计算机科学丛书

数据库系统实现

Hector Garcia-Molina
(美) Jeffrey D. Ullman 著
Jennifer Widom
(斯坦福大学)
杨冬青 唐世渭 徐其钧 等译

机械工业出版社
China Machine Press

需要全本请访问 www.ertongbook.com

本书是斯坦福大学计算机科学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。书中还对信息集成的最新技术，例如数据仓库、OLAP、数据挖掘、Mediator、数据立方体系统等进行了介绍。

本书适合于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。

Hector Garcia-Molina, Jeffrey D Ullman, and Jennifer Widom. Database System Implementation
Authorized translation from the English language edition published by Prentice Hall
Copyright © 2000 by Prentice Hall, Inc
All rights reserved
Chinese simplified language edition published by China Machine Press
Copyright © 2001 by China Machine Press

本书中文简体字版由美国Prentice Hall公司授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2000-0304

图书在版编目（CIP）数据

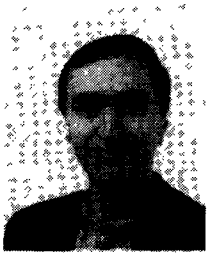
数据库系统实现/（美）加西亚（Garcia-Molina, H），（美）沃尔曼（Ullman, J D），（美）威德姆（Widom, J）著，杨冬青等译. -北京 机械工业出版社，2001 3
（计算机科学丛书）
书名原文. Database System Implementation

ISBN 7-111-07887-X
Ⅰ数 Ⅱ ①莫 ②沃 ③威 ④杨 Ⅲ 数据库系统-理论研究 Ⅳ TP311 13
中国版本图书馆CIP数据核字（2000）第80012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· 陈贤舜
北京忠信诚胶印厂印刷 · 新华书店北京发行所发行
2001年3月第1版第1次印刷
787mm × 1092mm 1/16 · 29 75印张
印数：0 001-5 000册
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

作者简介



Hector Garcıa-Molına是斯坦福大学计算机科学与电气工程系的Leonard Bosack和Sandra Lerner教授。他在数据库系统、分布式系统和数字图书馆领域中发表了大量论文。他的研究兴趣包括分布式计算系统、数据库系统和数字图书馆。



Jeffrey D Ullman是斯坦福大学的Stanford W Ascherman计算机科学教授。他作为作者或合作者出版了15本著作，发表了170篇技术论文，其中包括《A Fırst Course ın Database Systems》(Prentıce Hall 出版社，1997)和《Elements of ML Programmıng》(Prentıce Hall 出版社，1998)。他的研究兴趣包括数据库理论、数据库集成、数据挖掘和利用信息基础设施进行教育。他获得了Guggenheım Fellowshıp等多种奖励，并被推选进入国家工程院。他还被授予1996年Sıgmod贡献奖和1998年Karl V Karstrom杰出教育家奖。



Jennıfer Wıdom是斯坦福大学计算机科学与电气工程系的副教授。她是多个编辑委员会和程序委员会的成员，在计算机科学会议和杂志上发表了许多文章。她还是《A Fırst Course ın Database Systems》的作者之一。她的研究兴趣包括半结构化数据的数据库系统和XML、数据仓库以及主动数据库系统。

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 Garcıa-Molına、Jeffrey D Ullman和Jennıfer Wıdom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。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。书中还对信息集成的最新技术，例如数据仓库、OLAP、数据挖掘、Medıator(集成层软件)、数据立方体系统等进行了介绍。该书已经作为斯坦福大学计算机科学专业数据库系列课程第二门课的教科书使用。我们在北京大学计算机系研究生课程的教学中也使用了该书中的部分内容。

我们认为该书内容深入且全面，技术实用且先进，叙述深入浅出，是一本难得的高层次的教科书。我们将这本书译成中文，介绍给国内广大读者。我们认为这本书既适合于作为高等学校计算机专业研究生教材或本科生课程参考书，又适合于作为从事相关的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，唐世渭和徐其钧在本书的翻译和审校中做了大量的工作。参加翻译的还有杨良怀、王爱华、王腾蛟、叶茂盛、赵绍军、赵畅。另外，高桂英协助进行了译稿的整理、录入等工作。

