



高等院校计算机教材系列

数据库 原理与应用

何玉洁 编著

为教师提供教学课件

TP311.13
258

2007

高等院校计算机教材系列

数据库 原理与应用

何玉洁 编著



机械工业出版社
China Machine Press

本书系统讲解数据库的基本概念和应用技术,包括目前流行的SQL Server后台数据库管理系统以及Visual Basic 6.0可视化编程环境。附录中包含SQL Server系统提供的常用函数、Visual Basic应用程序的发布方法,并给出一个完整的课程设计题目,帮助读者综合运用所学知识。

本书为各章均配备习题,相关章配有上机练习,书后还提供习题答案,方便读者参考。本书重点突出、面向实用,并为教师配有教学课件,方便教学。本书适合作为高等院校计算机专业数据库原理课程的教材,也可供广大技术人员及自学者参考。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

数据库原理与应用/何玉洁编著.-北京:机械工业出版社,2007.1
(高等院校计算机教材系列)

ISBN 7-111-19871-9

I. 数… II. 何… III. 数据库系统-高等学校-教材 IV. TP311.13

中国版本图书馆CIP数据核字(2006)第106919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:朱 劼

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007年1月第1版第1次印刷

184mm×260mm·22印张

定价:32.00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

前 言

数据库技术是计算机科学中的一个非常重要的部分，数据库技术以及数据库的应用也正以日新月异的速度发展，因此作为现代的大学生，特别是计算机专业的学生，学习和掌握数据库知识是非常必要的。

本书是面向计算机专业学生学习数据库知识而编写的一本教材，其特点是内容全面，既包括数据库的基础理论知识，又包括数据库的前端和后端的应用技术。SQL Server是目前广泛应用的后台数据库管理系统，Visual Basic也是使用非常普遍、方便的可视化编程环境。将这些内容结合在一本书中，可以使读者系统、全面地学习数据库系统的整体概念和应用技术。在介绍数据库理论时，本书特别加强了解决实际问题的内容，包括在数据库管理系统中对索引的管理方法以及如何构建提高数据查询效率的索引，如何编写带参数的存储过程以及如何自定义函数以实现复杂的数据查询功能等。在实现数据完整性约束方面，本书除了介绍常用的完整性约束方法之外，还介绍了实现复杂的数据完整性约束的方法——触发器。

本书由三部分组成。第一部分介绍数据库系统的基本概念和基本理论，这部分由第1章~第9章组成，具体内容包括数据管理的发展过程、数据库系统的组成、关系代数、基本SQL语句的使用、数据完整性约束的实现方法、视图、存储过程和函数的概念及定义方法、关系规范化理论、数据库事务及并发控制、备份恢复机制以及数据库的设计过程。

第二部分主要介绍SQL Server 2000的功能和使用方法，这部分由第10章~第14章组成，具体内容包括安装和配置SQL Server、在SQL Server环境中创建数据库和表、安全管理、数据传输以及备份和恢复数据库。

第三部分主要介绍如何在Visual Basic 6.0环境中开发数据库的前端应用程序，这部分包括第15章~第17章。在这部分中，我们将介绍数据库的应用程序和数据访问接口技术，包括Visual Basic 6.0中的ADO数据控件以及ADO对象访问数据库的技术，并用四个例子说明使用这些技术开发数据库应用程序的过程，最后介绍Visual Basic自带的可以自动生成数据库应用程序的数据窗体向导的功能。

本书还包括四个附录，附录A介绍SQL Server 2000中系统提供的常用函数，目的是使读者能够更好地使用SQL Server提供的功能。附录B介绍如何发布已编制好的VB应用程序。当我们在Visual Basic 6.0开发环境中开发好应用程序之后，就应该交付给用户使用。在交付应用程序时，我们不能要求用户的计算机上也安装Visual Basic开发环境，因此，必须对已开发好的应用程序进行发布，使用户脱离Visual Basic开发环境也能够运行Visual Basic应用程序。附录C介绍一个应用实例，主要目的是综合运用已学习的知识开发一个比较完整的数据库应用程序，这个附录可作为学生的课程设计题目。附录D给出本书的习题解答。

为了便于教师使用本书进行教学，我们为本书制作了电子课件，需要的教师可登录华章网站（www.hzbook.com）下载。

本书是作者对多年从事数据库教学的经验和感受的总结。本书的出版得到了机械工业出版社华章分社的大力帮助和支持，特别是华章分社的总编温莉芳、编辑朱劼，她们在我编写此书的过程中均给予了极大的支持，并提出了许多宝贵的意见和建议，在此，我对她们及机械工业出版社华章分社表示诚挚的感谢。

