

大学本科教材·计算机教学丛书

数据库系统基础

姜翠霞 编著



北京航空航天大学出版社

大学本科教材·计算机教学丛书



数据库系统基础

姜翠霞 编者

北京航空航天大学出版社

内 容 简 介

本书系统地阐述数据库系统的基础理论、基本技术和基本方法。全书共 11 章。前 10 章主要内容包括数据库的基本概念、数据模型、关系数据库、数据库查询语言、关系数据理论、关系数据库设计、事务处理、数据库保护、分布式数据库、物理数据库设计等。第 11 章包括 SQL Server 的上机环境介绍以及学习本课程可以选做的 7 个实验。各章附有习题。知识结构科学,理论体系完整,注重实践能力的培养,实用性强。既适合高等学校计算机及相关专业教学使用,也可供自学者参考。

图书在版编目(CIP)数据

数据库系统基础/姜翠霞编著. —北京:北京航空航天大学出版社,2009.5

ISBN 978-7-81124-760-2

I. 数… II. 姜… III. 数据库系统 IV. TP311.13

中国版本图书馆 CIP 数据核字(2009)第 060309 号

数据库系统基础

姜翠霞 编著

责任编辑 幼 章

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail: bhpresse@263.net

北京时代华都印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:15 字数:384 千字

2009 年 5 月第 1 版 2009 年 5 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-760-2 定价:24.00 元

作者备有本书课件,需要者可与本社发行部,或作者联系。



总 前 言

随着科学技术、文化、教育、经济和社会的发展,计算机教学进入了我国历史上最火热的年代,欣欣向荣。就计算机专业而言,全国开办计算机本科专业的院校在2004年之初505所,到2006年已经发展到771所。另外,在全国高校中的非计算机专业,包括理工农医以及文科(文史哲法教、经管、文艺)等专业,按各自专业的培养目标都融入了计算机课程的教学。过去出版界出版了一大批计算机教学方面的各类教材,满足了一定时期的需求,但是还不能完全适应计算机教学深化改革的要求。

面对《国家科学技术中长期发展纲要(2006年—2020年)》制订的信息技术发展目标,计算机教学也要随之进行改革,以便提高培养质量。教学要改革,教材建设必须跟上。面对各层次、各类型的学校和各类型的专业都要开设计算机课程,就应有多样化的教材,以适应各专业教学的需要。北京航空航天大学出版社是以出版高等教育教材为主的,愿对计算机教学的教材建设做出贡献。

为计算机类教材的出版,北京航空航天大学出版社成立了“大学本科教材·计算机教学丛书”编审委员会。出版计算机教材,得到了北京航空航天大学计算机学院的大力支持。该院有三位教育部高等学校计算机科学与技术教学指导委员会(下称教指委)的成员参加编审委员会的工作。其他成员是北京航空航天大学、北京交通大学等6所院校和中科院计算技术研究所对计算机教育有研究的教指委成员、专家、学者和出版社的领导。

我们组织编写、出版计算机课程教材,以大多数高校实际状况为基点,使其在现有基础上能提高一步,追求符合大多数高校本科教学适用为目标。按照教指委制订的计算机科学与技术本科专业规范和计算机基础课教学基本要求的精神,我们组织身居教学第一线,具有教学实践经验的教师进行编写。在出书品种和内容上,面对两个方面的教学:一是计算机专业本科教学,包括计算机导论、计算机专业技术基础课、计算机专业课等;二是非计算机专业的计算机基础课程的本科教学,包括理工农医类、文史哲法教类、经管类、艺术类等的计算机课程。

教材的编写注重以下几点:

1. 基础性。具有基础知识和基本理论,以使学生在专业发展上具有潜力,便于适应社会的需求。
2. 先进性。融入计算机科学与技术发展的新成果;瞄准计算机科学与技术发展的新方向,内容应具有前瞻性。这样,以使学生扩展视野,以便与科技、社会发展的脉络同步。
3. 实用性。一是适应教学的需求;二是理论与实践相结合,以使学生掌握实用技术。

编写、出版的教材能否适应教学改革的需求,只有师生在教与学的实践中做出评价,我们期望得到师生的批评和指正。

“大学本科教材·计算机教学丛书”

