前　言

人类社会已步入信息时代，而信息时代最重要的特点就是信息技术的广泛使用。充分利用各项信息资源离不开处理信息和管理数据的高科技手段，数据库技术就是在这样的社会需求背景下成为信息时代的基础。

作者编写本书的目的，是为了使读者能够更好地掌握数据库的基本原理知识，熟悉常用的关系数据库管理系统，了解数据库应用系统的开发设计方法。第 1 章主要介绍数据库的相关概念、数据库技术的发展及数据模型。第 2 章介绍关系数据库的基础理论，包括关系模型的相关概念、关系操作的两种实现方式，即关系代数与关系演算。第 3 章介绍数据库设计的 6 个基本阶段、数据库设计理论与设计方法。第 4 章介绍标准通用查询语言 SQL。第 5 章介绍关系数据库规范化理论，包括函数依赖、范式的概念及分类、关系模式的分解算法。第 6 章介绍新一代的关系数据库管理系统 SQL Server 2005，包括软件的安装、主要管理工具和数据库的相关操作。第 7 章介绍 Transact-SQL 语言的内容，包括 Transact-SQL 语言基础、主要的流程控制语句和函数的使用。第 8 章介绍触发器和存储过程的内容，包括基本概念、使用特点和相关操作。第 9 章介绍 SQL Server 2005 数据库安全的内容，包括 SQL Server 2005 数据库安全控制机制及主要的安全控制工具。第 10 章通过一个具体的数据库应用系统实例介绍开发数据库软件的方法和主要技术。

本书概念清楚，内容安排合理，既有数据库的基本原理和方法的详细说明，又翔实地介绍了关系数据库管理系统 SQL Server 2005。本书注重理论与实践相结合，使学习者既能掌握基本的数据库理论，也能提高数据库系统应用与技术开发的水平。

本书由王琬茹编写第 1 章、第 4 章、第 8 章和第 9 章，温永刚编写第 2 章、第 3 章和第 5 章，孙杰编写第 7 章和第 8 章，山鹰编写第 10 章。

由于时间仓促与水平有限，书中难免有错误及不妥之处，恳请广大读者提出宝贵意见。

编　者

2008 年 3 月

需要全本请访问 www.ertongbook.com

目　　录

第 1 章　数据库系统概述 …… 1
1.1　概述 …… 1
1.1.1　数据库技术的基本概念 …… 1
1.1.2　数据库技术的应用及发展 …… 4
1.1.3　Web 网络数据库系统编程简介 …… 5
1.2　数据库技术的发展阶段 …… 6
1.2.1　人工管理阶段 …… 6
1.2.2　文件系统管理阶段 …… 7
1.2.3　数据库系统管理阶段 …… 8
1.3　数据库系统的体系结构 …… 9
1.3.1　模式的概念 …… 9
1.3.2　三级模式结构 …… 10
1.3.3　数据库模式的映像 …… 11
1.4　数据模型 …… 12
1.4.1　现实世界的信息化过程 …… 12
1.4.2　概念模型 …… 13
1.4.3　数据模型及其组成要素 …… 17
1.5　三种常见的数据模型 …… 18
1.5.1　层次模型 …… 18
1.5.2　网状模型 …… 20
1.5.3　关系模型 …… 21
习题 …… 23
第 2 章　关系数据库基础 …… 24
2.1　关系模型的基本概念 …… 24
2.1.1　关系模型 …… 24
2.1.2　数据结构 …… 27
2.1.3　关系操作 …… 27
2.1.4　关系的完整性规则 …… 28
2.2　关系代数 …… 30
2.2.1　传统的集合运算 …… 31
2.2.2　专门的关系运算 …… 32
2.2.3　关系数据检索实例 …… 36
2.3　关系演算 …… 38

2.3.1 元组关系演算语言 …… 38
2.3.2 域关系演算语言 …… 42
习题 …… 48
第 3 章 数据库设计 …… 49
3.1 数据库设计概述 …… 49
3.1.1 数据库设计内容及特点 …… 49
3.1.2 数据库设计的方法 …… 50
3.1.3 数据库设计的基本步骤 …… 51
3.2 需求分析 …… 53
3.2.1 需求分析的任务 …… 53
3.2.2 需求分析的方法 …… 55
3.3 概念结构设计 …… 58
3.3.1 概念结构的设计方法 …… 58
3.3.2 数据抽象与局部视图设计 …… 60
3.3.3 视图的集成 …… 61
3.4 逻辑结构设计 …… 64
3.4.1 概念模型向关系模型的转换 …… 64
3.4.2 数据模型的优化 …… 68
3.4.3 用户子模式的设计 …… 69
3.5 数据库的物理设计 …… 69
3.5.1 确定数据库的物理结构 …… 70
3.5.2 评价物理结构 …… 72
3.6 数据库实施 …… 72
3.7 数据库运行和维护 …… 74
习题 …… 75
第 4 章 结构化查询语言 SQL …… 77
4.1 SQL 概述 …… 77
4.1.1 SQL 的特点 …… 77
4.1.2 SQL 数据库的三级模式结构 …… 78
4.2 SQL 的数据定义 …… 79
4.2.1 基本表的定义 …… 79
4.2.2 索引的定义 …… 83
4.3 SQL 的数据查询 …… 84
4.3.1 简单查询 …… 86
4.3.2 连接查询 …… 91
4.3.3 嵌套查询 …… 94
4.3.4 组合查询 …… 98
4.4 SQL 的数据更新 …… 99
4.4.1 插入语句 …… 99

