

中等职业学校计算机系列规划教材  
根据教育部中等职业学校新教学大纲要求编写

# SQL Server 2000 数据库应用与开发

SQL Server 2000 SHUJUKU YINGYONG YU KAIFA

方 睿 李 飞 编著

北京工业大学出版社

中等职业学校计算机系列规划教材

# SQL Server 2000

## 数据库应用与开发

方 睿 李 飞 编著

北京工业大学出版社

## 内 容 提 要

全书围绕着一个网上玩具商店的范例展开,共分为三部分。第1章为第一部分,介绍数据库基础理论,讲解数据库基本概念,是学习数据库的理论基础。第2章~第8章为第二部分,主要介绍 Microsoft SQL Server 2000 的使用和管理,包括 Microsoft SQL Server 2000 的安装,Transact-SQL,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介绍等,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9章为第三部分,介绍一个基于 ASP.NET+SQL Server 2000 开发网上玩具商店的范例,也是将前面8章的例子综合起来实现一个完整的范例。

本书条例清晰,概念准确,讲解详细。可用做中等职业学校计算机等相关专业学生的教材,也可作为数据库初学者的培训教材,还可供数据库管理人员和网络管理员作参考。

### 图书在版编目(CIP)数据

SQL Server 2000 数据库应用与开发/方睿,李飞编著.

北京:北京工业大学出版社,2007.11

ISBN 978-7-5639-1789-1

I. S... II. ①方... ②李 III. 关系数据库—数据库管理系统, SQL Server 2000—专业学校—教材 IV.

TP311.138

中国版本图书馆 CIP 数据核字(2007)第 069718 号

## SQL Server 2000 数据库应用与开发

方睿 李飞 编著

\*

北京工业大学出版社出版发行

邮编:100022 电话:(010)67392308

各地新华书店经销

徐水宏远印刷有限公司印刷

\*

2007年11月第1版 2007年11月第1次印刷

787mm×1092mm 16开本 17印张 422千字

印数:1~4 000册

ISBN 978-7-5639-1789-1/T·304

定价:24.50元

# 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提高。Microsoft SQL Server 2000 是目前使用最广泛的数据库管理系统,由于它和 Windows 网络操作系统无缝集成,拥有智能化的内容管理等强大功能,因此得到大量用户的喜爱。

就目前市场上关于数据库的图书来看,大致分为两大类:一类是侧重于理论的介绍,以数据库理论为基础,重点在数据库的系统结构、关系模型、关系代数、SQL 程序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,主要介绍某种数据库管理系统的具体操作,有点类似于操作手册,没有相关理论基础的人员不易掌握。

本书总结作者多年来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,围绕一个范例展开,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,举出一个个例子来讲解。全书的例子以 ToyUniverse 为主线,让读者边学习理论边实践,易于读者掌握。

本书侧重于数据库系统的开发,全书分为三个部分。第 1 章为第一部分,介绍数据库基础理论,讲解数据库基本概念,是学习数据库的理论基础。第 2 章~第 8 章为第二部分,主要介绍 Microsoft SQL Server 2000 的使用和管理,包括 Microsoft SQL Server 2000 的安装,Transact-SQL,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介绍等,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 9 章为第三部分,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(ToyUniverse),重点介绍怎样用 Microsoft SQL Server 2000 和 .NET 的 C# 语言开发一个 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。

