网络数据库原理及应用

方睿　刁仁宏　吴四九　编著

四川大学出版社

需要全本请访问 www.ertongbook.com

前　言

随着计算机技术的不断发展，信息化管理程度的不断提高，数据库技术在信息管理中的作用日益重要。Microsoft SQL Server 2000 是目前使用最广泛的数据库，它与 Windows 网络操作系统的无缝集成，智能化的内容管理，强大的功能，使它得到大量用户的喜爱。

就目前市场上关于数据库的书籍来看，大致分为两大类：一类侧重于理论的介绍，以数据库理论为基础，重点在于数据库的系统结构、关系模型、关系代数、SQL 语言和规范化理论等，理论性强，实践不够；另一类以应用管理为主，主要介绍某种具体的操作，有点类似于操作手册，没有理论基础的人员不容易掌握。

本书总结作者多年从事数据库开发和教学的经验，结合作者讲授印度国际信息技术学院（NIIT）的数据库课程的一些体会，围绕一个实例展开，将网上玩具商店（ToyUniverse）的例子分解成各个不同的知识点，结合 Microsoft SQL Server 2000 中的各个具体技术，举出一个个例子来进行讲解。全书的实例以 ToyUniverse 为主线，让读者边学习理论边实践，易于读者掌握。

本书侧重于数据系统的开发，共分为三大部分。第一部分包括第 1 章，介绍数据库理论基础，这部分讲述数据库最基本的概念，是学习数据库的理论基础。第二部分包括第 2 章～第 8 章，主要介绍 Microsoft SQL Server 2000 的使用和管理，包括 SQL Server 2000 的安装，Transact-SQL 语言，数据库对象操作和管理，数据的查询和修改，数据库系统的安全性管理，实施数据完整性和常用实用工具等内容。在介绍本部分过程中，贯穿相应的数据库理论知识，使读者很容易将理论和实践结合起来。第三部分包括第 9 章，综合前面各章的内容，结合 Visual Studio .NET 2003 开发环境，给出一个范例（网上玩具商店），重点介绍怎样用 SQL Server 2000 和.NET 的 C＃开发一个 B/S 结构的应用程序。读者在学完本书后，可以依照第 9 章的提示，开发出自己的数据库系统。

本书条理清晰，概念准确，讲解详细，可作为大专院校相关专业的教材，也可作为数据库初学者、Microsoft SQL Server 2000 的培训教材或数据库管理人员和网络管理员的参考书。

在编写本书过程中得到了很多同事的关心和帮助，特别是得到了李飞、甘刚、陈灵、李中志、盛志伟、韩斌、苟智坚等教师的大力支持，在此表示深深地感谢。

鉴于作者水平有限，书中难免有错误和不妥之处，恳请读者批评指正。

编者

2005 年 8 月

目　录

第 1 章　数据库理论基础 1

1.1　数据库发展简史 1
1.1.1　数据管理的诞生 1
1.1.2　关系数据库的由来 2
1.1.3　结构化查询语言 3
1.1.4　面向对象数据库 3
1.2　数据库管理系统和数据库系统 4
1.2.1　文件管理系统 4
1.2.2　数据库管理系统 5
1.2.3　数据库系统 8
1.3　数据库系统的结构 9
1.3.1　三级模式结构 9
1.3.2　数据库的模式映像功能 12
1.4　数据库设计过程 13
1.4.1　数据和数据模型 13
1.4.2　概念层数据模型 16
1.4.3　组织层数据模型 19
1.4.4　E-R 模型转化为关系模型 22
1.5　关系数据库设计的规范化和非规范化 25
1.5.1　规范化设计 25
1.5.2　非规范化设计 31
1.6　数据库应用结构 32
1.6.1　客户机/服务器结构 32
1.6.2　互联网应用结构 32
思考题 34

第 2 章　SQL Server 2000 安装和常用工具 35

2.1　软件和硬件要求 35
2.1.1　SQL Server 2000 的版本 35
2.1.2　各个版本性能说明 36
2.1.3　操作系统要求 38
2.1.4　硬件要求 38
2.2　安装 SQL Server 2000 39