4.4.2 删除语句 101
4.4.3 修改语句 101
4.5 视图 102
4.5.1 视图的定义和删除 103
4.5.2 视图的查询和维护 104
4.6 SQL 的数据控制 104
习题 106
第 5 章 关系数据库规范化理论 108
5.1 规范化问题的提出 108
5.2 函数依赖 110
5.2.1 函数依赖的定义 110
5.2.2 函数依赖与属性 111
5.3 范式 112
5.3.1 第一范式（1NF） 112
5.3.2 第二范式（2NF） 113
5.3.3 第三范式（3NF） 113
5.3.4 BCNF 范式 114
5.3.5 多值函数依赖关系的第四范式（4NF） 115
5.3.6 规范化小结 117
5.4 关系模式分解算法 117
5.4.1 关系模式的分解 117
5.4.2 无损分解的定义和性质 118
5.4.3 无损连接的检验方法 119
5.4.4 保持函数依赖的分解 121
习题 121
第 6 章 SQL Server 2005 关系数据库系统 123
6.1 SQL Server 2005 简介 123
6.1.1 SQL Server 2005 新增功能 123
6.1.2 SQL Server 2005 产品系列 126
6.1.3 SQL Server 2005 的系统需求 127
6.1.4 SQL Server 2005 的安装过程 129
6.2 SQL Server 2005 系统结构 131
6.3 SQL Server 2005 主要管理工具 133
6.3.1 SQL Server Management Studio 133
6.3.2 SQL Server Configuration Manager 139
6.3.3 Analysis Services 140
6.3.4 Database Engine Tuning Advisor 141
6.3.5 Business Intelligence Development Studio 142
6.3.6 SQL Server Profiler 142

6.3.7 文档和教程 ····· 143
6.4 SQL Server 2005 数据库操作 ····· 145
6.4.1 创建数据库 ····· 145
6.4.2 修改数据库 ····· 150
6.4.3 删除数据库 ····· 156
6.4.4 管理数据表 ····· 157
6.4.5 数据完整性规则 ····· 162
6.4.6 索引 ····· 162
6.4.7 视图 ····· 168
习题 ····· 173
第 7 章 Transact-SQL ····· 174
7.1 Transact-SQL 语言基础 ····· 174
7.1.1 Transact-SQL 语法约定 ····· 174
7.1.2 Transact-SQL 数据类型 ····· 175
7.1.3 变量 ····· 178
7.1.4 运算符 ····· 179
7.2 流程控制语句 ····· 182
7.2.1 IF...ELSE 语句 ····· 182
7.2.2 BEGIN...END 语句 ····· 183
7.2.3 WHILE, BREAK 和 CONTINUE 语句 ····· 184
7.2.4 RETURN 语句 ····· 185
7.2.5 WAITFOR 语句 ····· 186
7.2.6 无条件转移语句 ····· 187
7.2.7 TRY...CATCH 语句 ····· 187
7.2.8 GO 语句 ····· 188
7.3 函数 ····· 188
7.3.1 聚合函数 ····· 189
7.3.2 数学函数 ····· 191
7.3.3 字符串函数 ····· 194
7.3.4 系统函数 ····· 198
7.3.5 日期和时间函数 ····· 201
7.3.6 文本和图像函数 ····· 203
习题 ····· 204
第 8 章 存储过程和触发器 ····· 205
8.1 存储过程 ····· 205
8.1.1 存储过程概述 ····· 205
8.1.2 存储过程的分类 ····· 206
8.1.3 用户存储过程的创建与执行 ····· 207
8.1.4 用户存储过程的查看和修改 ····· 212

需要全本请访问 www.ertongbook.com

8.1.5 用户存储过程的删除 213
8.2 触发器 214
8.2.1 触发器概述 214
8.2.2 触发器的创建 216
8.2.3 触发器的修改和删除 219
习题 221
第 9 章 数据库的安全保护 222
9.1 数据库安全性控制 222
9.1.1 SQL Server 身份验证模式 222
9.1.2 设置身份验证方式 224
9.2 登录名和角色 226
9.2.1 管理登录 226
9.2.2 角色管理 234
9.3 权限管理 243
习题 248
第 10 章 SQL Server 2005 应用 249
10.1 学生管理系统的设计 249
10.1.1 需求分析及主要任务 249
10.1.2 概念结构设计 250
10.1.3 逻辑结构设计 251
10.1.4 物理结构设计 251
10.1.5 数据库的实现 252
10.2 PowerBuilder 数据库应用程序的设计与实现 254
参考文献 270

第 1 章　数据库系统概述

数据库技术是计算机科学技术的一个重要分支，尽管数据库技术的产生与发展只有几十年的时间，但是数据库的理论和应用却取得了较大的突破。数据库技术作为当前信息社会中对大量数据进行组织与管理的重要技术手段和软件技术，伴随着网络技术的不断发展，已成为网络信息化管理系统的基础。本章主要介绍数据库技术的应用和发展，数据库的基本概念和特点，数据库系统的体系结构，以及数据模型的相关内容和三种常用数据模型。

1.1　概述

数据库技术是随着信息社会对数据处理业务的需要而产生的。在过去的几十年中，数据库技术应用的发展和重要作用有目共睹，数据库技术作为信息系统核心和基础的地位日益凸显重要。从小型单项事务处理系统到大型信息系统，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、办公信息系统、地理信息系统等，越来越多的应用领域使用数据库存储、处理和检索数据。正是因为有了数据库技术，显著地提高了各行业的工作效率和质量，并产生出极大的社会与经济效益。