本书在编写过程中得到了很多同事的关心和帮助,特别是陈灵的积极参与和帮助,在此表示衷心感谢。

鉴于作者能力水平有限,书中难免有谬误和不妥之处,恳请广大读者批评指正。

编 者

2007 年 11 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 关系型数据库的由来 .....                 | 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 .....                   | 6  |
| 1.2.3 数据库系统 .....                     | 8  |
| 1.3 数据库系统的体系结构 .....                  | 9  |
| 1.3.1 三级模式结构 .....                    | 9  |
| 1.3.2 二级映像功能 .....                    | 11 |
| 1.4 数据库设计过程 .....                     | 12 |
| 1.4.1 数据和数据模型 .....                   | 13 |
| 1.4.2 概念模型 .....                      | 15 |
| 1.4.3 组织模型 .....                      | 18 |
| 1.4.4 E-R 模型转换为关系模型 .....             | 21 |
| 1.5 关系型数据库设计的规范化和非规范化 .....           | 23 |
| 1.5.1 规范化设计 .....                     | 24 |
| 1.5.2 非规范化设计 .....                    | 30 |
| 1.6 数据库应用结构 .....                     | 30 |
| 1.6.1 客户机/服务器结构 .....                 | 30 |
| 1.6.2 互联网应用结构 .....                   | 31 |
| 【本章小结】 .....                          | 32 |
| 【习题】 .....                            | 32 |
| 第 2 章 SQL Server 2000 的安装和常用工具 .....  | 37 |
| 2.1 SQL Server 2000 对计算机软、硬件的要求 ..... | 37 |
| 2.1.1 SQL Server 2000 的版本 .....       | 37 |
| 2.1.2 SQL Server 2000 各版本性能说明 .....   | 38 |
| 2.1.3 操作系统要求 .....                    | 40 |
| 2.1.4 硬件要求 .....                      | 41 |

|        |                              |    |
|--------|------------------------------|----|
| 2.2    | SQL Server 2000 的安装 .....    | 41 |
| 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  | 客户端和服务端网络实用工具 .....          | 53 |
| 2.3.5  | 联机丛书 .....                   | 54 |
|        | 【本章小结】 .....                 | 55 |
|        | 【习题】 .....                   | 55 |
| 第 3 章  | Transact-SQL 基础 .....        | 57 |
| 3.1    | SQL 概述 .....                 | 57 |
| 3.1.1  | SQL 的特点 .....                | 57 |
| 3.1.2  | SQL 的组成 .....                | 58 |
| 3.1.3  | SQL 的语句结构 .....              | 59 |
| 3.1.4  | 常用的 SQL 语句 .....             | 59 |
| 3.2    | Transact-SQL 数据类型 .....      | 60 |
| 3.2.1  | 整数数据类型 .....                 | 61 |
| 3.2.2  | 浮点数据类型 .....                 | 62 |
| 3.2.3  | 二进制数据类型 .....                | 63 |
| 3.2.4  | 逻辑数据类型 .....                 | 63 |
| 3.2.5  | 字符数据类型 .....                 | 63 |
| 3.2.6  | 文本和图形数据类型 .....              | 64 |
| 3.2.7  | 日期和时间数据类型 .....              | 65 |
| 3.2.8  | 货币数据类型 .....                 | 66 |
| 3.2.9  | 特定数据类型 .....                 | 66 |
| 3.2.10 | 用户自定义数据类型 .....              | 66 |
| 3.2.11 | 新数据类型 .....                  | 67 |
| 3.3    | Transact-SQL 变量 .....        | 67 |
| 3.3.1  | 局部变量 .....                   | 67 |
| 3.3.2  | 全局变量 .....                   | 68 |
| 3.4    | Transact-SQL 编程基础 .....      | 68 |
| 3.4.1  | 注释符 .....                    | 68 |
| 3.4.2  | 运算符 .....                    | 68 |
| 3.4.3  | 通配符 .....                    | 70 |
| 3.5    | Transact-SQL 流程控制语句 .....    | 70 |
| 3.5.1  | IF...ELSE .....              | 70 |
| 3.5.2  | BEGIN...END .....            | 71 |
| 3.5.3  | CASE .....                   | 71 |