由于时间仓促加之本人水平所限，书中难免有不妥之处，望广大同仁给予批评指正。

何玉洁

2006年4月

目 录

前言

第一部分 数据库原理

第1章 数据库概述	2
1.1 数据管理的发展	2
1.1.1 以数据为中心的应用系统的特点	2
1.1.2 文件管理系统	3
1.1.3 数据库管理系统	5
1.1.4 数据独立性	7
1.2 什么是数据库系统	8
1.3 使用数据库系统的原因及数据库应用的前景	9
1.3.1 使用数据库系统的原因	9
1.3.2 数据库应用的前景	9
1.4 小结	10
习题	10
第2章 数据库系统结构	11
2.1 数据和数据模型	11
2.1.1 数据	11
2.1.2 数据模型	11
2.2 概念层数据模型	13
2.2.1 基本概念	13
2.2.2 实体-联系模型	13
2.3 组织层数据模型	15
2.3.1 层次数据模型	16
2.3.2 网状数据模型	17
2.3.3 关系数据模型	18
2.4 数据库系统的结构	18
2.4.1 三级模式结构	19
2.4.2 二级映像功能	21
2.4.3 数据库管理系统	22
2.5 小结	23
习题	24

第3章 关系数据库	25
3.1 关系模型概述	25
3.1.1 关系数据结构	25
3.1.2 关系操作	25
3.1.3 数据完整性约束	26
3.2 关系数据模型的基本术语与形式化定义	26
3.2.1 关系模型的基本术语	26
3.2.2 关系数据结构及其形式化定义	28
3.3 关系模型的完整性约束	30
3.3.1 实体完整性	30
3.3.2 参照完整性	31
3.3.3 用户定义的完整性	32
3.4 关系代数	33
3.4.1 传统的集合运算	33
3.4.2 专门的关系运算	36
3.5 小结	39
习题	39
第4章 SQL语言	40
4.1 基本概念	40
4.1.1 SQL语言的发展	40
4.1.2 SQL语言的特点	40
4.1.3 SQL语言功能概述	41
4.2 SQL的数据类型	41
4.2.1 数值型	42
4.2.2 字符串型	42
4.2.3 日期时间型	43
4.2.4 货币型	44
4.3 基本表的定义、删除及修改	44
4.3.1 基本表的定义与删除	44
4.3.2 修改表结构	46
4.4 数据查询功能	47
4.4.1 查询语句的基本结构	48
4.4.2 简单查询	48
4.4.3 多表连接查询	58

4.4.4 子查询	62	第7章 关系数据库规范化理论	100
4.5 数据更改功能	65	7.1 函数依赖	100
4.5.1 插入数据	65	7.1.1 函数依赖的基本概念	100
4.5.2 更新数据	66	7.1.2 一些术语和符号	101
4.5.3 删除数据	67	7.1.3 为什么要讨论函数依赖	102
4.6 建立与删除索引	67	7.2 关系规范化	102
4.6.1 索引的概念	67	7.2.1 关系模式中的码	102
4.6.2 索引的分类	69	7.2.2 范式	103
4.6.3 创建和删除索引	72	7.3 关系模式的分解准则	106
4.7 小结	73	7.4 小结	108
习题	73	习题	108
第5章 视图、存储过程和用户自定义 函数	75	第8章 数据库保护	110
5.1 视图	75	8.1 事务的基本概念	110
5.1.1 视图的概念	75	8.1.1 事务	110
5.1.2 定义视图	75	8.1.2 事务的特征	110
5.1.3 删除视图	77	8.1.3 SQL事务处理模型	111
5.1.4 视图的作用	77	8.2 并发控制	112
5.2 存储过程	78	8.2.1 并发控制概述	112
5.2.1 存储过程的概念	78	8.2.2 并发控制措施	114
5.2.2 创建和执行存储过程	79	8.2.3 封锁协议	115
5.3 用户自定义函数	84	8.2.4 死锁	117
5.3.1 函数的概念	84	8.2.5 并发调度的可串行性	118
5.3.2 创建和调用标量函数	84	8.2.6 两段锁协议	119
5.3.3 创建和调用内嵌表值函数	85	8.3 数据库备份与恢复	120
5.3.4 创建和调用多语句表值函数	87	8.3.1 数据库故障的种类	120
5.3.5 更改和删除函数	88	8.3.2 数据库备份	121
5.4 小结	89	8.3.3 数据库恢复	122
习题	89	8.4 小结	123
习题	89	习题	123
第6章 实现数据完整性约束	90	第9章 数据库设计	124
6.1 数据完整性基本概念	90	9.1 数据库设计概述	124
6.1.1 完整性约束条件的作用对象	90	9.1.1 数据库设计的特点	125
6.1.2 实现数据完整性的方法	90	9.1.2 数据库设计方法概述	126
6.2 实现声明完整性	91	9.1.3 数据库设计的基本步骤	126
6.3 实现过程完整性	93	9.2 数据库需求分析	127
6.3.1 事务基本概念	93	9.2.1 需求分析的任务	127
6.3.2 触发器	94	9.2.2 需求调查	128
6.4 小结	98	9.3 数据库结构设计	129
习题	98	9.3.1 概念结构设计	129
习题	98		