在本书的翻译过程中，译者参照该书的WWW主页中的勘误表，对书中的疏漏之处进行了更正。此外，对于未包括在勘误表中的明显的笔误和排版错误，我们也做了订正。

限于译者水平，译文中疏漏和错误难免，欢迎批评指正。

译者

2000年10月于北京大学

译者简介



杨冬青，北京大学计算机系教授，博士生导师，数据库与信息系统领域负责人。中国计算机学会数据库专委会委员，中国计算机学会普及工作委员会副主任。1969年毕业于北京大学数学力学系数学专业，从事数据库与信息系统领域研究、开发与教学二十余年，曾获国家科技进步二等奖等多项奖励。



唐世渭，北京大学计算机系教授，博士生导师。北京大学信息科学中心主任，视觉与听觉信息处理国家重点实验室主任。中国计算机学会数据库专委会副主任。1964年毕业于北京大学数学力学系计算专业，从事数据库与信息系统领域研究、开发与教学二十余年，曾获国家科技进步二等奖等多项奖励。



徐其钧，北京大学计算机系高级工程师。从事数据库与信息系统领域的研究、开发十余年。参加过国家科技攻关项目数个，主持过大型数据库应用系统项目，并编著过数据库领域的书籍数部。

需要全本请访问 www.ertongbook.com

前　言

本书是为斯坦福大学数据库系列课程的第二门课CS245设计的。第一门课程CS145的内容包括数据库设计和数据库编程，Jeff Ullman和Jennifer Widom为该课程写的教科书《数据库系统入门教程》（A First Course in Database Systems）于1997年由Prentice-Hall出版社出版。CS245的内容包括DBMS实现技术，特别是存储结构、查询处理和事务管理。

本书的使用

斯坦福大学实行每学年4个学期的制度，所以采用本书的主要课程CS245的教学时间仅为10周。在1999年冬季学期，Hector Garcia-Molina使用了本书的“试用”版，教学内容包括以下部分：2.1～2.4节，整个第3章和第4章，5.1节和5.2节，6.1～6.7节，7.1～7.4节，整个第8章，第9章去掉9.8节，10.1～10.3节，11.1节，以及11.5节。

第6章和第7章的剩余部分（查询优化）在高级课程CS346中讲授。在该课程中，要求学生实现他们自己的DBMS。本书中未包括在CS245中的其他部分可以在另一门高级课程CS347中讲授，该课程讨论分布式数据库和高级事务处理。

实行学期制的学校可以将本书与前一本教科书《数据库系统入门教程》结合使用。我们建议将《数据库系统入门教程》用于第一个学期，同时进行数据库应用程序设计实习。第二学期可以讲授本书的大部分或全部内容。将数据库的学习分为两门课程的好处是，不打算致力于DBMS研究的学生可以仅选修第一门课程，然后可以将数据库技术应用于他们所进入的计算机科学的任何分支。

选修要求

学生一般不会在大学的第一学年选修使用本书的课程，所以我们期望本书的读者具有计算机科学的传统领域中相当广泛的背景知识。我们假定读者已经学习过数据库程序设计，特别是SQL。读者最好了解关系代数，并且对于基本数据结构有一定程度的熟悉。同样地，关于文件系统和操作系统的知识也是很有帮助的。

习题

本书包括大量习题，几乎每一节都有习题。我们用惊叹号标记出难度较大的习题，或习题中难度较大的部分。对于特别难的习题，我们用两个惊叹号标记。

某些习题或习题中的部分用星号标记。对于这些习题，我们将努力通过本书的Web页面提供解答。这些解答向公众发布，读者可以用来进行自我测试。注意，在有些情况下，习题B要求你对习题A的解答进行修正或改编。如果习题A的某些部分有Web发布的解答，那么在Web页面上

也会有习题B的相应部分的解答。

WWW支持

本书的主页为

http://www-db.stanford.edu/~ullmam/dbsi.html

在主页上你可以找到标注星号的习题的解答、勘误表以及辅助材料。我们打算在每一次讲授CS245和其他数据库课程的相关部分时，将注释也提供到主页中，包括作业、考试和解答等。