2.3 SQL Server 2000 常用工具简介 48
2.3.1 服务管理器 48
2.3.2 企业管理器 49
2.3.3 查询分析器 52
2.3.4 客户端和服务器网络实用工具 54
2.3.5 联机丛书 55
思考题 56
第 3 章 Transact-SQL 语言基础 58
3.1 SQL 概述 58
3.1.1 SQL 语言的特点 58
3.1.2 SQL 语言的组成 59
3.1.3 SQL 语句的结构 59
3.1.4 常用的 SQL 语句 59
3.2 Transact-SQL 数据类型 61
3.2.1 整数数据类型 62
3.2.2 浮点数据类型 63
3.2.3 二进制数据类型 64
3.2.4 逻辑数据类型 65
3.2.5 字符数据类型 65
3.2.6 文本和图形数据类型 65
3.2.7 日期和时间数据类型 66
3.2.8 货币数据类型 67
3.2.9 特定数据类型 68
3.2.10 用户自定义数据类型 68
3.2.11 新数据类型 68
3.3 Transact-SQL 变量 68
3.3.1 局部变量 69
3.3.2 全局变量 69
3.4 Transact-SQL 编程基础 70
3.4.1 注释符（Annotation） 70
3.4.2 运算符（Operator） 70
3.4.3 通配符（Wildcard） 71
3.5 Transact-SQL 流程控制语句 71
3.5.1 IF…ELSE 71
3.5.2 BEGIN…END 72
3.5.3 CASE 72
3.5.4 WHILE…CONTINUE…BREAK 73
3.5.5 WAITFOR 74

3.5.6 GOTO.....74
3.5.7 RETURN.....74
思考题.....76

第 4 章 数据库基本对象操作和管理.....77

4.1 数据库的创建和管理.....77
4.1.1 SQL Server 数据库特点.....77
4.1.2 创建数据库.....79
4.1.3 管理数据库.....84
4.2 表的创建和管理.....87
4.2.1 列的属性.....87
4.2.2 创建表.....89
4.2.3 用户自定义类型.....93
4.2.4 管理表.....95
4.3 索引的创建和管理.....99
4.3.1 索引的相关概念.....99
4.3.2 管理索引.....103
4.4 关系图的创建和管理.....107
思考题.....110

第 5 章 数据的查询和修改.....115

5.1 数据的查询.....115
5.1.1 显示表中的所有数据.....115
5.1.2 显示一张表上指定列的所有数据.....115
5.1.3 显示指定的、带用户友好的列标题的列.....116
5.1.4 用条件来筛选表中指定的行.....116
5.1.5 按指定顺序显示数据（排序）.....119
5.1.6 对查询的结果进行分组计算.....119
5.1.7 用 TOP 限制结果集.....121
5.1.8 汇总数据.....122
5.1.9 字符串函数.....125
5.1.10 日期函数.....127
5.1.11 数学函数.....128
5.1.12 模糊查询.....129
5.1.13 内联接.....130
5.1.14 外联接.....131
5.1.15 合并查询结果集.....132
5.1.16 子查询.....133
5.1.17 查询语句小结.....135

需要全本请访问 www.ertongbook.com

5.2 数据的修改 135
5.2.1 数据的插入 135
5.2.2 数据的更新 138
5.2.3 数据的删除 139
5.2.4 数据修改时的完整性检查 140
思考题 143
第 6 章 实施数据完整性 145
6.1 数据完整性的概念 145
6.1.1 实现数据完整性的方法 145
6.1.2 完整性约束条件的作用对象 146
6.2 约束 146
6.2.1 主关键字约束 146
6.2.2 外关键字约束 147
6.2.3 唯一性约束 149
6.2.4 检查约束 150
6.2.5 缺省约束 151
6.2.6 在企业管理器中创建约束 152
6.2.7 系统对约束的检查 153
6.3 规则 154
6.3.1 创建规则 154
6.3.2 查看规则 155
6.3.3 规则的绑定与松绑 156
6.3.4 删除规则 157
6.4 默认值 158
6.4.1 创建默认值 158
6.4.2 查看默认值 159
6.4.3 默认值的绑定与松绑 159
6.4.4 删除默认值 159
思考题 160
第 7 章 数据库高级对象操作和管理 162
7.1 视图 162
7.1.1 视图的概念 162
7.1.2 创建视图 163
7.1.3 管理视图 165
7.1.4 通过视图管理数据 167
7.1.5 索引视图 168
7.2 存储过程 169

