .....                    | 342 |
| 10.5.4 在 mediator 上结合并视图 .....      | 312 | 11.3.4 最小散列与 Jaccard 相似度 .....       | 343 |
| 10.5.5 习题 .....                     | 313 | 11.3.5 为什么能用最小散列估计<br>相似度 .....      | 343 |
| 10.6 以局部作为视图的 mediator .....        | 314 | 11.3.6 最小散列的实现 .....                 | 343 |
| 10.6.1 LAV mediator 的动机 .....       | 314 |                                      |     |
| 10.6.2 LAV mediator 的术语 .....       | 315 |                                      |     |
| 10.6.3 扩展解决方案 .....                 | 316 |                                      |     |

|                                      |     |                                       |     |
|--------------------------------------|-----|---------------------------------------|-----|
| 11.3.7 习题 .....                      | 344 | 12.2.2 PageRank 的递归公式——<br>初步尝试 ..... | 364 |
| 11.4 局部敏感散列 .....                    | 345 | 12.2.3 爬虫陷阱和死角 .....                  | 366 |
| 11.4.1 LSH 实例：实体分辨 .....             | 345 | 12.2.4 考虑爬虫陷阱和死角的<br>PageRank .....   | 367 |
| 11.4.2 标签的局部敏感散列 .....               | 346 | 12.2.5 习题 .....                       | 368 |
| 11.4.3 最小散列法和局部敏感<br>散列的结合 .....     | 347 | 12.3 特定主题的 PageRank .....             | 369 |
| 11.4.4 习题 .....                      | 348 | 12.3.1 “远距离移动”集 .....                 | 369 |
| 11.5 大规模数据的聚簇 .....                  | 348 | 12.3.2 计算主题相关的 PageRank ...           | 370 |
| 11.5.1 聚簇的应用 .....                   | 349 | 12.3.3 链接作弊 .....                     | 371 |
| 11.5.2 距离的定义 .....                   | 350 | 12.3.4 主题相关的 PageRank 和<br>链接作弊 ..... | 371 |
| 11.5.3 凝聚式聚簇 .....                   | 352 | 12.3.5 习题 .....                       | 372 |
| 11.5.4 $k$ -Means 算法 .....           | 353 | 12.4 数据流 .....                        | 372 |
| 11.5.5 大规模数据的 $k$ -Means<br>方法 ..... | 354 | 12.4.1 数据流管理系统 .....                  | 372 |
| 11.5.6 内存中满载点后的<br>处理过程 .....        | 355 | 12.4.2 数据流应用 .....                    | 373 |
| 11.5.7 习题 .....                      | 356 | 12.4.3 数据流数据模型 .....                  | 374 |
| 11.6 小结 .....                        | 357 | 12.4.4 数据流转换为关系 .....                 | 374 |
| 11.7 参考文献 .....                      | 358 | 12.4.5 关系转换为数据流 .....                 | 375 |
| 第 12 章 数据库系统与互联网 .....               | 360 | 12.4.6 习题 .....                       | 376 |
| 12.1 搜索引擎体系结构 .....                  | 360 | 12.5 数据流挖掘 .....                      | 377 |
| 12.1.1 搜索引擎的组成 .....                 | 360 | 12.5.1 动机 .....                       | 377 |
| 12.1.2 Web 爬虫 .....                  | 361 | 12.5.2 统计二进制位数 .....                  | 378 |
| 12.1.3 搜索引擎中的查询处理 .....              | 363 | 12.5.3 统计不同元素的个数 .....                | 381 |
| 12.1.4 对网页进行排名 .....                 | 363 | 12.5.4 习题 .....                       | 381 |
| 12.2 用于识别重要网页的 PageRank ...          | 364 | 12.6 小结 .....                         | 382 |
| 12.2.1 PageRank 的直观思想 .....          | 364 | 12.7 参考文献 .....                       | 383 |

# 第 1 章 DBMS 系统概述

数据库对于当今的任何部门都是至关重要的。无论何时访问一个主要的网站——Yahoo!、Amazon.com 或者数以千计的提供信息的小网站，总有一个数据库在幕后服务，提供你所需要的信息。公司用数据库来维护其所有重要的记录。在许多科学研究的核心中也同样需要数据库。天文学家、人类学研究人员、探索蛋白质的医药特性的生物化学家以及许多其他的科学研究人员收集到的数据也是用数据库表示的。

