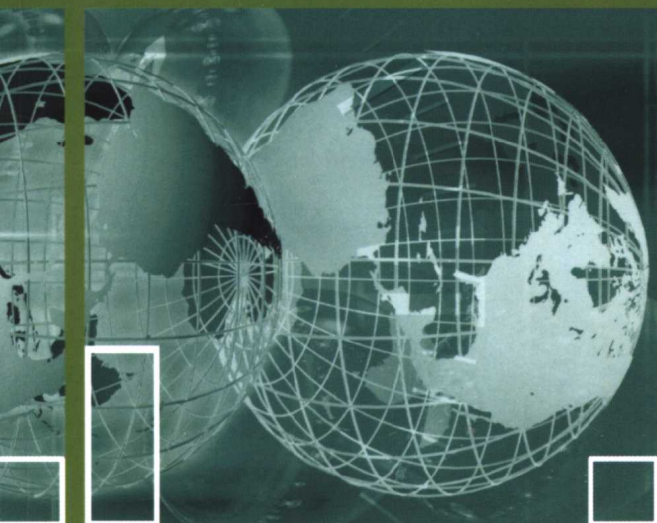




Visual FoxPro

数据库设计教程

杨佩理 陶瑜 编著



机械工业出版社
China Machine Press

计算机应用技术系列教材



Visual FoxPro 数据库设计教程

杨佩理 陶瑜 编著



机械工业出版社
China Machine Press

本书全面介绍了Visual FoxPro (VFP) 数据库编程语言, 内容涵盖从数据库的基本概念和基本操作到应用程序设计的各个方面。本书有序地组织了VFP所包含的主要知识, 讲解了VFP的操作设计环节, 使读者能够快速掌握VFP平台上的数据处理方法。从编程部分开始, 本书紧密结合面向对象的主题, 以培养读者面向对象编程的能力。书中给出了大量经过调试运行的实例, 便于初学者入门。本书结尾还包含一个课程设计, 以提高读者的实际应用能力。本书可作为高等院校相关专业VFP课程的入门教材, 也可用于自学。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Visual FoxPro数据库设计教程 /杨佩理等编著. -北京: 机械工业出版社, 2004.7

(计算机应用技术系列教材)

ISBN 7-111-14390-6

I. V… II. ① 杨… III. 关系数据库-数据库管理系统, Visual FoxPro-教材
IV. TP311.138

中国版本图书馆CIP数据核字 (2004) 第041899号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 朱 劼

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2004年7月第1版第1次印刷

787mm × 1092mm 1/16 · 20印张

印数: 0 001 -4 000册

定价: 29.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换
本社购书热线: (010) 68326294

前 言

Visual FoxPro (VFP) 是微软公司开发的一款适用于微机的数据库管理系统产品。VFP提供了强大的项目及数据库管理功能,有许多实用的设计向导,可快捷地设计查询、视图、表单、报表以及菜单。VFP充分贯彻了“面向对象”的编程思想,围绕着对象展开程序设计,并通过事件的驱动得以执行。VFP中内嵌了调试器以便于程序调试,对设计完成的应用程序可连编成可执行程序,运用VFP提供的“安装向导”可将应用程序制作成安装盘,从而对自己开发的软件进行发布。

目前,介绍VFP数据库编程语言的书籍有很多,但符合“简明扼要、系统化、理论与实践并重”的书籍还是很少。本书正是遵循这样的原则而编写的,相信本书会成为数据库语言的初学者的优秀向导。

本书可以作为VFP数据库编程语言课程一个学期或一个学年的教材,也适合用于自学。本书希望培养读者在以下方面的能力:

- 1) 透彻理解基于Windows操作系统的可视化编程思想。
- 2) 熟练掌握关系数据库的设计和操作方法。
- 3) 掌握开发数据处理类应用程序的全过程,基本掌握应用程序的调试方法。

本书的主要特色:

1) 开门见山。本书舍弃了许多数据库书籍开头惯有的、繁难的数据库术语,只是介绍了必要的有关数据库的基本概念,而后直接进入VFP的大门。

2) 用最直观的表作为VFP语言的切入点。从这里切入,读者能快速地熟悉VFP的操作环境,理解VFP的操作方式。由表到数据库,再引出项目管理器,顺理成章,并且每个环节都不至落入空泛。

3) 贯彻VFP面向对象的编程思想。本书从第6章程序设计的介绍开始,就将读者主要的注意力引入到面向对象编程的思路。尽管不得不预先用到后面章节的内容,但这样的预先应用不会引起读者的困惑。后续的几章不断强化面向对象编程的概念和思想,并通过大量例子使读者清晰掌握VFP面向对象编程的思想。

4) 与实践密切结合。书中包含众多的示例,示例附有设计分析和具体设计的环节,这些示例和附录A中的20个实验,可为读者掌握VFP语言提供很好的实践机会。

本书的第1章~第5章由杨佩理、陶瑜编写;附录A由陶瑜编写;第6章~第12章和附录B、C由杨佩理编写。全书由杨佩理统稿,常晋义、周蓓审阅。

本书在编写过程中得到了江苏省各高校教授VFP课程的老师的大力支持,我们向在本书编写和出版过程中给予帮助的同志表示衷心感谢。特别是无锡南洋学院的许伯康老师,他为本书的编写提出了许多宝贵的意见和建议,在此表示深深的敬意和感谢。

由于作者水平有限,本书还存在诸多不足之处,恳请读者指正。

最后,衷心希望这本书能伴随读者轻松学习,学有收获!