9.3.2 逻辑结构设计	133	12.1.1 安全控制模型	178
9.3.3 物理结构设计	136	12.1.2 数据库权限的种类及 用户的分类	179
9.4 数据库行为设计	138	12.2 SQL Server的安全控制	179
9.4.1 功能需求分析	138	12.3 管理SQL Server登录帐户	181
9.4.2 功能设计	139	12.3.1 建立登录帐户	182
9.4.3 事务设计	139	12.3.2 修改登录帐户的属性	184
9.5 数据库的实施和维护	140	12.3.3 删除登录帐户	184
9.5.1 数据库数据的加载和试运行	140	12.4 管理数据库用户	185
9.5.2 数据库的运行和维护	141	12.4.1 建立数据库用户	185
9.6 小结	141	12.4.2 删除数据库用户	186
习题	142	12.5 管理权限	186
		12.5.1 SQL Server权限种类	186
		12.5.2 权限的管理	187
		12.6 角色	190
		12.6.1 建立用户自定义的角色	190
		12.6.2 为用户定义的角色授权	191
		12.6.3 添加和删除用户自定义 角色的成员	192
		12.7 小结	193
		习题	193
		上机练习	193
		第13章 数据传输	194
		13.1 DTS功能概述	194
		13.2 利用DTS向导实现数据的 导入和导出	195
		13.3 小结	201
		习题	202
		上机练习	202
		第14章 备份和恢复数据库	203
		14.1 备份数据库	203
		14.1.1 为什么要进行数据库备份	203
		14.1.2 备份的内容及备份时间	203
		14.1.3 SQL Server的备份设备	204
		14.1.4 SQL Server 2000的备份类型	205
		14.1.5 备份策略	207
		14.1.6 实现备份	209
		14.2 恢复数据库	216
		14.2.1 恢复前的准备	216
第二部分 SQL Server 2000			
基础及使用			
第10章 SQL Server 2000基础	146		
10.1 SQL Server 2000概述	146		
10.2 安装与测试	147		
10.2.1 安装前的准备	147		
10.2.2 安装及安装选项	148		
10.2.3 测试安装	156		
10.3 SQL Server 2000常用工具简介	157		
10.4 卸载SQL Server 2000	159		
10.5 小结	160		
习题	160		
上机练习	160		
第11章 数据库与基本表的创建和管理	161		
11.1 数据库的创建与管理	161		
11.1.1 SQL Server数据库的构成	161		
11.1.2 创建数据库	162		
11.1.3 删除数据库	166		
11.1.4 修改数据库	166		
11.2 基本表的创建与管理	167		
11.2.1 定义表及约束	168		
11.2.2 修改表结构	173		
11.2.3 删除表	174		
11.3 小结	175		
习题	176		
上机练习	176		
第12章 安全管理	178		
12.1 安全控制	178		

第一部分 数据库原理

本部分主要介绍数据库的基础理论知识，包括数据和数据模型，关系数据库的标准操作语言——SQL，数据库的安全性和完整性，如何设计性能优越的关系表，如何实现事务的并发控制以及如何对数据库应用系统进行分析 and 设计。

本部分由以下7章组成：

- 第1章 数据库概述
- 第2章 数据库系统结构
- 第3章 关系数据库
- 第4章 SQL语言
- 第5章 视图、存储过程和用户自定义函数
- 第6章 实现数据完整性约束
- 第7章 关系数据库规范化理论
- 第8章 数据库保护
- 第9章 数据库设计

第 1 章 数据库概述

随着信息管理水平的不断提高，信息资源已成为企业的重要财富和资源，用于信息管理的数据库技术也得到了很大的发展，其应用领域也越来越广泛。数据库的应用形式日益多样，从小型事务处理到大型信息系统，从联机事务处理到联机分析处理，从一般企业管理到计算机辅助设计与制造（CAD/CAM），乃至地理信息系统等都应用了数据库技术。数据库技术已经渗透到我们日常生活的方方面面，比如用信用卡购物，飞机、火车订票系统，图书馆对书籍及借阅的管理等，无一不使用了数据库技术。数据库的建设规模、数据库中信息量的大小以及使用的程度已经成为衡量企业的信息化程度的重要标志。

