21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

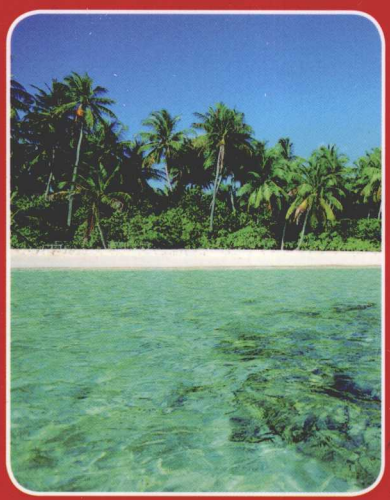
数据库原理及应用（第2版）

Database Principle and Application (2nd Edition)

何玉洁 刘福刚 主编

于绍娜 余阳 张荣梅 副主编

- 突出数据库的理论性，对内容精心选择和安排
- 注重数据库的实用性，配有大量的实例和习题
- 体现数据库的应用性，深入浅出地分析和说明



精品系列

人民邮电出版社
POSTS & TELECOM PRESS

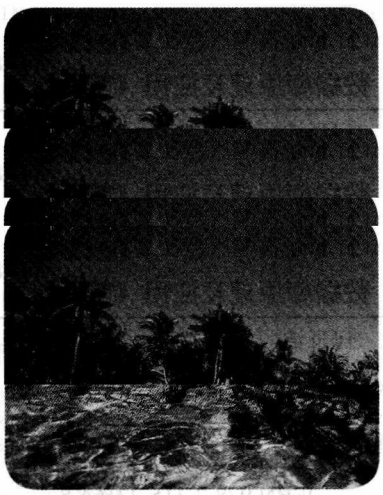
21世纪高等学校计

21st Century University Planned Textbooks of Computer Science

数据库原理及应用（第2版）

Database Principle and Application (2nd Edition)

何玉洁 刘福刚 主编
于绍娜 余阳 张荣梅 副主编



精品系列

人民邮电出版社
北京

图书在版编目（CIP）数据

数据库原理及应用 / 何玉洁，刘福刚主编. -- 2版
-- 北京 : 人民邮电出版社，2012.3
21世纪高等学校计算机规划教材
ISBN 978-7-115-27164-8

Ⅰ. ①数… Ⅱ. ①何… ②刘… Ⅲ. ①数据库系统－高等学校－教材 Ⅳ. ①TP311.13

中国版本图书馆CIP数据核字(2012)第011595号

内 容 提 要

本书由11章、2个附录组成，主要内容包括关系数据库基础、SQL语言、关系数据理论、数据库设计、事务与并发控制、后台数据库编程、视图和索引、安全管理、备份和恢复数据库等，在附录部分给出了SQL Server 2008的安装以及该平台支持的常用系统函数。

本书条理清晰、语言简洁，适合作为高等院校计算机及理工科类多用计算机学科的大学本科数据库教材，也可作为相关人员学习数据库知识的参考书。

21世纪高等学校计算机规划教材

数据库原理及应用（第2版）

◆ 主　　编　何玉洁　刘福刚
副 主 编　于绍娜　余　阳　张荣梅
责任编辑　武恩玉

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：787×1092　1/16
印张：18　　2012年3月第2版
字数：474千字　　2012年3月河北第1次印刷

ISBN 978-7-115-27164-8

定价：35.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

本书第1版出版于2008年4月，到现在已经过去了3年多，从第1版的使用反馈意见及个人的教学实际情况看，有必要对第1版的很多内容进行修订，以使本书更符合高校教学的要求，符合技术的发展需求。

本书修订内容如下：

（1）将后台的数据库实践平台从SQL Server 2000升级到SQL Server 2008。直接升级到SQL Server 2008而没有升级到SQL Server 2005的考虑是，SQL Server 2005和SQL Server 2008的操作界面基本是一样的，占用的系统资源也差不多，但SQL Server 2008在操作细节上更加人性化，提供了与Visual Studio编程环境类似的微帮助功能，更便于用户使用。