作者Email: yangpl@szit.edu.cn。

作 者

2003年11月

目 录

前言

第1章 数据库系统及VFP概述1

1.1 数据处理技术发展概述1

1.1.1 数据处理1

1.1.2 数据处理技术的发展1

1.2 数据库系统的基本概念2

1.2.1 数据库2

1.2.2 数据库管理系统3

1.2.3 数据库系统4

1.2.4 数据库系统的发展4

1.2.5 关系数据库系统简介5

1.3 数据模型6

1.3.1 三个世界6

1.3.2 数据模型7

1.3.3 关系数据库9

1.4 Visual FoxPro的安装和启动9

1.4.1 安装VFP中文版10

1.4.2 启动Visual FoxPro10

1.4.3 Visual FoxPro系统界面11

1.5 Visual FoxPro的特点15

1.5.1 Visual FoxPro的平台特点15

1.5.2 Visual FoxPro的语言特点15

1.6 本章小结17

思考与练习17

第2章 表的基本操作19

2.1 表的创建19

2.1.1 字段类型19

2.1.2 表结构的创建20

2.1.3 表结构的修改23

2.1.4 表的打开和关闭24

2.1.5 表结构的显示25

2.2 表记录的基本操作26

2.2.1 记录的追加26

2.2.2 记录的显示28

2.2.3 记录的定位30

2.2.4 记录的修改33

2.2.5 记录的删除35

2.2.6 记录的筛选37

2.3 工作区和工作区中的表39

2.3.1 工作区39

2.3.2 “数据工作期”窗口40

2.4 表索引的创建和使用41

2.4.1 索引的作用41

2.4.2 表索引的创建42

2.4.3 索引的修改和删除44

2.4.4 索引的使用45

2.4.5 利用索引快速定位记录46

2.5 本章小结46

思考与练习47

第3章 数据库管理49

3.1 数据库的设计49

3.1.1 VFP数据库的基本概念49

3.1.2 数据库的设计过程51

3.2 数据库的创建与组成52

3.2.1 数据库的创建53

3.2.2 数据库的组成54

3.3 数据库表的扩展设置58

3.3.1 数据库表字段的扩展属性59

3.3.2 数据库表的表属性61

3.4 永久关系与参照完整性的操作64

3.4.1 永久关系的操作64

3.4.2 参照完整性的定义及操作66

3.5 建立表之间的临时关系68

3.5.1 临时关系的认识68

3.5.2 临时关系的创建68

3.6 数据库的维护70

3.6.1 数据库的打开与关闭70

3.6.2 数据库的修改与删除71

3.6.3 数据库信息的显示	71	5.3.4 视图的维护	117
3.7 本章小结	72	5.4 本章小结	118
思考与练习	72	思考与练习	119
第4章 用项目管理器管理项目	73	第6章 可视化程序设计基础	121
4.1 项目管理器	73	6.1 Visual FoxPro编程概述	121
4.1.1 创建一个项目文件	73	6.1.1 VFP的两种数据处理方式	121
4.1.2 打开和关闭项目管理器	75	6.1.2 VFP的两种程序设计模式	121
4.2 项目管理器下的项目	76	6.1.3 VFP程序的三种基本结构	122
4.2.1 数据文件	76	6.2 简单程序设计	123
4.2.2 文档文件	77	6.2.1 一个简单的程序示例	123
4.2.3 类库文件	77	6.2.2 程序文件的基本操作	124
4.2.4 代码文件	78	6.2.3 常用的命令和函数	125
4.2.5 其他文件	78	6.3 分支结构的程序设计	129
4.3 项目管理器的界面和设置	78	6.3.1 单分支选择语句	130
4.3.1 项目管理器的折叠与展开	78	6.3.2 多分支选择语句	132
4.3.2 项目管理器的界面定制	79	6.4 循环结构的程序设计	133
4.3.3 在项目间共享文件	80	6.4.1 SCAN...ENDSCAN命令	134
4.3.4 查看项目信息	80	6.4.2 FOR...ENDFOR命令	135
4.4 “项目管理器”的基本管理	81	6.4.3 DO WHILE...ENDDO命令	136
4.4.1 “项目管理器”对数据库的操作	81	6.4.4 循环控制	137
4.4.2 “项目管理器”对数据库表的操作	83	6.5 过程和自定义函数	138
4.4.3 “项目管理器”对其他文件的操作	84	6.5.1 过程	138
4.5 本章小结	85	6.5.2 自定义函数	139
思考与练习	86	6.5.3 过程文件	140
第5章 数据的查询检索	87	6.5.4 变量的作用域	141
5.1 查询的创建和使用	87	6.6 本章小结	142
5.1.1 初识查询	87	思考与练习	142
5.1.2 利用查询向导创建查询	88	第7章 面向对象编程入门	145
5.1.3 利用查询设计器创建查询的步骤	93	7.1 VFP中的对象	145
5.1.4 查询的运行和修改	102	7.1.1 对象的概念	145
5.2 SELECT-SQL语句	102	7.1.2 对象的属性	146
5.2.1 SQL语句简介	102	7.1.3 对象的事件	147
5.2.2 从查询文件认识SELECT-SQL语句	103	7.1.4 对象的方法	147
5.2.3 SELECT-SQL语句的基本语法	105	7.2 用表单向导设计表单	148
5.2.4 SELECT-SQL语句的标准语法格式	109	7.2.1 表单概述	148
5.3 视图的创建和使用	110	7.2.2 用表单向导创建单表表单	148
5.3.1 视图的认识	110	7.2.3 用表单向导创建一对多表单	151
5.3.2 用视图设计器创建视图	112	7.3 表单设计器	153
5.3.3 用CREATE SQL VIEW命令创建视图	117	7.3.1 表单设计器概述	153