1.1.1　数据库技术的基本概念

与数据库技术密切相关的五个基本概念是信息、数据、数据库、数据库管理系统和数据库系统。

1．信息

在当今社会中，人类的所有活动都离不开信息。信息已经成为一种重要的资源，它与能源、材料共同组成了现代社会的三大资源。不同的领域对信息有着不同的定义。通常认为，信息是人对现实世界事务存在方式或运动状态的某种认识，是客观事物属性和相互联系特性的表现，反映了客观事物的存在形式和运动状态。例如，图书的名称、出版社、价格及作者都是关于图书的信息，它们组合在一起从不同的角度共同反映出图书的存在形式。

2．数据

数据是把事件的某些属性规范化后的表现形式，可以被识别，也可以被描述。“数据”的概念包括两个方面。一方面，数据内容是事物特性的反映或描述；另一方面，数据是符号的集合。信息与数据既有区别，又有联系。两者的关系是：数据是信息的载体，信息则是数据内在含义，两者可以相互转换。两者也是抽象与客观现实的关系，即信息是抽象的，不随数据设备所决定的数据表示方式而改变，而数据的表示方式及存在方式却是客观现实的。

3．数据库

数据库（Database，DB）是长期存储在计算机中的、有组织的、统一管理的相关数据的集合。数据库能被各种用户共享，可以为多种应用提供服务，数据库不存在有害的或是不必

要的冗余，数据间联系紧密但数据的存储独立于使用它的程序。

4．数据库管理系统

数据库管理系统（Database Management Systems, DBMS）是专门用于管理数据库的计算机系统软件，位于用户与操作系统之间，为用户或应用程序提供访问数据库的方法。数据库管理系统能够为数据库提供数据的定义、建立、维护、查询和统计等操纵功能，并完成数据完整性检查、安全性检查、并发控制及数据故障恢复等控制功能。

在数据库管理系统的操纵功能中，数据的定义功能是指建立数据库结构的操作，可以通过此项功能实现数据库框架的建立。建立功能是指可以将大批数据录入到数据库的操作，它使得数据库含有需要保存的数据记录。维护功能是指对数据的插入、删除和修改操作，可以满足数据库中信息变化或更新需求。数据查询和统计功能是指通过对数据库的访问，为实际应用提供需要的数据。数据库管理系统提供数据操纵语言（Data Manipulation Language, DML）来实现对数据库的操纵功能。有两类 DML：一类是嵌入在宿主语言中的，如嵌入在 VC, Java, Power Builder, Delphi 等高级语言中，称为宿主型 DML；另一类是可以独立地交互使用的 DML，它可以不依托高级语言直接对数据库进行操作，称为自主型或自含型 DML，常用的有 SQL Plus 等。大多数的，如 Oracle 和 SQL Server 等都包含有这两种 DML 供用户使用。

DBMS 提供了数据控制功能，以适应共享数据的环境。数据控制功能主要包括对数据库中数据的安全性、完整性、并发和故障恢复的控制。安全性控制是指保护数据库以防止不合法的使用造成数据泄漏、更改或破坏，用户只能按照规定对数据进行处理，按照相应的权限操纵数据库。数据的完整性是指数据库的正确性和相容性。数据的完整性控制是指控制合法用户使用数据库时向数据库加入不符合语义的数据，保证数据库中数据是正确的，避免非法的更新。数据的并发控制是指在多个用户共享的系统中，许多用户可能同时对同一数据进行操作，从而导致数据的不一致性，主要包括丢失更新、不可重复读和读脏数据三种类型。故障恢复是指在发生事务内部故障、系统故障、介质故障及计算机病毒时，可以恢复数据库本身。恢复机制是建立在冗余备份的基础之上。

目前国内外比较流行的数据库管理系统有 Oracle, DB2, Sybase 和 SQL Server 等。SQL Server 是由 Sybase, Microsoft 和 Ashton-Tate 联合开发的 OS/2 系统上的数据库系统。1988 年正式投入使用，1990 年 Ashton-Tate 公司退出了 SQL Server 的开发，1994 年 Sybase 公司也将重点投入到 UNIX 版本的 SQL Server 开发上，而 Microsoft 公司则致力于将 SQL Server 移植到 NT 平台上。1996 年 Microsoft 公司推出 SQL Server 6.5；1998 年升级到 7.0 版本；2000 年 SQL Server 2000 投入市场；2005 年 11 月 Microsoft 公司推出最新一代的数据库产品 SQL Server 2005。

从 Microsoft SQL Server 2000 到 Microsoft SQL Server 2005，数据库系统不仅仅具有更高的性能、处理能力，而且具备了一些新的特性，这些特性也是与当前数据库的发展方向一致的，例如在企业环境中，更注意安全性、高可用性、集成的管理工具等。

5．数据库系统

数据库系统（Database Systems, DBS）是采用了数据库技术的计算机系统，它能够按照数据库的方式存储和维护数据，并且能够向应用程序提供数据。狭义地讲，数据库和数据库管理系统加在一起就构成了数据库系统；广义地讲，数据库系统通常由数据库、硬件、软件和人员 4 部分组成。