（2）去掉了第1版中的VB编程部分。由于现在使用VB编写数据库应用程序的人越来越少，VB的市场使用率越来越低，因此去掉了这部分内容。

（3）习题形式多样、题量丰富。为每章精心设置大量习题，每章习题都包含选择题、填空题和简答题3种；对第6章关系数据库理论和第7章数据库设计还设置了设计题；对能够上机实践的内容，如第3章SQL语言基础及数据定义功能、第4章数据操作语句、第5章视图和索引、第9章数据库编程、第10章安全管理和第11章备份和恢复数据库均给出了上机练习题。

（4）将索引从第1版的“第3章数据定义功能”中分离出来，将视图从第1版“第4章数据操作”中分离出来，将索引和视图合并为新的一章。第2版扩充了对索引和视图的讲解，使其知识更全面，学生学习完之后能了解其应用原理。

（5）对第1版第4章的数据查询功能的讲解，增加了CASE表达式、集合并运算（UNION）和将查询结果保存到新表（SELECT…INTO…）功能的介绍。

（6）增加了数据库后台编程技术的介绍，内容包括存储过程、触发器和游标。

（7）将在SQL Server 2008平台上进行的实践操作的介绍与知识讲解紧密结合在一起。在第1版中，实践和理论知识的讲解是分开进行的，在第2版中，为使学生在学习完理论知识后更便于实践，将实践的讲解分散到涉及相关内容的各个章中，如数据表的定义、视图和索引的定义等。

（8）将SQL Server基础知识的介绍、安装以及常用工具的介绍从书的正文中分离出来，放到附录中，使书的正文内容更加协调。

（9）增加了对SQL Server提供的常用系统函数的介绍，这部分内容也放在附录中。

除上述这些大的调整之外，在每一章对示例、内容也做了一些调整和补充，使

全书的知识内容更加完整，条理更加清晰，更便于学生学习掌握数据库知识。

本书的特点是内容涵盖全面，立足学以致用，在内容选取上既包括了数据库的基础理论知识，又包括了数据库的实践和后台编程技术。

本书可作为计算机专业以及非计算机专业多用计算机（电子、通信、管理及信息处理）学科的大学本科教材。随着计算机软硬件技术的不断发展，各企业和部门管理水平的不断提高和规范化，计算机的应用水平取得了长足的进步，特别是数据库技术，其应用水平及普及速度更是日新月异，数据库技术已经不再仅是计算机专业学生必须学习的课程，而且也成为非计算机专业，特别是多用计算机学科的大学生必须学习和掌握的知识。作为新时代的大学生，为了能够适应社会对人才的需要，有必要全面地掌握数据库知识。

本书适合32到48学时的教学，各学校和各专业可根据对学生的计算机水平掌握程度的要求，在授课中对本书的一些内容进行筛选。比如要求比较低的学校，可选讲数据库编程、有关关系数据库理论和极小函数依赖集的内容。

本书为授课教师提供电子教案，同时还提供了本书的习题解答供教师下载。

作者在编写本书的过程中，得到了人民邮电出版社领导和编辑的大力支持和帮助，同时也得到了很多使用过本书第1版的高校教师的帮助，他们对本书的内容提出了许多宝贵的意见和建议，是他们一直以来的鼓励与帮助，促使我们完成了这本书的编写。在此，对人民邮电出版社及所有帮助过我们的教师、学生及各方面人士表示诚挚的感谢。

本书的作者都是从事数据库教学多年并一直致力于数据库技术及应用和研究的一线教师，在数据库的设计与使用方面都有一些自己的经验和感受，本书是这些作者多年教学经验和感受的总结。虽然作者尽了自己应尽的努力，但由于水平所限，书中难免有不妥之处，望广大同仁能给予批评和指正。