编审委员会

2008年1月

“大学本科教材·计算机教学丛书”
编审委员会成员

主 任 马殿富

副主任 麦中凡

陈炳和

委 员（以音序排列）

陈炳和 邓文新 金茂忠

刘建宾 刘明亮 罗四维

卢湘鸿 马殿富 麦中凡

乔少杰 谢建勋 熊 璋

张 莉



前言

计算机科学与技术的发展异常迅猛,几乎无处不在、无处不用。教育部高等教育司为适应计算机发展的形势,组织计算机专业以及理工类专业、文科类专业的计算机教学指导委员会,对各类型的计算机教学提出了基本要求,并制订了规范(2006年先后公布)。数据库技术的教学在各种类型的计算机教学中都占有重要的地位。作者在多年的教学实践和编写教材的基础上,按照高等学校计算机教学指导委员会制订的专业规范,编写了这本《数据库系统基础》教材,旨在满足本科生,或研究生的教学需要。

全书共11章。第1章至第4章介绍数据库系统的基本概念、数据模型和数据库语言。第5章和第6章介绍关系数据理论和关系数据库设计方法。第7章和第8章介绍数据库保护方面的知识,包括数据库恢复、并发控制、数据库安全性和数据库完整性。第9章第10章介绍了分布式数据库系统和数据库的存储结构。第11章介绍了本课程的配套实验课的内容,共7个实验,可以选择其中的5个实验来做。本书的参考学时数为40~60学时。实验参考学时数为10~20学时。本书除供给课堂教学使用,还便于自学,也可供广大科技人员学习、参考。

在本书编写过程中,本校教师李大辉、韩金库、赵鑫和潘海珠阅读了部分初稿为本书提出了许多宝贵意见,在此对他们表示诚挚的谢意。许多同行和同事或提供资料,或给予支持和鼓励,对他们也表示感谢。

由于编者水平有限,书中难免存在一些缺点和错误,殷切希望广大读者批评指正。

作者

于2008年夏

E-mail: cypf2004@yeah.net



目 录

第1章 绪 论	1
1.1 数据库系统的产生和发展	1
1.1.1 信息、数据与数据处理	1
1.1.2 数据管理的发展	2
1.2 数据视图与数据模型	9
1.2.1 数据抽象	9
1.2.2 数据模型	10
1.3 数据库语言	11
1.3.1 数据定义语言	11
1.3.2 数据操纵语言	12
1.3.3 数据控制语言	12
1.4 数据库的体系结构	13
1.4.1 模式的概念	13
1.4.2 数据库系统的三级模式结构	13
1.4.3 逻辑模式	14
1.4.4 外模式	14
1.4.5 内模式	15
1.4.6 外模式/模式映像	15
1.4.7 模式/内模式映像	15
1.5 数据库系统的组成	15
1.6 数据库系统的全局结构	17
1.6.1 查询处理器	18
1.6.2 存储管理器	19
1.6.3 磁盘存储器中的数据结构	19
1.7 小 结	19
习题/思考题	20
第2章 数据模型	21
2.1 数据模型的组成要素	21
2.2 概念数据模型	21
2.2.1 概念数据模型的基本术语	21
2.2.2 实体联系模型	23
2.3 层次数据模型	24
2.3.1 层次数据模型的数据结构	25
2.3.2 层次模型的数据操纵与完整性约束	25
2.3.3 层次模型的优缺点	25
2.4 网状数据模型	26



2.4.1 网状数据模型的数据结构	26
2.4.2 网状数据模型的操纵与完整性约束	26
2.4.3 网状模型的优缺点	27
2.5 关系数据模型	27
2.6 对象模型	28
习题/思考题	29
第3章 关系数据库	30
3.1 关系模型	30
3.1.1 基本概念	30
3.1.2 关系形式化定义	31
3.1.3 关系的完整性	34
3.2 关系代数	35
3.2.1 传统的集合运算	36
3.2.2 专门的关系运算	38
3.2.3 关系代数查询实例	42
3.3 关系演算	43
3.3.1 元组关系演算语言 ALPHA	44
3.3.2 域关系演算语言 QBE	48
习题/思考题	53
第4章 关系数据库语言 SQL	55
4.1 SQL 概述	55
4.1.1 SQL 的产生与发展	55
4.1.2 SQL 的特点	55
4.1.3 SQL 数据库的体系结构	57
4.2 SQL 的数据定义	58
4.2.1 SQL 模式的创建和撤销	58
4.2.2 基本数据类型	58
4.2.3 基本表的创建、修改和撤销	59
4.2.4 索引的创建和撤销	61
4.3 SQL 的数据查询	62
4.3.1 SELECT 查询语句	62
4.3.2 SELECT 语句用法	63
4.4 SQL 的数据更新	73
4.4.1 数据插入	73
4.4.2 数据删除	74
4.4.3 数据修改	75
4.5 视图	76
4.5.1 定义视图	76
4.5.2 查询视图	78
4.5.3 更新视图	78
4.5.4 视图的作用	79