1）数据库

数据库的体系结构可分为两个部分：一部分是存储应用所需的数据，称为物理数据库部分；另一部分是描述部分，描述数据库的各级结构，这部分由数据字典管理。

2）硬件

数据库系统建立在计算机系统之上，涉及的硬件资源主要包括 CPU、内存和外部存储器等。与其他的软件系统相比，数据库系统对计算机的硬件有三个方面的特殊要求。首先，要求计算机系统要有足够大的内存存放操作系统、数据库管理系统的核心模块、数据库数据缓冲区及用户的工作区。通常内存越大，数据库的性能越好，运行越稳定。其次，要求计算机系统应具有足够大容量的磁盘空间。数据库文件就存放在磁盘中，它需要尽可能高的磁盘存取速度满足数据读取的需要，另外还需要充足的磁盘空间或存储设备实现数据库的备份。最后，随着网络数据库的不断应用，目前大多数的数据库系统对网络服务能力要求较高，要求能够满足网络数据库的数据传输需要。当然，不同的数据库产品对硬件的要求也是不尽相同的，其他的一些因素也是需要考虑的，如可支持的终端数、安全性和设备可靠性等。

3）软件

软件主要包括操作系统、数据库管理系统和一些开发工具，例如带有数据库接口的各种高级语言和应用开发软件等。数据库管理系统是数据库系统软件的核心部分，但它需要将操作系统作为运行的基础平台。操作系统则提供对数据库管理系统的支持。当前比较流行的应用开发语言主要有 Visual Basic、C++、Java、PowerBuilder 和 Delphi 等，这些高级语言统称为第 4 代程序设计语言。它们的一个共同特点是：都具有与数据库交互的接口，可以为数据库应用系统的开发和应用提供良好的开发和应用环境，都是以数据库管理系统为核心。

4）人员

人员是指管理、开发、使用数据库系统的所有人员，通常包括数据库管理员（Database Administrator, DBA）、系统分析员、应用程序员和终端用户。

数据库管理员负责全面地管理和控制数据库系统，其职责包括以下几个方面。

① 参与数据库和应用系统的设计。数据库中究竟要存放哪些信息由数据库管理员决定。因此，他们必须参与数据库设计的全过程，与用户、应用程序员、系统分析员一起设计概念模式，决定与应用有关的实体、实体间的联系和实体的属性。只有参与数据库和应用系统的设计，数据库管理员才可能对数据库结构及程序设计方法了解清楚，并为今后数据库管理工作打下基础，才能保证数据库设计的合理性。

② 决定数据库的存储结构和存取策略。数据库管理员要综合各用户的应用需求，与数据库设计人员共同决定数据的物理组织、存放方式及数据存取方法等，使数据的存储空间利用率和存取效率两个方面都较优。

③ 负责定义数据的安全性要求和完整性约束条件。保证数据库的安全性和完整性是数据库管理员的重要职责，数据库管理员要负责定义各用户的数据使用权限、数据保密级别和数据完整性的约束条件。

④ 负责监督和控制数据库的使用和运行，负责系统的维护和数据恢复工作。数据库管理员负责监视数据库系统的运行情况，如果出现问题，数据库管理员要及时处理。在日常工作中，数据库管理员要负责记录数据库使用的日志文件，通过日志文件了解数据库的使用和更改的情况。数据库管理员还要定期对数据进行备份，为以后的数据使用和数据恢复做准备。

当数据库出现故障时，数据库管理员需要使用日志文件对数据库进行恢复工作，使数据库能在最短时间恢复到正确状态。

⑤ 数据库系统的改进和重组。数据库管理员负责通过各种日志和统计数字监视、分析数据库系统的性能。数据库管理员要负责对系统运行时的系统空间利用率和处理效率进行记录、统计、分析，并根据实际应用和一定的策略对数据库进行重组并不断改进数据库的设计。当用户的需求增加或改变时，数据库管理员还要对数据库进行重新构造。

1.1.2 数据库技术的应用及发展

随着社会生产技术的不断发展，人类所面对的信息量越来越大，如何快速、高效地处理数据信息是摆在人们面前的一项重要课题。伴随计算机技术快速发展起来的数据库技术已经成为数据处理的有力工具，成为信息技术中的一个重要支撑。可以这样认为：没有数据库技术，人们在浩瀚的信息世界中将显得手足无措。

数据库技术是计算机科学技术的一个重要分支。从 20 世纪 50 年代中期开始，计算机应用从科学研究部门扩展到企业管理及政府行政部门，人们对数据处理的要求也越来越高。从 1968 年世界上第一个商品化的数据管理系统 IMS 开始，数据库技术得到了迅猛发展。今天，随着因特网普及，数据库技术的重要性更为显著。现在，数据库已经成为信息管理、办公自动化、计算机辅助设计等应用的主要软件工具之一，帮助人们处理各种各样的信息数据。

数据库技术的优势体现在众多方面，例如数据冗余度的减少，降低了数据存储成本；数据资源可以实现充分的共享，让越来越多的用户在网络上分享数据；以及为用户提供方便简洁的数据交互方式，提供数据库编程接口，使用户易于编写有关数据库应用程序等。数据库技术的最初应用领域主要是信息管理，例如教育卫生、政府机关、企业管理、经济金融、图书检索等各种行业的信息处理。在当今社会，哪里有大量的数据信息需要处理，哪里就会出现数据库技术的身影，它已经广泛地渗透到社会方方面面的应用领域当中。目前数据库技术的主要应用领域包括 Web 数据库技术、多媒体数据库技术和数据情报检索服务技术等。

