

Visual FoxPro

程序设计教程

```
FORMMyform1...
FORMMyform2=createobject...
FORMMyform2.show
READ EVENTS
DEFINE CLASS myForm1 AS form
    Top=40
    Left=100
    Caption=' 菜单1'
    ForeColor=RGB(255,0,0)
    **添加形状Shape1对象:
    ADD OBJECT Shape1 AS SHAPE WITH,
        Top=34,,
        Left=163,,
        Height=63,,
        FillStyle=7,,
        FillColor=RGB(255,0,0)
    **定义CLICK事件:
    PROCEDURE CLICK
        THISFORM.Shape1.FillColor=RGB(255,0,0)
        WAIT WINDOW ~ 菜单1被单击
        THISFORM.Shape1.FillColor=RGB(255,0,0)
        WAIT WINDOW ~ 菜单1被单击
    ENDPROC
    PROCEDURE DESTROY
        CLEAR EVENTS
    ENDPROC
ENDDEFINE

DEFINE CLASS myForm2 AS myForm1
    Caption=' 菜单2'
    Left=400
    ForeColor=RGB(0,0,0)
    **定义myForm2的CLICK事件:
    PROCEDURE CLICK
        MYFORM2.click
        WAIT WINDOW ~ 菜单2被单击
    ENDPROC
ENDDEFINE
```

主编 张吉春

计算机系列教材

郭施祎

高等学校计算机基础教育规划教材

西北工业大学出版社

高等学校计算机基础教育规划教材

Visual FoxPro 程序设计教程

主 编 张吉春 郭施祎

副主编 杜春玲 孙秋厚

崔忠凤 赵春燕

西北工业大学出版社

【内容提要】本书是高等学校计算机基础教育规划教材，是一本通用的关系型数据库管理系统实用教材，主要介绍数据库基本原理和数据库管理系统软件 Visual FoxPro 6.0 的相关知识。具体内容包括：数据库与 Visual FoxPro 6.0 的概述、数据与数据运算、数据表的操作、索引与查询、报表和标签、表单、程序设计基础、结构化查询语言 SQL、面向对象的程序设计、应用程序连编和发布、多用户与数据共享。书末附录给出了课程设计实例和总复习题。

本书思路全新，图文并茂，练习丰富，既可作为高等院校计算机专业“数据库原理与应用”课程的教材，也可作为各类成人教育计算机专业的教材或从事数据库应用开发技术人员的参考资料，还可以作为 Visual FoxPro 培训班教材。

图书在版编目 (CIP) 数据

Visual FoxPro 程序设计教程/张吉春，郭施祎主编. —西安：西北工业大学出版社，2007.8

(高等学校计算机基础教育规划教材)

ISBN 978-7-5612-2267-6

I. V… II. ①张…②郭… III. 关系数据库—数据库管理系统, Visual FoxPro—程序设计—高等学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2007) 第 121170 号

出版发行：西北工业大学出版社

通信地址：西安市友谊西路 127 号 邮编：710072

电 话：(029) 88493844 88491757

网 址：www.nwpup.com

电子邮箱：computer@nwpup.com

印 刷 者：陕西丰源印务有限公司

开 本：787 mm×1 092 mm 1/16

印 张：19.5

字 数：523 千字

版 次：2007 年 8 月第 1 版

2007 年 8 月第 1 次印刷

定 价：28.00 元

前 言

Visual FoxPro 6.0 是 Microsoft 公司推出的关系数据库管理系统及面向对象和可视化的数据库应用系统开发工具,它具有强大的性能、功能完备的工具、极其友好的用户界面、简单的数据存取方式、良好的兼容性、真正的可编译性、独一无二的跨平台技术和较强的安全性,是目前最实用的数据库管理系统软件之一。

与早期的 FoxBase 或 FoxPro 相比,它引入了面向对象程序设计思想,可以进行面向对象和可视化的程序设计,无论是从数据库的概念、使用和管理,还是从数据库应用系统的开发、速度、能力和灵活性等方面,都是早期的软件无法比拟的。

本书每章末均配有习题和上机实验,便于读者练习、自学。全书在编排上注意依据教学特点,突出重点,删繁就简,力求通俗易懂、简洁实用。本书适合大学、高职高专、成人教育及各类中等学校选作教材,也可用于从事计算机工作的科技人员专业培训和自学参考。

全书共分 11 章。第 1 章主要介绍了数据库与 Visual FoxPro 概述,包括数据库系统基本概念、关系数据库系统、数据库应用系统等内容;第 2 章介绍了数据与数据运算;第 3~5 章介绍了数据表的操作、索引与查询、报表与标签;第 6 章详细介绍了表单,主要包括表单的创建、添加表单控件以及美化表单;第 7 章详细介绍了 Visual FoxPro 6.0 程序设计基础;第 8 章介绍了结构化查询语言 SQL;第 9 章介绍了面向对象的程序设计;第 10 章介绍了应用程序的连编和发布;第 11 章介绍了多用户与数据共享。书末附录给出了课程设计实例和总复习题。