习题/思考题	80
第 5 章 关系数据理论	81
5.1 冗余导致的问题	81
5.2 规范化	81
5.2.1 函数依赖	82
5.2.2 码	82
5.2.3 范 式	83
5.2.4 2NF	84
5.2.5 3NF	84
5.2.6 BCNF	85
5.2.7 多值依赖	85
5.2.8 4NF	87
5.2.9 连接依赖	87
5.2.10 第五范式	87
5.3 数据依赖的公理系统	88
5.4 模式分解	91
5.4.1 无损连接分解	91
5.4.2 保持依赖分解	91
习题/思考题	92
第 6 章 关系数据库设计	94
6.1 数据库设计概述	94
6.1.1 数据库设计问题	94
6.1.2 数据库设计方法和步骤	94
6.2 实体-联系模型	96
6.2.1 基本属性与组合属性	96
6.2.2 存在依赖性(existence - dependent)	96
6.2.3 弱实体(weak entity)	97
6.2.4 单值属性与多值属性	97
6.2.5 递归实体	98
6.2.6 实体的超类与子类	99
6.3 E-R 模型向关系模型的转换	99
6.4 E-R 模型设计实例	103
6.4.1 概念结构设计(设计 E-R 模型)	104
6.4.2 逻辑结构设计	105
6.4.3 数据库的实施	108
6.4.4 在数据库设计中应注意的问题	110
习题/思考题	111
第 7 章 数据库的恢复	112
7.1 事 务	112
7.1.1 事务的概念	112
7.1.2 事务的操作	112



7.1.3 事务的状态	113
7.1.4 事务的性质	114
7.2 SQL 的事务管理	115
7.2.1 SQL 的事务执行	115
7.2.2 可能破坏事务特性的因素	115
7.3 数据库恢复及故障	116
7.3.1 事务故障的种类	116
7.3.2 存储设备分类	117
7.4 数据库恢复的原理	117
7.4.1 数据转储	117
7.4.2 登记日志文件	118
7.5 数据库恢复技术	119
7.5.1 基于故障类型的恢复	119
7.5.2 具有检查点的恢复技术	119
7.5.3 数据库镜像	119
7.5.4 永久存储器的恢复	120
7.5.5 影子页面技术	120
7.6 SQL Server 的备份策略与方案	121
7.7 并发控制	122
7.7.1 并发控制引起的问题	122
7.7.2 封锁	125
7.7.3 封锁协议	125
7.7.4 活锁与死锁	126
7.7.5 并发调度的可串行性	128
7.7.6 协议	129
7.7.7 封锁粒度	129
7.7.8 插入和删除操作	130
习题/思考题	132
第8章 数据库管理	133
8.1 引言	133
8.2 数据库的完整性	133
8.2.1 数据库完整性分类	134
8.2.2 完整性约束条件与 DBMS 的完整性控制	135
8.2.3 SQL Server 的数据完整性	136
8.2.4 存储过程	140
8.2.5 触发器	141
8.3 数据库的安全性	147
8.3.1 定义视图	147
8.3.2 访问控制	148
8.3.3 数据加密	150
8.3.4 数据库审计	150