进入 20 世纪 80 年代后，数据库技术以前所未有的速度向前发展着，并且与各种新兴学科技术有机结合，衍生出新一代数据库技术，例如分布式数据库技术、工程数据库技术、面向对象数据库技术、人工智能数据库技术、并行数据库技术、模糊数据库技术等都是这方面的实例，广泛应用于多个领域，如计划统计、数据挖掘、信息决策等。新一代的数据库技术都具备以下几个主要特征。

① 支持数据管理、对象管理和知识管理。

② 数据库系统必须保持或者继承原来数据库系统的技术。

③ 数据库系统必须对其他系统开放，支持数据库语言标准，支持标准网络协议，有良好的可移植性、可连接性、可扩展性和互操作性。

新一代的数据库技术中较为典型的代表是分布式数据库技术与面向对象数据库技术，下面分别对它们进行简单介绍。

1．分布式数据库技术

分布式数据库是数据库系统和计算机网络相结合的产物，它是针对地理上分散，而管理上又需要不同程度集中管理的需求而提出的一种数据库管理技术。分布式数据库系统由一组

数据组成，这些数据物理上分布在计算机网络的不同节点（也称场地）上，逻辑上则属于同一个系统。分布式数据库系统在物理上是分布的，在逻辑上则统一的。如果数据库能够同时满足下列几个特性，则可以称其为分布式数据库。

① 分布性：即数据存储在多个不同的节点上。

② 逻辑相关性：数据库系统内的数据在逻辑上具有相互关联的特性。

③ 场地透明性：使用分布式数据库中的数据时不需要指明数据所在的位置。

④ 场地自治性：每一个单独的节点能够执行局部的应用请求。

2．面向对象数据库技术

数据库技术与面向对象程序设计方法相结合形成了面向对象数据库技术，因此一个面向对象的数据库系统首先是一个数据库系统，同时又具有面向对象的特征。其主要特征体现在以下三个方面。

① 具有表达和管理对象的能力。面向对象数据库系统通过对象及它们之间的相互联系来描述现实世界。

② 面向对象数据库系统中的对象可以具有任意复杂度的对象结构。只有这样，对象才能具备所有描述该对象的必要信息。

③ 必须具有与面向对象编程语言交互的接口。面向对象数据库系统可以通过与面向对象编程语言交互的接口，提供持久化对象和共享对象的能力，从而允许多个程序和应用共享这些对象。

④ 应具有表达和管理数据库变化的能力。

伴随着数据库新技术的不断涌现，数据库系统结构正在由传统的终端/主机的集中式结构发展成为网络环境的分布式结构，并逐步产生了两层、三层或多层客户/服务器结构及因特网环境下的浏览器/服务器结构和移动环境下的动态结构，以适应不同的应用需求环境。

1.1.3　Web 网络数据库系统编程简介

随着 Web 网络技术在当今社会中的广泛应用，以及众多企业由传统的客户机/服务器结构向浏览器/服务器结构的大规模转型，网络应用正随着数据库技术与 Web 技术的结合进行着深刻变革。正是在数据库的支持下，扩展网页功能、设计交互式页面、构造功能强大的后台管理系统，以及更新或维护网站都将变得轻而易举。随着网络应用的深入，Web 数据库技术日益显示出其重要地位。这里仅简单介绍 Web 数据库开发的两种主要技术：动态服务器页面技术与 Java 服务器页面技术。

1．动态服务器页面技术

动态服务器页面（Active Server Pages, ASP）是微软公司推出的取代通用网关接口（Common Gateway Interface, CGI）编程的一种技术，是一种简便易学、功能强大的服务器编程技术。ASP 实际上是微软公司开发的一套服务器端脚本运行环境。利用 ASP，设计人员可以建立动态的、交互的、高效的 Web 服务器应用程序。用 ASP 编写的程序在服务器端执行，程序执行完毕后，再将执行的结果返回给客户端浏览器，这样不仅减轻了客户端浏览器的负担，大大提高了交互速度，而且避免了 ASP 程序源代码的外泄，提高了程序的安全性。

需要全本请访问 www.ertongbook.com

2．Java 服务器页面技术

Java 服务器页面（Java Server Pages, JSP）是 Sun 公司发布的 Web 应用开发技术，一经推出，就受到了人们的广泛关注。JSP 技术为创建高度动态的 Web 应用提供了一个独特的开发环境，它能够适应市场上绝大多数的服务器产品。

JSP 使用 Java 语言编写服务器端程序，当客户端向服务器发出请求时，JSP 源程序被编译成 servlet 并由 Java 虚拟机执行。这种编译操作仅在对 JSP 页面的第一次请求时发生。因此，JSP 程序能够提供更快的交互速度，安全性和跨平台性也很优秀。

目前，ASP 技术与 JSP 技术是市场上并驾齐驱的两种 Web 应用开发技术，各自都占有一定的市场份额。两种技术各有所长，各有所短，在进行选择时用户可以从工作平台的搭建、开发技术的难易程度、后台数据库的连接方式及工作环境的具体要求等方面进行考虑。

1.2 数据库技术的发展阶段

数据库技术研究的问题是如何科学地组织和存储数据，如何高效地获取和处理数据，而数据处理的中心问题是数据管理。因此，数据管理的发展是数据库技术发展过程中的重要内容。

数据管理指的是数据的分类、组织、编码、存储、检索和维护。从数据管理的角度来看，自 20 世纪 60 年代末 70 年代初以来，数据库技术在 40 多年中主要经历了三个阶段：人工管理阶段、文件系统管理阶段和数据库系统管理阶段。目前，数据库技术仍在日新月异地发展，而数据库技术的应用也将进一步深入到人类社会的各个层面。

