

中文 Access 2002 应用基础教程

毛文毅 主编

刘冬宁 刘浩钊 编著

北 京

冶金工业出版社

2002

内容简介

Access 2002 是微软公司最新推出的关系型数据库管理系统。本书由浅入深、循序渐进地对 Access 2002 进行了介绍, 主要内容有: 数据库基础、Access 2002 应用基础、表的设计与应用、使用查询、窗体、报表、宏、Visual Basic for Applications、Access 与 Web、SQL Server 2000 Desktop Engine、数据库的安全与维护以及 Access 开发实例等。

本书编写本着可读性、可操作性和实用性的原则, 力求做到语言描述形象化, 理论、概念简单化, 尽量把 Access 2002 管理数据库的各方面知识和方法介绍给读者, 以便读者在短时间内能真正掌握这个功能强大的数据库开发工具。本书不仅适合于初中级水平的读者、专业技术人员, 还可以作为大、中专院校的专业技能培训教材。

图书在版编目 (CIP) 数据

中文 Access 2002 应用基础教程 / 毛文毅主编. —北京: 冶金工业出版社, 2002.6

ISBN 7-5024-3033-4

I. 中... II. 毛... III. 关系数据库—数据库管理系统, Access 2002 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2002) 第 031635 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 戈兰

广东出版技校彩印厂印刷; 冶金工业出版社发行; 各地新华书店经销

2002 年 6 月第 1 版, 2002 年 6 月第 1 次印刷

787mm×1092mm 1/16; 12.25 印张; 277 千字; 187 页; 1-2500 册

20.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

1. 关于 Access 2002

Access 2002 是微软公司最新推出的关系型数据库管理系统。自 2001 年问世以来, Access 2002 凭借着易学、易用、功能强大的特点, 已逐步成为桌面数据库领域的领导者, 并拥有广泛的用户。它与其他应用程序高度集成, 具有强大的网络功能; 它提供了大量的管理工具和许多简便有效的向导, 使用户无需花大量时间和精力去学习即可开发出高效实用的数据库应用系统。

Access 2002 在功能方面较以往版本有了较大改进, 它提供了全新的“数据透视图”视图、“数据透视表”视图、完全的 XML 支持和 SQL Server 2000 的扩展属性支持等, 并且其内部控件还都做到了“Web 化”, 适用于网络化、信息化的需要。

2. 本书的结构

本书共分 12 章, 由简单到复杂、从理论到实践, 循序渐进地讲述数据库软件 Access 2002 的各项功能与应用。各章内容安排如下:

第 1 章数据库基础。从数据库的基本理论入手, 讲述了数据库的概念、数据库技术的发展、数据库模型、数据库设计以及 Access 2002 和数据库的关系等内容。

第 2 章 Access 2002 应用基础。从这一章开始接触 Access 2002 方面的知识, 该章首先对 Access 2002 作了简单介绍, 接着介绍了 Access 2002 的新特性, 重点讲述了数据库中的几个对象。

第 3 章表的设计与应用。该章重点讲述 Access 2002 基本对象——表的创建、设计、管理, 定制数据表视图, 数据表之间的关系, 获取外部数据及子表的使用等内容。

第 4 章使用查询。作为数据库的一个基本特征, 查询是最常见的操作之一, 该章介绍了查询的概念, 分析了查询的各个类型及其实现的步骤。

第 5 章窗体的设计。窗体是数据库系统与用户之间联系的窗口, 怎样创建和设置好一个窗体是该章讲述的重点内容。

第 6 章创建报表。创建报表是导出数据库数据的一个前奏, 该章重点讲述了报表的创建与设计。

第 7 章宏的应用。宏即可使用户简单快捷地操纵数据库系统, 也可以加强用户对数据库系统的管理应用, 该章就宏的创建、运行、调试等操作做了重点讲解, 并列出一一些常见的宏操作。

第 8 章 VBA 的应用。VBA 具有高级程序设计语言的语句结构, 简单易用。该章重点讲述了 VBA 中的过程、VBA 中的数据、VBA 中的表达式、VBA 程序的调试以及运行 VBA 代码。

第 9 章 Access 与 Web。Access 2002 在网络上的应用越来越显得重要, 而该章讲述的数据访问页的创建和发布也是 Access 数据库在 Web 页上应用的基本内容。

第 10 章 SQL Server 2000 Desktop Engine。SQL Server 2000 是一个功能强大的数据库系统, 怎样把 Access 2002 与它协同使用是该章讲述的内容。