7.3.2 表单的设计思想	154	9.3 子类的创建和使用	199
7.4 本章小结	160	9.3.1 利用程序创建类	199
思考与练习	161	9.3.2 利用类设计器创建子类	202
第8章 常用控件的设计	163	9.3.3 利用表单设计器创建子类	204
8.1 深入了解对象和数据	163	9.3.4 子类的使用	206
8.1.1 数据绑定型控件与非数据 绑定型控件	163	9.3.5 类库的操作	209
8.1.2 ControlSource属性设置的作用	163	9.4 深入了解事件模型	211
8.2 显示信息的对象	164	9.4.1 VFP中的事件	211
8.2.1 标签	164	9.4.2 追踪事件序列	212
8.2.2 文本框	165	9.4.3 事件代码的设计和執行	213
8.2.3 编辑框	166	9.5 本章小结	214
8.2.4 图像	166	思考与练习	214
8.2.5 形状	167	第10章 报表设计	217
8.2.6 微调器	168	10.1 利用报表向导创建报表	218
8.3 提供预定数据的对象	169	10.1.1 报表向导	218
8.3.1 选项组	170	10.1.2 一对多报表向导	223
8.3.2 复选框	172	10.2 利用报表设计器设计报表	225
8.3.3 列表框	173	10.2.1 报表设计器简介	225
8.3.4 组合框	175	10.2.2 设计报表	227
8.4 允许用户执行特定任务的对象	177	10.3 报表输出	239
8.4.1 命令按钮	177	10.3.1 报表页面设置	239
8.4.2 命令按钮组	178	10.3.2 输出报表	239
8.5 表格、页框和时钟	181	10.4 本章小结	241
8.5.1 表格	181	思考与练习	241
8.5.2 页框	183	第11章 设计应用程序界面	243
8.5.3 时钟	186	11.1 菜单设计	243
8.6 本章小结	187	11.1.1 菜单设计思想	243
思考与练习	187	11.1.2 菜单设计	246
第9章 面向对象编程进阶	191	11.1.3 快捷菜单的设计和使用的	251
9.1 类的概念	191	11.1.4 顶层菜单的设计和使用的	253
9.1.1 类的特性	191	11.2 创建自定义工具栏	254
9.1.2 控件类和容器类	192	11.2.1 工具栏的特性	254
9.1.3 VFP的基类	193	11.2.2 定义工具栏类	255
9.1.4 类的层次结构	194	11.2.3 创建自定义工具栏	257
9.1.5 类库	194	11.3 本章小结	258
9.2 对象和对象的操作	195	思考与练习	258
9.2.1 类和对象的关系	195	第12章 应用程序的集成和测试	259
9.2.2 对象的操作	195	12.1 创建应用程序	259
9.2.3 对象的引用	198	12.1.1 应用程序的规划	259

12.1.2 数据库设计	260	12.3.1 应用程序的测试	269
12.1.3 应用程序的主菜单设计	260	12.3.2 应用程序的连编	270
12.1.4 详细设计	261	12.4 应用程序的包装出品	271
12.1.5 创建应用程序的主程序	261	12.5 本章小结	275
12.2 应用程序的调试	262	思考与练习	276
12.2.1 调试的前期准备	262	附录A 上机实验任务和指导	277
12.2.2 调试应用程序	263	附录B 学生成绩管理系统样表	307
12.2.3 处理应用程序的运行错误	268	附录C 课程设计任务和指导	309
12.3 应用程序的测试和连编	269		

第1章 数据库系统及VFP概述

自1946年第一台计算机诞生以来,随着软、硬件的发展,计算机的功能已从最初的纯数值计算延伸至各个领域,数据处理已成为计算机最广泛的应用领域之一。数据库技术组成的数据库系统成为数据处理的一个最佳途径,成为了现代计算机系统的一个重要的、必不可少的组成部分。而数据库管理系统作为对数据资源进行统一管理的软件,也引起了人们广泛的关注,本书所介绍的Visual FoxPro (VFP)就是一种在微机上普遍使用的、面向对象的关系数据库管理系统。

来吧,让我们进入数据库的领域,领略一下数据处理技术的发展过程;然后一步一步地进入到Visual FoxPro的多彩世界中!