1.2.1 人工管理阶段

人工管理阶段是指计算机诞生的初期（20 世纪 50 年代中期以前），这时的计算机主要用于科学计算。硬件状况是外存只有纸带、卡片、磁带，没有磁盘等直接存取的存储设备。软件状况是没有操作系统，没有管理数据的软件。数据处理方式采用的是批处理。数据管理较为简单，数据管理的主要任务就是对使用的穿孔卡片进行物理的存储和处理。

数据管理在人工管理阶段具有以下几个特点。

1．数据不保存

由于当时计算机主要用于科学计算，加上当时的计算机软硬件条件比较差，一般不需要也不允许将数据长期保存，只是在完成某一计算任务时将应用程序与数据一起装入，两者结合在一个整体，用完后就将它们撤销，不保存原始数据，也不保存计算结果。

2．使用应用程序管理数据

数据需要由应用程序自己管理，没有相应的软件系统负责数据的管理工作。应用程序不仅要规定数据的逻辑结构，还要设计物理结构，包括存储结构、存取方法、输入方式等。应用程序与数据之间的依赖性太强，一旦数据在存储器上改变物理地址，就需要改变相应的用户程序。这给程序员的工作增加了负担，为程序的设计与维护都带来一定的困难。

3．数据不共享

数据是面向应用的，一组数据只能对应于一个程序。即使两个应用程序使用相同的数据，也必须各自定义自己的数据存储和存取方式，不能共享相同的数据定义，因此造成程序和程

序之间可能有大量的重复数据，出现数据冗余。

4．数据不具有独立性

当数据的逻辑结构或物理结构发生变化时，必须对应用程序做出相应的修改，因此程序员的负担较重。

人工管理阶段的数据管理模型如图 1.1 所示。

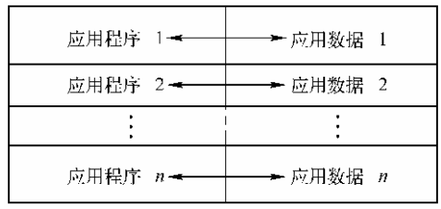


图 1.1　人工管理阶段数据管理模型

1.2.2　文件系统管理阶段

20 世纪 50 年代后期到 60 年代中期，在硬件方面，已经出现了磁盘、磁鼓等可以进行直接存储的外部存储设备；在软件方面，出现了高级语言和操作系统，操作系统中已经有了专门的数据管理软件，一般称为文件系统。此阶段的数据处理方式不再是单一的批处理，而是产生了联机实时处理的方式。

数据管理在文件系统管理阶段中具有以下几个特点。

1．数据可以长期保存

由于计算机大量用于数据处理，数据需要长期保留在外存上，反复进行查询、修改、插入和删除等操作。

2．由专门的软件即文件系统管理数据

文件系统是操作系统的一个子系统。在文件系统中，按一定的规则将数据组织成一个文件，利用“按文件名访问，按记录进行存取”的管理技术，可以对文件进行修改、插入和删除等操作。文件系统实现了记录内的结构性，但整体无结构。程序和数据之间由文件系统提供存取方法进行转换，使应用程序与数据之间有了一定的独立性，程序员不必关心数据的物理位置细节，可以集中精力考虑算法，并且降低了维护程序的工作量。

3．数据共享性差，冗余度大

文件系统中的数据还是面向应用的，一个文件通常对应于一个应用程序，文件与应用程序之间仍旧保持着密切的联系。当不同的应用程序使用相同的数据时，也必须建立各自的数据文件，而不能共享相同的数据，因此数据的冗余度大，浪费存储空间。例如，学校中人事处管理的职工人事档案，教务处理管理的教师教学情况，所用到的数据很多都是重复的，但也要为每个应用建立单独的数据文件。

由于相同数据的重复存储、各自管理，容易造成系统数据的不一致性，加大了数据修改和维护的困难程度。例如，教务处更改了某个教师的相关信息，但人事处有关此职工的信息却没有修改，这时就出现了同一个职工在不同的部门结果不一样的情况。

4．数据独立性差

文件系统中的文件是为某一特定应用服务的，文件的逻辑结构对该应用程序来说是优化的，因此很难对现有的数据再增加一些新的应用，系统不易扩充。同时，一旦数据的逻辑结构发生改变，则必须修改应用程序，而应用程序的改变又会影响到数据结构的改变。因此文件系统仍然是一个不具有弹性的、无结构的数据集合，即文件之间还是保持着一种彼此孤立的关系，尽管可以反映现实世界中客观存在的事物，但是不能反映这些事物之间的联系。

文件系统阶段的数据管理的模型如图 1.2 所示。

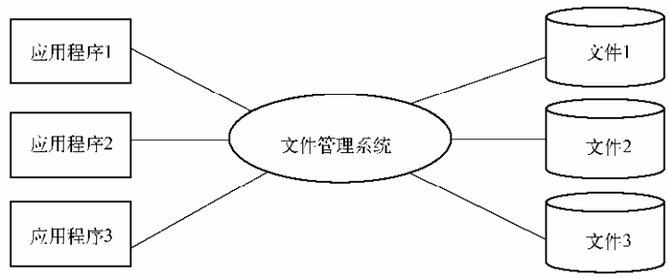


图 1.2　文件系统管理阶段数据管理模型

1.2.3　数据库系统管理阶段

20 世纪 60 年代后期以来，计算机用于管理的规模越来越大，应用越来越广泛，处理的数据量也急剧增长。同时，多种应用、多种语言互相覆盖地共享数据集合的要求也越来越强烈。传统的文件系统已经不能满足人们的需要，能够统一管理和共享数据的专门软件系统——数据库管理系统便应运而生。因为数据模型是数据系统的核心和基础，各种数据库系统都是基于某种数据模型的。