7.2.1 存储过程的概念 169
7.2.2 创建存储过程 170
7.2.3 管理存储过程 175
7.2.4 系统存储过程 176
7.3 用户自定义函数 176
7.3.1 创建用户自定义函数 177
7.3.2 管理用户自定义函数 181
7.4 事务和锁 182
7.4.1 事务的概念 182
7.4.2 事务的回滚 183
7.4.3 锁的概念 184
7.4.4 查看锁 188
7.4.5 死锁及其防止 189
7.5 触发器 191
7.5.1 触发器的概念 191
7.5.2 创建触发器 193
7.5.3 INSTEAD OF 触发器 196
7.5.4 管理触发器 198
7.6 游标 199
7.6.1 游标的概念 199
7.6.2 游标的使用 200
7.6.3 管理游标 207
思考题 209
第 8 章 数据库系统的安全性管理 211
8.1 SQL Server 安全控制机制 211
8.1.1 数据库系统的安全控制模型 211
8.1.2 数据库权限和用户分类 212
8.1.3 SQL Server 的安全机制 212
8.1.4 设置 SQL Server 的认证模式 215
8.2 管理 SQL Server 登录账户 215
8.2.1 系统的登录账户 216
8.2.2 管理登录账户 216
8.3 管理数据库用户 218
8.3.1 数据库用户简介 218
8.3.2 管理数据库用户 218
8.4 管理权限 220
8.4.1 管理权限简介 220
8.4.2 权限的管理 221

8.5 管理角色.....222
8.5.1 固定的服务器角色.....222
8.5.2 固定的数据库角色.....224
8.5.3 用户自定义角色.....225
8.6 SQL Server 安全性管理的途径.....227
8.6.1 使用视图作为安全机制.....228
8.6.2 使用存储过程作为安全机制.....229
思考题.....229
第 9 章 ASP.NET+SQL Server 2000 开发网上玩具商店范例.....231
9.1 网上玩具商店解决方案.....231
9.2 .NET 与 SQL Server 2000 开发环境集成.....233
9.2.1 新建 Web 项目.....233
9.2.2 在服务器资源管理器中连接到 SQL Server.....233
9.2.3 管理数据库对象和数据.....234
9.3 网上玩具商店部分关键源代码分析.....235
9.3.1 创建应用程序首页 default.aspx.....235
9.3.2 .NET 应用的数据访问程序块.....236
9.3.3 ASP.NET 配置文件 Web.config.....241
9.4 范例程序的安装调试.....244
9.4.1 数据库的安装.....244
9.4.2 应用程序调试环境.....244
9.5 其他部分文件.....245
9.5.1 项目各文件夹中的内容.....246
9.5.2 项目各文件之间的导航关系.....247

第 1 章　数据库理论基础

数据库的诞生和发展给计算机信息管理带来了一场巨大的革命。数据管理经历了从手工管理阶段、文件管理阶段到数据库管理阶段的变迁。随着信息处理的日益发展，信息管理水平的不断提高，计算机管理数据方式的不断改进，数据库技术正逐步渗透到我们日常生活的各个方面。从超市的货物管理，书店的图书管理，飞机、火车的售票系统，网上购物，到关系到我们每个人身份的户籍管理、电信移动的通信管理，都离不开数据库技术。数据库技术正在不知不觉地影响着我们的生活。

有了大量的数据，还需要对这些数据进行有效科学的管理，合理的分析，才能服务于人。一个网上购物网站，经过长时间的运行，记录了大量的顾客消费的记录，不加分析，这些数据是毫无用处的。如果分析这些数据，得出顾客的消费习惯，例如某段时间内什么商品最好卖，什么商品最不好卖，那么，这些结果对商家是十分有用的。数据库技术就是研究对数据进行科学的管理，合理的分析，为人们提供安全、准确数据的技术。

本章重点介绍如何对数据进行科学的管理，主要介绍目前使用得最多的关系数据库的基本概念。

1.1　数据库发展简史

1.1.1　数据管理的诞生

数据库的历史可以追溯到 50 年前，那时的数据管理非常简单，通过大量的分类、比较和表格绘制，机器运行数百万穿孔卡片来进行数据的处理，其运行结果可在纸上打印出来或者制成新的穿孔卡片。数据管理就是对所有这些穿孔卡片进行物理的储存和处理。

1951 年，雷明顿兰德公司（Remington Rand Inc）的一种叫做 Univac I 的计算机推出了 1 秒钟可以输入数百条记录的磁带驱动器，从而引发了数据管理的革命。1956 年，IBM 生产出第一个磁盘驱动器——Model 305 RAMAC。此驱动器有 50 个盘片，每个盘片直径是 2 英尺，可以储存 5MB 的数据。使用磁盘最大的好处是可以随机地存取数据，而穿孔卡片和磁带只能顺序存取数据。