|                                      |     |
|--------------------------------------|-----|
| 3.5.4 WHILE...CONTINUE...BREAK ..... | 71  |
| 3.5.5 WAITFOR .....                  | 72  |
| 3.5.6 GOTO .....                     | 73  |
| 3.5.7 RETURN .....                   | 73  |
| 【本章小结】 .....                         | 74  |
| 【习题】 .....                           | 74  |
| 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 数据库的特点 .....        | 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 .....                    | 86  |
| 4.2.1 列的属性 .....                     | 87  |
| 4.2.2 创建表 .....                      | 88  |
| 4.2.3 用户自定义类型 .....                  | 92  |
| 4.2.4 管理表 .....                      | 93  |
| 4.3 索引的创建和管理 .....                   | 97  |
| 4.3.1 索引的相关概念 .....                  | 97  |
| 4.3.2 管理索引 .....                     | 101 |
| 4.4 关系图的创建和管理 .....                  | 104 |
| 【本章小结】 .....                         | 106 |
| 【习题】 .....                           | 106 |
| 第5章 数据的查询和修改 .....                   | 114 |
| 5.1 数据的查询 .....                      | 114 |
| 5.1.1 显示表中的所有数据 .....                | 114 |
| 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、由用户命名的列标题的列 .....        | 115 |
| 5.1.4 用条件来筛选表中的行 .....               | 116 |
| 5.1.5 按指定顺序显示数据（排序） .....            | 118 |
| 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 .....                     | 121 |
| 5.1.9 字符串函数 .....                    | 124 |
| 5.1.10 日期函数 .....                    | 126 |
| 5.1.11 数学函数 .....                    | 127 |
| 5.1.12 模糊查询 .....                    | 128 |
| 5.1.13 内联接 .....                     | 129 |
| 5.1.14 外联接 .....                     | 130 |

|        |                    |     |
|--------|--------------------|-----|
| 5.1.15 | 合并查询结果集 .....      | 131 |
| 5.1.16 | 子查询 .....          | 132 |
| 5.1.17 | SELECT 语句小结 .....  | 134 |
| 5.2    | 数据的修改 .....        | 135 |
| 5.2.1  | 数据的添加 .....        | 135 |
| 5.2.2  | 数据的修改 .....        | 137 |
| 5.2.3  | 数据的删除 .....        | 139 |
| 5.2.4  | 数据修改时的完整性检查 .....  | 140 |
| 【本章小结】 | .....              | 141 |
| 【习题】   | .....              | 141 |
| 第 6 章  | 实施数据完整性 .....      | 150 |
| 6.1    | 数据完整性的概念 .....     | 150 |
| 6.1.1  | 实现数据完整性的方法 .....   | 150 |
| 6.1.2  | 完整性约束条件的作用对象 ..... | 151 |
| 6.2    | 约束 .....           | 151 |
| 6.2.1  | 主关键字约束 .....       | 151 |
| 6.2.2  | 外关键字约束 .....       | 152 |
| 6.2.3  | 唯一性约束 .....        | 154 |
| 6.2.4  | 检查约束 .....         | 155 |
| 6.2.5  | 默认约束 .....         | 156 |
| 6.2.6  | 在企业管理器中创建约束 .....  | 157 |
| 6.2.7  | 系统对约束的检查 .....     | 158 |
| 6.3    | 规则 .....           | 159 |
| 6.3.1  | 规则的创建 .....        | 159 |
| 6.3.2  | 规则的查看 .....        | 160 |
| 6.3.3  | 规则的绑定与松绑 .....     | 161 |
| 6.3.4  | 规则的删除 .....        | 162 |
| 6.4    | 默认值 .....          | 162 |
| 6.4.1  | 默认值的创建 .....       | 163 |
| 6.4.2  | 默认值的查看 .....       | 163 |
| 6.4.3  | 默认值的绑定与松绑 .....    | 163 |
| 6.4.4  | 默认值的删除 .....       | 164 |
| 【本章小结】 | .....              | 164 |
| 【习题】   | .....              | 164 |
| 第 7 章  | 数据库高级对象操作和管理 ..... | 170 |
| 7.1    | 视图 .....           | 170 |
| 7.1.1  | 视图的概念 .....        | 170 |