本书由张吉春、郭施玮担任主编,杜春玲、孙秋厚、崔忠凤、赵春燕担任副主编。参加本书编写的还有(以姓氏笔画为序):元静、刘彤、李环宇、张蓓、高玲玲、殷玉法。

本书在编写和出版过程中,参阅了一些专家和同行的科研资料、专著、教材,在此一并表示感谢!由于编者水平有限,书中难免有不足之处,敬请广大读者批评指正。

编 者
2007 年 6 月

目 录

第1章 数据库与 Visual FoxPro 概述..... 1	
1.1 数据库系统基本概念..... 1	
1.1.1 数据、信息和数据库..... 1	
1.1.2 数据库系统..... 2	
1.1.3 数据模型..... 4	
1.1.4 数据库管理系统..... 6	
1.2 关系数据库系统..... 6	
1.2.1 关系术语..... 6	
1.2.2 关系的特点..... 7	
1.2.3 关系设计..... 7	
1.2.4 关系运算..... 9	
1.3 数据库应用系统..... 10	
1.4 Visual FoxPro 概述..... 10	
1.4.1 Visual FoxPro 的特点..... 10	
1.4.2 Visual FoxPro 的运行环境和安装..... 12	
1.4.3 Visual FoxPro 的启动与退出..... 13	
1.4.4 Visual FoxPro 的用户界面及 工作方式..... 14	
1.4.5 Visual FoxPro 的辅助设计工具..... 19	
1.5 Visual FoxPro 的命令格式及文件 类型..... 21	
1.5.1 Visual FoxPro 的命令格式..... 22	
1.5.2 Visual FoxPro 的文件类型..... 23	
本章小结..... 24	
习题一..... 25	
上机实验一..... 25	
第2章 数据与数据运算..... 26	
2.1 数据类型..... 26	
2.2 常量与变量..... 27	
2.2.1 常量..... 28	
2.2.2 变量..... 28	
2.2.3 内存变量赋值命令..... 29	
2.3 表达式..... 30	
2.3.1 运算符..... 30	
2.3.2 表达式..... 33	
2.4 常用函数..... 34	
2.4.1 字符及字符串处理函数..... 34	
2.4.2 数学运算函数..... 35	
2.4.3 转换函数..... 36	
2.4.4 日期函数..... 37	
2.4.5 测试函数..... 38	
2.4.6 键值函数..... 41	
2.4.7 数组函数..... 42	
2.4.8 其他函数..... 43	
本章小结..... 45	
习题二..... 45	
上机实验二..... 46	
第3章 数据表的操作..... 47	
3.1 项目管理器..... 47	
3.1.1 项目管理器简介..... 47	
3.1.2 管理项目的数据..... 49	
3.1.3 管理项目的文档..... 50	
3.1.4 管理项目的各种操作..... 50	
3.2 使用工具栏..... 54	
3.2.1 工具栏简介..... 54	
3.2.2 创建新工具栏..... 55	
3.2.3 修改 Visual FoxPro 6.0 工具栏..... 55	
3.3 创建数据库与表..... 56	
3.3.1 表和数据库..... 56	
3.3.2 建立工作目录..... 57	
3.3.3 建立数据库文件..... 58	
3.3.4 建立表文件..... 59	
3.4 编辑数据表..... 63	
3.4.1 显示表记录..... 63	
3.4.2 浏览窗口的分隔..... 66	
3.4.3 添加表记录..... 67	
3.4.4 不同类型文件的复制..... 69	
3.4.5 查找与替换..... 70	
3.5 维护数据表..... 74	

3.5.1 修改表文件结构	74	5.3 创建标签	137
3.5.2 输入字段中的记录	76	5.3.1 标签概念	137
3.5.3 删除表文件记录	78	5.3.2 利用标签向导创建标签	138
3.5.4 数据表字段特性	80	5.3.3 利用标签设计器创建标签	140
本章小结	83	5.3.4 打印报表或标签	142
习题三	83	本章小结	142
上机实验三	84	习题五	143
第4章 索引与查询	86	上机实验五	143
4.1 索引数据表	86	第6章 表单设计	144
4.1.1 索引关键字	86	6.1 简单表单的创建	144
4.1.2 建立索引	87	6.1.1 表单创建方法与种类	144
4.1.3 查看和修改索引	90	6.1.2 利用向导创建表单	144
4.1.4 使用索引	90	6.1.3 保存及运行表单	147
4.2 信息查询	92	6.1.4 创建一对多表单	149
4.2.1 查询含义	92	6.2 快速创建表单	152
4.2.2 创建查询文件	92	6.2.1 启动表单设计器	152
4.2.3 建立数据表间的联接	94	6.2.2 快速创建表单	154
4.2.4 设定查询条件	95	6.2.3 利用命令创建表单	156
4.2.5 设置数据排序	96	6.3 添加表单控件	157
4.2.6 设置分组查询	97	6.3.1 表单控件简介	157
4.2.7 杂项设置	97	6.3.2 添加文本框和标签控件	158
4.3 视图	99	6.3.3 添加组合框和列表框控件	159
4.3.1 视图含义	99	6.3.4 添加命令按钮控件	163
4.3.2 创建本地视图	99	6.3.5 添加页框控件	164
4.3.3 创建远程视图	104	6.3.6 添加表格控件	165
本章小结	107	6.4 美化表单	169
习题四	107	6.4.1 完善控件按钮	169
上机实验四	108	6.4.2 改变控件的格式	170
第5章 报表与标签	109	6.4.3 改变控件的颜色	171
5.1 创建简单报表	109	6.4.4 布局表单中的对象	172
5.1.1 报表布局	109	6.4.5 设置对象的“Tab”键次序	173
5.1.2 利用向导创建报表	111	本章小结	174
5.1.3 利用一对多向导创建报表	115	习题六	174
5.2 快速创建报表	118	上机实验六	175
5.2.1 快速创建报表	118	第7章 Visual FoxPro 6.0 程序设计	
5.2.2 设置报表布局	121	基础	177
5.2.3 报表控件的使用	124	7.1 结构化程序设计纲要	177
5.2.4 美化报表设计	129		