数据库系统的萌芽出现于 20 世纪 60 年代。当时计算机开始广泛地应用于数据管理，对数据的共享提出了越来越高的要求。传统的文件系统已经不能满足人们的需求了，能够统一管理和共享数据的数据库管理系统（DBMS）应运而生。数据模型是数据库系统的核心和基础，各种 DBMS 软件都是基于某种数据模型的。所以，通常也按照数据模型的特点将传统数据库系统分成网状数据库、层次数据库和关系数据库三类。

最早出现的是网状 DBMS。1961 年，通用电气公司（General Electric Co.）的 Charles

Bachman 成功地开发出世界上第一个网状 DBMS，也是第一个数据库管理系统——集成数据存储（Integrated DataStore IDS），奠定了网状数据库的基础，并在当时得到了广泛的发行和应用。IDS 具有数据模式和日志的特征，但它只能在 GE 主机上运行，并且数据库只有一个文件，数据库所有的表必须通过手工编码来生成。

之后，通用电气公司的一个客户——BF Goodrich Chemical 公司重写了整个系统，并将重写后的系统命名为集成数据管理系统（IDMS）。

网状数据库模型对于层次和非层次结构的事物都能比较自然地模拟，在关系数据库出现之前，网状 DBMS 要比层次 DBMS 用得普遍。在数据库发展史上，网状数据库占有重要的地位。

层次型 DBMS 是紧随网络型数据库而出现的。最著名、最典型的层次数据库系统是 IBM 公司在 1968 年开发的 IMS（Information Management System），这是一种适合其主机的层次数据库，是 IBM 公司研制的最早的大型数据库系统程序产品。从 20 世纪 60 年代末产生，到如今已经发展到 IMSV6，它提供群集、N 路数据共享、消息队列共享等先进特性的支持。这个具有 30 年历史的数据库产品在如今的 WWW 应用连接、商务智能应用中扮演着新的角色。

1973 年，Cullinane 公司（也就是后来的 Cullinet 软件公司）开始出售 Goodrich 公司的 IDMS 改进版本，并且逐渐成为当时世界上最大的软件公司。

1.1.2 关系数据库的由来

网状数据库和层次数据库已经很好地解决了数据的集中和共享问题，但是在数据的独立性和抽象级别上仍有很大缺陷。用户在对这两种数据库进行存取数据时，仍然需要明确数据的存储结构，指出存取路径，而关系数据库较好地解决了这些问题。

1969 年，E.F. Codd 发明了关系数据库。

关系模型有严格的数学基础，抽象级别较高，而且简单清晰，便于理解和使用。但是当时也有人认为关系模型是理想化的数据模型，用来实现 DBMS 是不现实的，尤其担心关系数据库的性能令人难以接受，更有人视其为是对当时正在进行中的网状数据库规范化工作的严重威胁。

1970 年，关系模型建立之后，IBM 公司在 San Jose 实验室增加了大量的研究人员研究这个项目，这个项目就是著名的 System R。其目标是论证一个全功能关系 DBMS 的可行性。该项目结束于 1979 年，完成了第一个实现 SQL（Structured Query Language，结构化查询语言）的 DBMS。然而 IBM 对 IMS 的承诺阻止了 System R 的投产，一直到 1980 年，System R 才作为一个产品正式推向市场。IBM 产品化步伐缓慢有三个原因：一是 IBM 重视信誉，重视质量，尽量减少故障；二是 IBM 是一个大公司，体系庞大；三是 IBM 内部已经有层次数据库产品，相关人员不积极，甚至反对。

1973 年，加州大学伯克利分校的 Michael Stonebraker 和 Eugene Wong 利用 System R 已发布的信息开始开发自己的关系数据库系统 Ingres。他们开发的 Ingres 项目最后由 Oracle 公司、Ingres 公司以及硅谷的其他厂商所商品化。后来，System R 和 Ingres 系统双双获得 ACM 的 1988 年“软件系统奖”。

1976 年，霍尼韦尔公司（Honeywell）开发了第一个商用关系数据库系统——Multics

Relational Data Store。关系型数据库系统以关系代数为坚实的理论基础，经过几十年的发展和实际应用，技术越来越成熟和完善。其代表产品有 Oracle、IBM 公司的 DB2、微软公司的 MS SQL Server 以及 Informix、ADABASD 等。