|       |                             |     |
|-------|-----------------------------|-----|
| 7.1.2 | 视图的创建 .....                 | 171 |
| 7.1.3 | 视图的管理 .....                 | 173 |
| 7.1.4 | 通过视图管理数据 .....              | 174 |
| 7.1.5 | 索引视图 .....                  | 175 |
| 7.2   | 存储过程 .....                  | 177 |
| 7.2.1 | 存储过程的概念 .....               | 177 |
| 7.2.2 | 存储过程的创建 .....               | 178 |
| 7.2.3 | 存储过程的管理 .....               | 183 |
| 7.2.4 | 系统存储过程 .....                | 183 |
| 7.3   | 用户自定义函数 .....               | 184 |
| 7.3.1 | 用户自定义函数的创建 .....            | 184 |
| 7.3.2 | 用户自定义函数的管理 .....            | 188 |
| 7.4   | 事务和锁 .....                  | 189 |
| 7.4.1 | 事务的概念 .....                 | 189 |
| 7.4.2 | 事务的回滚 .....                 | 190 |
| 7.4.3 | 锁的概念 .....                  | 191 |
| 7.4.4 | 锁的查看 .....                  | 194 |
| 7.4.5 | 死锁及其防止 .....                | 196 |
| 7.5   | 触发器 .....                   | 197 |
| 7.5.1 | 触发器的概念 .....                | 198 |
| 7.5.2 | 触发器的创建 .....                | 199 |
| 7.5.3 | INSTEAD OF 触发器 .....        | 202 |
| 7.5.4 | 触发器的管理 .....                | 204 |
| 7.6   | 游标 .....                    | 205 |
| 7.6.1 | 游标的概念 .....                 | 205 |
| 7.6.2 | 游标的使用 .....                 | 206 |
| 7.6.3 | 游标的管理 .....                 | 212 |
|       | 【本章小结】 .....                | 214 |
|       | 【习题】 .....                  | 214 |
| 第 8 章 | 数据库系统的安全性管理 .....           | 228 |
| 8.1   | SQL Server 安全控制机制 .....     | 228 |
| 8.1.1 | 数据库系统的安全控制模型 .....          | 228 |
| 8.1.2 | 数据库权限和用户分类 .....            | 229 |
| 8.1.3 | SQL Server 的安全机制 .....      | 229 |
| 8.1.4 | 设置 SQL Server 的身份验证模式 ..... | 232 |
| 8.2   | 管理 SQL Server 登录账户 .....    | 232 |
| 8.2.1 | 内置系统账户 .....                | 232 |
| 8.2.2 | 登录账户的管理 .....               | 233 |

|       |                                          |     |
|-------|------------------------------------------|-----|
| 8.3   | 管理数据库用户 .....                            | 234 |
| 8.3.1 | 数据库用户简介 .....                            | 234 |
| 8.3.2 | 数据库用户的管理 .....                           | 235 |
| 8.4   | 权限的管理 .....                              | 236 |
| 8.4.1 | 权限管理简介 .....                             | 236 |
| 8.4.2 | 权限的管理 .....                              | 237 |
| 8.5   | 管理角色 .....                               | 238 |
| 8.5.1 | 固定的服务器角色 .....                           | 238 |
| 8.5.2 | 固定的数据库角色 .....                           | 240 |
| 8.5.3 | 用户自定义角色 .....                            | 241 |
| 8.6   | SQL Server 安全性管理的途径 .....                | 243 |
| 8.6.1 | 使用视图作为安全机制 .....                         | 243 |
| 8.6.2 | 使用存储过程作为安全机制 .....                       | 244 |
|       | 【本章小结】 .....                             | 244 |
|       | 【习题】 .....                               | 244 |
| 第9章   | ASP.NET+SQL Server 2000 开发网上玩具商店范例 ..... | 246 |
| 9.1   | 网上玩具商店解决方案 .....                         | 246 |
| 9.2   | .NET 与 SQL Server 2000 开发环境集成 .....      | 248 |
| 9.3   | 网上玩具商店部分关键源代码分析 .....                    | 249 |
| 9.3.1 | 创建应用程序首页 .....                           | 249 |
| 9.3.2 | .NET 应用程序的数据访问程序块 .....                  | 251 |
| 9.3.3 | ASP.NET 配置文件 web.config .....            | 255 |
| 9.4   | 范例程序的安装调试 .....                          | 258 |
| 9.4.1 | 数据库的安装 .....                             | 258 |
| 9.4.2 | 应用程序调试环境 .....                           | 258 |
| 9.5   | 其他部分文件 .....                             | 259 |
| 9.5.1 | 项目各文件夹里面的内容 .....                        | 259 |
| 9.5.2 | 项目各文件之间的导航关系 .....                       | 261 |