何玉洁

2011年12月

目 录

第1章 数据库概述

随着信息管理水平的不断提高，应用范围的日益广泛，信息资源已成为企业的重要财富，而作为管理信息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。数据库管理系统正日益进入到我们的日常生活中，人们在不知不觉中扩展着对数据库的使用，比如信用卡购物，飞机、火车订票系统、图书馆对书籍的借阅管理等，无一不使用了数据库技术。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）、地理信息系统等，数据库系统已经渗透到我们日常生活中的方方面面，数据库中信息量的大小以及使用程度已经成为衡量企业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对数据进行科学管理以为人们提供可共享的、安全的、可靠的数据。数据库技术一般包含数据管理和数据处理两部分内容。

数据库系统本质上是一个用计算机存储数据的系统，数据库本身可以看作是一个电子文件柜，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

从计算机产生之后，人们就希望能够使用计算机存储和管理数据。最初对数据的管理是用文件方式进行的，也就是人们通过编写应用程序来实现对数据的存储和管理。后来，随着数据量的不断增大，计算机处理能力的不断增强，人们对数据的要求越来越多，希望达到的目的也越来越复杂，而文件管理方式已经很难满足人们对数据的需求，由此产生了数据库系统，也就是用数据库系统来存储和管理数据。数据管理的发展因此也就经历了文件管理和数据库管理两个阶段。

本节介绍文件管理和数据库管理在管理数据上的主要差别。

1.1.1 文件管理

理解今日数据库特征的最好办法是看一下在数据库技术产生之前，人们是如何通过文件的方式对数据进行管理的。

20世纪50年代后期到60年代中期，在计算机硬件方面，已经有了磁盘等直接存取存储设备；在软件方面，操作系统中已经有了专门的数据管理软件，一般称为文件管理系统。文件管理系统把数据组织成相互独立的数据文件，利用“按文件名访问，按记录进行存取”的管理技术，可以对文件中的数据进行修改、插入和删除操作。

在出现了程序设计语言之后，开发人员不但可以创建自己的文件并将数据保存在文件中，而

且还可以编写应用程序来处理文件中的数据，即编写应用程序来定义文件的结构，实现对文件内容的插入、删除、修改和查询操作。当然，真正实现磁盘文件的物理存取操作的是操作系统中的文件管理系统，应用程序只是告诉文件管理系统对哪个文件的哪些数据进行哪些操作。我们将开发人员编写应用程序，并借助文件管理系统的功能，实现对用户数据处理的方式称为文件管理。在本章后边的讨论中将忽略掉文件管理系统，假定应用程序是直接对磁盘文件进行操作。

现在看一下文件管理方式下对数据的操作模式。假设某学校要用文件管理系统实现对学生的管理。在此系统中主要实现两部分功能：学生基本信息管理和学生选课情况管理。假设教务部门对学生选课情况进行管理，各系部对学生基本信息进行管理。在学生基本信息管理中保存学生的基本信息数据，假设这些数据保存在 F11 文件中。对学生选课情况的管理涉及学生的部分基本信息、课程的基本信息和学生的选课信息，假设用 F2 和 F3 两个文件分别保存课程基本信息和学生选课基本信息数据。

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。由于学生选课管理中要用到一些 F1 文件中的数据，为减少冗余，假设这个功能就使用学生基本信息管理中的 F1 文件中的数据，如图 1-1 所示。