数据库的能力来源于一个知识和技术的结合体，这是数十年研究开发的结果，并且已经嵌入到专门的软件中，这个软件称作数据库管理系统或 DBMS，或更通俗地称为“数据库系统”。DBMS 是一个强有力的工具，用于高效地创建和管理大量的数据，并使得数据能够安全地长期保存。DBMS 是当前最复杂的软件系统之一。在本书中，我们主要学习数据库管理系统实现技术，还将探讨当前数据库系统研究的一些新课题。

## 1.1 数据库系统的发展

什么是数据库？本质上讲，数据库就是信息的集合，它可以存在很长时间，往往是很多年。一般来讲，“数据库”这个词指的是由数据库管理系统管理的数据的集合。数据库管理系统将满足：

1. 允许用户使用专门的数据定义语言来创建新的数据库并指定其模式(数据的逻辑结构)。
2. 给予用户使用适当的语言来查询数据(“查询”是数据库术语，指关于数据的问题)和修改数据的能力，这种语言通常称为查询语言(query language)或数据操纵语言(data-manipulation language)。
3. 支持对非常大量的数据(许多 TB 或者更多)长期地进行存储，允许高效地存取数据以进行查询和数据库修改。
4. 使数据具有持久性(durability)，即能够从故障、多种类型的错误或者故意滥用中进行恢复。
5. 控制多个用户同时对数据进行访问，不允许用户间有不恰当的相互影响(称作孤立性(isolation))，并且不会发生在数据上进行了部分的而不是完整的操作的情况(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产生于 20 世纪 60 年代末。这些系统是由文件系统演变而来的，提供了对上述第(3)条的部分支持：文件系统可以长时间地存储数据，并且允许存储大量的数据。但是，文件系统并不能保证数据不丢失，如果没有备份的话；而且它们也不支持对数据项的高效存取，如果不知道它在特定的文件里的存储位置的话。

文件系统不直接支持第(2)条，即针对文件中数据的查询语言。它们对第(1)条(数据模式)的支持仅限于创建文件的目录结构。第(4)条并不总被文件系统支持，你可能会丢失并没有备份的数据。最后，文件系统也不满足第(5)条。虽然它们允许几个用户或进程并发地访问文件，但文件系统一般并不阻止两个用户同时修改同一个文件，从而导致一个用户的修改不能出现在文件中。

在 DBMS 最初的重要的应用中, 数据由许多小的数据项组成, 对数据有许多查询或修改。这些应用的实例如下:

1. 银行系统, 维护账户和确保系统故障时并不引起钱的丢失。
2. 航空公司订票系统, 这些系统就像银行系统一样, 要求确保数据不会丢失, 并且必须能接受顾客大量的小操作。
3. 企业记录系统, 雇员和税务记录、财产清单、销售记录和大量的其他类型的信息, 大部分是很关键的。

早期的 DBMS 要求程序员直接面对数据的存储格式。这些数据库系统用一些不同的数据模型来描述信息在数据库中的结构, 其中主要有“层次型的”或基于树的模型和基于图的“网状”模型。后者在 20 世纪 60 年代末通过了 CODASYL(数据系统和语言委员会)报告<sup>①</sup>, 从而实现标准化。

这些早期的模型和系统有一个问题, 就是它们并不支持高级查询语言。例如, CODASYL 查询语言具有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的语句。即使对于非常简单的查询, 也需要相当大的工作量来写这样的程序。

### 1.1.2 关系数据库系统

在 1970 年 Ted Codd 发表了一篇著名的论文<sup>②</sup>后, 数据库系统发生了显著的变化。Codd 认为数据库系统应该呈现给用户组织成叫做“关系”的表的数据。在幕后, 应该有一个复杂的数据结构, 允许对各式各样的查询进行快速响应。但是, 与早期数据库系统的程序员不同的是, 关系数据库的程序员并不需要关心存储结构。查询可以用很高级的语言来表达, 这样可以极大地提高数据库程序员的效率。我们将在这本书的大部分章节涵盖数据库系统的关系模型, 并将广泛涵盖 SQL(结构化查询语言)这一基于关系模型的最重要的查询语言。

到 1990 年, 关系数据库系统已经成为标准。然而数据库领域一直在发展, 而且针对数据管理的新的问题和方法也不断浮出水面。面向对象的特性渗入了关系模型。一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一些新趋势。

### 1.1.3 越来越小的系统

最初, DBMS 是庞大的、昂贵的、在大型计算机上运行的软件系统。因为保存吉字节(GB)的数据需要大的计算机系统, 所以大容量是必需的。今天, 几百 GB 的数据都可以放在单个磁盘上, 在个人计算机上运行 DBMS 已无任何问题。所以, 基于关系模型的数据库系统甚至可以在非常小的机器上运行, 而且它们也开始作为一种计算机应用的常见工具出现, 如同电子表格和文字处理器一样。