1968 年，IBM 公司推出了商品化软件数据库信息管理系统 IMS，这是数据库技术发展中的里程碑，它使用了一种树状结构的层次模型，并广泛应用于商业领域。一年后，1969 年美国数据系统语言协会 CODASYL 下属的数据库任务组 DBTG 发表了网状数据库系统的标准文本，为网状数据模型的发展奠定了基础。但是以层次模型与网状模型为基础的数据库系统并没有一个完整和坚固的数学理论基础，相应的数据库产品直接按实际应用的要求来设计。直到 1970 年，IBM 公司的 SAN Jose 研究室的高级研究员 E. F. Code 发表了一篇题为“大型数据库的数据关系模型”的著名文章。他在文章中引进关系代数，首次提出数据库的关系模型的概念，成为数据库技术发展的一个重大飞跃。从此，数据库系统的发展进入了“关系型数据库系统”时期。几年后 IBM 公司研制成功第一个关系数据库系统 System R。20 世纪 80 年代后，关系型数据库系统逐步取代网状和层状模型，占领了市场。

与人工管理和人工系统相比，数据库系统的特点主要体现在以下几个方面。

1．数据库中的数据是结构化的

数据结构化是数据库与文件系统的根本区别，文件系统中的数据是无结构的，数据库系统不仅要描述数据本身，还要描述数据之间的联系。数据不再面向某个应用，而是面向整个应用系统，数据冗余明显减少并能够较好地实现数据共享。

2．具有较高的数据独立性

数据独立性包括两个方面，即数据的物理独立性和数据的逻辑独立性。数据的物理独立性指存储在磁盘上的数据库数据与使用数据的用户应用程序之间是相互独立的。用户应用程序只需要关心数据的逻辑结构，而数据的存储由数据库管理系统去实现，而不需要用户应用程序去设计。当数据物理的存储结构发生改变时，应用程序不需要进行改变。数据的逻辑独立性指用户的应用程序与数据库的逻辑结构是相互独立的。当数据的逻辑结构发生改变时，用户的应用程序可以不变。数据独立性是由数据库管理系统的二级映像功能来保证的。

数据与程序的独立，使得数据的定义可以从程序中分离出去，加上数据由数据库管理系统管理，从而简化了应用程序的编制和程序员的负担。

3．具有较低的数据冗余

数据不再面向某个应用而是面向整个系统，数据库系统中的数据可以被多个应用、多个用户共享使用，这样就可以使得数据库系统中的重复数据减少到最低程度，可以较好地减少数据冗余，节约存储空间以便存放更多的数据。另外由于是面向多用户、多应用的，系统可以很容易地增加新的应用，使得数据库系统弹性变大，易于扩充。

4．数据控制功能

数据库系统具有安全性控制功能，能够防止数据的丢失和被非法使用。具有数据完整性控制功能，保护数据的正确、有效和相容。具有数据的并发控制功能，能够避免并发程序之间引起的相互干扰而得到错误的数据。具有数据的恢复控制功能，保障在数据库被破坏或数据出现不可靠的情况时，系统有能力把数据库恢复到最近某个时刻的正确状态。

数据库系统阶段的数据管理模型如图 1.3 所示。

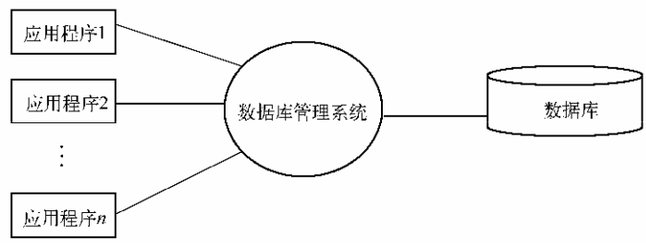


图 1.3　数据库阶段数据管理模型

1.3　数据库系统的体系结构

所有的数据库都需要有一个总体框架，即数据库系统的体系结构。数据库系统的体系结构可以从不同的层次或不同的角度来进行考察。从数据库用户来看，数据库系统的结构分为集中式结构（单用户或主从结构）、分布式结构、客户/服务器结构和并行结构。从数据库管理系统的角度看，数据库系统通常采用三级模式结构：外模式、模式与内模式。本小节主要介绍后者，它是数据库系统的一般概念，是学习数据库的基础和纲要。

1.3.1　模式的概念

在数据模型中有“型”与“值”的概念。“型”指对某一类数据的结构和属性的说明，

是一个总框。而“值”则是对某个型的具体赋值。例如：（张明，男，22，山东，173）就是一个值，它对应的“型”就是（姓名，性别，年龄，籍贯，身高）。

“模式”是数据库中全体数据的逻辑结构和特征的描述，它是对数据模型中型的描述，不涉及具体的值。模式的一个具体的值即是该模式的一个实例。同一个模式可以有很多实例。与实例相比，模式是稳定的，不会经常变动；而由于数据库中的数据处于不断更新的状态，实例是处在相对变动的情况。模式反映的是数据的结构和数据之间的联系，而实例反映的是数据库某一时刻的状态。