简单地说，数据库技术就是研究如何科学地管理数据以便为人们提供可共享的、安全的、可靠的数据的技术。数据库技术一般包括数据管理和数据处理两部分内容。

数据库系统实质上是一个用计算机存储数据的系统，可以将数据库看做一个电子文件柜，也就是说，数据库是收集数据文件的仓库或容器。

1.1 数据管理的发展

在介绍数据管理的发展之前，我们有必要先了解一下以数据为中心的应用系统的特点。

1.1.1 以数据为中心的应用系统的特点

随着计算机的普及和信息量的不断增加，在众多的计算机应用中，数据密集型的应用的发展速度非常迅速。数据密集型的应用也就是我们所说的以数据为中心的应用，这种应用具有如下三个特点：

1. 涉及的数据量大

以图书馆和银行的信息管理为例，如果要将全部信息保存起来，则数据量是很大的，我们不可能将这些信息全部保留在内存中。内存只能暂时存放其中的很小一部分信息，而其余的大量数据则要存放在辅助存储设备上。

2. 数据不随程序的结束而消失

需要长期保留在计算机中的数据称为持久性数据。比如图书馆和银行的信息，必须要持久地保存，这些数据就是持久性数据。持久保存的数据是有价值的，人们可以通过分析积累的数据，制定出合适的方针和决策。例如通过分析一段时间内哪些图书借出的次数比较多，可以帮助图书管理人员决定下次要多采购哪些书。这就是我们经常说的辅助决策支持。持久性数据不能存放在内存中，必须存储在永久性存储设备上，比如存放在磁盘或磁带上。

3. 数据可以被多个应用程序共享

数据不是某个用户专有的，而是可被许多用户使用，而且还必须允许多个用户同时使用这些数据。以飞机订票系统为例，一个地区有成百上千个订票点，在一个订票点工作时，其它的订票点也必须能同时工作，否则设立这么多个订票点就没有意义了。

如何很好地管理这种大量的、持久的、共享的数据是计算机科学技术领域中一门重要的技术和重要的研究课题。

1.1.2 文件管理系统

要想很好地理解现代数据库的特征，最好先看一下在数据库技术产生之前，人们如何保存和使用数据。

早期的数据是采用文件系统进行管理的，即将数据保存在文件中。用户的应用程序直接操作文件中的数据。在文件系统中，数据按其内容、结构和用途分成若干个命名的文件。文件一般为某一用户或用户组所有，但也可以指定与其它用户共享。用户可以通过操作系统对文件进行打开、读、写、关闭等操作。

假设现在用文件系统来实现对学生进行管理的程序。在此系统中，要对学生的基本信息和选课情况进行管理。在管理学生基本信息时要用到学生的基本信息数据，假设此数据存在F1文件中。学生选课情况的管理包括管理学生的基本信息、课程的基本信息和学生的选课信息，假设用F2和F3两个文件分别存储课程基本信息和学生选课基本信息数据。学生选课情况管理中涉及的学生基本信息可以使用学生基本信息

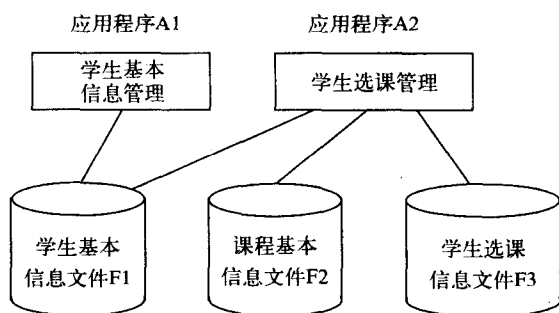


图1-1 文件管理系统示例

信息管理系统中的F1文件。假设实现学生基本信息管理功能的应用程序叫A1，实现学生选课管理功能的应用程序叫A2，则学生的基本信息和选课情况可用图1-1表示。

假设F1、F2和F3文件分别包含如下信息：

- F1包含学号、姓名、性别、出生日期、所在系、专业、所在班、特长、家庭住址。
- F2包含课程号、课程名、授课学期、学分、课程性质。
- F3包含学号、姓名、专业、课程号、课程名、修课类型、修课时间、考试成绩。