# 第 1 章 数据库理论基础

## 【学习目标】

- (1) 了解数据库发展简史。
- (2) 了解数据库、数据库管理系统、数据库系统和数据库系统体系结构。
- (3) 理解信息的三个领域：现实世界、信息世界和计算机世界。
- (4) 理解关系模型、层次模型的概念。
- (5) 了解数据库技术的研究领域。

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。随着信息处理技术的日益发展，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人们日常生活的各个方面。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，关系到每个人身份的户籍管理以及电信移动的通信管理等都离不开数据库技术。数据库技术正在不知不觉地影响着人们的生活。

有了大量的数据，还需要对这些数据进行合理的分析，才能使数据服务于人。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记录。若不加分析，这些数据毫无用处。而通过分析这些数据，得出顾客的消费习惯，例如某段时间内什么商品最好卖，什么最不好卖，无疑这些结果对商家十分有用。数据库技术就是研究如何对数据进行科学的管理，合理的分析，为人们提供安全、准确数据的一种技术。

## 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，再制成穿孔卡片，从而放到计算机上处理数据，其运行结果打印在纸上或者制成新的穿孔卡片。而数据管理就是对所有这些穿孔卡片进行物理的存储和处理。

然而，1951 年雷明顿·兰德公司 (Remington Rand Inc.) 推出了一种 1s 可以输入数百条记录的磁带驱动器并用于所生产的 Univac I 计算机，从而引发了数据管理的革命。1956 年 IBM 公司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 2ft (约 0.6m)，可以存储 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领域，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(Database Management System, DBMS)应运而生。数据模型是数据库系统的核心和基础,各种数据库管理系统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、层次型数据库和关系型数据库三类。

最早出现的是网状数据库管理系统,1961年通用电气公司(General Electric Co.)的查尔斯·巴克曼(Charles Bachman)成功地开发出世界上第一个网状数据库管理系统,这也是第一个数据库管理系统——集成数据存储(Integrated Data Store, IDS)。它奠定了网状数据库的基础,并在当时得到了大量的发行和广泛的应用。IDS具有数据模式和日志的特征,但它只能在通用电气公司的主机上运行,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公司最终不得不重写整个系统。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,在关系型数据库出现之前网状数据库管理系统要比层次型数据库管理系统用得普遍。在数据库发展史上,网状数据库占有重要的地位。

层次型数据库管理系统是紧随网状数据库而出现的。最著名的层次型数据库系统是IBM公司在1968年开发的IMS(Information Management System),一种适合其主机的层次型数据库。这是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,到今天已经发展到IMS V6,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多年历史的数据库产品在如今的WWW应用、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BF Goodrich Chemical公司的IDMS改进版,并且逐渐成为当时世界上最大的软件公司。

### 1.1.2 关系型数据库的由来

网状数据库和层次型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。当用户在对这两种数据库进行数据存取时,仍然需要明确数据的存储结构,指出存取路径。而关系型数据库较好地解决了这些问题。

1969年E.F. Codd发明了关系型数据库。

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,用来实现数据库管理系统是不现实的。尤其担心关系型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来研究这个项目,这个项目就是著名的System R。其目标是论证一个全功能关系型数据库管理系统的可行性。该项目结束于1979年,完成了第一个实现结构化查询语言(Structured Query Language, SQL)的数据库管理系统。然而IBM公司对IMS的承诺阻止了System R的投产,一直到1980年System R才作为一个商品正式推向市场。

1973年美国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型数据库系统——Ingres。最后他们开发的Ingres项目由