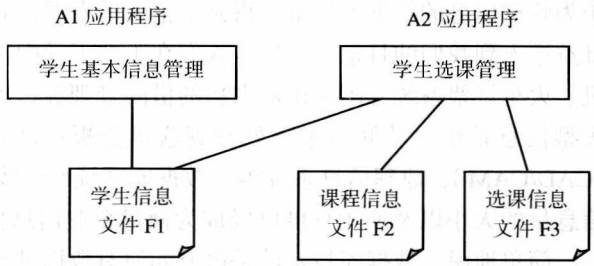


图 1-1　文件管理系统示例

假设 F1、F2 和 F3 文件分别包含如下信息：

F1 文件包含：学号、姓名、性别、出生日期、联系电话、所在系、专业、班号。

F2 文件包含：课程号、课程名、授课学期、学分、课程性质。

F3 文件包含：学号、姓名、所在系、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”的处理过程大致为：在学生选课管理中，若有学生选课，则先查 F1 文件，判断有无此学生；若有则再访问 F2 文件，判其所选的课程是否存在；若一切符合规则，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，就会发现用文件管理数据有如下缺点。

1. 编写应用程序不方便

应用程序编写者必须掌握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）。操作系统只提供了打开、关闭、读、写等几个底层的文件操作命令，而对文件中数据的查询、修改等操作都必须在应用程序中编程实现。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

由于 A2 应用程序需要在学生选课信息文件（F3 文件）中包含学生的一些基本信息，比如，除了学号之外，还需要包含姓名、性别、专业、所在系等信息，而学生信息文件（F1 文件）中也包含了这些信息，因此 F3 文件和 F1 文件中有重复的数据。但这些重复的数据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或称数据的冗余。

数据冗余所带来的问题不仅仅是存储空间的浪费（其实，随着计算机硬件技术的飞速发展，存储容量的不断扩大，空间问题已经不是我们解决问题需要关心的主要问题），还会造成数据的不一致（inconsistency）。例如，假设某个学生所学的专业发生了变化，一般只会想到在 F1 文件中进行修改，而往往忘记了在 F3 中同样需要进行修改。由此造成同一名学生在 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，尤其是当系统中存在多处数据冗余时，情况更是如此。这样数据就失去了其可信性。

文件本身并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. 应用程序依赖性

就文件管理而言，应用程序对数据的操作要依赖于存储数据的文件结构。文件和记录的结构通常是应用程序代码的一部分，如 C 程序的 struct。文件结构的每一次修改，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），都将导致应用程序的修改，因为我们在打开文件进行读取数据时，必须要将文件记录中的不同字段的值对应到应用程序的变量中。而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，这些都需要在应用程序中做相应的修改，而（频繁）修改应用程序是很麻烦的。因为人们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。甚至是修改了文件的存储位置或者是文件名，也都需要对应用程序进行修改，这显然给程序维护人员带来很多麻烦。

所有这些都是由于应用程序对文件结构以及文件的物理特性过分依赖造成的，换句话说，用文件管理数据时，其数据独立性（data independence）很差。

4. 不支持对文件的并发访问

在现代计算机系统中，为了有效利用计算机资源，一般都允许多个应用程序同时运行（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想打开此文件，他会得到什么信息呢？他只能以只读的方式打开此文件，而不能在第一用户打开的同时对此文件进行修改。我们再假设，如果用某种程序设计语言编写了一个对某文件中的内容进行修改的程序，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。在文件被关闭之前，不管是在其他的程序中，还是在同一个程序中都不允许再次打开此文件，这就是文件管理方式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. 数据间联系弱

当用文件管理数据时，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。比如对上述的 F1 文件和 F3 文件，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 文件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实际应用当中所要求的自然联系，但文件本身不具备自动实现这些联系的功能，我们必须编写应用程序来保证这些联系，也就是必须手工保证这些联系。这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，用文件管理数据时很难反映现实世界事物间客观存在的联系。

6. 难以按不同用户的愿望表示数据

按用户的愿望表示数据分为两种情况：第一种是有些用户可能只关心全体信息中的部分信息，比如分配学生宿舍的人可能只关心学生基本信息中的学号、姓名、性别；第二种情况是用户需要的信息可能来自于多个不同文件的部分信息内容的组合，例如可能有用户希望得到如下信息：