7.1.1 对问题进行说明	177	上机实验八	229
7.1.2 分解问题	177	第 9 章 面向对象的程序设计	230
7.1.3 编制模块	178	9.1 基本概念	230
7.1.4 示例程序	179	9.1.1 对象与类	230
7.1.5 测试模块	180	9.1.2 类的特征和基类	233
7.1.6 组装全部模块	180	9.1.3 事件	235
7.1.7 整体测试	182	9.2 定义类的程序方式	237
7.2 基本语句和状态设置命令	182	9.2.1 定义类的命令	237
7.2.1 输出命令	182	9.2.2 向容器类中添加对象	239
7.2.2 基本输入	186	9.2.3 指定方法和事件程序	241
(ACCEPT/INPUT/WAIT)	186	9.2.4 保护和隐藏类成员	242
7.3 程序的控制结构	188	9.2.5 按类层次调用方法或事件代码	244
7.3.1 顺序结构	188	9.3 利用类设计器定义类	246
7.3.2 选择结构	189	9.4 创建和使用类库	250
7.3.3 循环结构	195	9.4.1 创建类库的方法	250
7.4 过程与函数	199	9.4.2 修改库中的类定义	250
7.4.1 函数	199	9.4.3 使用类库的命令	251
7.4.2 过程和过程设计	200	9.5 面向对象程序设计与实例	252
本章小结	203	9.5.1 设计一个应用系统的具体实现	252
习题七	203	过程	252
上机实验七	205	9.5.2 一个数据库应用系统实例	253
第 8 章 结构化查询语言 SQL	206	本章小结	267
8.1 SQL 概要	206	习题九	267
8.2 SQL 数据定义功能	206	上机实验九	268
8.2.1 建立表结构	207	第 10 章 应用程序的连编和发布	269
8.2.2 修改表结构	210	10.1 调试器	269
8.2.3 建立临时表	212	10.1.1 调试器的设置	269
8.2.4 删除表	213	10.1.2 调试器的使用	270
8.3 SQL 的数据修改功能	213	10.2 应用程序的连编	274
8.3.1 插入数据	213	10.2.1 设置主文件	274
8.3.2 删除数据	214	10.2.2 连编应用程序	276
8.3.3 更新数据	215	10.3 应用程序的发布	277
8.4 SQL 的数据查询功能	216	本章小结	282
8.4.1 简单查询	218	习题十	282
8.4.2 嵌套查询	222	上机实验十	283
8.4.3 连接查询	224	第 11 章 多用户与数据共享	284
8.4.4 分组及使用库函数查询	225	11.1 多用户环境中的数据访问	284
本章小结	227		
习题八	228		

11.1.1 多用户环境中数据访问的方式	284	11.4 冲突处理	295
11.1.2 数据锁定的方法	285	11.4.1 死锁	295
11.1.3 数据解锁的方法	289	11.4.2 出错处理例程	296
11.1.4 数据工作期	289	11.4.3 冲突检测的函数	296
11.2 数据缓冲技术	290	本章小结	296
11.2.1 缓冲方法与锁定方式	290	习题十一	296
11.2.2 使用缓冲技术更新数据	292	上机实验十一	297
11.3 事务处理	293	附 录	298
11.3.1 事务处理的控制命令	293	附录 1 信息管理课程设计	298
11.3.2 使用事务	294	附录 2 总复习题	299
11.3.3 事务处理的嵌套	295		

第 1 章 数据库与 Visual FoxPro 概述

1.1 数据库系统基本概念

1.1.1 数据、信息和数据库

数据 (Data) 在一般意义上被认为是对客观事物特征所进行的一种抽象化、符号化表示。例如, 某人出生日期是 1985 年 6 月 28 日, 身高 1.72 m, 体重 66 kg, 其中 1985 年、6 月、28 日、1.72 m 和 66 kg 等就是数据, 它们描述了该人的某些特征。数据可以有不同的形式。例如, 出生日期也可以表示成 “1985.6.28”、“{06/28/85}” 等形式。需要明确的是, 这里数据的概念比在科学计算领域中已大大地拓宽了, 不仅包括数字、字母、文字及其他特殊字符组成的文本形式的数据, 而且还包括图形、图像、声音等多媒体数据。总之, 凡是能够被计算机处理的对象都称为数据。