Oracle 公司、Ingres 公司以及硅谷的其他厂商进一步商品化。后来，System R 和 Ingres 系统双双获得 ACM（美国计算机协会）1988 年的“软件系统奖”。

1976 年霍尼韦尔（Honeywell）公司开发了第一个商用关系型数据库系统——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、Microsoft 公司的 Microsoft SQL Server 以及 Informix、ADABASD 等。

### 1.1.3 结构化查询语言

1974 年，IBM 公司的 Ray Boyce 和 Don Chamberlin 将 E.F.Codd 的关系型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。SQL 的功能包括数据查询、数据操纵、数据定义和数据控制，是一个综合的、通用的关系型数据库语言，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 集成数据库生命周期中的全部操作。SQL 提供了与关系型数据库进行交互的方法，它可以与标准的程序设计语言一起工作。自产生之日起，SQL 便成了检验关系型数据库的试金石，而 SQL 标准的每一次变更都指导关系型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（美国国家标准学会）把 SQL 作为关系型数据库语言的美国标准，同年公布了标准 SQL 文本。目前 SQL 标准有三个版本。ANSIX3135-89，“Database Language - SQL with Integrity Enhancement” [ANS89]，给出了基本 SQL 的定义，一般称做 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，现在称做 SQL-92 标准。SQL-92 包括模式操作，动态创建和执行 SQL 语句，网络环境支持等增强特性。在完成 SQL-92 标准后，ANSI 和 ISO（国际标准化组织）即开始合作开发 SQL3 标准。SQL3 的主要特点在于对抽象数据类型的支持，为新一代对象关系型数据库提供了标准。

### 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，美国的 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 公司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在市场的发展情况并不理想。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

主要设计思想是试图用新型数据库系统来取代现有的数据库系统。这对许多已经多年运用数据库系统并积累了大量工作数据的客户，尤其对大客户来说，是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的。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，都对其产生畏惧感。

一般把数据库系统分为三代：

- (1) 支持层次模型和网状模型的第一代数据库系统。
- (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。

在介绍数据库管理系统之前，下面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

早期的数据管理采用文件形式进行，即数据保存在文件中，文件由操作系统和特定的软件及程序共同管理。在文件系统中，数据按其内容、结构和用途等分成若干个文件。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对文件进行打开、读、写等操作。

假设现在要用某种程序设计语言编写对网上玩具商店的信息进行管理的一个系统。玩具的基本信息、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。即用 File1 存储玩具的基本信息，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。现在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；在购物者购买玩具管理中要用到玩具的基本信息，购物者的基本信息和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 中的数据。设实现“玩具基本信息管理”功能的应用程序是 App1，实现“购物者购买玩具管理”功能的应用程序是 App2，则这两个应用程序和三个文件的关系如图 1-1 所示。

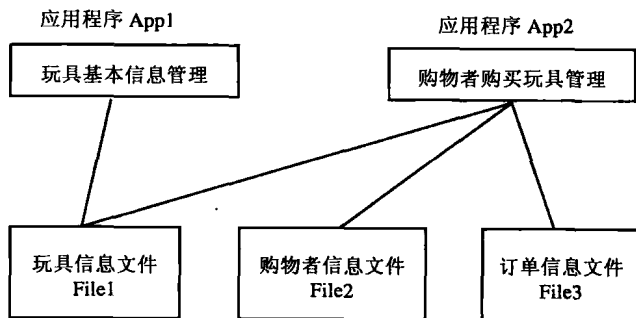


图 1-1 文件管理系统示例

若

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，玩具质量等字段。

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，截止日期等字段。

File3 包含：订单编号，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，订单处理等字段。

则对购物者购买玩具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 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；如果有此玩具，则将订单信息写到 File3 中。

一切看起来似乎很完美，但仔细考虑一下，就会发现在基于这样的文件管理系统中有如下一些缺点：

(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) 应用程序的依赖性强。在处理文件时，应用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，文件管理系统中数据的独立性差。

(3) 不支持文件的并发访问。在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，就会出现 App1 和 App2 同时访问 File1 的情况。从而使系统报错。