H. G.-M
J.D.U.
J. W.
加州，斯坦福大学

目　录

作者简介
译者序
前言
第1章　DBMS实现概述 ……1
1.1　Megatron 2000数据库系统介绍 ……1
1.1.1　Megatron 2000实现细节 ……2
1.1.2　Megatron 2000如何执行查询 ……3
1.1.3　Megatron 2000有什么问题 ……4
1.2　数据库管理系统概述 ……4
1.2.1　数据定义语言命令 ……4
1.2.2　查询处理概述 ……5
1.2.3　主存缓冲区和缓冲区管理器 ……6
1.2.4　事务处理 ……6
1.2.5　查询处理器 ……7
1.3　本书梗概 ……8
1.3.1　预备知识 ……8
1.3.2　存储管理概述 ……8
1.3.3　查询处理概述 ……9
1.3.4　事务处理概述 ……9
1.3.5　信息集成概述 ……9
1.4　数据库模型和语言回顾 ……10
1.4.1　关系模型回顾 ……10
1.4.2　SQL回顾 ……10
1.4.3　关系的和面向对象的数据 ……12
1.5　小结 ……13
1.6　参考文献 ……14
第2章　数据存储 ……15
2.1　存储器层次 ……15
2.1.1　高速缓冲存储器 ……16
2.1.2　主存储器 ……16
2.1.3　虚拟存储器 ……17
2.1.4　第二级存储器 ……18
2.1.5　第三级存储器 ……19
2.1.6　易失和非易失存储器 ……20
习题 ……21
2.2　磁盘 ……21
2.2.1　磁盘结构 ……21
2.2.2　磁盘控制器 ……23
2.2.3　磁盘存储特性 ……24
2.2.4　磁盘访问特性 ……25
2.2.5　块的写入 ……28
2.2.6　块的修改 ……28
习题 ……28
2.3　第二级存储器的有效使用 ……29
2.3.1　计算的I/O模型 ……30
2.3.2　第二级存储器中的数据排序 ……30
2.3.3　归并排序 ……31
2.3.4　两阶段多路归并排序 ……32
2.3.5　扩展多路归并以排序更大的关系 ……34
习题 ……35
2.4　改善第二级存储器的访问时间 ……36
2.4.1　按柱面组织数据 ……37
2.4.2　使用多磁盘 ……38
2.4.3　磁盘镜像 ……39
2.4.4　磁盘调度和电梯算法 ……39
2.4.5　预取和大规模缓冲 ……42
2.4.6　各种策略及其优缺点 ……43
习题 ……44
2.5　磁盘故障 ……45
2.5.1　间断性故障 ……45
2.5.2　校验和 ……46
2.5.3　稳定存储 ……47
2.5.4　稳定存储的错误处理能力 ……47
习题 ……48

2.6 从磁盘崩溃中恢复 ……48
2.6.1 磁盘的故障模型 ……48
2.6.2 作为冗余技术的镜像 ……49
2.6.3 奇偶块 ……50
2.6.4 一种改进：RAID 5 ……52
2.6.5 多个盘崩溃时的处理 ……53
习题 ……55
2.7 小结 ……57
2.8 参考文献 ……58
第3章 数据元素的表示 ……60
3.1 数据元素和字段 ……60
3.1.1 关系型数据库元素的表示 ……60
3.1.2 对象的表示 ……61
3.1.3 数据元素的表示 ……62
3.2 记录 ……65
3.2.1 定长记录的构造 ……65
3.2.2 记录首部 ……67
3.2.3 定长记录在块中的放置 ……68
习题 ……68
3.3 块和记录地址的表示 ……69
3.3.1 客户机-服务器系统 ……69
3.3.2 逻辑地址和结构地址 ……70
3.3.3 指针混写 ……71
3.3.4 块返回磁盘 ……74
3.3.5 被固定的记录和块 ……74
习题 ……75
3.4 变长数据和记录 ……77
3.4.1 具有变长字段的记录 ……77
3.4.2 具有重复字段的记录 ……78
3.4.3 可变格式记录 ……79
3.4.4 不能装入一个块中的记录 ……80
3.4.5 BLOBS ……81
习题 ……82
3.5 记录的修改 ……83
3.5.1 插入 ……83
3.5.2 删除 ……84
3.5.3 更新 ……85
习题 ……85
3.6 小结 ……86
3.7 参考文献 ……87
第4章 索引结构 ……88
4.1 顺序文件上的索引 ……89
4.1.1 顺序文件 ……89
4.1.2 稠密索引 ……90
4.1.3 稀疏索引 ……92
4.1.4 多级索引 ……93
4.1.5 重复查找键的索引 ……94
4.1.6 数据修改期间的索引维护 ……97
习题 ……101
4.2 辅助索引 ……102
4.2.1 辅助索引的设计 ……103
4.2.2 辅助索引的应用 ……104
4.2.3 辅助索引中的间接 ……105
4.2.4 文档检索和倒排索引 ……107
习题 ……109
4.3 B树 ……111
4.3.1 B树的结构 ……111
4.3.2 B树的应用 ……114
4.3.3 B树中的查找 ……116
4.3.4 范围查询 ……116
4.3.5 B树的插入 ……117
4.3.6 B树的删除 ……119
4.3.7 B树的效率 ……122
习题 ……123
4.4 散列表 ……124
4.4.1 辅存散列表 ……124
4.4.2 散列表的插入 ……125
4.4.3 散列表的删除 ……126
4.4.4 散列表索引的效率 ……126
4.4.5 可扩展散列表 ……127
4.4.6 可扩展散列表的插入 ……127
4.4.7 线性散列表 ……129
4.4.8 线性散列表的插入 ……130
习题 ……132