8.3.5 SQL Server 的安全性机制	150
习题/思考题	164
第9章 分布式数据库	165
9.1 分布式数据库系统的一般概念	165
9.1.1 分布式数据库	165
9.1.2 分布式数据库管理系统	166
9.1.3 分布式数据库管理系统的分类	167
9.1.4 分布式数据库管理系统的基本特点	167
9.2 分布式数据库设计	168
9.2.1 数据的分割	169
9.2.2 数据的分布	170
9.2.3 数据的冗余	171
9.2.4 设计方法	171
9.3 分布式事务	172
9.3.1 网络的可恢复性	172
9.3.2 局部数据项和全局数据项	172
9.3.3 全局事务、局部子事务和可串行性	172
9.4 分布式加锁	173
9.4.1 分布式加锁方法	173
9.4.2 分布式两阶段加锁协议	176
9.5 分布式事务的交付	177
9.5.1 事务阻塞	178
9.5.2 两阶段交付协议	178
9.5.3 两阶段交付中的事务恢复	179
9.6 基于时印的分布式并发控制	180
9.6.1 分布式时印	180
9.6.2 并发存取数据的方法	180
9.6.3 锁和时印的比较	181
9.7 分布式数据库的安全	181
9.7.1 身份验证	181
9.7.2 保密通信	182
9.7.3 访问控制	182
9.7.4 库文加密	182
9.7.5 密码体制与密码管理	183
习题/思考题	183
第10章 数据库的存储结构	185
10.1 物理存储介质概述	185
10.2 磁 盘	186
10.2.1 磁盘的物理特性	187
10.2.2 磁盘性能的度量	188
10.2.3 磁盘块存取的优化	188



10.3 RAID	190
10.3.1 通过冗余提高可靠性	190
10.3.2 通过并行提高性能	191
10.4 第三级存储	192
10.4.1 光 盘	192
10.4.2 磁 带	193
10.5 存储访问	193
10.5.1 缓冲区管理器	194
10.5.2 缓冲区替换策略	194
10.6 文件组织	195
10.6.1 定长记录	195
10.6.2 变长记录	196
10.7 文件中记录的组织	198
10.7.1 顺序文件组织	198
10.7.2 聚集文件组织	199
10.8 数据字典	199
10.9 索引技术	200
10.9.1 索引技术的分类	200
10.9.2 有序索引的分类	200
10.9.3 主索引	201
10.9.4 辅助索引	202
10.10 散列技术	203
10.10.1 散列机制	203
10.10.2 散列索引	204
10.10.3 静态散列中的问题	205
10.10.4 可扩充散列结构	205
习题/思考题	206
第 11 章 实验与指导	208
实验一 SQL Server 的安装及管理工具的使用	208
实验二 创建数据库和表	215
实验三 表数据的插入、修改和删除	217
实验四 数据库的查询	219
实验五 存储过程和触发器的使用	221
实验六 视图的定义、使用实验	222
实验七 数据库设计	224
附录 1	226
附录 2	227
参考文献	228



第1章

绪论

数据库是数据管理的最新技术。数据库技术在信息系统的研究中一直是非常重要的主题。从小型单项事务处理系统到大型信息系统,从一般企业管理系统到计算机辅助设计与制造(CAD/CAM)、办公信息系统等,越来越多新的应用领域采用数据库存储和处理它们的信息资源。特别是 Internet 技术飞速的发展,使得数据库技术知识成为最热门的职业途径之一。数据库技术能使 Internet 应用超越具有早期应用特点。

很多同学发现这个课题虽然可能具有挑战性,但它非常有趣。数据库的设计和开发既包括艺术,又包括工程。了解用户需求,然后,把它们转变成有效的数据库设计是一个艺术过程。把设计转变成实际的数据库,并且这些数据库带有功能完备、高性能的应用,是一个工程过程。两方面都充满着既有挑战性,又是非常有趣的智力难题。

数据库技术的需求量非常大,这门课中所学到的技能和知识将是非常必要的。本书的目标是提供牢固的数据库技术基础。如果选择从事数据库领域的职业,那么,本书会有助于读者的成功。



1.1 数据库系统的产生和发展

1.1.1 信息、数据与数据处理

1. 信息

信息是指现实世界事物的存在方式或运动状态的反映。例如,一支铅笔,颜色是绿的,形状是圆柱形的,这些都是关于铅笔的信息,是铅笔存在状态的反映。具体地说,信息是一种已经被加工为特定形式的数据。这种数据形式对接收者来说是有意义的,而且对当前和将来的决策具有明显的或实际的价值。在信息社会中,信息是一种资源,其重要性可以与物质和能量相提并论,是企业赖以生存和发展所必需的。

2. 数据

几乎与信息同样广泛使用的另一个概念是“数据”。所谓数据,是将现实世界中的各种信息记录下的、可以识别的符号,是信息的载体,是信息的具体表示形式。例如,描述铅笔的信息,用一组数据“绿色、圆柱形”表示。由于这些符号已被人们赋予了特定的语义,因此它们就具有传递信息的功能。数据可用多种不同的形式来表示一种同样的信息,信息不随它的数据形式不同而改变。数据的表现形式多种多样,不仅有熟知的数字和文字,还可以有图形、图像、声音等多种表现形式。它们都可以经过数字化后存入计算机。