第 11 章数据库的安全与维护。庞大的数据要想不出任何问题是很难的，所以该章讲述了 Access 数据库的安全维护知识以及优化其性能的方法。

第 12 章 Access 开发实例。讲述一个 Access 数据库组建的全过程。

3. 本书的特点

本书的内容全面，结构合理严谨，语言通俗易懂，图片、示例清晰明了，并且各章均配有习题，读者可以一边学习一边实践。

4. 本书的适用对象

本书的主要对象是：初、中级水平的软件使用人员、计算机专业技术人员，大、中专院校师生以及各类专业技能培训班、计算机爱好者等。

本书由毛文毅主编，其中第 1 章、第 8~11 章由刘冬宁编写，第 2~7 章、第 12 章由刘浩钊编写。由于编者水平有限，时间仓促，缺点错误之处在所难免，恳请广大读者批评指正。

编 者

2002 年 5 月

目 录

第 1 章 数据库基础..... 1

1.1 什么是数据库..... 1

1.2 数据库技术的发展..... 1

1.2.1 人工管理阶段..... 1

1.2.2 文件系统阶段..... 2

1.2.3 数据库系统阶段..... 3

1.3 数据库管理系统 (DBMS) 4

1.4 数据库模型..... 5

1.4.1 层次模型..... 6

1.4.2 网状模型..... 7

1.4.3 关系模型..... 8

1.5 关系数据库标准语言——SQL..... 9

1.6 数据库设计..... 10

1.6.1 数据库设计和信息系统..... 11

1.6.2 数据库设计的特点..... 11

1.6.3 数据库设计方法..... 12

1.6.4 数据库设计的基本步骤..... 12

1.7 Access 2002 和数据库的关系..... 13

1.8 小结..... 14

综合练习一..... 14

一、基础题..... 14

二、思考题..... 15

三、上机操作..... 15

第 2 章 Access 2002 应用基础.....16

2.1 Access 简介..... 16

2.1.1 使用方便的 Access 2002 向导..... 16

2.1.2 动态数据交换 (DDE) 及对象的链接和嵌入 (OLE) 16

2.1.3 在 Internet 上的应用和客户/服务器..... 16

2.1.4 语言的标准化..... 17

2.2 Access 2002 的新特性..... 17

2.2.1 数据透视图和数据透视表..... 17

2.2.2 XML 支持..... 17

2.2.3 Microsoft SQL Server 2000 扩展属性的支持..... 18

2.2.4 Access 2002 的其他新功能..... 18

2.3 数据库对象..... 19

2.3.1 表..... 19

2.3.2 查询..... 19

2.3.3 窗体..... 19

2.3.4 报表..... 20

2.3.5 Web..... 20

2.3.6 宏..... 20

2.3.7 模块..... 20

2.4 创建数据库..... 21

2.4.1 使用模板创建数据库..... 21

2.4.2 创建空数据库..... 22

2.5 打开数据库..... 22

2.6 小结..... 23

综合练习二..... 24

一、基础题..... 24

二、思考题..... 24

三、上机操作..... 24

第 3 章 表的应用..... 25

3.1 使用设计器创建一个新表..... 25

3.2 使用表向导创建表..... 28

3.3 输入数据创建表..... 31

3.4 表字段的设计..... 32

3.4.1 数据类型..... 32

3.4.2 定义字段的输入掩码..... 36

3.5 管理数据表中的数据..... 38

3.5.1 增加记录..... 38

3.5.2 选定并修改数据..... 39

3.5.3 删除记录..... 39

3.5.4 复制和粘贴数据..... 39

3.5.5 查找和替换数据..... 40

3.5.6 数据的排序..... 41

3.5.7 数据筛选..... 42

3.6 定制数据表视图..... 45

3.6.1 改变行高和列宽..... 45

3.6.2 移动列..... 46

3.6.3 冻结和解冻列..... 46

3.6.4 隐藏/显示列..... 46

3.6.5 改变字体的样式、大小..... 46

目 录

和颜色	47	三、上机操作	73
3.6.6 设置数据表的格式	47	第5章 窗体的设计	74
3.7 数据表之间的关系	48	5.1 窗体的功能和特点	74
3.7.1 查看已有的关系	48	5.2 使用向导创建窗体	75
3.7.2 建立表间关系	49	5.3 快速创建窗体	76
3.8 获取外部数据	50	5.3.1 纵栏式窗体	76
3.8.1 导入表	50	5.3.2 表格式窗体	77
3.8.2 链接表	51	5.3.3 数据表窗体	77
3.9 子表的使用	51	5.3.4 图表窗体	77
3.10 小结	52	5.3.5 数据透视表窗体	77
综合练习三	52	5.4 使用窗体设计器创建窗体	78
一、基础题	52	5.4.1 打开窗体的设计视图	78
二、思考题	53	5.4.2 页眉、页脚	82
三、上机操作	53	5.4.3 标签	83
第4章 使用查询	54	5.4.4 文本框	84