4.5 小结 ……133
4.6 参考文献 ……134
第5章 多维索引 ……136
5.1 需要多维的应用 ……136
5.1.1 地理信息系统 ……137
5.1.2 数据立方体 ……138
5.1.3 SQL多维查询 ……138
5.1.4 使用传统索引执行范围查询 ……139
5.1.5 利用传统索引执行最邻近查询 ……140
5.1.6 传统索引的其他限制 ……141
5.1.7 多维索引结构综述 ……141
习题 ……142
5.2 多维数据的类散列结构 ……143
5.2.1 网格文件 ……143
5.2.2 网格文件的查找 ……144
5.2.3 网格文件的插入 ……145
5.2.4 网格文件的性能 ……146
5.2.5 分段散列函数 ……147
5.2.6 网格文件和分段散列的比较 ……148
习题 ……149
5.3 多维数据的类树结构 ……151
5.3.1 多键索引 ……151
5.3.2 多键索引的性能 ……152
5.3.3 kd树 ……153
5.3.4 kd树的操作 ……154
5.3.5 使kd树适合辅存 ……156
5.3.6 四叉树 ……157
5.3.7 R树 ……158
5.3.8 R树的操作 ……159
习题 ……161
5.4 位图索引 ……162
5.4.1 位图索引的诱因 ……163
5.4.2 压缩位图 ……164
5.4.3 游程长度编码位向量的操作 ……166
5.4.4 位图索引的管理 ……166
习题 ……168
5.5 小结 ……168
5.6 参考文献 ……169
第6章 查询执行 ……171
6.1 一种查询代数 ……172
6.1.1 并、交和差 ……173
6.1.2 选择操作符 ……174
6.1.3 投影操作符 ……175
6.1.4 关系的积 ……176
6.1.5 连接 ……177
6.1.6 消除重复 ……179
6.1.7 分组和聚集 ……179
6.1.8 排序操作符 ……181
6.1.9 表达式树 ……181
习题 ……183
6.2 物理查询计划操作符介绍 ……185
6.2.1 扫描表 ……186
6.2.2 扫描表时的排序 ……186
6.2.3 物理操作符计算模型 ……186
6.2.4 衡量代价的参数 ……187
6.2.5 扫描操作符的I/O代价 ……188
6.2.6 实现物理操作符的迭代器 ……188
6.3 数据库操作的一趟算法 ……191
6.3.1 一次多元组操作的一趟算法 ……192
6.3.2 全关系的一元操作的一趟算法 ……193
6.3.3 二元操作的一趟算法 ……195
习题 ……197
6.4 嵌套循环连接 ……198
6.4.1 基于元组的嵌套循环连接 ……198
6.4.2 基于元组的嵌套循环连接的迭代器 ……199
6.4.3 基于块的嵌套循环连接算法 ……200
6.4.4 嵌套循环连接的分析 ……201
6.4.5 迄今为止的算法小结 ……201
习题 ……202
6.5 基于排序的两趟算法 ……202
6.5.1 利用排序消除重复 ……202
6.5.2 利用排序进行分组和聚集 ……204
6.5.3 基于排序的并算法 ……204
6.5.4 基于排序的交和差算法 ……205