目前，多数的数据库管理系统产品在体系结构上通常都具有相同的特征，即采用三级模式结构，并提供二级映像功能。

1.3.2 三级模式结构

数据库的数据模式是由外模式、模式和内模式三级模式构成，其结构如图 1.4 所示。

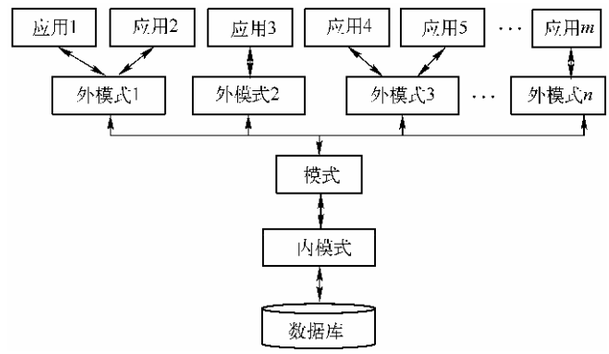


图 1.4 数据库系统的三级模式结构

1．模式

“模式”也称为逻辑模式或概念模式，它是数据库的总框，是对数据库中全体数据的逻辑结构和特性的描述，是所有用户的公共数据视图。模式是数据库系统模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序、高级语言和物理结构无关。模式实际上是数据库数据在逻辑级上的视图。一个数据库只有一个模式，它通常是以某一种数据模型为基础的，综合统一地考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体。数据库管理系统提供模式描述语言进行模式的定义。其定义的内容不仅包括对数据库的记录型、数据项的型、记录间的联系等的描述，同时也包括对数据的安全性定义（如数据使用权限、安全保护方式等）、数据应满足的完整性条件等内容。

2．外模式

“外模式”也称为用户模式或子模式，它是对各个用户或程序所涉及的数据的逻辑结构和数据特征的描述，即数据库用户所看到的和使用的数据库的描述。外模式是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示，是模式的子集。由于用户的应用需求、数据的安全需求、数据的使用方式等方面的不同，可以从一个数据库模式推导出多个不同的用户外模式，有利于数据的独立性和共享性。即使对模式中的同一数据，在外模式中的结构、

类型、长度、安全级别等方面都可以不同；同一外模式也可以为某一用户的多个应用系统所使用，但一个应用程序只能使用一个外模式。

外模式是数据库系统保证数据库安全性的一个重要手段。每个用户只能看见和访问自身所对应的外模式中的数据，无法看到数据库中其他用户模式所涉及的数据。同时，由于用户使用外模式，用户不必考虑那些与自己无关的数据，也无须了解数据的存储结构，使得用户使用数据的工作和程序设计的工作都得到了简化。

数据库管理系统提供子模式描述语言进行外模式的定义。

3．内模式

“内模式”也称为存储模式或物理模式，它是对内层数据的物理结构和存储方式的描述，是数据在数据库文件内部的表示方式，它是用设备介质语言来定义的。例如，数据库记录的存储方式是顺序存储、链式存储、B 树方式存储还是按照 HASH 方式进行存储，是否使用索引，数据是否进行压缩存储，是否对数据进行加密等。内模式通常对一般用户来说是透明的。

数据库管理系统提供内模式描述语言进行内模式的定义。

综上所述，外模式、模式与内模式之间的关系是：模式是内模式的逻辑抽象形式；内模式是模式的具体实现；外模式是概念数据库的子集，也是内模式子集的描述。

1.3.3　数据库模式的映像

数据库系统提供的三级模式对应着数据的三个抽象级别，它们把数据的具体组织留给数据库管理系统管理，使用户能逻辑地、抽象地处理数据，而不需要关心数据在计算机中的具体表示方法和存储方式。为了实现这三个层次上的联系和转换，数据库系统在这三级模式中提供了两层映像：外模式/模式映像、模式/内模式映像。正是在这两层映像机制的帮助下，数据库系统中的数据才具有了较高的逻辑独立性和物理独立性。

1．外模式/模式映像

外模式描述的是数据的局部逻辑结构，模式描述的是数据的全局逻辑结构。对应于同一个模式可以有多个外模式，而对于每一个外模式，数据库系统都对应一个外模式/模式映像，它定义并保证了外模式与模式之间的对应关系。这些映像定义通常包含在各自的外模式中。

当模式改变时（例如，增加新的关系、新的属性、改变属性数据类型等），由数据库管理员对各个外模式/模式映像做相应的改变，可以使外模式保持不变。应用程序是依据数据的外模式进行编写的，外模式/模式映像可以保证模式的变化不会影响到应用程序的执行，实现了数据与程序的逻辑独立，即数据的逻辑独立性。

2．模式/内模式映像

数据库中只有一个模式，也只有一个内模式，所以模式/内模式映像是唯一的，它定义了数据库全局逻辑结构与存储结构之间的对应关系。映像定义通常包含在模式描述中。当数据库的存储结构改变时（如由原来的顺序存储改为链式存储），模式/内模式映像也必须做出相应的改变，使得模式保持不变，保证数据与程序的物理独立，即数据的物理独立性。同外模式/模式映像一样，模式/内模式映像的调整也是由数据库管理员来实现的。

数据库的二级映像保证了数据库外模式的稳定性，从而从底层保证了应用程序的稳定

需要全本请访问 www.ertongbook.com