1.1 数据处理技术发展概述

1.1.1 数据处理

所谓数据,是现实世界中信息的载体,是客观事物的符号表示。

在计算机世界中,数据是所有能输入到计算机中,且能被计算机识别、存储,并由计算机程序进行处理的符号的总称,即计算机程序加工处理的对象。

数据处理,顾名思义是指对各种类型的数据进行收集、存储、分类、排序、计算或加工、检索、传输等的过程。简单地说,就是由原始数据转换为结果数据的过程。

数据在计算机中的具体处理过程如图1-1所示。

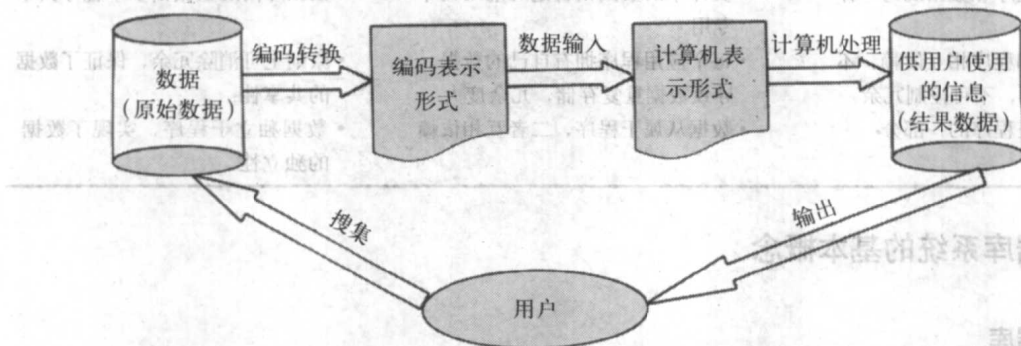


图1-1 数据处理过程

1.1.2 数据处理技术的发展

数据处理的效率与数据存储结构和处理方式有着密切的关系。从数据的存储结构和处理方式角度而言,通常把计算机数据处理技术的发展分为三个阶段:

- 无管理阶段

- 文件管理方式阶段
- 数据库管理阶段

1. 无管理阶段（20世纪50年代中期之前）

20世纪50年代中期之前是计算机数据处理的初级阶段。数据的管理完全由程序员进行考虑和安排，因此该阶段也叫作纯人工管理阶段。在这一阶段，程序和数据结合成一个整体，两者相互依赖，当程序之间出现重复数据时，这些重复数据不能共享，计算机在数据管理中还没有发挥作用。

2. 文件管理方式阶段（20世纪50年代中期~60年代末）

随着计算机硬件性能的改进和软件技术的发展，计算机数据处理进入了文件管理系统阶段。此时，文件管理系统是应用程序和数据文件的接口，应用程序通过该接口完成数据文件的建立、存取、修改和撤销等操作。该阶段在理论上实现了程序和数据的分离，但实际上还是存在着致命的缺点：一个数据文件只能为一个应用程序所拥有。当另一个应用程序也需这个数据文件时，此数据文件不能共享给第二个应用程序。因此，文件管理系统所管理的基本上还是分散的、相互独立的数据文件。此时的数据处理存在数据冗余大、易造成数据的不一致性、程序与数据的相互依赖等诸多问题。

3. 数据库管理阶段（20世纪60年代末以后）

到20世纪60年代末，针对数据的文件管理方式存在的缺点，计算机软件工作者经过长期不懈的努力，提出了数据库的概念，由此进入了数据库管理阶段。数据库技术组成的数据库管理方式提供了一种较完善的高级管理。它实现了数据的独立性、共享性，并且保证了数据的一致性。

从表1-1中，我们可以很清楚地看到三个阶段数据性能的对照。

表1-1 三个阶段的数据性能对照表

无管理阶段	文件管理方式阶段	数据库管理阶段
• 应用程序和数据混为一体 • 数据和程序相互依赖，不可分割，不可控制冗余 • 数据是程序的一部分	• 文件中的数据由特定的应用程序专用 • 每个应用程序拥有自己的数据，导致数据重复存储，冗余度大 • 数据从属于程序，二者互相依赖	• 数据库内的数据由多个程序共享 • 原则上可消除冗余，保证了数据的共享性 • 数据独立于程序，实现了数据的独立性