6.5.5 基于排序的一个简单的连接算法 …206
6.5.6 简单排序连接的分析 …208
6.5.7 一种更有效的基于排序的连接 …208
6.5.8 基于排序的算法小结 …209
习题 …209
6.6 基于散列的两趟算法 …211
6.6.1 通过散列划分关系 …211
6.6.2 基于散列的消除重复算法 …211
6.6.3 基于散列的分组和聚集算法 …212
6.6.4 基于散列的并、交、差算法 …212
6.6.5 散列连接算法 …213
6.6.6 节省一些磁盘I/O …213
6.6.7 基于散列的算法小结 …215
习题 …216
6.7 基于索引的算法 …216
6.7.1 聚簇和非聚簇索引 …217
6.7.2 基于索引的选择 …217
6.7.3 使用索引的连接 …219
6.7.4 使用有排序索引的连接 …220
习题 …221
6.8 缓冲区管理 …222
6.8.1 缓冲区管理结构 …222
6.8.2 缓冲区管理策略 …223
6.8.3 物理操作符选择和缓冲区管理的关系 …225
习题 …226
6.9 使用超过两趟的算法 …226
6.9.1 基于排序的多趟算法 …227
6.9.2 基于排序的多趟算法的性能 …227
6.9.3 基于散列的多趟算法 …228
6.9.4 基于散列的多趟算法的性能 …228
习题 …229
6.10 关系操作的并行算法 …229
6.10.1 并行模型 …229
6.10.2 一次一个元组的并行操作 …232
6.10.3 全关系操作的并行算法 …233
6.10.4 并行算法的性能 …233
习题 …235
6.11 小结 …235
6.12 参考文献 …237
第7章 查询编译器 …238
7.1 语法分析 …238
7.1.1 语法分析与语法分析树 …239
7.1.2 SQL的一个简单子集的语法 …239
7.1.3 预处理器 …243
习题 …243
7.2 用于改进查询计划的代数定律 …244
7.2.1 交换律与结合律 …244
7.2.2 涉及选择的定律 …246
7.2.3 下推选择 …248
7.2.4 涉及投影的定律 …249
7.2.5 有关连接与积的定律 …252
7.2.6 有关消除重复的定律 …252
7.2.7 涉及分组与聚集的定律 …253
习题 …254
7.3 从语法分析树到逻辑查询计划 …255
7.3.1 转换成关系代数 …255
7.3.2 从条件中去除子查询 …256
7.3.3 逻辑查询计划的改进 …261
7.3.4 结合/交换操作符的分组 …262
习题 …263
7.4 操作代价的估计 …264
7.4.1 中间关系大小的估计 …264
7.4.2 投影大小的估计 …265
7.4.3 选择大小的估计 …266
7.4.4 连接大小的估计 …268
7.4.5 多连接属性的自然连接 …269
7.4.6 多个关系的连接 …271
7.4.7 其他操作的大小估计 …272
习题 …273
7.5 基于代价的计划选择介绍 …274
7.5.1 大小参数估计值的获取 …275
7.5.2 统计量的增量计算 …277
7.5.3 减少逻辑查询计划代价的启发式 …278