1.1.3　结构化查询语言

1974 年，IBM 的 Ray Boyce 和 Don Chamberlin 将 Codd 关系数据库的数学定义以简单的关键字语法表现出来，里程碑式地提出了 SQL 语言。SQL 语言的功能包括查询、操纵、定义和控制，是一个综合的、通用的关系数据库语言，同时又是一种高度非过程化的语言，只要求用户指出做什么，而不需要指出怎么做。SQL 集成实现了数据库生命周期中的全部操作。SQL 提供了与关系数据库进行交互的方法，它可以与标准的编程语言一起工作。自产生之日起，SQL 语言便成了检验关系数据库的试金石，而 SQL 语言标准的每一次变更都指导关系数据库产品的发展方向。然而，直到 20 世纪 70 年代中期，关系理论才通过 SQL 在商业数据库 Oracle 和 DB2 中使用。

1986 年，ANSI 把 SQL 作为关系数据库语言的美国标准，同年公布了标准 SQL 文本。目前 SQL 标准有 3 个版本。基本 SQL 定义是 ANSIX3135-89，“Database Language-SQL with Integrity Enhancement”[ANS89]，一般叫做 SQL-89。SQL-89 定义了模式定义、数据操作和事务处理。SQL-89 和随后的 ANSIX3168-1989，“Database Language-Embedded SQL”构成了第一代 SQL 标准。ANSIX3135-1992[ANS92]描述了一种增强功能的 SQL，现在叫做 SQL-92 标准。SQL-92 包括模式操作、动态创建、SQL 语句动态执行、网络环境支持等增强特性。在完成 SQL-92 标准后，ANSI 和 ISO 即开始合作开发 SQL3 标准。SQL3 的主要特点在于抽象数据类型的支持，为新一代对象关系数据库提供了标准。

1.1.4　面向对象数据库

随着信息技术和市场的发展，人们发现关系型数据库系统虽然技术很成熟，但其局限性也是显而易见的。它能很好地处理所谓的“表格型数据”，却对技术界出现的越来越多的复杂类型的数据无能为力。20 世纪 90 年代以后，技术界一直在研究和寻求新型数据库系统，但在什么是新型数据库系统的发展方向的问题上，产业界一度是相当困惑的。受当时技术风潮的影响，在相当一段时间内，人们把大量的精力花在研究“面向对象的数据库系统（Object Oriented database）”或简称“OO 数据库系统”。值得一提的是，美国 Stonebraker 教授提出的面向对象的关系型数据库理论曾一度受到产业界的青睐，而 Stonebraker 本人也在当时被 Informix 花大价钱聘为技术总负责人。

然而，数年的发展表明，面向对象的关系型数据库系统产品的市场发展的情况并不理想，理论上的完美性并没有带来市场的热烈反应。其不成功的主要原因在于，这种数据库产品的主要设计思想是企图用新型数据库系统来取代现有的数据库系统。这对许多已经运用数据库系统多年并积累了大量工作数据的客户，尤其是大客户来说，就无法承受新旧数据间的转换而带来的巨大工作量及巨额开支。另外，面向对象的关系型数据库系统使查询语言变得极其复杂，使得无论是数据库的开发商还是应用客户，都产生畏惧感。

一般把数据库系统分为以下三代：

（1）支持层次模型和网状模型的第一代数据库系统。

需要全本请访问 www.ertongbook.com

（2）支持关系模型的第二代数据库系统。

（3）支持面向对象的数据模型的第三代数据库系统。

1.2 数据库管理系统和数据库系统

本节主要介绍数据库系统的组成和数据库管理系统的功能。在了解数据库管理系统之前，我们先了解文件管理系统和数据库管理系统的区别。

1.2.1 文件管理系统

早期的数据管理是采用文件形式进行管理的，即数据保存在文件中，文件是由操作系统和特定的软件和程序共同管理的。在文件系统中，数据按其内容、结构和用途等分成若干个命名的文件，文件一般为某一个或某一组用户所有。用户可以通过操作系统和特定的程序来对文件进行打开、读、写等操作。

假设我们现在要用某种程序设计语言编写对玩具商店的信息进行管理的一个系统，玩具的信息、购物者的信息以及购物者购买玩具订单的信息等都以文件的形式保存起来。现在假设在此系统中，要对玩具的基本信息和购物者购买玩具订单的信息进行管理。在玩具基本信息管理中要用到玩具的基本信息数据，假设此数据保存在 File1 文件中。购物者购买玩具管理系统中要用到玩具的基本信息，购物者的基本信息，购物者购买玩具订单的信息。假设用 File2 存储购物者的基本信息，用 File3 存储购物者购买玩具订单的信息。此部分的玩具的基本信息同样用 File1 文件中的数据。假设实现“玩具基本信息管理”功能的应用程序叫 App1，实现“购物者购买玩具管理”功能的应用程序叫 App2，那么这两个应用程序和三个文件的关系如图 1-1 所示。