我们将文件中所包含的每一个子项称为文件结构中的字段或列，将每一行数据称为一个记录。

“学生选课管理”系统的处理过程大致为：在学生选课管理系统中，若有学生选课，则先查F1文件，判断有无此学生。若有此学生，则再访问F2文件，判断其所选的课程是否存在。若课程也存在，就将学生选课信息写到F3文件中。

这看起来似乎很好，但仔细分析一下，就会发现使用文件管理系统管理数据有如下一些缺点。

1. 编写应用程序不方便

应用程序编写者必须对所用文件的逻辑及物理结构（文件中包含多少个字段，每个字段的数据类型，采用何种存储结构，比如链表或数组等）有清楚的了解。操作系统只提供了打开、关闭、读、写等几个低级的文件操作命令，而文件的查询、修改等处理都必须在应用程序中通过编程实现。这样也容易造成各应用程序在功能上的重复，比如图1-1中的“学生基本信息管理”和“学生选课管理”都要对F1文件进行操作，但这两个功能相同的操作却很难共享。

2. 数据冗余不可避免

假设A2需要用到F3文件中包含的学生的所有或大部分信息，比如，除了学号之外，还需

要姓名、性别、专业、所在系等信息，而F1中也包含了这些信息，因此F3和F1文件中有重复的信息。但这些重复的信息只是不同文件的部分内容，因此很难在两个文件中公用这些公共信息，从而造成数据的重复（也叫数据的冗余）。

数据冗余不仅会造成存储空间的浪费（其实，随着计算机硬件技术的飞速发展，存储容量不断扩大，空间问题已经不是解决问题时需要关心的主要问题），更为严重的是造成了数据的不一致。例如，假设某个学生所学的专业发生了变化，我们一般只会想到在F1文件中进行修改，而往往忘记了在F3中要进行同样的修改。这样就会造成同一名学生在F1文件和F3文件中的“专业”不一样，也就是数据不一致。人们不能判定哪个数据是正确的，尤其当数据冗余很多的时候，情况更是如此。这样数据就失去了其可信性。

文件系统本身不具备维护数据一致性的功能，这些功能完全由用户（应用程序开发者）负责维护。这在简单的系统中还可以勉强应付，但在复杂的系统中，若让开发者来保证数据的一致性，几乎是不可能的。

3. 应用程序有依赖性

就文件系统而言，应用程序要依赖于文件的结构。文件和记录的结构通常是应用程序代码的一部分，如C语言的struct。文件结构每进行一次修改，比如添加字段、删除字段甚至是修改字段的长度（如电话号码从7位扩到8位），都要对应用程序进行相应的修改，因为我们在打开文件读取数据时，必须要将文件记录中的不同字段的值对应到程序变量中。随着应用环境和需求的变化，修改文件的结构是不可避免的事情，这样就需要在应用程序中进行相应的修改，而频繁修改应用程序是很麻烦的。这是因为首先要熟悉原有程序，修改后还需要对程序进行测试、安装等。

以上弊端都是由于应用程序对文件结构过分依赖造成的。换句话说，文件系统的数据库独立性不好。

4. 不支持对文件的并发访问

在现代计算机系统中，为了有效地利用计算机资源，系统一般允许多个应用程序并发运行（尤其是在现在的多任务操作系统环境中）。文件最初是作为程序的附属数据出现的，它一般不支持多个应用程序同时对同一个文件进行访问。我们可以回忆一下，假设某个用户打开了一个Excel文件，如果第二个用户在第一个用户没有关闭此文件之前就想打开此文件，他会得到什么信息？他只能以只读方式打开此文件，而不能在第一个用户打开文件的同时对此文件进行修改。我们再回忆一下，如果我们用C语言编写一个修改某文件内容的程序，其过程是先以写的方式打开文件，然后写入新内容，最后再关闭文件。在文件关闭之前，无论在其它的程序中还是在同一个程序中都是不能再打开此文件的，这就是文件系统不支持并发访问的含义。

对于以数据为中心的应用系统来说，必须要支持多个用户对数据的并发访问。

5. 数据间的联系弱

在文件系统中，文件与文件之间是彼此独立、毫不相干的，文件之间的联系必须通过程序来实现。比如在上述的F1和F3文件中，F3文件中的学号、姓名等学生的基本信息必须是F1文件中已经存在的（即选课的学生必须是已经存在的学生）。同样，F3中的课程号等与课程有关的基本信息也必须是F2文件中已经存在的（即学生选的课程也必须是已经存在的课程）。这些数据之间的联系是客观需求当中所要求的很自然的联系，但文件系统本身不具备自动实现这些联系的功能，所以必须通过应用程序来保证这些联系，也就是说必须编写代码来手工地保证这些联系。这样不但增加了编写代码的工作量和复杂度，而且当联系很复杂时，也难以