4.1 什么是查询	54	5.4.5 列表框、组合框	86
4.2 查询的类型	54	5.4.6 选项组	88
4.3 使用向导创建查询	55	5.5 窗体控件的格式设置	89
4.3.1 简单查询向导	55	5.5.1 更改控件标题和间距	89
4.3.2 交叉表查询向导	57	5.5.2 改变控件的大小和位置	89
4.3.3 查找重复项查询向导	59	5.5.3 设置控件的颜色	90
4.3.4 查找不匹配项查询向导	60	5.5.4 改变控件的外观	90
4.4 在查询设计视图中创建		5.6 小结	90
和修改查询	62	综合练习五	90
4.4.1 设计查询	63	一、基础题	90
4.4.2 添加或删除表	63	二、思考题	91
4.4.3 修改字段	64	三、上机操作	91
4.4.4 设定查询的条件	65	第6章 创建报表	92
4.4.5 统计字段数据	67	6.1 报表的功能和特点	92
4.5 创建查询	68	6.1.1 表格式报表	92
4.5.1 选择查询	68	6.1.2 纵栏式报表	92
4.5.2 操作查询	68	6.1.3 标签	92
4.5.3 参数查询	70	6.2 快速创建报表	92
4.5.4 SQL 查询	71	6.2.1 纵栏式报表	93
4.6 小结	72	6.2.2 表格式报表	93
综合练习四	72	6.3 使用报表向导创建报表	94
一、基础题	72	6.4 创建标签	95
二、思考题	73		

目 录

6.5 在设计视图中创建报表	97	8.4.2 常量	125
6.5.1 报表的几个组成部分	97	8.4.3 数据类型	126
6.5.2 在报表中使用控件	98	8.4.4 转换数据类型	127
6.6 小结	100	8.4.5 数组	127
综合练习六	101	8.5 VBA 中的表达式	128
一、基础题	101	8.5.1 算术表达式	129
二、思考题	101	8.5.2 逻辑表达式	129
三、上机操作	101	8.5.3 条件表达式	129
第 7 章 宏的应用	102	8.5.4 字符串表达式	130
7.1 创建宏	102	8.6 VBA 中的赋值语句	130
7.1.1 在宏操作列表中选择 所需的操作	102	8.7 VBA 中的子程序	131
7.1.2 在宏窗口的“操作”栏中 键入操作名	103	8.7.1 创建子程序	131
7.1.3 从数据库窗口中拖动对象 到宏窗口的操作列	103	8.7.2 调用子程序	132
7.2 创建宏组	103	8.7.3 传递参数	132
7.3 运行和调试宏	104	8.8 VBA 控制程序代码的流程	133
7.3.1 运行宏	104	8.8.1 在程序中做出判断	133
7.3.2 调试宏	105	8.8.2 使用循环	135
7.4 常用的宏操作	106	8.9 VBA 程序的调试	137
7.5 小结	111	8.9.1 添加断点暂停程序的运行	138
综合练习七	112	8.9.2 单步执行 VBA 代码	138
一、基础题	112	8.9.3 调试与快速监视	139
二、思考题	112	8.9.4 跟踪 VBA 过程的调用	139
三、上机操作	112	8.9.5 语法检查及禁用	139
第 8 章 VBA 的应用	113	8.9.6 忽略错误处理	140
8.1 Visual Basic 与 VBA	113	8.10 运行 VBA 代码	140
8.2 VBA 中的模块	114	8.11 小结	141
8.2.1 建立新模块	114	综合练习八	141
8.2.2 创建与窗体和报表不相关 的模块	114	一、基础题	141
8.3 VBA 中的过程	115	二、思考题	141
8.3.1 创建过程	115	三、上机操作	141
8.3.2 Sub、Function 和 Property	117	第 9 章 Access 与 Web	142
8.4 VBA 中的数据	124	9.1 Microsoft Access 2002 中的 Web... ..	142
8.4.1 变量	124	9.1.1 Web 简介	142
8.4.2 常量	125	9.1.2 超链接	142
8.4.3 数据类型	126	9.1.3 数据访问页	143
8.4.4 转换数据类型	127	9.2 数据访问页的创建	144
8.4.5 数组	127	9.2.1 数据访问页的自动创建	144
8.5 VBA 中的表达式	128	9.2.2 运用向导创建数据访问页	145
8.5.1 算术表达式	129		
8.5.2 逻辑表达式	129		
8.5.3 条件表达式	129		
8.5.4 字符串表达式	130		
8.6 VBA 中的赋值语句	130		
8.7 VBA 中的子程序	131		
8.7.1 创建子程序	131		
8.7.2 调用子程序	132		
8.7.3 传递参数	132		
8.8 VBA 控制程序代码的流程	133		
8.8.1 在程序中做出判断	133		
8.8.2 使用循环	135		
8.9 VBA 程序的调试	137		
8.9.1 添加断点暂停程序的运行	138		
8.9.2 单步执行 VBA 代码	138		
8.9.3 调试与快速监视	139		
8.9.4 跟踪 VBA 过程的调用	139		
8.9.5 语法检查及禁用	139		
8.9.6 忽略错误处理	140		
8.10 运行 VBA 代码	140		
8.11 小结	141		
综合练习八	141		
一、基础题	141		
二、思考题	141		
三、上机操作	141		