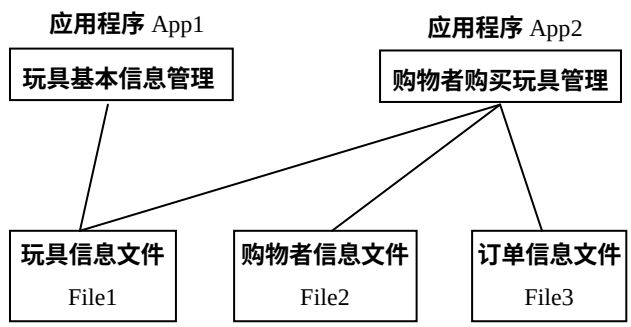


图 1-1 文件管理系统示例

假设 File1 包含玩具 ID、玩具名称、玩具描述、玩具价格、商标、照片、数量、玩具适合的最低年龄、最大年龄和玩具重量等，File2 包含购物者 ID、姓名、密码、邮件地址、地址、邮政编码、电话、信用卡编号、信用卡类型和截止日期等，File3 包含定单编号、购物者姓名、玩具 ID、玩具名称、运货方式、运货费用、礼品包装费用和定单处理等，则“购物者购买玩具管理”的处理过程大致如下：

在购物者购买玩具的过程中，若有购物者购买玩具，先查找文件 File2，判断此用户是否合法；如果合法，则访问 File1，判断有无此玩具；如果有，则将订单信息写到文件 File3 中。

一切看起来似乎很完美，但我们仔细考虑，就会发现基于这样的文件管理系统有以下一些缺点：

（1）编写应用程序不方便。程序员必须对所使用文件的逻辑结构和物理结构（文件中有多少个字段，每个字段的数据类型，采用什么存储结构等）有清楚的了解。对文件的查询、修改等处理都必须在程序中实现。

（2）应用程序的依赖性。在处理文件时，程序依赖于文件的格式。随着应用环境（例如操作系统的变化）和需求的变化，就需要修改文件结构，比如增加一个字段，修改某个字段的类型等，结果导致应用程序就要做相应的变化。频繁地修改、安装、调试应用程序是很麻烦的事，也就是说，文件管理系统的数据独立性差。

（3）不支持文件的并发访问。现代的计算机应用系统中，为了高效地利用资源，一般要求多个应用程序并发访问。在上例中，如果管理者在向玩具信息文件中添加玩具信息，同时又有购物者在购买玩具，就有 App1 和 App2 同时访问 File1 文件。如果第一个用户没有访问结束，第二个用户来访问时系统就会报错。

（4）数据间的耦合度差。在文件管理系统中，文件与文件之间是彼此相互独立的，文件之间的联系必须通过程序来实现。例如图 1-1 中的 File1 和 File3 文件，File3 中的玩具 ID 必须是在 File1 中已经存在的信息。这在现实中是一件自然的事，但文件管理系统本身并不具备这种耦合功能，必须由程序员编写程序来实现。手工编写比较繁琐，而且还容易出错。

（5）数据表示的单一化。如果用户需要的信息来自于多个不同文件的部分信息内容的组合，我们就需要对多个文件进行提取、比较、组合和表示。当数据量很大、涉及的表比较多时，这个过程很复杂。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。

（6）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权，比如只能读和修改数据而不能删除数据，或者对文件中的某个或某些字段不能读取或修改等。而在实际生活中，数据的安全性是非常重要且不可缺少的，就像在玩具管理中，我们不允许购物者修改玩具的价格一样。

随着人们对数据需求的增加和计算机科学的不断发展，如何能对数据进行有效、科学、正确、方便的管理已成为人们的迫切需求。针对文件系统的这些缺陷，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。

1.2.2　数据库管理系统

数据库技术的发展主要是用于克服文件管理系统在管理数据上的诸多缺陷。对于上述的玩具基本信息管理和购买玩具管理，如果使用数据库管理系统来实现，其实现方式与文件系统有很大区别，如图 1-2 所示。

比较图 1-1 和图 1-2，可以直观地发现两者有如下差别：

使用文件系统时，应用程序是直接访问存储数据的文件；而使用数据库系统时，则是

通过数据库管理系统（Database Management System，DBMS）访问数据。这个变化使得应用编程的工作变得很简单，因为应用程序开发人员不再需要关心数据的物理存储方式和存储结构，这些都交给了数据库管理系统来完成。