1.2 数据库系统的基本概念

1.2.1 数据库

1. 数据库概念

什么是数据库？“数据库”（DataBase，简称DB）一词最早在20世纪60年代出现于美国，此后慢慢流行起来。我们现在所说的数据库，是指以一定的组织形式存放在计算机存储介质上的相互关联的数据的集合。

例如，把一个班的学生的基本信息（学号、姓名、性别、出生日期、联系电话等）、所在专业开设课程的基本信息（课程代号、课程名、是否必修课、学分等）有序地组织起来，以

一定的形式（比如表格）存放在计算机存储介质上，我们就可以把这些关联的信息看作一个数据库，它可以为多个用户所共享。

在VFP中，数据库是一些表的集合以及这些表之间存在的关系，而表就如我们平时经常使用的表格。如学生表、成绩表等，见表1-2、表1-3。

表1-2 学生表

学号	姓名	性别	出生日期	联系电话	班级代号
02011101	李林	男	02/12/83	0512-58221363	020111
02011102	陆海涛	男	04/15/82	0510-63529242	020111
...	...				

表1-3 课程表

课程代号	课程名	理论学时	实验学时	必修课	学分	课程备注
01	中文Windows95	30	40	T	4	此课程为基础课
02	Visual FoxPro	50	40	T	6	此课程需要大量上机操作
...	...					

2. 数据库的主要特点

数据库主要有如下特点：

- **数据共享性** 数据资源为多个应用程序服务，由此实现数据的共享。
- **数据独立性** 数据和应用程序之间相互分离，相互独立，不存在相互依赖关系。
- **可控冗余度** 数据共享的实现，使得重复数据大大减少，但有时为提高查询效率，也可保留少量重复数据。
- **安全性保护** 可以对数据库系统实施不同的安全检查，从而保证数据的安全。
- **完整性保证** 对网络环境下多用户系统数据的完整性检查。
- **标准化实施** 由于数据库的集中控制，保证遵循统一的标准，有助于数据交换或系统间的交流。

1.2.2 数据库管理系统

1. 数据库管理系统的定义

假定我们现在想要在计算机中建立由表1-2、表1-3组成的一个数据库，并将它存放在计算机的存储设备上（如软盘、硬盘等）。如果要在纸上绘制这两张表，那么显然是轻而易举的，但现在要把它们作为数据库的一部分绘制在计算机中，则显然比较困难了。

那我们究竟怎样建立这样的—个数据库呢？答案是：需要借助一种软件来完成上述工作。这种软件就是所谓的“数据库管理系统”（DataBase Management System，简称DBMS）。而本书所介绍的Visual FoxPro就是一种在个人计算机上非常流行的数据库管理系统。

数据库管理系统是对数据库进行各种管理的一种系统软件，是用户与数据库之间的接口。

2. 数据库管理系统的作用

数据库管理系统的功能当然并非仅限于建立几张表，它的作用类似于仓库系统中的管理机构，要负责处理用户（应用程序）存取数据的各种请求。借助DBMS，用户可以完成各种各样的数据管理工作。

一般地说, DBMS应该具有以下功能:

- **数据定义功能** DBMS向用户提供“数据定义语言”(DDL), 用于描述数据库结构。
- **数据操纵功能** DBMS向用户提供“数据操纵语言”(DML), 可对数据进行查询、更新(包括增加、删除、修改)等操作。
- **控制和管理功能** DBMS可在多用户使用的情况下对数据进行“并发控制”, 对用户权限实施“安全性检查”, 并具有数据备份、恢复等功能。

目前在PC机上使用的DBMS都是关系数据库管理系统, 简称RDBMS, 我们学习的VFP就是一种面向对象的关系数据库管理系统。借助VFP这样一种功能完善的可视化数据处理平台, 我们可以很容易地开发出面向不同任务的、实用的数据库应用软件。

1.2.3 数据库系统

数据库系统(DataBase System, 简称DBS)是实现有组织地、动态地存储大量关联数据、方便用户访问的计算机软硬件资源所组成的具有管理数据库功能的计算机系统。它由计算机软件(数据库、数据库管理系统、操作系统)和计算机硬件资源等组成。