目 录

9.3 数据访问页的编辑	146	二、思考题	161
9.3.1 使用设计视图	146	三、上机操作	161
9.3.2 Microsoft 脚本编辑器	147	第 11 章 数据库的安全与维护	162
9.4 在 Web 中发布和访问		11.1 数据库安全	162
Access 数据库	148	11.1.1 设置数据库密码	162
9.5 小结	150	11.1.2 数据库密码的清除	163
综合练习九	150	11.1.3 用户级安全机制	164
一、基础题	150	11.1.4 安全机制向导	166
二、思考题	151	11.1.5 删除用户级安全机制	169
三、上机操作	151	11.2 数据库加密	169
第 10 章 SQL Server 2000 Desktop		11.3 MDE 文件	170
Engine.....	152	11.4 优化数据库	171
10.1 SQL Server 2000 Desktop		11.4.1 压缩数据库	171
Engine 的功能	152	11.4.2 性能分析器	172
10.2 创建 Access 项目	152	11.5 小结	172
10.3 使用升迁向导	153	综合练习十一	173
10.3.1 升迁前的准备	154	一、基础题	173
10.3.2 升迁向导的使用方法	154	二、思考题	173
10.3.3 升迁向导报表	154	三、上机操作	173
10.3.4 升迁数据库对象	154	第 12 章 Access 开发实例	174
10.4 交互式 SQL	155	12.1 系统分析	174
10.4.1 表	156	12.2 表的设计	174
10.4.2 视图	156	12.3 查询的设计	179
10.4.3 约束	156	12.4 窗体的设计	180
10.4.4 触发器	159	12.5 安全机制	184
10.5 小结	160	12.6 小结	187
综合练习十	160		
一、基础题	160		

第1章 数据库基础

人们对计算机的应用从科学计算进入数据处理是一个划时代的转折,使计算机从单一的科学计算机工具变为人们日常工作中的得力助手。为了借助计算机科学地保存和管理大量复杂的数据,以便人们能方便而充分地利用这些宝贵的信息资源,经过不断地发展,逐渐形成了数据库技术。

1.1 什么是数据库

数据库是什么呢?实质上,数据库只不过是一些存在了很长时间——常常是许多年信息的聚集。

通常意义下,“数据库”(DataBase,简称DB)这个术语是指由数据库管理系统(Database Management System,简称DBMS)管理的数据聚集。一个数据库系统应该是:

1) 允许用户用一种叫做数据定义语言(Data Definition Language)的专用语言,建立新的数据库和指定它们的模式(Schema)。

2) 使用户能够用适当的语言查询数据。“查询”(Query)是一个数据库术语,指对数据的某种询问和更新,使用的语言通常称为“查询语言”或“数据操作语言”(Data Manipulation Language)。

3) 支持存储大量的数据。经过很长一段时间以后,仍保证安全,使其免遭意外或非授权的使用,同时允许对数据库查询和更新的有效访问。

4) 控制多用户的同时访问。使得一个用户的访问不影响其他用户,保证同时访问不会损坏数据。

1.2 数据库技术的发展

数据库技术的发展大致经历了人工管理阶段、文件系统阶段、数据库系统阶段。

1.2.1 人工管理阶段

这一阶段(20世纪50年代以前)计算机应用的背景是:计算机主要用于科学计算。从硬件上看,外存只有磁带、卡片、纸带,没有磁盘等直接存取的存储设备;从软件上看,没有操作系统,没有管理数据的软件,数据处理的方式是批处理。

这个时期数据管理的特点是:

1. 数据不保存

因为计算机主要应用于科学计算,一般不需要将数据长期保存。只是在计算某一课题时将数据输入,用完就撤走,不仅对用户数据这样处理,有时对系统软件也是这样。

2. 没有专用的软件对数据进行管理

程序员不仅要规定数据的逻辑结构,而且还要在程序中设计物理结构,包括存储结构,存取方法,输入输出方式等。因此程序中存取数据的子程序随着存储的改变而改变。即数

据与程序不具有独立性，这样不仅程序员必须花费许多精力在数据的物理布置上，而且数据在存储上有一点改变，就必须修改程序。

3. 只有程序概念，没有文件概念

数据的组织方式必须由程序员自行设计。

4. 数据不具有独立性

一组数据对应一个程序，数据是面向应用的。即使两个应用程序涉及某些相同的数据，也必须各自定义，无法互相利用，互相参照。所以程序和程序之间有大量重复的数据。

在人工管理阶段，程序与数据之间的对应关系如图 1-1 所示。

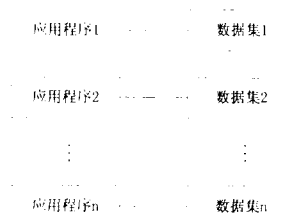


图 1-1 程序与数据之间的对应关系

1.2.2 文件系统阶段

20 世纪 50 年代后期到 60 年代中期，这时硬件方面已有了磁盘、磁鼓等直接存取存储设备；软件方面，操作系统中已经有了专门的数据管理软件，一般称为文件系统；处理方式上不仅有了批处理，而且能够联机实时处理。