另一个重要的趋势是文档的使用, 经常使用 XML(可扩展的建模语言)来标记。大量小文档的集合可以作为一个数据库, 而查询和操作它们的方法与关系数据库系统不同。

### 1.1.4 越来越大的系统

另一方面, 吉字节也不再是大的数据了。公司数据库系统经常存储太字节(TB,  $10^{12}$ B)数据。还有许多数据库的数据量达到拍字节(PB,  $10^{15}$ B), 并把它们都提供给用户。一些重要的例子如下:

1. 卫星向下发送 PB 的信息, 存储在特殊的系统中。

① CODASYL Data Base Task Group April 1971 Report, ACM, New York.

② Codd, E. F., “A relational model for large shared data banks”, *Comm. ACM*, 13:6, pp. 377-387, 1970.

2. 一张图片比 1000 个字的存储空间大。只用 5kB 或 6kB 可以存储 1000 个字, 而存储一张图片需要更大的空间。一些像 Flickr 这样的信息库存储了数以百万计的图片, 并且提供对这些图片的搜索。甚至像 Amazon 这样的数据库也存有数以百万计的产品图片。

3. 如果静态的图片耗费空间, 电影则耗费更多。一小时的视频需要至少 1GB。像 YouTube 这样的网站拥有数十万或是上百万的电影, 并且能轻易地提供它们。

4. P2P 文件共享系统使用普通计算机构成的大网络来存储和发布各种各样的数据。虽然网络中每个节点可能只存储几百 GB 的数据, 但它们所存储的数据合起来非常巨大。

### 1.1.5 信息集成

在很大的程度上, 建立和维护数据这个老问题已经变成了信息集成的问题: 把许多相关的数据库所包含的信息连接成一个整体。例如, 一个大公司有许多分部。每个分部可能都建有各自独立的产品记录或员工记录的数据库。这些分部中可能有一些曾经是独立的公司, 它们有自己做事情的方式。这些分部可能会用不同的 DBMS 和不同的结构来存储信息, 可能会用不同的术语来表示同一件事或用相同的术语来表示不同的事。更糟糕的是, 因为存在使用这些不同的数据库的遗留应用程序, 使得几乎不可能废弃这些数据库。

结果是, 在现存数据库上建立结构体变得越来越必要, 其目的是把分布在这些数据库中的信息集成起来。一种常用的方法是建立数据仓库 (data warehouse), 将众多的遗留数据库中的信息进行适当的翻译, 周期性地拷贝到一个中心数据库中。另一种方法是实现一个 Mediator 或“中间件”, 它的功能是支持各个不同数据库中数据的一个集成的模型, 并在这个模型和每个数据库所使用的实际模型之间进行翻译。

## 1.2 数据库管理系统概述

在图 1-1 中, 我们可以看到一个完整的 DBMS 的轮廓。单线框表示系统成分, 双线框表示内存中的数据结构。实线指明控制和数据流, 虚线指明仅是数据流。由于该图很复杂, 我们分几个步骤来考虑它的细节。首先, 我们说对于 DBMS 有两个不同的命令来源:

1. 普通用户和应用程序, 他们要求对数据进行访问或修改。

2. 数据库管理员 (database administrator, DBA), 负责建立数据库的结构或模式的一个人或一组人。

### 1.2.1 数据定义语言命令

第二类命令处理起来相对比较简单, 我们以图 1-1 的右上部分为始点给出它的踪迹。例如, 一个大学注册管理数据库的数据库管理员 (或 DBA) 可能决定需要一个表或关系, 它的列是学生、学生选的课程和学生修该课程的成绩。DBA 可能还决定所允许的成绩只能是 A、B、C、D 和 F。此结构和约束信息都是数据库模式的一个部分。如图 1-1 所示, 由 DBA 输入这些信息, DBA 需要特殊的权限才能执行模式修改命令, 因为这些命令对于数据库会发生深刻的影响。这些进行模式修改的数据定义语言 (DDL) 命令由 DDL 处理程序进行分析, 然后传给执行引擎, 由执行引擎经过索引/文件/记录管理器去改变元数据 (metadata), 即数据库的模式信息。

### 1.2.2 查询处理概述

与 DBMS 的大部分交互都沿着图 1-1 左侧的路径进行。用户或应用程序使用数据操纵语言 (DML) 启动某个活动, 该活动不影响数据库模式, 但是可能影响数据库内容 (如果该活动是一个修改命令), 或者会从数据库中抽取数据 (如果该活动是一个查询)。DML 语句被如下两个分离的