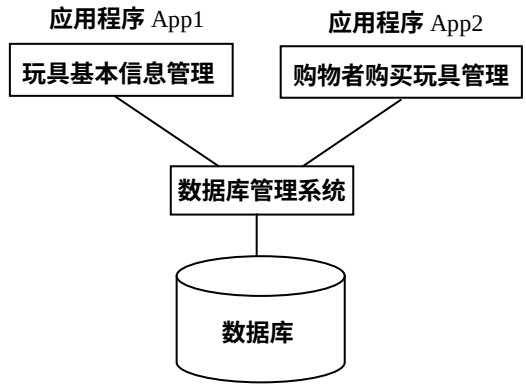


图 1-2　数据库管理系统示例

在数据库系统中，数据不再仅仅服务于某个程序或用户，而是成为一定业务范围的共享资源，由数据库管理系统软件统一管理。

数据库管理系统与文件系统相比，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，这样用户应用程序时不再需要关心数据的存储方式。另外，数据的存储方式的变化也不再影响应用程序，这些变化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，应用程序也不需要进行任何修改。

与文件系统管理数据的局限性进行比较，数据库系统具有下面一些优点。

（1）将相互关联的数据集成在一起。

在数据库系统中，所有的数据都存储在数据库中，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，将图 1-1 中的 File1、File2 和 Fiel3 文件中的重复数据挑选出来，单独进行管理，这样就可以形成以下几部分信息：

File1 包含：玩具 ID，玩具名称，玩具描述，玩具价格，商标，照片，数量，最低年龄，最大年龄，玩具重量等。

File2 包含：购物者 ID，姓名，密码，邮件地址，地址，邮政编码，电话，信用卡编号，信用卡类型，截止日期等。

File3 包含：定单编号，购物者 ID，玩具 ID，运货方式，运货费用，礼品包装费用，定单处理等。

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将在后

面介绍），重复的信息只存储一份。当我们在购买玩具时，需要玩具的名称，我们根据购物者购买玩具的 ID 号，可以很容易地在玩具基本信息中找到此 ID 对应的名称。因此，消除数据的重复存储而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，当某个玩具的描述发生变化时，我们只需在“玩具基本信息”进行修改即可。

（3）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。

多用户系统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。可以设想一下火车订票系统，如果多个订票点同时对一列火车进行订票操作，那么必须要保证不同订票点订出票的座位不能重复。

（4）程序与数据相互独立。

在数据库中，所有数据项以及数据的存储格式都存储其中，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数据文件的存储结构。

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（逻辑存储方式或物理存储方式）发生变化，比如从链表结构改为哈希结构，或者是顺序和非顺序之间的转换，不必对应用程序做任何修改。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。

（5）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的、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），能够很快将数据库恢复到正确的状态，并使数据不丢失或少丢失，从而保证系统能够连续、可靠地运行。

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，经过以上的讨论，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。

具体来说，一个数据库管理系统应该具备如下功能：

（1）数据定义功能：定义数据的结构、数据与数据之间的关联关系、数据的完整性约束等。

（2）数据操纵功能：实现对数据库中数据的操纵，包括插入、删除和修改数据。

（3）数据查询功能：实现灵活的数据查询功能，使用户可以方便地使用数据库中的数

据。

（4）数据控制功能：实现对数据库数据的安全性控制、完整性控制等各方面的控制功能。

（5）数据管理功能：实现数据库的备份和恢复。

（6）数据通信功能：在分布式数据库或提供网络操作功能的数据库中提供数据的通信功能。

除此之外，还有性能优化、并发控制等许多其他功能。

数据库管理系统（DBMS）是数据库系统的核心。上面的优点和功能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的。数据库管理系统是运行在操作系统之上的系统软件。数据库管理系统的任务就是对数据资源进行管理，并使之能为多个用户共享，同时保证数据的安全性、可靠性、完整性、一致性，保证数据的高度独立性。

1.2.3 数据库系统

上面讲述了数据库管理系统是一个系统软件，比如 SQL Server、Oracle、DB2 等都是著名的数据库管理系统软件。但有了数据库管理系统这个软件之后，并不意味着我们已经具备了数据库管理系统管理数据带来的优点，我们还必须在这个软件基础之上进行一些必要的工作，以便把数据库管理系统提供的功能发挥出来。首先，必须在这个系统中存放用户自己的数据，让数据库管理系统帮助我们把这些数据管理起来；其次，我们还应有对这些数据进行操作并让这些数据发挥应有作用的应用程序；最后，我们还需要一个维护整个系统正常运行的管理人员，比如当数据库出现故障或问题时，应该如何处理以使数据库恢复正常，这个管理人员就称为数据库系统管理员（DBA）。