（班号，学号，姓名，课程名，学分，考试成绩）

对于第一种情况，如果有多个不同用户希望看到的数据是不同的，如果为每个这样的用户建立一个文件，势必造成很多的数据冗余。我们希望的是，用户关心哪些信息就为他生成哪些信息，对用户不关心的数据将其屏蔽掉。

对第二种情况，由于这些信息涉及了3个文件：从F1文件中得到“班号”信息，从F2文件中得到“学分”，从F3文件中得到“考试成绩”；而“学号”、“姓名”可以从F1文件或F3文件中得到，“课程名”可以从F2文件或F3文件中得到。在生成结果数据时，必须对从3个文件中读取的数据进行比较，然后组合成一行有意义的数据。比如，将从F1文件中读取的学号与从F3文件中读取的学号进行比较，学号相同时，才可以将F1文件中的“班号”、F3文件中的“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文件和F3文件的组合后，还需要将组合的结果再与F2文件中的内容进行比较，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，涉及的文件比较多时，我们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，在文件管理方式中是很难处理的。

7. 无安全控制功能

在文件管理方式中，很难控制某个人对文件的操作，比如只允许某个人查询和修改数据，但不能删除数据，或者对文件中的某个或者某些字段不能修改等。而在实际应用中，数据的安全性是非常重要且不可缺少的。比如，在学生选课管理中，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技术的不断发展，如何对数据进行有效、科学、正确、方便的管理已成为人们的迫切需求。针对文件管理的这些缺陷，人们逐步发展了以统一管理和共享数据为主要特征的数据库管理系统。

1.1.2 数据库管理

20世纪60年代后期以来，计算机管理数据的规模越来越大，应用范围越来越广泛，数据量急剧增加，同时多种应用同时共享数据集合的要求也越来越强烈。

随着大容量磁盘的出现，硬件价格的不断下降，软件价格的不断上升，编制和维护系统软件及应用程序的成本相应的不断增加。在数据处理方式上，联机实时处理要求更多，并开始提出和考虑分布处理。在这种背景下，以文件方式管理数据已经不能满足应用的需求，于是出现了新的管理数据的技术——数据库技术，同时出现了统一管理数据的专门软件——数据库管理系统。

例如，对于上述的学生基本信息管理和学生选课管理两个子系统，使用数据库技术来管理，其实现方式与文件管理有很大的区别。用数据库技术来管理的实现过程如图1-2所示。

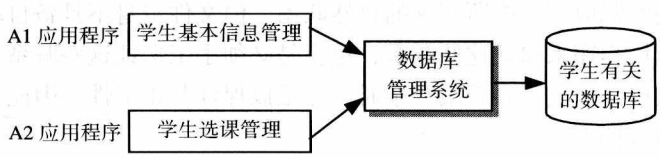


图1-2 数据库管理实现示例

比较图 1-1 和图 1-2，可以直观地发现两者有如下差别。

（1）在文件管理中，应用程序是直接访问存储数据的文件；而在数据库管理中，应用程序则是通过数据库管理系统（DataBase Management System，DBMS）来访问数据的。

（2）在数据库管理中，用户在访问数据时不再是逐一文件进行访问，而是针对一个存储用户所需全部信息的数据库进行访问，数据具体存储文件的信息被数据库隐藏了，而且这些文件的具体操作和存储位置等细节信息也由数据库管理系统统一进行管理。

数据库管理与文件管理相比实际上是在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）之间增加了一层，即数据库管理系统。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，正是因为有了这个系统软件，才使得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，交给了这个系统软件来完成，这样应用程序或者用户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，这些变化都交给数据库管理系统来处理。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，应用程序也不需要进行任何修改。

与文件管理数据的局限性进行比较，数据库管理具有以下优点。

1. 相互关联的数据集成

在数据库系统中，所有相关的应用数据都存储在数据库环境中，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，将本书 1.1.2 节中的 F1 文件、F2 文件和 F3 文件中的重复数据挑选出来，进行合理的管理，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、联系电话、所在系、专业、班号。