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. 由文件系统管理数据

由专门的软件即文件系统进行数据管理，文件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，可以对文件进行修改、插入和删除的操作。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，程序员可以不必过多地考虑物理细节，将精力集中于算法。而且数据在存储上的改变不一定反映在程序上，大大节省了维护程序的工作量。

但是，文件系统仍存在以下缺点：

- 数据共享性差，冗余度大

在文件系统中，一个文件基本上对应于一个应用程序，即文件仍然是面向应用的。当不同的应用程序具有部分相同的数据时，也必须建立各自的文件，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。同时由于相同数据的重复存储、各自管理，容易造成数据的不一致性，给数据的修改和维护带来了困难。

- 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, 因此要想对现有的数据再增加一些新的应用会很困难, 系统不容易扩充。一旦数据的逻辑结构改变, 必须修改应用程序, 修改文件结构的定义。应用程序的改变 (例如应用程序改用不同的高级语言等), 也将引起文件的数据结构的改变。因此数据与程序之间仍缺乏独立性。可见, 文件系统仍然是一个不具有弹性的无结构的数据集合, 即文件之间是孤立的, 不能反映现实事物之间的内存联系。

在文件系统阶段, 程序与数据之间的关系如图 1-2 所示。

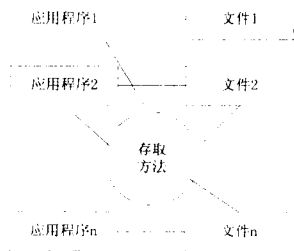


图 1-2 程序与数据之间的关系

1.2.3 数据库系统阶段

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, 数据量急剧增长, 同时多种应用、多种语言互相覆盖地共享数据集合的要求越来越强烈。这时硬件已有大容量磁盘, 硬件价格下降, 软件价格上升, 为编制和维护系统软件及应用程序所需的成本相对增加; 在处理方式上, 联机实时处理要求更多, 并开始剔除和考虑分布处理。在这种背景下, 以文件系统作为数据管理手段已经不能满足应用的需求, 于是为解决多用户、多应用共享数据的需求, 使数据为尽可能多的应用服务, 数据库技术便应运而生, 出现了统一管理数据的专门软件系统——数据库系统。

用数据库系统来管理数据比文件系统具有明显的优点, 从文件系统到数据库系统, 标志着数据管理技术的飞跃。其特点如下:

1. 采用复杂的数据模型 (结构)

数据模型不仅描述数据本身的特点, 还要描述数据之间的联系。这种联系通过存取路径实现。通过所有存取路径表示自然的数据联系时数据库与传统文件的根本区别。这样数据不再面向特定的某个或多个应用, 而是面向整个应用系统。数据冗余明显减少, 实现了数据共享。

2. 有较高的数据独立性

数据的物理结构与逻辑结构间差别可以很大。用户以简单的逻辑结构操作数据而无需考虑数据的物理结构。

数据库的结构分成用户的逻辑结构、整体逻辑结构、物理结构三级。用户的数据和外存中的数据之间转换由数据管理系统实现。

在物理结构改变时, 尽量不影响整体逻辑结构、用户的逻辑结构以及应用程序, 这就是物理数据独立性。在整体逻辑结构改变时, 尽量不影响用户的逻辑结构以及应用程序, 这就是逻辑数据独立性。

3. 提供方便的用户接口

数据库系统为用户提供了方便的用户接口，用户可使用查询语言或简单的中断命令操作数据库，也可以用程序方式操作数据库。

4. 提供数据控制功能

数据库管理系统（DBMS）提供以下四方面的数据控制功能：

1）数据完整性（Integrity）：保证数据库始终包含正确的数据。用户可以设计一些完整性规则以确保数据值的正确性。

2）数据安全性（Secutity）：保证数据的安全和机密，防止数据丢失或被窃取。

3）数据库的并发控制（Concurrency）：避免并发程序之间的相互干扰，防止数据库被破坏，杜绝提供给用户不正确的数据。

4）数据库的恢复（Recovery）：在数据库被破坏或数据不可靠时，系统有能力把数据库恢复到最近某时刻的正确状态。

这一阶段程序和数据的关系如图 1-3 所示。

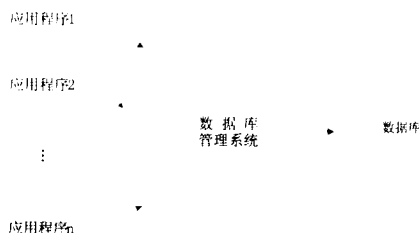


图 1-3 程序与数据的关系

综上所述，数据库是长期存储在计算机内有组织的大量的共享的数据集合。它可以供各种用户共享，具有最小冗余度和较高的数据独立性。DBMS 在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对系统进行恢复。