需要全本请访问 www.ertongbook.com

7.5.4 枚举物理计划的方法 ……279
习题 ……281
7.6 连接顺序的选择 ……283
7.6.1 连接的左右变元的意义 ……283
7.6.2 连接树 ……283
7.6.3 左深连接树 ……284
7.6.4 通过动态编程来选择连接顺序和分组 ……286
7.6.5 带有更具体的代价函数的动态编程…289
7.6.6 选择连接顺序的贪婪算法 ……290
习题 ……291
7.7 物理查询计划选择的完成 ……292
7.7.1 选取选择方法 ……292
7.7.2 选取连接方法 ……294
7.7.3 流水线操作与物化 ……294
7.7.4 一元流水线操作 ……295
7.7.5 二元流水线操作 ……296
7.7.6 物理查询计划的符号 ……298
7.7.7 物理操作的顺序 ……300
习题 ……300
7.8 小结 ……301
7.9 参考文献 ……302
第8章 系统故障对策 ……304
8.1 可回复操作的问题和模型 ……304
8.1.1 故障模式 ……304
8.1.2 关于事务的进一步讨论 ……306
8.1.3 事务的正确执行 ……307
8.1.4 事务的原语操作 ……308
习题 ……310
8.2 undo日志 ……310
8.2.1 日志记录 ……311
8.2.2 undo日志规则 ……312
8.2.3 使用undo日志的恢复 ……314
8.2.4 检查点 ……315
8.2.5 非静止检查点 ……316
习题 ……318
8.3 redo日志 ……319
8.3.1 redo日志规则 ……320
8.3.2 使用redo日志的恢复 ……320
8.3.3 redo日志的检查点 ……321
8.3.4 使用带检查点的redo日志的恢复 …322
习题 ……323
8.4 undo/redo日志 ……323
8.4.1 undo/redo规则 ……323
8.4.2 使用undo/redo日志的恢复 ……324
8.4.3 undo/redo日志的检查点 ……325
习题 ……326
8.5 防备介质故障 ……327
8.5.1 备份 ……327
8.5.2 非静止转储 ……328
8.5.3 使用备份和日志的恢复 ……329
习题 ……330
8.6 小结 ……330
8.7 参考文献 ……331
第9章 并发控制 ……333
9.1 串行调度和可串行化调度 ……333
9.1.1 调度 ……333
9.1.2 串行调度 ……334
9.1.3 可串行化调度 ……335
9.1.4 事务语义的影响 ……335
9.1.5 事务和调度的一种记法 ……336
习题 ……337
9.2 冲突可串行性 ……337
9.2.1 冲突 ……337
9.2.2 优先图及冲突可串行性判断 ……339
9.2.3 优先图测试发挥作用的原因 ……341
习题 ……341
9.3 使用锁的可串行性实现 ……343
9.3.1 锁 ……343
9.3.2 封锁调度器 ……345
9.3.3 两阶段封锁 ……346
9.3.4 两阶段封锁发挥作用的原因 ……346
习题 ……347
9.4 用多种锁方式的封锁系统 ……349

9.4.1 共享锁与排他锁 ……349
9.4.2 相容性矩阵 ……350
9.4.3 锁的升级 ……351
9.4.4 更新锁 ……352
9.4.5 增量锁 ……353
习题 ……354
9.5 封锁调度器的一种体系结构 ……356
9.5.1 插入锁动作的调度器 ……356
9.5.2 锁表 ……358
习题 ……360
9.6 数据库元素层次的管理 ……360
9.6.1 多粒度的锁 ……360
9.6.2 警示锁 ……361
9.6.3 幻像与插入的正确处理 ……363
习题 ……364
9.7 树协议 ……364
9.7.1 基于树的封锁的动机 ……365
9.7.2 访问树结构数据的规则 ……365
9.7.3 树协议发挥作用的原因 ……366
习题 ……368
9.8 使用时间戳的并发控制 ……369
9.8.1 时间戳 ……369
9.8.2 物理上不可实现的行为 ……369
9.8.3 脏数据的问题 ……370
9.8.4 基于时间戳调度的规则 ……371
9.8.5 多版本时间戳 ……373
9.8.6 时间戳与封锁 ……374
习题 ……374
9.9 使用有效性确认的并发控制 ……375
9.9.1 基于有效性确认的调度器的结构 …375
9.9.2 有效性确认规则 ……376
9.9.3 三种并发控制机制的比较 ……378
习题 ……379
9.10 小结 ……379
9.11 参考文献 ……380
第10章 再论事务管理 ……382
10.1 读未提交数据的事务 ……382
10.1.1 脏数据问题 ……382
10.1.2 级联回滚 ……384
10.1.3 回滚的管理 ……384
10.1.4 成组提交 ……385
10.1.5 逻辑日志 ……387
习题 ……388
10.2 视图可串行性 ……389
10.2.1 视图等价性 ……389
10.2.2 多重图与视图可串行性的判断 ……390
10.2.3 视图可串行性的判断 ……393
习题 ……393
10.3 死锁处理 ……394
10.3.1 超时死锁检测 ……394
10.3.2 等待图 ……394
10.3.3 通过元素排序预防死锁 ……396
10.3.4 时间戳死锁检测 ……397
10.3.5 死锁管理方法的比较 ……399
习题 ……400
10.4 分布式数据库 ……401
10.4.1 数据的分布 ……401
10.4.2 分布式事务 ……402
10.4.3 数据复制 ……402
10.4.4 分布式查询优化 ……403
习题 ……403
10.5 分布式提交 ……404
10.5.1 分布式原子性的支持 ……404
10.5.2 两阶段提交 ……404
10.5.3 分布式事务的恢复 ……406
习题 ……407
10.6 分布式封锁 ……408
10.6.1 集中封锁系统 ……408
10.6.2 分布式封锁算法的代价模型 ……409
10.6.3 封锁多副本的元素 ……410
10.6.4 主副本封锁 ……410
10.6.5 局部锁构成的全局锁 ……410
习题 ……411
10.7 长事务 ……412