3. 信息与数据的联系

- 数据是信息的符号表示,或称为载体;
- 信息是数据的内涵,是对数据语义的解释;



- 数据是符号化的信息；
- 信息是语义化的数据。

4. 数据处理

数据处理是指对数据进行一系列收集、加工、存储、合并、分类、计算、检索、传输等操作过程。在当今的信息社会,所说的信息处理实际上就是利用计算机进行数据处理的过程。该过程包括:数据的采集、整理、编码和输入,有效地把数据组织到计算机中,由计算机系统对数据进行一系列的加工、存储、合并、分类、计算、检索、传输、输出等操作过程。所以信息处理也称为数据处理。数据管理则是指对数据进行分类、组织、编码、存储、检索和维护,它是数据处理的中心问题。

最初研制计算机是为了用它进行科学计算,在应用需求的推动下,随着计算机技术的发展,人们开始使用计算机进行数据处理。

人们把用计算机对数据进行处理的应用系统称为计算机信息系统。

1.1.2 数据管理的发展

数据管理技术的发展,与计算机硬件(主要是外部存储器)、软件及计算机应用的范围有着密切的联系。数据管理技术的发展大致经过以下4个阶段:人工管理阶段、文件系统阶段、数据库系统阶段和高级数据库系统阶段。

1. 人工管理阶段

20世纪50年代中期以前,计算机主要用于科学计算,其他工作还没有展开。外部存储器只有磁带、卡片和纸带等,没有磁盘。软件只有汇编语言,没有操作系统,没有相应的软件系统负责管理数据。数据需要由应用程序自己管理。这个时期的数据管理有以下特点:

- 1) 数据不保存。计算机主要用于计算,一般情况下,数据不需要保存。
- 2) 没有专用的软件对数据进行管理。每个应用程序都要包括存储结构、存取方法、输入输出方式等内容。程序中的存取子程序随着存储结构的改变而改变,因而数据与程序不具有独立性。
- 3) 只有程序的概念,没有文件的概念。数据的组织方式必须由程序员自己设计与安排。
- 4) 数据面向应用,即一组数据对应一个程序。

2. 文件系统阶段

20世纪50年代后期到60年代中期,计算机不仅用于科学计算,还用于数据管理。此时,出现了磁盘、磁鼓等外部存储器。出现了操作系统和高级语言。操作系统中的文件系统是专门管理外存的数据管理软件。

早期的数据管理都采用文件系统(File System,FS)。在文件系统中,数据按其内容、结构和用途组成若干命名的文件。文件一般为某一用户或用户组所有,但可供指定的其他用户共享。用户可以通过操作系统(Operating System,OS)对文件进行打开、读、写和关闭等操作。这个时期的数据管理有以下特点。

- 1) 数据可长期保存在外部存储器上。由于计算机的应用转向信息管理,因此需要以文件的方式保存大量数据,以便对文件中的数据进行大量的查询、修改和插入等操作。
- 2) 程序和数据之间有了一定的独立性。程序只须用文件名就可与数据打交道,不必关心数据的物理位置,由操作系统的文件系统提供存取方法(读/写)。



3) 文件组织多样化。有索引文件、链接文件和直接存取文件等。但文件之间相互独立,缺乏联系。数据之间的联系要通过程序去构造。

4) 数据不再属于某个特定的程序,可以重复使用。文件结构的设计仍然是基于特定的用途,程序是基于特定的物理结构和存取方法,因此程序与数据结构之间的依赖关系并未根本改变。

下面以一个图书馆的图书管理系统为例,来说明利用“操作系统”的文件系统开发简单的数据管理应用的过程。

基于文件系统的简单数据管理应用开发过程如下:

1) 开发任务 简单图书管理系统包括新书入库、读者增删、图书出借、催还通知等模块,分别完成图书信息、读者信息、出借信息的录入、修改、删除及查询等管理。

2) 开发工具及环境 C/C++ 程序设计语言、Windows 操作系统的文件系统。

3) 系统分析及设计 根据开发任务指定的模块,需要管理的“数据”(data)包括:图书信息、读者信息、出借信息。C/C++ 语言中,有一个“结构体”数据类型,可用来存放相关数据,可以在程序中定义 3 个结构体类型的数据结构来存放图书信息、读者信息和出借信息。