(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独立的，文件之间的联系必须通过应用程序来实现。例如图 1-1 中的 File1 和 File3，File3 中的玩具 ID 必须是在 File1 中已经存在的信息。这在现实中是件很必然的事，但文件管理系统本身并不具备这种耦合功能，必须由程序员编写应用程序来实现。而手工编写应用程序比较烦琐，而且容易出错。

(5) 数据表示单一化。如果用户需要的信息来自于多个不同文件的部分信息时，文件管理系统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，这个过程就会很复杂。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，很难控制用户对文件的操作权，比如只能读和修改数据而不能删除数据，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的，就像在玩具商店管理系统中，不能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，也随着计算机科学技术的不断发展，对数据进行有效、科学、正确、方便地管理已成为人们的迫切需求。针对文件管理系统的这些缺陷，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

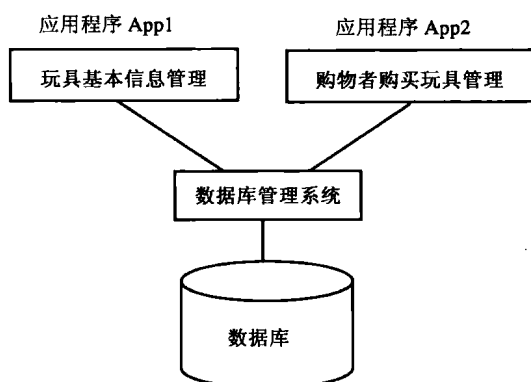


图 1-2 数据库管理系统示例

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。对于上述玩具的基本信息管理和购物者购买玩具管理，如果使用数据库管理系统来实现，其实现方式与文件管理系统有很大区别，如图 1-2 所示。

比较图 1-1 和图 1-2，可发现两者有如下差别：

使用文件管理系统时，应用程序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访问数据。这个变化使得应用程序编写的工作变得很简单，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。

在数据库系统中，数据不再仅仅服务于某个应用程序或用户，而是看成一定业务范围内的共享资源，由数据库管理系统统一管理。

数据库系统与文件管理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烦琐的操作和功能，交给了这个系统软件，这样应用程序不再需要处理数据的存储方式。反之，数据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，应用程序也不需要再进行任何修改。

与文件管理系统管理数据的局限性进行比较，数据库系统有如下优点：

（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，应用程序可通过数据库管理系统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 的 File1、File2 和 File3 中的重复数据挑选出来，单独进行管理，这样就可以形成如下的几部分信息。

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，玩具质量等字段。

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，截止日期等字段。

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

在关系型数据库中，可以将每一种信息存储在一张表中（关系型数据库的概念将在后边章节介绍），重复的信息只存储一份，当用户在购买玩具需要玩具的名称时，可根据用户购买玩具的 ID 号，很容易地在玩具基本信息中找到此 ID 对应的玩具名称。因此，消除数据的重复存储并不影响用户对信息的提取，同时还可以避免由于数据重复存储而造成数据不一致的



问题。比如，当某个玩具的描述发生变化时，程序开发人员只需在“玩具基本信息”一个地方进行修改即可。

(3) 数据库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管理系统而言的，对于单用户系统而言，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。这就要求数据库管理系统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，那么必须保证不同订票点出票的座位号不能重复。

数据集成与数据共享是大型环境中数据库管理系统的主要优点。

数据库系统技术发展到今天已经是一门比较成熟的技术了，经过以上讨论，可以概括出数据库系统的如下特征：

数据库系统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，能有效、及时地处理数据，并能保证数据的一致性和完整性。

应用程序与数据相互独立。在数据库系统中，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，它们通过数据库管理系统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的存储方式。

应用程序与数据相互独立有两个方面的含义，一方面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，应用程序不必作任何修改。另一方面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。这些变化都由数据库管理系统负责维护。大多数情况下，应用程序并不知道数据的存储方式或数据项已经发生了变化。

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；数据库系统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），能够很快将数据库恢复到正常状态，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。

数据库管理系统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，还要保证数据的高度独立性。

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。