保证其正确性。因此，文件系统不能反映现实世界事物间的联系。

6. 难以按不同用户的需要表示数据

如果用户需要的信息来自多个不同的数据文件，我们就需要对多个文件的信息内容进行提取、比较、组合和表示。例如，假设有用户希望得到如下信息：

(所在班，学号，姓名，课程名，学分，考试成绩)

这些信息涉及三个文件。从F1文件中可以得到“所在系”信息，从F2文件中得到“学分”信息，从F3文件中得到“考试成绩”信息。而“学号”、“姓名”信息可以从F1或F3文件中得到，“课程名”可以从F2或F3文件中得到。我们在生成一行数据时，必须比较从三个文件中读取的数据，然后组合成一行有意义的信息。比如，将从F1文件中读取的学号与从F3中读取的学号进行比较，学号相同时，才可以将F1中的“所在系”、F3中的“考试成绩”以及当前所对应的学号和姓名组合成一行数据的内容。同样，在处理完F1和F3文件的组合后，我们可以将组合的结果再与F2文件的内容进行比较，找出课程号相同的课程的学分，再与已有的结果组合起来。如果数据量很大、涉及的表比较多时，可以想象这个过程有多复杂。因此，这种大容量复杂信息的查询，在文件管理系统中是很难处理的。

7. 无安全控制功能

在文件管理系统中，很难控制某个人对文件的操作，比如控制某人只能读和修改文件，不能删除文件，或者不能读或修改文件中的某个或者某些字段等等。而在实际应用中，数据的安全性是非常重要的且不可缺少的。比如，在学生选课管理系统中，我们不允许学生修改他的考试成绩。在银行系统中，更是不允许一般用户修改其存款数额。

随着人们对数据需求的增加以及计算机科学技术的不断发展，对数据进行有效、科学、正确、方便的管理就成为人们的迫切需求。针对文件系统的这些缺陷，人们逐步开发出了以统一管理和共享数据为主要特征的数据库管理系统。

1.1.3 数据库管理系统

数据库技术的出现主要是为了克服文件管理系统在管理数据上的诸多缺陷，满足人们对数据管理的需求。对于上述的学生基本信息管理和学生选课管理系统来说，如果使用数据库来管理，其实现方式与文件系统有很大区别。用数据库进行管理的实现过程如图1-2所示。

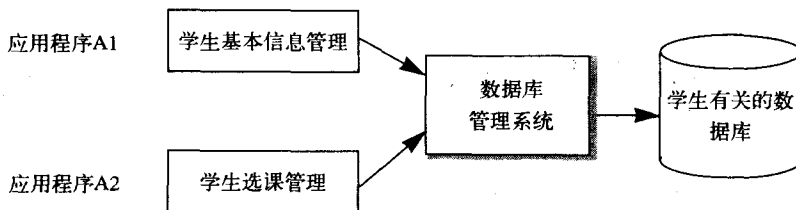


图1-2 数据库管理系统实现示例

比较一下图1-1和图1-2，可以发现两者有如下差别：

- 在文件系统中，应用程序直接访问存储数据的文件。而在数据库系统中，应用程序则是通过数据库管理系统（DataBase Management System，简称DBMS）来访问数据。
- 在数据库系统中，数据不再仅仅为某个程序或用户服务，存储数据的文件也不再需要直接被应用程序管理，而是由一个称为数据库管理系统的软件统一管理。

与文件系统相比,数据库管理系统实际上是应用程序和存储数据的数据库(在某种意义上也可以把数据库看成是一些文件的集合)之间的接口。数据库管理系统实际上是一个系统软件。不要小看这个变化,有了这个系统软件后,以前在应用程序中由开发人员实现的很多繁琐的操作和功能,都可以交给这个系统软件。这样应用程序不再需要关心数据的存储方式,而且数据的存储方式的变化也不再影响应用程序。将需要进行的修改交给数据库管理系统,经过数据库管理系统处理后,应用程序感觉不到这些变化,因此,应用程序也不需要任何修改。

与文件系统管理数据比较后,我们会发现数据库系统具有以下优点。

1. 将相互关联的数据集成在一起

在数据库管理系统中,所有相关的应用数据都存储在一个称为数据库的环境中,应用程序可通过DBMS访问数据库中的所有数据。

2. 较少的数据冗余

由于数据是统一管理的,因此可以从全局着眼,合理地组织数据。例如,将1.1.2节中的F1、F2和F3文件中的重复数据去掉,只在一个地方进行管理,这样就可以形成如下所示的几部分信息:

学生基本信息:学号、姓名、性别、出生日期、所在系、专业、所在班、特长、家庭住址。

课程基本信息:课程号、课程名、授课学期、学分、课程性质。

学生选课信息:学号、课程号、选课类型、选课时间、考试成绩。

在关系数据库中,可以将每一种信息存储在一个表中(关系数据库的概念我们在后面介绍),重复的信息只存储一份。若在学生选课时需要学生的名字,根据学生选课信息中的学号,可以很容易地在学生基本信息中找到此学号对应的名字。因此,消除数据的重复存储不影响我们对信息的提取,同时还可以避免由于数据重复存储而造成的数据不一致问题。比如,当某个学生所学的专业发生变化时,我们只需在“学生基本信息”中进行修改即可。

同1.1.2节中的问题一样,当我们要检索(所在班,学号,姓名,课程名,学分,考试成绩)信息时,这些信息也需要从三个地方(关系数据库中称为三张表)得到,也需要对信息进行适当的组合,即学生选课信息中的学号只能与学生基本信息中学号相同的信息组合在一起。同样,学生选课信息中的课程号也必须与课程基本信息中的课程号相同的信息组合在一起。过去在文件管理系统中,这个工作是由开发者编程实现的,而有了数据库管理系统后,这些繁琐的工作就完全交给了数据库管理系统来完成。

因此,在数据库管理系统中,减少了数据冗余和开发者的负担。

3. 程序与数据相互独立

在数据库中,数据所包含的所有数据项以及数据的存储格式都与数据一起存储在数据库中,它们通过DBMS而不是应用程序来访问和管理,应用程序不再需要包含要处理的文件和记录格式。

程序与数据相互独立有两个方面的含义。一方面是指若数据的存储方式发生变化(这里包括逻辑存储方式和物理存储方式)时,比如从链表结构改为哈希结构,或者从顺序存储转换为非顺序存储,应用程序不必进行任何修改。另一方面是指当数据的结构发生变化时,比如增加或减少了一些数据项,如果应用程序与这些修改的数据项无关,则应用程序也不用进行修改。这些变化都由DBMS负责维护。大多数情况下,应用程序并不知道数据存储方式或数据项已经发生了变化。

4. 保证数据的安全可靠

数据库技术能够保证数据库中的数据是安全可靠的。它有一套安全控制机制,可以有效

地防止数据库中的数据被非法使用或非法修改。数据库中还有一套完整的备份和恢复机制，当数据遭到破坏时（由软件或硬件故障引起的），能够很快地将数据库恢复到正确的状态，并使数据不丢失或只有很少的丢失，从而保证系统能够连续、可靠地运行。

5. 最大限度地保证数据的正确性

保证数据的正确性是指存放到数据库中的数据必须符合实际情况，比如人的性别只能是“男”和“女”，人的年龄应该在0~150之间（假设没有年龄超过150岁的人）。如果我们在“性别”中输入了其它的值，或者将一个负数输入到“年龄”中，显然是不符合实际情况的。数据库系统能够保证进入到数据库中的数据都是正确的数据，这就是数据完整性。数据完整性是通过在数据库中建立约束来实现的。当我们建立好保证数据正确性的约束之后，如果有不符合约束条件的数据进入到数据库中，数据库就能主动拒绝这些数据。

6. 数据可以共享并能保证数据的一致性

数据库中的数据可以被多个用户共享，共享是指允许多个用户同时操作相同的数据。当然这个特性是针对大型的多用户数据库系统而言的，对于单用户系统，在任何时候最多只有一个用户访问数据库，因此不存在共享的问题。

多用户系统问题是数据库管理系统在内部解决的问题，它对用户是不可见的。这就要求数据库能够对多个用户进行协调，保证多个用户对数据进行的操作不发生矛盾和冲突，即在多个用户同时使用数据库时，能够保证数据的一致性和正确性。可以设想一下，在飞机订票系统中，如果多个订票点同时对一架航班订票，那么必须保证不同订票点订出票的座位不能重复。

数据集成与数据共享是大型环境中数据库系统的主要优点。

数据库技术发展到今天已经是一门比较成熟的技术，经过上面的讨论，我们可以概括出数据库具备如下特征：数据库是相互关联的数据的集合，它用综合的方法组织数据，具有较小的数据冗余，可供多个用户共享，具有较高的数据独立性，具有安全控制机制，能够保证数据的安全性，允许并发地使用数据库，能有效、及时地处理数据，并能保证数据的一致性和完整性。