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、修课类型、修课时间、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边介绍），重复的信息只存储一份，当在学生选课中需要学生的姓名等其他信息时，根据学生选课中的学号，可以很容易地在学生基本信息中找到此学号对应的姓名等信息。因此，消除数据的重复存储不影响对信息的提取，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，只需在“学生基本信息”一个地方进行修改即可。

同 1.1.2 节中的问题一样，当要检索班号，学号，姓名，课程名，学分，考试成绩信息时，这些信息也需要从 3 个地方（关系数据库为 3 张表）获取，也需要对信息进行适当的组合，即学生选课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，学生选课中的课程号也必须与课程基本信息中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，而现在有了数据库管理系统，这些烦琐的工作可以完全交给数据库管理系统来完成。

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，应用程序不再需要处理文件和记录的格式。

程序与数据相互独立有两个方面的含义。一方面是指当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），比如从链表结构改为哈希表结构，或者是顺序和非顺序之间的转换，应用程序不必作任何修改；另一方面是指当数据的逻辑结构发生变化时，比如增加或减少一

些数据项，如果应用程序与这些修改的数据项无关，则应用程序无需修改。这些变化都由 DBMS 负责维护。在大多数情况下，应用程序并不知道数据存储方式或数据项何时已经发生了变化。

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，以保证当数据遭到破坏时（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储到数据库中的数据必须符合现实世界的实际情况，比如人的性别只能是“男”或“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他的值，或者将一个负数输入到年龄中，在现实世界中显然是不对的。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，这就是数据的正确性，也称为数据完整性。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。当然这个特点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。

多用户共享问题是数据库管理系统内部解决的问题，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，保证多个用户之间对数据的操作不会产生矛盾和冲突，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。可以设想一下火车订票系统，如果多个订票点同时对某一天的同一列火车进行订票，那么必须要保证不同订票点订出票的座位不能重复。

数据库技术发展到今天已经是一门比较成熟的技术，经过上边的讨论，可以概括出数据库管理系统具备如下特征。

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、可靠，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固有的，而是靠数据库管理系统提供和保证的。

1.2 数据独立性

数据独立性包含两个方面，即逻辑独立性和物理独立性。物理独立性是指当数据的存储结构发生变化时（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（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，需要先搞清什么是非数据独立性。在数据库技术出现之前，也就是在使用文件管理数据的时候，实现的应用程序常常是数据依赖的，也就是数据的物理存储方式以及有关的存取技术都是在开发应用程序时需要考虑的方面，而且，数据的物理存储方式和访问技术直接体现在应用程序的代码中。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，也

要知道记录的顺序是索引的，应用程序的代码必须基于这些知识而设计和实现。我们称这样的应用程序是数据依赖的。在这种方式中，一旦数据的物理存储结构改变了，应用程序必须做相应的调整。比如，将数据的存储从顺序存储改为链式存储，则应用程序必须调整数据的访问方式。而修改是与数据管理密切联系的部分，与应用程序最初要解决的问题毫不相干。

在数据库系统中，尽量避免应用程序依赖于数据的情况，这有如下两个原因：

- 不同的用户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。例如我们在 1.1.2 节中介绍的文件系统很难按不同用户的要求显示数据的例子。如果基本数据发生了变化，势必要修改应用程序。
- 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的要求。比如，添加新的数据列，改变数据列的类型，或者增加新的存储设备等。理想的情况下，这些变化不应该影响应用程序。但遗憾的是，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，数据独立性的提出主要是客观应用的要求。数据独立性可以描述为：应用程序不会因数据的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。

1.3　数据库系统的组成

前面我们介绍了数据库系统具有的各种特征，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，一般包括 3 个主要部分：数据库、数据库管理系统和应用程序，如图 1-3 所示。其中，数据库是数据的汇集，它以一定的组织形式保存在存储介质上；DBMS 是管理数据库的系统软件，可以实现数据库系统的各种功能；应用程序指以数据库和数据库数据为基础的程序。