信息 (Information) 通常被认为是有一定含义的、经过加工处理的、对决策有价值的信息。请看一个简单例子: 某排球队中, 每个队员的身高数据为 1.85 m, 1.97 m, 1.86 m……经过计算得到平均身高为 1.88 m, 这便是该排球队的一条重要信息。又如, 今年入学的所有新生中, 每个人的出生日期为原始数据, 用今年年份减去出生日期中的年份, 得到每个人的年龄 (可视为二次数据), 再由每个人的年龄求出平均年龄, 即得到有用的信息, 它反映出本届学生整体的年龄状况。数据与信息之间的关系可以表示为

$$\text{信息} = \text{数据} + \text{处理}$$

其中, 处理是指将数据转换为信息的过程, 包括数据的收集、存储、加工、排序、检索等一系列活动。数据处理的目的是从大量的现有数据中, 提取对人们有用的信息, 作为决策的依据。可见, 信息与数据是密切相关的。数据是信息的载体, 它表示了信息; 信息是数据的内涵, 是数据的语义解释。信息是有价值的。信息的价值取决于它的准确性、及时性、完整性和可靠性。为了提高信息的价值, 就必须用科学的方法来管理信息, 这种方法就是数据库技术。

数据库 (DataBase, DB) 是指存储在计算机存储设备上、结构化的相关数据的集合。请注意, 这些数据是以二进制形式存储在磁盘、光盘、半导体存储器等存储介质上的。那么, 它们又是如何存储的呢? 众所周知, 图书馆书库中的图书是按一定规则 (即藏书模型) 分门别类整齐地排列在书架上的, 读者查阅起来十分方便。试想, 如果数以百万计的图书杂乱无章地堆放在一起, 要从中找出一本所需要的书, 那简直如同大海捞针! 同理, 为了便于检索和使用数据, 数据库中的大量数据也必须按照一定的规则 (即数据模型) 来存放, 这就是所谓的 “结构化”。此外, 存储在数据库中的数据彼此之间是有一定联系的, 而不是毫不相干的。例如, 学生成绩数据库中存储的是学号、姓名、性别及其各门课程的成绩等, 这些数据是相关的。可见, 数据库不仅包括描述事物的数据, 而且反映了相关事物之间的联系。数据以什么形式存储, 什么是结构化, 何谓相关数据, 这几个问题明确了, 你对数据库也就有了一个全面而深入的理解, 而不是简单地将它看作是存储数据的仓库。

1.1.2 数据库系统

1. 数据管理技术的发展

计算机数据管理技术随着计算机硬件和软件技术的发展而不断发展,大致经历了人工管理、文件系统和数据库系统三个发展阶段。

(1) 人工管理阶段: 20 世纪 50 年代中期以前, 计算机主要用于科学计算, 还没有专门用于管理数据的软件。数据与计算或处理它们的程序在一起。如果数据的类型、格式、数量或输入输出方式改变了, 程序也必须作相应的修改, 数据与程序不具有独立性。一个程序中的数据, 其他程序不能使用, 因此, 各程序之间存在大量的重复数据, 被称为数据冗余。

(2) 文件系统阶段: 20 世纪 50 年代后期至 60 年代, 计算机开始大量地用于管理中的数据处理工作。在软件方面, 出现了高级语言和操作系统。操作系统中的文件系统是专门管理外存储器的数据管理软件。程序和数据可以分别存储为程序文件和数据文件, 因而程序与数据有了一定的独立性。常用的高级语言 FORTRAN、BASIC、C 等都支持使用数据文件。这个阶段称为文件系统阶段。

文件系统阶段对数据的管理虽然有了长足的进步, 但是, 一些根本性问题并没有得到解决。例如, 数据冗余度大, 同一数据项在多个文件中重复出现; 缺乏数据独立性, 数据文件只是为了满足专门需要而设计的, 供某一特定应用程序使用, 数据和程序相互依赖; 数据无集中管理, 各个文件没有统一管理机制, 无法相互联系, 各自为政, 其安全性与完整性无法保证。诸如此类的问题造成了文件系统管理的低效率、高成本, 促使人们研究新的数据管理技术。

(3) 数据库系统阶段: 从 20 世纪 60 年代后期开始, 随着社会信息量的迅速增长, 需要计算机管理的数据量急剧增长, 文件系统越来越不能适应管理大量数据的需要。同时, 人们对数据共享的需求日益增强。计算机技术的迅猛发展, 特别是大容量磁盘开始使用, 在这种社会需求和技术成熟的条件下, 数据库技术应运而生, 使得数据管理技术进入崭新的数据库系统阶段。

数据库系统克服了文件系统的种种弊端, 它能够有效地储存和管理大量的数据, 使数据得到充分共享, 使数据冗余大大减少, 使数据与应用程序彼此独立, 并提供数据的安全性和完整性统一机制(数据的安全性是指防止数据被窃取和失密, 数据的完整性是指数据的正确性和一致性)。用户可以以命令方式或程序方式对数据库进行操作, 方便而高效。数据库系统的优越性使其得到迅速发展和广泛应用。从大型机到微型机, 从 UNIX 到 Windows, 推出了许多成熟的数据库管理软件, 如 Oracle、Sybase、FoxBASE、FoxPro 和 Visual FoxPro, 等等。今天, 数据库系统已成为计算机数据管理的主要方式, 而由文件系统支持的数据文件, 仅在数据量较小的场合下使用。