需要再次强调的是，所有这些特征并不是数据库中的数据所固有的，而是由数据库管理系统提供和保证的。

1.1.4 数据独立性

数据独立性包含逻辑独立性和物理独立性两个方面。物理独立性是指当数据的存储结构发生变化时，比如从链表存储改为哈希表存储，不影响应用程序的特性。逻辑独立性是指当表达现实世界的信息内容发生变化时，比如增加一些列、删除无用列等，也不影响应用程序的特性。要理解数据独立性的含义，必须先搞清什么是非数据独立性。在数据库系统出现之前，也就是在使用文件系统管理数据的时候，实现的应用程序常常是数据依赖的。也就是说，数据的物理表示方式和有关的存取技术都是在应用程序中要考虑的，而且有关物理表示的知识和访问技术直接体现在应用程序的代码中。例如，如果数据文件使用了索引，那么应用程序也必须知道有索引存在，而且还要知道记录的顺序是索引的，这样应用程序的内部结构就是基于这些知识而设计的。特别地，各种数据访问的准确形式和应用程序的异常检查程序也在很大程度上依赖于数据管理软件提供给应用程序的接口。我们称这样的应用程序是数据依赖的。因为一旦改变数据的物理表示，就会对应用程序产生很大的影响。例如，如果用哈希

表来重建数据的索引,则应用程序不得不进行很大的修改。而且这种情况下,应用程序修改的部分恰恰是与数据管理软件密切联系的部分,而与应用程序最初要解决的问题毫不相干,这是由数据管理接口的特点所引起的。

在数据库系统中,我们要尽量避免应用程序依赖于数据的情况,原因如下:

- 不同的应用程序看数据的角度是不同的,即使是对同样的数据也存在这样的问题。如我们在1.1.2节中介绍的文件系统很难按不同用户的要求显示同样的数据。
- 随着科学技术的进步以及应用业务的变化,有时必须要改变数据的物理表示和访问技术,以适应技术发展及需求变化。比如,添加新的数据列,改变列的类型,或者是增加新的存储设备等。理想情况下,这些变化不应该影响应用程序。但是,如果应用程序是数据依赖的,则这些改变都会要求应用程序做相应的改变。这种维护的代价不亚于创建一个新的应用程序。

因此,数据独立性是一种客观应用的要求。数据独立性可以描述为:应用程序不会因物理表示和访问技术的改变而改变,即应用程序不依赖于任何特定的物理表示和访问技术。数据库技术的出现正好克服了应用程序与数据的物理表示和访问技术间的依赖问题。

1.2 什么是数据库系统

前面我们介绍了数据库系统所具有的各种特征,那么什么是数据库系统呢?简单地说,数据库系统就是基于数据库的计算机应用系统。数据库系统一般包括四个部分:数据库、数据库管理系统、应用程序和系统管理员,如图1-3所示。

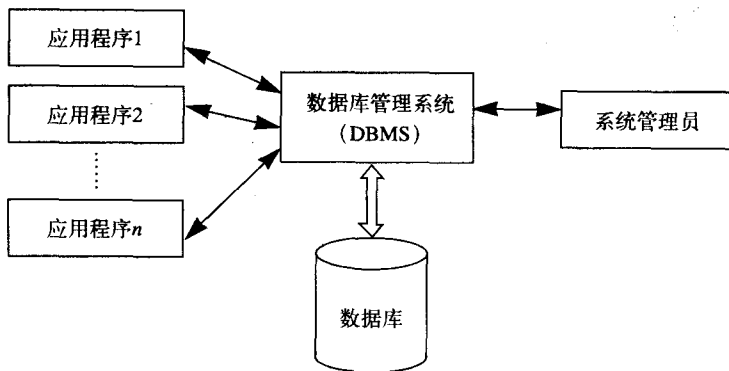


图1-3 数据库系统简图

其中,数据库是数据的汇集,它以一定的组织形式保存于存储介质上。DBMS是管理数据库的系统软件,它实现数据库系统的各种功能,是数据库系统的核心。系统管理员负责数据库的规划、设计、协调、维护和管理等工作。应用程序是指以数据库以及数据库数据为基础的应用程序。

除此之外,数据库系统还包括支持系统运行的计算机的硬件和操作系统环境以及使用数据库系统的用户。硬件环境是指保证数据库系统正常运行的最基本的内存、外存等硬件资源。由于数据库管理系统是一种系统软件,所以它必须建立在一定的操作系统环境上,没有合适的操作系统,数据库管理系统是无法正常运转的。比如SQL Server 2000的企业版就需要Windows操作系统的服务器版的支持。