数据库系统的出现使信息系统从以加工数据的程序为中心，转向围绕共享的数据库为中心的新阶段。这样既便于数据的集中管理，又有利于应用程序的研制和维护，提高了数据的利用率和相容性，提高了决策的可靠性。

目前，数据库已经成为现代信息系统的不可分离的重要组成部分。具有数百万甚至数十亿字节信息的数据库已经普遍存在于科学技术、工业、农业、商业、服务业和政府部门的信息系统。

20 世纪 80 年代后不仅在大型机上，在多数微机上也配置了 DBMS，使数据库技术得到更加广泛的应用和普及。

数据库技术是计算机领域中发展最快的技术之一。数据库技术的发展是沿着数据模型的主线展开。

1.3 数据库管理系统（DBMS）

要了解数据库管理系统（Data Base Management System，简称 DBMS），先要了解数据

库系统 (Data Base System, 简称 DBS)。

数据库系统是一个复杂的系统,它是采用了数据库技术的计算机系统,是一个实际可运行的,按照数据库方式存储、维护和向应用系统提供数据或信息支持的系统。它是存储介质,处理对象和管理系统的集合体,通常由数据库、硬件、软件和数据管理员四部分组成。

数据库管理系统是指在 DBS 中对数据进行管理的软件系统,它是 DBS 的核心组成部分。DBS 中所有与数据库打交道的操作,包括建库、查询、更新及各种数据控制,都是通过 DBMS 进行的。

DBMS 把用户对数据库的操作从应用程序带到外部级、概念级,然后导向内部级,把数据作为可管理资源进行处理,实现了不同用户的共享,并增进了数据的安全性、完整性和可用性、独立性。

因此, DBMS 的主要功能主要有:

1) 对数据库定义。

在 DBMS 中包括了数据定义语言 (DDL) 的便宜程序,它定义了数据库的机构、模式及模式间的映象。

2) 数据库存取 DBMS 提供数据操纵语言 (Data Manipulation Language, 简称 DML) 实现对数据的基本操作:检索、插入、修改和删除。

3) 运行控制数据库的恢复、数据完整性控制、多用户环境下的并发控制和数据安全性控制都由 DBMS 实现。

4) 数据库的维护,数据库的原始数据录入、数据转换、数据转储、数据库修改,以及性能监视和分析功能等都由 DBMS 实现。

通常来说, DBMS 的功能根据系统的不同而不同,大型系统功能较强,小型系统功能较弱。

1.4 数据库模型

目前,数据库领域中最常用的数据模型有四种,它们是:

1) 层次模型 (Hierarchical Model)。

2) 网状模型 (Network Model)。

3) 关系模型 (Relational Model)。

4) 面向对象模型 (Object Oriented Model)。

其中层次模型和网状模型统称为非关系模型。

非关系层次模型的数据库系统在 20 世纪 70 年代至 80 年代初非常流行,在数据库系统产品中占据了主导地位,现在已逐渐被关系模型的数据库系统取代,但在美国等一些国家里,由于早期开发的应用系统都是基于层次数据库或网状数据库系统的,因此目前仍有不少层次数据库或网状数据库系统在使用。

20 世纪 80 年代以来,面向对象的方法和技术在计算机各个领域,包括程序设计语言、软件工程、信息系统设计、计算机硬件设计等各方面都产生了深远的影响,也促进数据库中面向对象数据模型的研究和发展。

下面我们逐一介绍前三种模型。

1.4.1 层次模型

层次模型是数据库系统中最早出现的数据模型，层次数据库系统采用层次模型作为数据的组织方式。

层次数据库系统的典型代表是 IBM 公司的 IMS (Information Management System) 数据库管理系统，这是 IBM 公司在 1968 年推出的第一个大型的商用数据库管理系统，曾经得到广泛使用。

层次模型用树形结构来表示各类实体以及实体间的联系。显示世界中许多实体之间的联系本来就呈现出一种很自然的层次关系，如行政机构、家族关系等。

1. 层次模型的数据结构

在数据库中定义满足下面两个条件的基本层次联系的集合为层次模型：

- 1) 有且只有一个节点没有双亲结点，这个结点称为根结点。
- 2) 根以外的其他结点有且只有一个双亲结点。

在层次模型中，每个结点表示一个记录类型，记录（类型）之间的联系用结点之间的连线（有向边）表示，这种联系是父子间的一对多的联系。这就使得层次数据库系统只能处理一对多的实体联系。