一个完整的数据库系统是基于数据库的一个计算机应用系统，数据库系统一般包括五个主要部分：数据库、数据库管理系统、应用程序、数据库系统管理员和用户，如图 l-3 所示。

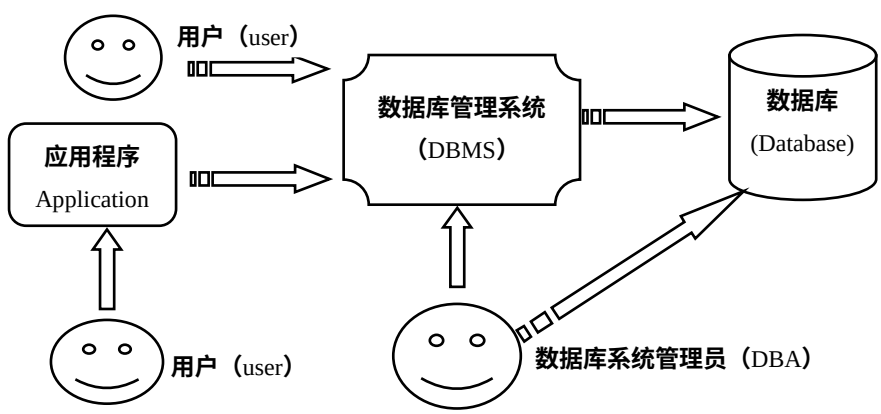


图 1-3 数据库系统组成

数据库是数据的集合，它以一定的组织形式存于存储介质上；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，是整个数据库系统的核心；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；数据库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；用户是使用数据库系统的一般人员。

数据库系统的运行还要有计算机的硬件和软件环境的支持，同时还要有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。软件环境是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上的，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 的企业版就需要服务器版的操作系统的支持。如图 1-4 所示。

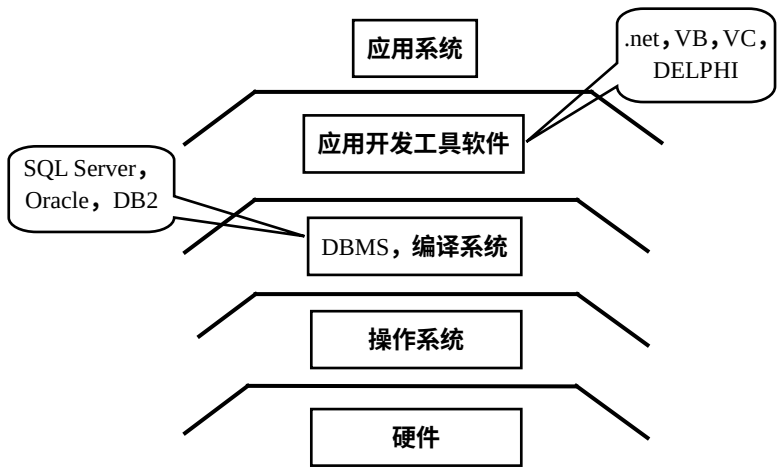


图 1-4　数据库系统软硬件层次

至此，我们可以看出，数据库、数据库管理系统和数据库系统是三个不同的概念，数据库强调的是数据，数据库管理系统是系统软件，而数据库系统强调的是整个应用系统。

1.3　数据库系统的结构

数据库系统的结构是一个框架结构，这个框架用于描述一般数据库系统的概念，但并不是说所有的数据库系统都一定使用这个框架。这个框架结构在数据库中并不是唯一的，特别是一些“小”的系统将难以支持这个体系结构的所有方面。这里我们介绍的数据库系统的体系结构能很好地适应大多数系统，而且它基本上和 ANSI/SPARC DBMS 研究组提出的数据库管理系统的体系结构（称为 ANSI/SPARC 体系结构）是相同的。本节主要介绍数据库系统的三级模式结构。

1.3.1　三级模式结构

模式是数据库中全体数据的逻辑结构和特征的描述，它涉及具体的数据值。数据库系统的三级模式结构是指数据库系统的外模式、模式和内模式，图 1-5 说明了各级模式之间的关系。广义地讲，内模式是最接近物理存储的，也就是数据的物理存储方式；外模式是最接近用户的，也就是用户所看到的数据视图；模式是介于内模式和外模式之间的中间层次。

需要全本请访问 www.ertongbook.com