图 1-3　数据库系统简图

除了上述这些最基本的组成之外，数据库系统还需要运行这些软件所需的计算机硬件环境和操作系统环境的支持。硬件环境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。操作系统环境指数据库管理系统作为系统软件是建立在一定的操作系统环境之上的，没有合适的操作系统，数据库管理系统是无法运行的。

在数据库系统中还包括一部分内容，即用户。一般可将用户分为 3 类。第一类是系统管理员，这类人员负责数据库的规划、设计、协调、维护和管理等工作，主要保证数据库正确和高效的运行。第二类是应用程序开发人员，这类人员负责在某种环境下使用某个程序设计语言编写数据库应用程序。第三类用户是最终用户，是数据库应用程序的使用者，他们在联机工作站或终端上通过数据库应用程序完成与数据库的交互以及对数据的操作。

简单地说，数据库系统包括了以数据为主体的数据库，管理数据库的系统软件——数据库管理系统，支持数据库系统运行的计算机硬件环境和操作系统环境以及使用数据库系统的人。

1.4 数据库应用结构

数据库应用结构是指数据库运行的软硬件环境。通过这个环境，用户可以访问数据库中的数据，可以通过数据库内部环境访问数据库，也可以通过外部环境访问数据库，可以执行不同的操作。用户的目的也可以各不相同，可以查询数据、修改数据或者插入新的数据。

不同的数据库管理系统可以具有不同的应用结构。我们将介绍4种最常见的应用结构，即集中式应用结构、文件服务器结构、客户/服务器结构和互联网应用结构。

1.4.1 集中式应用结构

在20世纪60～70年代，数据库系统环境是大型机环境。大型机代表一种“集中式”的环境。这种环境主要由一台功能强大、允许多用户连接的计算机组成。多个哑终端（一般只包括显示器和键盘，没有存储处理的能力）通过网络连接到大型机上，并可以与大型机进行通信。终端一般只是大型机的扩展，它们并不是独立的计算机。终端本身并不能完成任何操作，主要依赖于大型机来完成所有的操作。用户从终端键盘键入的信息传到主机，然后由主机将执行的结果以字符方式返回到终端上。这个时期计算机的所有资源（数据）都在主机上，所有处理（程序）也在主机上完成。图1-4所示为大型机结构的数据访问模式。

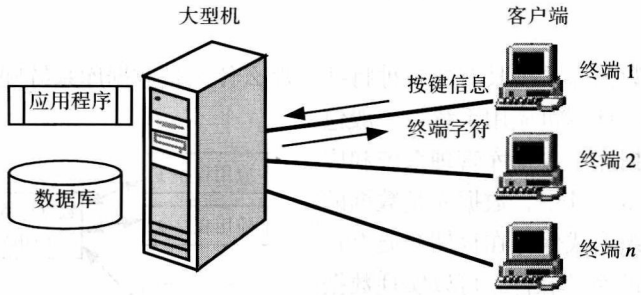


图1-4 大型机结构的数据访问模式

集中式应用结构的优点是可以实现集中管理，安全性好；但其缺点是费用昂贵，不能真正划分应用程序的逻辑。大型机的另一个主要问题就是对最终用户的限制，终端只能与大型机进行通信。而其他的一些任务，如用户的手工处理、字处理软件的使用或者是个人电脑等都无法与大型机交互。

1.4.2 文件服务器结构

到20世纪80年代，个人计算机进入了商用舞台，同时计算机应用的范围和领域也日趋广泛。这对那些没有能力实现大型机方案的企业来说，个人计算机无疑有了用武之地。在个人计算机进入商用领域不久，局域网也问世了，同时也诞生了文件服务器技术。图1-5所示为文件服务器结构的数据访问模式。