计算机网络技术的迅速发展为数据库提供了更好的运行环境, 使数据库系统从集中式发展到分布式。所谓集中和分布是对数据存放地点而言的。分布式数据库把数据分散存储在网络的多个节点上, 各个节点上的计算机可以利用网络通信功能访问其他节点上的数据库资源。例如, 一个银行有众多储户。如果所有储户的数据都存放在一个集中式数据库中, 所有储户存款、取款时都要访问这个数据库, 数据传输量必然很大。如果使用分布式数据库, 将众多储户的数据分散存储在离各自住所较近的储蓄所, 则大多数储户就可以就近存取, 仅有少量数据需要远程调用, 从而大大减少了网上的数据传输量, 提高了运行效率。

值得一提的是, 近年来, 智能数据库的研究取得了可喜的进展。传统数据库存储的数据都是已知的事实, 智能数据库除了存储已知的事实外, 还能存储用于逻辑推理的规则, 故又称为“基于规则的数据库”(rule-based database)。例如, 某智能数据库中存有“科长领导科员”的规则, 如果同时存有

“甲是科长”、“乙是科员”等数据，它就能够推理得出“甲领导乙”的新事实。随着人工智能逐步走向实用化，对智能数据库的研究日趋活跃。演绎数据库、专家数据库和知识库系统等都属于智能数据库的范畴。

2. 数据库系统的组成

通常把引进了数据库技术的计算机系统称为数据库系统 (DataBase System, DBS)。数据库系统是由以下几部分组成的：

(1) 计算机硬件系统：需要有容量足够大的内存和外存，用来运行操作系统、数据库管理系统核心模块和应用程序，以及存储数据库。

(2) 数据库集合：存储在计算机外存设备上的若干个设计合理、满足应用需要的数据库。

(3) 数据库管理系统 (DBMS)：这是帮助用户创建、维护和使用数据库的软件系统，是数据库系统的核心。比较流行的微机数据库管理系统有：FoxBASE、FoxPro for Windows、Visual FoxPro、Access 等。

(4) 相关软件：包括操作系统、编译系统、应用开发工具软件和计算机网络软件等。较大型的数据库系统，通常是建立在多用户系统或网络环境中的。

(5) 人员：包括数据库管理员和用户。在大型数据库系统中，需要有专人负责数据库系统的建立、维护和管理的工作，承担该任务的人员称为数据库管理员。用户可分为两类：专业用户和最终用户。专业用户侧重设计数据库、开发应用系统程序，为最终用户提供友好的用户界面。最终用户侧重对数据库的使用，主要是对数据库进行联机查询，或者通过数据库应用系统提供的界面使用数据库。

3. 数据库系统的特点

数据库系统的主要特点是：数据结构化、数据共享、数据独立性和统一的数据控制功能。

(1) 数据结构化：数据库中的数据是有结构的，这种结构是由数据库管理系统所支持的数据模型表现出来的。数据库系统不仅可以表示事物内部各数据项之间的联系，而且可以表示事物与事物之间的联系。

(2) 数据共享：数据共享就是说数据库中的数据可以被多个用户、多种应用访问，这是数据库系统最重要的特点。数据冗余是指数据的重复。

由于数据库中的数据被集中管理，统一组织、定义和存储，可以避免不必要的冗余，因而也避免了数据的不一致性。与此同时，还带来了对数据的灵活应用方式，可以取整体数据的各种合理子集用于不同的应用系统。正因为如此，才使得多用户、多应用共享数据成为现实。

(3) 具有较高的数据独立性：在数据库系统中，数据与应用程序之间的相互依赖大大减小，数据的修改对程序不会产生大的影响或没有影响，具有较高的数据独立性。

(4) 具有统一的数据控制功能：数据共享必然伴随着并发操作，即多个用户同时使用同一个数据库。为此，数据库系统必须提供必要的保护措施，主要是以下 3 种数据控制功能：

1) 安全性控制：数据库系统设置了一套安全保护措施，只有合法用户才能进行指定权限的操作，以防止不合法的使用所造成数据的泄密和被破坏。

2) 完整性控制：数据的完整性是指数据的正确性、有效性和相容性。系统提供必要的功能来保证数据库中的数据在输入、修改过程中始终符合原来的定义和规定。此外，当计算机系统发生故障而破坏了数据或对数据的操作发生错误时，系统能提供相应机制，将数据恢复到正确状态。

3) 并发控制：当多个用户的并发进程同时存取、修改数据库时，可能会相互干扰而得到错误的

结果，并使数据库的完整性遭到破坏，因此必须对多用户的并发操作予以控制和协调。

并发控制中有一概念称为事务 (Transaction)，它是并发控制的基本单位与控制对象。事务是一操作系列，这些操作要么都做，要么都不做。两事务的并发操作可能造成数据的错误，通常采用封锁措施来保证数据的正确性。例如，事务 T1 要修改数据 A，首先封锁它，执行完读写操作之后才解锁 A。在事务 T1 的执行过程中，如果事务 T2 也提出对数据 A 的封锁要求，则必须等待，直到事务 T1 解锁数据 A 后，T2 才能获得对数据 A 的控制权。