4) 系统实现过程

- 定义数据结构:在程序中,定义 3 个数据结构,以存储图书管理中所要用到或产生的数据。
- 构造链表:图书管理中会涉及大量图书信息、读者信息和出借信息。如果采用数组存放数据,必须事先定义固定的长度。如果事先难以确定数组的长度,则必须把数组定得足够大,显然这会浪费内存空间。链表没有这种缺点,它根据需要开辟内存单元。所以,需要在程序中构造 3 个链表,分别是图书信息链表、读者信息链表和出借信息链表。有了链表后,可根据管理操作(增加、删除、修改和查询),分别对链表进行节点的插入(insert)、删除(delete)、修改(update)和查询(query)。为加快检索速度,还可以对链表进行“排序”(sort)。
- 设计用户操作界面:为方便用户的数据操作或操纵,可为用户设计,并提供操作“界面”(interface)。
- 创建数据存储文件:为保证在下次进入管理软件时数据依然可用,必须将各链表中的数据以文件系统中的文件形式存放在磁盘上。文件的打开、读、写和关闭等操作,可利用 C/C++ 语言中的文件 I/O 操作函数完成。

根据系统任务要求,可分别创建 3 个数据文件,即图书信息文件、读者信息文件和出借信息文件。

除第一次运行外,程序每次运行时,均需先将数据文件中的数据读出,放入程序中的对应链表中,以方便数据的操作;程序退出前,则应将链表中的数据分别写入对应的数据文件,以保存并留待下次使用。

至此,一个简单的图书管理系统即告完成。这个系统可以完成简单的图书管理任务。

这样的管理应用系统明显地具有下列缺点:

(1) 大容量数据的处理与存储

前面所开发的图书管理系统只能处理和存储少量数据,或只能应用于数据量少的场合。然而,当计算机数据管理向其他具有巨数据量(如 GB 级、TB 级或 PB 级)的领域(如证券、银

行、航空等)延伸时,基于文件系统的应用系统就需要从以下几方面进行相应修改。

注: 1 KB(kilobyte) = 1024 Byte; 1 MB(megabyte) = 1024 KB; 1 GB(gigabyte) = 1024 MB; 1 TB(terabyte) = 1024 GB; 1 PB(petabyte) = 1024 TB。

1) 内存不够。前面示例中,在进行数据处理时,一般先将数据文件中的所有数据读到数据链表中,这对于小数据量的情形是可以的。但是,对巨数据量场合,如 500 GB 或几个 TB,不能一次将所有数据都读入内存。于是,就需要对前面的程序进行改造,按需分批地将磁盘上的数据读入内存,以解决内存不够的问题。

2) 32 位计算机直接访问的地址为 4 GB。前面示例是基于操作系统的文件系统开发的。然而,32 位机上的 Linux、Windows NT/2000/XP 等操作系统,不允许硬盘上单个文件超过 4 GB。因此,当存储的数据量超过 4 GB 时,前面的程序也需要修改。办法之一,是创建多个文件来存放同一类数据,但同时需要协调同一类数据多个文件之间的处理关系。

(2) 大数据量下的查询速度

管理活动中,大量的操作是查询。然而,对于大数据量场合,既不能一次将所有数据读入内存进行查询,又不能只在一个数据文件中查询。在这种情况下,如何保证查询速度,也是程序编制时需要考虑的重要环节。

(3) 数据冗余和不一致

由于文件和程序是在很长的一段时间内由不同的程序员创建的,因此,不同文件可能采用不同格式,不同程序可能采用不同语言写成。此外,相同的信息可能在几个地方(文件)重复存储。例如,某个读者的地址可能既在读者信息文件中出现,又在出借信息文件中出现。这种冗余除了导致存储和访问开销增大外,还可能导致数据不一致性,即同一数据的不同副本不一致。例如,某个读者地址的更改可能在读者信息文件中得到反映而在系统的其他地方(如:出借信息文件)却没有。