每个记录类型可包含若干个字段，这里，记录类型描述的是实体，字段描述的是实体的属性。

各个记录类型及其字段都必须命名。各个记录类型、同一记录类型中各个字段不能同名。

每个记录类型可以定义一个排序字段，也称为码字段，如果定义该排序字段的值是惟一的，则它能惟一地标识一个记录值。

一个层次模型在理论上可以包含任意有限个记录型和字段，但任何实际的系统都会因为存储容量或实现复杂度而限制层次模型中包含的记录型个数和字段的个数。

在层次模型中，同一双亲的子女结点称为兄弟结点（Twin 或 Sibling），没有子女结点的结点称为叶结点。

图 1-4 给出了一个层次模型的例子。其中 R1 为根结点；R2 和 R3 为兄弟结点，是 R1 的子女结点；R4 和 R5 为兄弟结点，是 R2 的子女结点；R3，R4 和 R5 为叶结点。

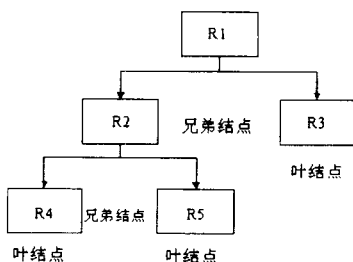


图 1-4 层次模型

从图上可以看出层次模型像一棵倒立的树，结点的双亲是惟一的。

层次模型的一个基本的特点是，任何一个给定的记录值只有按其路径查看时，才能显出它的全部意义，没有一个子女记录值能够脱离双亲记录值而独立存在。

2. 层次模型的优缺点

层次模型的优点主要有：

- 1) 层次模型本身比较简单。
- 2) 对于实体间联系是固定的，且预先定义好的应用系统，采用层次模型来实现，其性能优于关系模型，不低于网状模型。
- 3) 层次数据模型提供了良好的完整性支持。

层次模型的缺点主要有：

- 1) 现实世界中有很多联系是非层次性的，如多对多联系、一个结点具有多个双亲等，层次模型表示这类联系的方法很笨拙，只能通过引入冗余数据（易产生不一致性）或创建非自然的数据组织（引入虚拟结点）来解决。
- 2) 对插入和删除操作的限制比较多。
- 3) 查询子女结点必须通过双亲结点。
- 4) 由于结构严密，层次命令趋于程序化。

可见用层次模型对具有一对多的层次关系的部门描述非常自然、直观，容易理解。这是层次数据库的突出优点。

1.4.2 网状模型

在现实中事物之间的联系更多是非层次关系，用层次模型表示非树形结构是很不直接的，网状模型则可以克服这一弊病。

网状数据库系统采用网状模型作为数据的组织方式。网状数据模型的典型代表是 DBTG 系统，亦称 CODASYL 系统。这是 20 世纪 70 年代数据系统语言研究会 CODASYL（Conference on Data System Language）下属的数据库任务组（Data Base Task Group，简称 DBTG）提出的一个系统方案。DBTG 系统虽然不是实际的软件系统，但是它提出的基本概念、方法和技术具有普遍意义。它对于网状数据库系统的研制和发展起了重大的影响。后来不少的系统都采用 DBTG 模型或者简化的 DBTG 模型。例如，Cullinet Software 公司的 IDMS、Univac 公司的 DMS1100、Honeywell 公司的 IDS/2、HP 公司的 IMAGE 等。

1. 网状数据模型的数据结构

在数据库中，把满足以下两个条件的基本层次联系集合称为网状模型：

- 1) 允许一个以上的结点无双亲。
- 2) 一个结点可以有多个的双亲。

网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，允许多个结点没有双亲结点，允许结点有多个双亲结点，此外它还允许两个结点之间有多种联系（称之为复合联系）。因此网状模型可以更直接地去描述现实世界。而层次模型实际上是网状模型的一个特例。

与层次模型一样，网状模型中每个结点表示一个记录类型（实体），每个记录类型可包含若干字段（实体的属性），结点间的连线表示记录类型（实体）之间一对多的父子联

系。

从定义可以看出,层次模型中子女结点与双亲结点的联系是惟一的,而在网状模型中这种联系可以不惟一。因此,要为每个联系命名,并指出与该联系有关的双亲记录和子女记录。

2. 网状数据模型的优缺点

网状数据模型的优点主要有:

- 1) 能够更为直接地描述现实世界, 如一个结点可以有多个双亲。
- 2) 具有良好的性能, 存取效率较高。

网状数据模型的缺点主要有:

- 1) 结构比较复杂, 而且随着应用环境的扩大, 数据库的结构就变得越来越复杂, 不利于最终用户掌握。
- 2) DDL、DML 语言复杂, 用户不容易使用。

由于记录之间联系是通过存取路径实现的, 应用程序在访问数据时必须选择适当的存取路径, 因此, 用户必须了解系统结构的细节, 加重了编写应用程序的负担。

1.4.3 关系模型

关系模型是目前最重要的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式。

1970 年美国 IBM 公司 San Jose 研究室的研究员 E.F.Codd 首次提出了数据库系统的关系模型, 开创了数据库关系方法和关系数据理论的研究, 为数据库技术奠定了理论基础。由于 E.F.Codd 的杰出工作, 他于 1981 年获得 ACM 图灵奖。

20 世纪 80 年代以来, 计算机厂商新推出的数据库管理系统几乎都支持关系模型, 非关系系统的产品也大都加上了关系接口。数据库领域的当前研究工作也都是以关系方法为基础。

1. 关系数据模型的数据结构

关系模型与以往的模型不同, 它是建立在严格的数学概念的基础上。在用户观点下, 关系模型中数据的逻辑结构是一张二维表, 它由行和列组成。现在以学生登记表(如表 1-1 所示)为例, 介绍关系模型中的一些术语。

表 1-1 学生登记表

学号	姓名	年龄	性别	专业	年级
200115	张凡	18	男	微电子	2001
200116	王莹	18	女	计算机	2001
200117	李明	19	男	自动化	2001
...	...	...	...	...	...

1) 关系 (Relation): 一个关系对应通常好比是一张表, 如表 1-1 的这张学生登记表。

2) 元组 (Tuple): 表中的一行即为一个元组。

3) 属性 (Attribute): 表中的一列即为一个属性, 给每一个属性起一个名称即属性名。如表 1-1 有六列, 对应六个属性 (学号、姓名、年龄、性别、专业和年级)。

4) 主码 (Key): 表中的某个属性组, 它可以惟一确定一个元组, 如表 1-1 中的学号, 可以惟一确定一个学生, 也就成为本关系的主码。

5) 域 (Domain): 属性的取值范围, 如人的年龄一般在 1~90 岁之间, 大学生年龄属性的域是 (14~38), 性别的域是 (男、女), 专业域是一个学校所有系名的集合。

6) 分量: 元组中的一个属性值。

7) 关系模式: 对关系的描述, 一般表示为:

关系名 (属性 1, 属性 2, ..., 属性 n)

例如上面的关系可描述为:

学生 (学号、姓名、年龄、性别、专业和年级)

在关系模型中, 实体以及实体间的联系都是用关系来表示。例如学生、课程, 学生与课程之间的多对多联系在关系模型中可以如下表示:

学生 (学号、姓名、年龄、性别、专业和年级)

课程 (课程号、课程名、学分)

选修 (学号、课程号、成绩)

关系模型要求关系必须是规范化的, 即要求关系必须满足一定的规范条件, 这些规范条件中最基本的一条就是, 关系的每一个分量必须是一个不可分的数据项, 也就是说, 不允许表中还有表。

2. 关系数据模型的优缺点

关系数据模型具有下列优点:

1) 关系模型与非关系模型不同, 它是建立在严格的数学概念的基础上的。

2) 关系模型的概念单一。无论实体还是实体之间的联系都用关系表示。

对数据的检索结果也是关系 (即表)。所以其数据结构简单、清晰, 用户易懂易用。

3) 关系模型的存取路径对用户透明, 从而具有更高的数据独立性、更好的安全保密性, 也简化了程序员的工作和数据库开发建立的工作。

所以, 关系数据模型诞生以后发展迅速, 深受用户的喜爱。

当然, 关系数据模型也有缺点, 其中最主要的缺点是: 由于存取路径对用户透明, 查询效率往往不如非关系数据模型。因此为提高性能, 必须对用户的查询请求进行优化, 增加开发数据库管理系统的难度。

1.5 关系数据库标准语言——SQL

SQL (Structured Query Language, 结构化查询语言) 是 1974 年由 Boyce 和 Chamberlin 提出的, 并在 IBM 公司的关系 DBMS SYSTEM R 上实现的。它的前身是 1972 年提出的 SQUARE (Specifying Queries As Relational Expression) 语言, 在 1974 年作了修改, 改名为 SEQUEL (Structure English Query Language) 语言, 后简称为 SQL。

1975 年 ~ 1979 年 IBM 公司 San Jose Research Laboratory 研制了著名的关系数据库管理原型 System R 并实现了这种语言。由于它功能丰富、语言简洁, 倍受用户及计算机工业界欢迎, 被众多计算机公司和软件公司所采用。经各公司的不断修改和完善, SQL 语言最终发展为关系数据库的标准语言。