10.7.1 长事务的问题 ……412
10.7.2 saga（系列记载） ……414
10.7.3 补偿事务 ……414
10.7.4 补偿事务发挥作用的原因 ……416
习题 ……416
10.8 小结 ……417
10.9 参考文献 ……418
第11章 信息集成 ……420
11.1 信息集成的方式 ……420
11.1.1 信息集成的问题 ……420
11.1.2 联邦数据库系统 ……421
11.1.3 数据仓库 ……423
11.1.4 Mediator ……425
习题 ……426
11.2 基于Mediator系统的包装器 ……427
11.2.1 查询模式的模板 ……428
11.2.2 包装器生成器 ……429
11.2.3 过滤器 ……429
11.2.4 其他在包装器上进行的操作 ……430
习题 ……431
11.3 联机分析处理 ……432
11.3.1 OLAP应用 ……433
11.3.2 OLAP数据的多维视图 ……433
11.3.3 星型模式 ……434
11.3.4 切片和切块 ……436
习题 ……437
11.4 数据立方体 ……438
11.4.1 立方体操作符 ……438
11.4.2 通过物化视图实现立方体 ……441
11.4.3 视图的格 ……443
习题 ……444
11.5 数据挖掘 ……445
11.5.1 数据挖掘的应用 ……446
11.5.2 关联规则的挖掘 ……447
11.5.3 A-Priori算法 ……449
11.6 小结 ……450
11.7 参考文献 ……451
索引 ……453

第1章 DBMS实现概述

数据库对于当今的任何部门都是至关重要的。数据库被用来保存内部记录，在WWW中向顾客和客户展示数据，以及支持许多其他的商务处理。同样地，在许多科学研究的核心中也需要数据库。数据库被用来表示天文学家、基因组研究人员、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。

数据库的能力是由于知识和技术的结合，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，或更口语化地称为“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。这些系统是当前最复杂的软件类型之一。DBMS为用户提供的能力包括：

1) *持久存储*。与文件系统类似，DBMS支持对非常大量的数据进行存储，这些数据独立于使用数据的任何处理程序而存在。然而，DBMS在提供灵活性方面比文件系统做得更好，例如，它提供支持对非常大量的数据进行高效存取的数据结构。
2) *编程接口*。DBMS使得用户能够通过强有力的查询语言访问和修改数据。DBMS在数据操纵的灵活性方面也比文件系统强。与文件读写相比，DBMS提供的对存储的数据进行操作的方式要复杂得多。
3) *事务管理*。DBMS支持对数据的并发存取，即多个不同的进程（称作“事务”）同时对数据进行存取。为避免同时访问所造成的不良后果，DBMS支持*隔离*，即看起来事务是一次一个地在执行，以及*原子性*，即要求事务或者完全执行；或者完全不执行。DBMS还支持*回复性*(resilience)，即能够从多种类型的故障或错误中恢复的能力。

核心技术回顾

本书的读者应已学习过Ullman和Widom的《数据库系统入门教程》这一层次的数据库系统方面的内容，并了解SQL编程方面的知识。他们应该熟悉下列术语：

- *数据*：值得保留的任何信息，一般来说是电子形式的。
- *数据库*：为了访问和修改而组织的、在长时期内保留的数据的集合。
- *查询*：从数据库中抽取特定数据的操作。
- *关系*：将数据组织到二维表中的组织方式，表中的行（元组）表示基本的实体或某种事实，表中的列（属性）表示实体的特性。
- *模式*：数据库中数据结构的描述，通常称作“元数据”。