从图1-5可以看出，在文件服务器结构中，应用程序是在客户端的工作站上运行的，而不是在服务器上运行的，文件服务器只提供了资源（数据）的集中管理和访问途径。这种结构的特点是将共享数据资源集中管理，而将应用程序分布在各个客户工作站上。文件服务器结构的优点在于实现

的费用比较低廉，而且配置非常灵活。在一个局域网中可以方便地增减客户端工作站。文件服务器结构的缺点是，由于文件服务器只提供文件服务，所有的应用处理都要在客户端完成，这就意味着客户端的个人计算机必须要有足够的能力，以便执行需要的任何程序。这可能经常需要对客户端的计算机进行升级，否则就很难改进应用程序的功能或提高应用程序的性能。特别要提出的是，虽然应用程序可以存放在网络文件服务器的硬盘上，但它每次都要传送到客户端的个人计算机的内存中执行。另外，所有的处理都是在客户端完成的，因此网络上就要经常传送大量无用的数据。

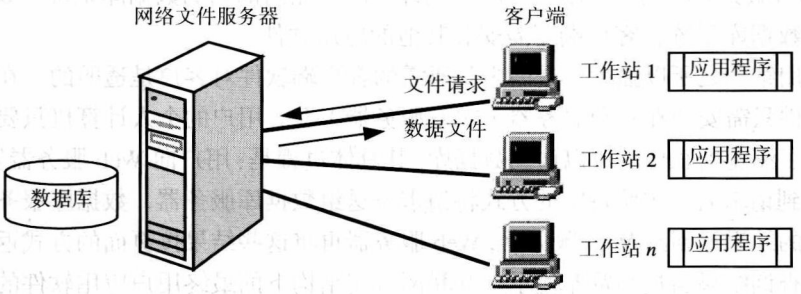


图 1-5　文件服务器结构的数据访问模式

Microsoft 的 FoxPro 就是曾经非常流行的支持文件服务器结构的数据库管理系统。

1.4.3　客户/服务器结构

文件服务器结构的费用虽然低廉，但是和大型机的“集中式”相比，它缺乏足够的计算和处理能力。为了解决费用和性能的矛盾，客户/服务器结构应运而生，这种结构允许应用程序分别在客户工作站和服务器（注意，不再是文件服务器）上执行，这样就可以合理地划分应用逻辑，充分发挥客户工作站和服务器两方面的性能。图 1-6 所示为客户/服务器结构的数据访问模式。

在客户/服务器结构中，应用程序或应用逻辑可以根据需要划分在服务器和客户工作站中。这样，为了完成一个特定的任务，客户工作站上的程序和服务器上的程序可以协同工作。从图 1-6 可以看出客户/服务器结构和文件服务器结构的区别。客户/服务器结构的客户工作站向服务器发送的是处理请求，而不是文件请求；服务器返回的是处理的结果，而不是整个文件，从而极大地减少了网络流量。

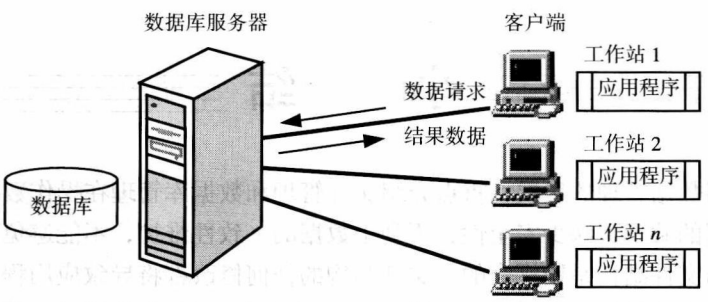


图 1-6　客户/服务器结构的数据访问模式

目前，常用的数据库管理系统都支持客户/服务器结构，如 Microsoft 的 SQL Server、Sybase、Oracle、DB2 等。

综上所述，大型机集中式应用结构的所有程序都在主机内执行，而文件服务器结构的所有程