1.1.3 数据模型

客观存在并且可以相互区别的事物称为实体。实体可以是实际事物 (例如，一个学生，一个部门等)，也可以是抽象事件 (如借阅若干本书，一场足球比赛等)。同类型实体的集合构成一个实体集。

实体的特性称为属性。例如，学生实体可以用学号、姓名、性别、出生日期等属性来描述；图书实体可以用编号、书名、作者、出版社、定价等属性来描述。

实体之间的对应关系称为联系，它反映现实世界事物之间的相互关联。联系分为两种，一种是实体内部各属性之间的联系，另一种是实体之间的联系。实体之间的联系有以下 3 种类型：

(1) 一对一联系：例如，如果一所学校只有一位校长，他不能同时在其他学校兼任校长，那么学校与校长这两个实体之间存在一对一的联系。

(2) 一对多联系：例如，一所学校有多名学生，而一名学生只能在一所学校里注册，学校与学生两个实体之间便存在一对多的联系。

(3) 多对多联系：例如，我们来看学生和课程两个实体，一名学生可以选修多门课程，而一门课程可以被多名学生选修，可见，学生与课程之间存在多对多的联系。又如，科研课题与科研人员两个实体之间也存在多对多的联系。

在一个数据库系统中，为了反映事物本身及事物之间的各种联系，数据库中的数据必须具有一定的结构，这种结构用数据模型来表示。任何一个数据库管理系统都是基于某种数据模型的。数据模型主要有 3 种：层次模型、网状模型和关系模型。

1. 层次模型

利用树型结构表示实体及其之间联系的模型称为层次模型。图 1.1.1 是一个层次模型的例子，它体现出实体之间一对多的联系。

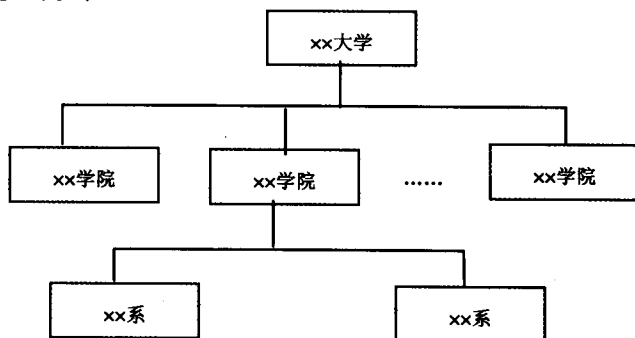


图 1.1.1 层次模型

2. 网状模型

利用网状结构表示实体及其之间联系的模型称为网状模型。该模型体现多对多的联系，具有很大的灵活性。图 1.1.2 给出了一个用网状模型表示某汽车制造厂中技术人员、图纸、工人和工件之间的联系。

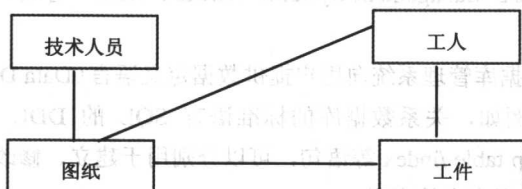


图 1.1.2 网状模型

3. 关系模型

用二维表结构表示实体以及实体之间联系的模型称为关系模型。关系模型把各种联系都统一描述成一些二维表，即由若干行和若干列组成的表格。每一个这样的二维表格就称为一个关系。例如，表 1.1 就是一个关系。

表 1.1 学生关系实例

学 号	姓 名	性 别	出生日期	个人简历	照 片
0205001	道明寺	男	02/25/81	memo	Gen
0205002	花泽类	男	05/23/80	memo	Gen
0205003	林杉菜	女	03/23/78	memo	Gen
0205004	西门	男	05/04/79	memo	Gen
0205005	美作	男	08/24/79	memo	Gen
0205006	李真	女	07/21/80	memo	Gen
0205007	腾堂静	女	01/14/80	memo	Gen

对我们来说，无论是浏览还是设计这样一张二维表格都没有什么困难，即是说，关系模型很容易被用户所接受，并且关系模型以关系数学理论为基础，因此基于关系模型的关系型数据库管理系统成为当今最为流行的数据库管理系统。这就是本书重点介绍关系型数据库管理系统的根本原因。

经过 30 多年的发展，基于不同数据模型的数据库系统先后经历了第一代和第二代，现在正向第三代前进。

第一代数据库系统是非关系型数据库系统，包括层次型和网状型数据库系统。第二代数据库系统是关系型数据库系统。与第一代数据库系统相比，第二代数据库系统的突出优点有两个：一是采用二维表作为数据结构，简单明了、易学易用；二是查询效率高，仅用一条命令即可访问整个二维表，而第一代数据库每次仅能访问一条记录。此外，通过多表联合操作还能对有联系的若干个二维表实现“关联”查询。

随着多媒体技术的发展，要求数据库能够存储图形、声音等复杂对象。将数据库技术与面向对象技术相结合的第三代数据库系统，即对象—关系数据库系统 ORDBS (Object Relational DataBase Systems) 于 20 世纪 80 年代中期开始研究，虽然目前尚处于发展过程之中，但已经显示出其光明的前景。

1.1.4 数据库管理系统

在数据库系统中,数据成为多个用户或多个应用程序的共享资源,它已经从应用程序中独立出来,由数据库管理系统(DataBase Management System, DBMS)来统一管理。一般地说,数据库管理系统具有下列功能:

(1)数据定义功能:数据库管理系统向用户提供数据定义语言(Data Definition Language, DDL),用于描述数据库的结构。例如,关系数据库的标准语言 SQL 的 DDL 语言,一般设置有 create table/index, alter table, drop table /index 等语句,可以分别用于建立、修改或删除关系数据库的二维表结构,或者定义或删除数据库表的索引。

(2)数据操作功能:数据库管理系统向用户提供数据操作语言(Data Manipulation Language, DML),支持用户对数据库中的数据进行查询、追加、插入、删除、修改等操作。在不同的数据库管理系统中,数据操作语言的语法格式也不同。按其实现方法可分为两类:一类数据操作语言可以独立使用,不依赖于任何其他程序设计语言,称为自含型或自主型语言;另一类是宿主型数据操作语言,需要嵌入到宿主语言(例如 FORTRAN, COBOL, C 等)中使用。在使用高级语言编写应用程序时,如果需调用数据库中的数据,则需要用宿主型数据操作语言的语句来实现。因此,数据库管理系统必须包含数据操作语言的编译或解释程序。

(3)控制和管理功能:数据库中的数据是宝贵的共享资源,必须有一定的控制手段来保障资源不受破坏。因此,数据库管理系统必须具有控制和管理功能。如前所述,在多用户使用数据库时对数据进行的“并发控制”,对用户权限实施监督的“安全性检查”,数据的备份、恢复和转储功能等。数据库系统的规模越大,这类功能也就越强。需要指出的是,一种数据库管理系统只能支持一种模型的数据库系统。

目前,微机上使用的数据库管理系统都是关系型数据库管理系统,简称 RDBMS (Relational DBMS)。它们提供的数据库语言都具有“一体化”的特点,即集数据定义语言 DDL 和数据操作语言 DML 于一体,在数据库管理系统 DBMS 的统一管理下完成上述各种功能。

(4)数据字典:数据字典(Data Dictionary)中存放着系统中所有数据的定义和设置信息。例如,字段属性、记录规则、表间关系等,提供了对数据库数据描述的集中管理手段。它使用户在设计和修改数据库时有更大的灵活性。例如用户可设置字段默认值,创建数据库表间的永久关系等。

1.2 关系数据库系统

目前,关系数据库理论日趋成熟,在微机数据库系统中得到普遍的应用。关系数据库可分为单用户数据库和多用户数据库。早期的微机数据库都是单用户系统,只允许一人使用。随着局域网应用的发展,供网络用户共享的多用户数据库开始流行。Visual FoxPro 就是一种多用户数据库系统。在它以前,已有 FoxBASE+、FoxPro 等多用户数据库供微机用户选用。

1.2.1 关系术语

如前所述,一个二维表就是一个关系。常用的关系术语如下:

记录:二维表中每一行称为一个记录或称为一个元组。

字段：二维表中每一列称为一个字段或称为一个属性。

例如，在表 1.1 中，学号、姓名、性别、出生日期等都是字段名。字段名是变量，它们有不同的类型和宽度，对于数值型字段，还有小数位数。字段名、字段类型、字段宽度和小数位数称为字段的定义。一条记录中各字段的具体内容称为字段值。

域：即属性的取值范围。例如，在考试成绩关系中，“数学”字段的域为 0~100。在学生关系中，“性别”字段的域是“男”和“女”。

关键字：在一个关系中有这样一个或几个字段，它（们）的值可以唯一地标识一条记录，称之为关键字（Key）。例如，在学生关系中，学号是关键字。

关系模式：对关系的描述称为关系模式，其格式为

关系名(属性名 1, 属性名 2, ..., 属性名 n)

一个关系模式对应一个关系结构，是命名的属性集合。

1.2.2 关系的特点

在关系模型中，每一个关系模式都必须满足一定的要求，即关系必须规范化。规范化的关系应具有以下特点：

(1) 每一个字段均不可再分，即是说，表中不能再包含表。例如，手工制表时经常绘制复合表，如表 1.2 所示的职工工资表。这种表格不是二维表，因而不能直接存放到数据库中。如果删除表中的“应发部分”和“扣除部分”，变成表 1.3 的形式，则就是二维表了。

表 1.2 职工工资表

职工号	姓 名	应发部分			扣除部分		实发金额
		工资	津贴	奖金	水电费	公积金	
110	林小优	500	100	200	50	60	690
111	祁剑波	650	120	230	60	40	900
112	惠天地	550	200	300	50	50	950

表 1.3 职工工资表

职工号	姓 名	工 资	津 贴	奖 金	水电费	公积金	实发金额
110	林小优	500	100	200	50	60	690
111	祁剑波	650	120	230	60	40	900
112	惠天地	550	200	300	50	50	950