1.1 Megatron 2000数据库系统介绍

如果你使用过某个DBMS，或许是一个支持常见的SQL查询语言的DBMS，你可以想象实现这样

的一个系统并不困难。你心里可能会想到Megatron系统公司最近（虚构）提供的一个实现：Megatron 2000数据库管理系统。该系统在UNIX和其他操作系统上运行，采用关系方法，支持SQL语言。

1.1.1 Megatron 2000实现细节

首先，Megatron 2000 采用文件系统来存储它的关系。例如，关系`students(name, id, dept)`会存储在文件`/usr/db/students`中。对于每个元组，在文件`Students`中有一行。一个元组中各个分量的值存储成由特殊的标记字符`#`互相分开的字符串。例如，文件`/usr/db/students`可能像下面这样：

```
Smith#123#CS
Johnson#522#EE
    .
```

数据库模式存储在特定的文件`/usr/db/schema`中。对于每一个关系，文件`schema`中有一个以该关系的名字起始的行，在该行中，属性名和类型交替出现。字符`#`作为行中元素间的分隔符。例如，文件`schema`中可能包含如下的行

```
Students#name#STR#id#INT#dept#STR
Depts#name#STR#office#STR
```

在这里对关系`Students(name, id, dept)`进行了描述；属性`name`和`dept`的类型是字符串，属性`id`的类型是整数。同时还有对模式为`Depts(name, office)`的关系的描述。

例1.1　下面是使用Megatron 2000 DBMS的一个对话的例子。我们运行在一台称作`dbhost`的机器上，通过UNIX层次的命令`megatron 2000`启动DBMS。

```
dbhost> megatron2000
```

产生如下响应

```
WELCOME TO MEGATRON 2000!
```

现在我们与Megatron 2000的用户界面对话，对于Megatron 提示符（`&`），我们可以键入SQL查询[⊖]作为响应。用`#`结束查询。例如，

```
& SELECT *
  FROM Students #
```

产生下表作为回答

name	*id*	*dept*
Smith	123	CS
Johnson	522	EE

Megatron 2000还允许我们执行一个查询，然后把结果存在一个新的文件中，做法是用一条竖线

⊖ 在1.4.2节中对于SQL有一个简单的回顾。

和文件名来结束查询。例如.

```
& SELECT *
  FROM Students
  WHERE id >= 500 | HighId #
```

创建一个新的文件/usr/db/HighId，文件中只有一行

```
Johnson #522 #EE
```

□

1.1.2 Megatron 2000如何执行查询

让我们考虑SQL查询的一般形式

```
SELECT * FROM R WHERE <Condition>
```

Megatron 2000将做下列事情·

1) 读文件schema，以确定关系*R*中有哪些属性，以及它们的类型。

2) 检查<条件>对于关系*R*的语义合法性。

3) 显示每一个属性的名字作为列的头，并画一条线。

4) 读名字为*R*的文件，对于每一行.

（a）检查是否符合条件，

（b）若符合条件，则显示该行为一个元组。

要执行

```
SELECT * FROM R WHERE <condition> | T
```

Megatron 2000做以下事情：

1) 如上所述处理查询，但省略第3步，该步骤是产生列的头并画一条线来分隔开头和元组。

2) 将结果写到一个新文件/usr/db/T中。

3) 往文件/usr/db/schema中添加一个表示*T*的新的条目，该条目和关于*R*的条目一样，只是关系名为*T*，而不是*R*。也就是说，关系*T*的模式与关系*R*的模式相同。

例1.2 现在让我们考虑一个更复杂的查询，该查询涉及我们的两个样例关系Students和Depts的连接：

```
SELECT office
FROM Students, Depts
WHERE Students name = 'Smith' AND
      Students dept = Depts.name #
```

这个查询要求Megatron 2000对关系Students和Depts进行“连接”。即系统必须逐个考虑从两个关系中各取一个元组所组成的每一对元组，并检验是否满足下列条件：

a）这两个元组表示相同的系；

b）学生的名字是Smith。

该算法可以非形式化地描述如下：

```
for(each tuple s in Students)
    for(each tuple d in Depts)
```

需要全本请访问 www.ertongbook.com