然而,在文件系统中,无维护数据一致性的监控机制,数据的一致性完全由用户负责维护。这在简单系统中还可勉强应付,在复杂的系统中,要保证数据的一致性,几乎是不可能的。

(4) 数据访问困难

假设图书馆的某位高级职员想要找出所有居住在某个特定邮编地区的读者的姓名,于是他要求数据处理部门生成这样的一个列表。由于原始系统的设计者并未预料到会有这样的需求,因此没有现成的应用程序去满足这个需求。但是,系统中却有一个产生所有读者列表的应用程序。这时该高级职员有两种选择:一种是取得所有读者的列表,并从中手工提取所需信息;另一种是要求数据处理部门让某个系统程序员书写相应的应用程序。这两种方案显然都不太令人满意。假设编写了相应的程序,几天以后这位高级职员可能又需要将该列表减少到只列出借书数量不少于 3 本的那些读者。可以预见,产生这样一个列表程序又不存在,这位高级职员就再一次面临着前面那两种不尽人意的选择。

(5) 数据孤立

由于数据分散在不同文件中,这些文件又可能具有不同的格式,因此编写一个检索适当数据的新应用程序是很困难的。

(6) 应用程序依赖性

就文件处理而言,程序依赖于文件的格式。在文件处理系统中,文件和记录的物理格式通



常是应用代码的一部分。当文件格式变化时,应用程序也必须变化。

例如,如果对一个读者记录做修改,把邮政编码从5位扩展成9位,所有使用读者记录的应用程序,即使不使用邮政编码域,也必须修改。因为可能有20个程序处理读者文件,这样的变化意味着程序员必须识别所有受影响的程序,修改并测试它们。所有这些都是耗时易错的事情。另外,如果程序没有使用发生格式变化的域,要求程序员修改这些程序就是一种浪费。

(7) 完整性问题

由于计算机化的管理应用系统是现实业务系统的替代者。其管理的数据也应符合现实业务系统中的各种规章制度的要求。事实上,许多规章制度可以转化为对各种数据的“约束”(constraint)。例如,企业人事制度可能规定在职人员的年龄不得超过60岁(即职工的退休年龄),一个部门的主管不能在其他部门兼主管(即不能一身兼多职),一个学生的学号不能重复等。管理应用系统中管理的数据,就应遵从这种由制度转化而来的数据约束,从而保证数据的完整性(integrity)。开发者通过在各种不同应用程序中加入适当的代码来实现系统中的这些约束。前面基于文件系统的管理应用系统也应在这方面进行改进,以使其真实、完整地成为现实业务系统的替代者,提高管理的水平和效率。然而,当新的约束加入时,很难通过修改程序来体现这些新的约束。尤其是在约束涉及不同文件中的多个数据项时,问题就变得更加复杂。

(8) 原子性问题

管理应用系统在日常管理运行中,不可避免地会遇到各种各样的故障,如突然断电、系统死机、程序崩溃、磁盘不能读写等。在故障情况下,可能存在数据未完全写入磁盘的现象,从而出现数据丢失,甚至数据的破坏,这对于视管理数据为生命的各行业,特别是商业、证券、银行、保险、电信等,是一大灾难。因此,一旦故障发生,数据就应该被恢复到故障发生以前的状态,简称“故障恢复”(crash recovery)。对很多应用来说,这样的保证是至关重要的。现在看看把甲账户的100元转入乙账户这样的一个程序。假设在程序的执行过程中发生了系统故障,很可能甲账户上减去的100元还没来得及存入乙账户,这就造成了数据库状态的不一致。显然,为了保证数据库的一致性,这里的借和贷两个操作必须是要么都发生,要么都不发生。也就是说,转账这个操作必须是原子的——它要么全部发生,要么根本不发生。在传统的文件处理系统中必须增加大量代码来实现故障恢复,即使这样,要保持原子性也是很难做到的。

(9) 并发访问异常

为了提高系统的总体性能和加快响应速度,许多系统允许多个用户同时更新数据。在这样的环境中,并发的更新操作相互影响,可能导致数据的不一致。为了消除这种情况发生的可能性,系统必须进行某种形式的管理。但是,由于数据可能被多个不同的应用程序访问,这些程序相互间事先又没有协调,管理就很难进行。

(10) 安全性问题

企事业单位的日常管理转用计算机管理后,用户最担心的问题之一是保存在计算机中数据的“安全性”(security)。

数据的安全性主要在于对访问数据用户的授权,并非数据库系统的所有用户都可以访问所有数据。没有授权的用户不能访问系统的数据;得到授权的用户只能访问他所能访问的数据。因此,要实现现实管理系统所要达到的安全性要求,需要对基于文件系统的管理应用系统加大改进,以满足数据安全性要求。然而,由于应用程序总是即席加入到系统中来,因此这