(2) 同一个关系中不能有相同的字段名。

(3) 同一个关系中不能有内容完全一样的记录。

(4) 任意两行或任意两列互换位置，不影响数据的实际含义。

1.2.3 关系设计

关系设计就是对描述客观事物的数据进行分析、归纳，设计出符合上述基本特点的二维表。下面通过实例来说明如何设计出适用的关系。

表 1.4 是一个描述学生基本情况和学习情况的关系。

表 1.4 关系设计实例

学 号	姓 名	性 别	出生日期	课程号	课程名	学 分	学时数	成 绩
0205001	道明寺	男	02/25/81	101	高等数学	3	50	85
0205001	道明寺	男	02/25/81	102	VFP	4	55	78
0205003	林杉菜	女	03/23/78	103	会计原理	2	45	81
0205005	美作	男	08/24/79	102	VFP	4	55	89
0205005	美作	男	08/24/79	105	英语	3	48	75

这张二维表虽然符合关系的基本特点,但是仔细观察一下就会发现如下问题:

表中道明寺和美作都各占了 2 行,其中学号、姓名、性别和出生日期 4 个字段的值是重复的,造成数据冗余。这不仅浪费存储空间,而且容易导致数据的不一致性。例如,要将“道明寺”改为“道明诗”,就必须修改两条记录。若一旦疏忽大意,只修改了 1 条记录,就导致了姓名的不一致。当按姓名查询“道明诗”的记录时,就只能查到 1 条记录。此外,要查询一个学生某门课程的成绩,不仅要指定学号,还必须指定课程号,给使用带来不便。

这类问题如何解决呢?要解决问题,首先应该找出造成上述问题的原因。这类问题是由于设计者将学生基本情况、学习情况和课程情况这 3 种不同范畴的数据放在同一个二维表中造成的。如果把 3 种不同范畴的数据分开,分别设计学生关系、课程关系和成绩关系 3 个二维表,如表 1.5、表 1.6 和表 1.7 所示,问题便迎刃而解了。

表 1.5 学生关系

学 号	姓 名	性 别	出生日期
0205001	道明寺	男	02/25/81
0205003	林杉菜	女	03/23/78
0205005	美作	男	08/24/79

表 1.6 课程关系

课程号	课程名	学 分	学时数
101	高等数学	3	50
102	VFP	4	55
103	会计原理	2	45
105	英语	3	48

表 1.7 成绩关系

学 号	姓 名	课程号	成 绩
0205001	道明寺	101	85
0205001	道明寺	102	78
0205003	林杉菜	103	81
0205005	美作	102	89
0205005	美作	105	75

总之,要设计出水平较高的关系,除了使之符合关系的基本特点之外,还应该把不同范畴的数据放在不同的关系中。实践表明,这样做不仅可以减少数据冗余,使访问关系中的记录更加方便,而且还可以提高运行效率。

1.2.4 关系运算

集合论是关系数据库的数学基础。从一个关系中找出用户所需要的数据,就要使用关系数据库管理系统提供的专门的关系运算。关系运算包括选择、投影和连接等。

1. 选择 (Selection)

从一个关系中选出满足给定条件的记录的操作称为选择或筛选。选择是从行的角度进行的运算,选出满足条件的那些记录构成原关系的一个子集。例如,若从学生关系表(表 1.5)中找出所有男同学,则可按如下条件对学生关系进行选择操作:

性别="男"

得到的结果是:

学号	姓名	性别	出生日期
0205001	道明寺	男	02/25/81
0205005	美作	男	08/24/79

2. 投影 (Projection)

从一个关系中选出若干指定字段的值的操作称为投影。投影是从列的角度进行的运算,所得到的字段个数通常比原关系少,或者字段的排列顺序不同。例如,若从学生关系(表 1.5)中选出姓名和出生日期两个字段,则投影得到的结果是:

姓名	出生日期
道明寺	02/25/81
林杉菜	03/23/78
美作	08/24/79

3. 连接 (Join)

连接是把两个关系中的记录按一定条件横向结合,生成一个新的关系。最常用的连接运算是自然连接,它是利用两个关系中共有的字段,将该字段值相等的记录连接起来。

系统在执行连接运算时,要进行大量的比较操作。设关系 1 有 m 条记录,关系 2 有 n 条记录,连接运算先从关系 1 的第 1 条记录开始,依次与关系 2 中的第 1、第 2、第 3、...、第 n 条记录按照连接条件进行比较,凡是符合条件的两条记录就构成一条记录并存入新关系中,这一次扫描要进行 n 次比较。下次扫描是从关系 1 的第 2 条记录开始,依次与关系 2 中的 n 条记录按照连接条件进行比较,依次类推,总共需要进行 m 次扫描,整个连接过程要访问 $m \times n$ 条记录。可见,当 m 和 n 都很大时,连接运算所用的时间就会很长。为了提高连接操作的效率,在连接之前应该先进行连接优化。例如,先进行选择运算,以减少关系中的记录数目。如果能够进行投影运算,则进行投影以减少关系中的字段数目,然后再进行连接运算。

不同关系中的公共字段或者具有相同语义的字段是实现连接运算的“纽带”。例如,在表 1.6 和表 1.7 的两个关系中,公共字段为课程号。在表 1.7 中,如果将字段名“课程号”改为“所选课程号”,则“所选课程号”与表 1.6 中的“课程号”就是具有相同语义的字段。

需要明确的是,选择和投影属于单目运算,它们的操作对象只是一个关系。连接则为双目运算,其操作对象是两个关系。当然,通过两两连接,即可实现多个关系的连接。