数据库、数据库管理系统和数据库系统三者之间既有联系又有区别, 它们之间的关系如图1-2所示。从图中我们可以清楚地看出:

- 用户通过DBMS建立和使用数据库。
- 数据库中包含了DBMS所面向的数据。
- 数据库管理系统是实现数据处理的手段和工具。
- 数据库和数据库管理系统都是数据库系统的组成部分。

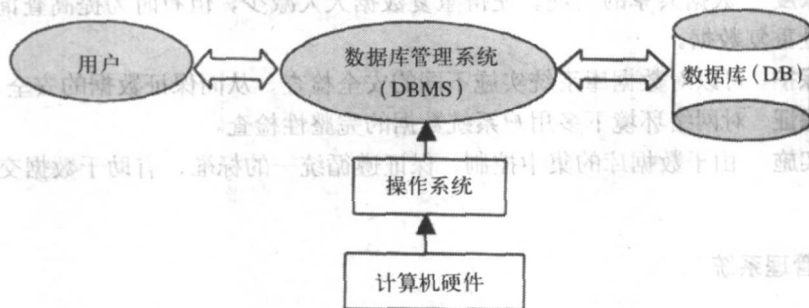


图1-2 用户、DBMS、DB、DBS的关系

1.2.4 数据库系统的发展

经过近30年的发展, 数据库系统已经走过了第一代——非关系数据库系统, 完善了第二代——关系数据库系统, 现在正向第三代——对象-关系数据库系统迈进。

1. 非关系数据库系统

非关系数据库系统是第一代数据库系统的总称, 其中包括层次数据库系统与网状数据库系统两种类型。

非关系数据库的基本模型如图1-3所示。

第一代数据库系统采用“记录”作为基本数据结构, 一次查询只能访问数据库中的一条记录。其中层次数据库系统采用“层次模型”, 其总体结构为“树形”, 在不同的记录之间只

允许单线联系；网状数据库系统采用“网状模型”，总体结构呈网形，在两个记录之间可以存在两种或多于两种的联系。

2. 关系数据库系统

关系数据库系统 (Relational DataBase System, 简称RDBS) 是第二代数据库系统, 是对第一代数据库系统的改进。目前在PC机上使用的数据库系统大部分都是第二代数据库系统。

与第一代数据库系统相比, RDBS具有下列优点:

1) 用人们习惯用的表格作为基本数据结构, 通过公共关键字来实现不同二维表之间的数据联系。以上面所示的表1-2、表1-3为例, 我们可以再建一张成绩表, 成绩表中存放一些成绩信息 (学号、课程代号、成绩、学期等), 其中成绩表中的学号和表1-2 (学生表) 中的学号为公共关键字, 成绩表中的课程代号和表1-3 (课程表) 中的课程代号为公共关键字, 通过这两对公共关键字, 我们就可以把这三张表相互联系起来。

2) 一次查询仅用一条命令或语句, 就可访问整个二维表, 因而查询效率较高, 通过多表联合操作, 还能对有联系的若干二维表实现“关联”查询。

3. 对象-关系数据库系统

随着多媒体应用的广泛利用, 可处理的数据类型更加丰富, 处理的形式也更加灵活。这其实对数据库技术提出了更高的要求, 数据库系统不仅要能存储和处理表格等相对静态的数据, 还能存储图形、声音等复杂的对象, 并能实现复杂对象的复杂行为。将数据库与面向对象技术有效结合, 就顺理成章地成为研究数据库技术的新课题和新方向, 构成第三代数据库系统的基础。

虽然目前对象-关系数据库系统 (Object-Relational DataBase System, 简称ORDBS) 还处在发展的过程中, 在技术上和应用上还有许多工作要做, 但它已经显示出光明的发展前景。它除了支持现有的基本类型之外, 还能支持文本、图像、声音等新的数据类型, 支持类、继承、函数/方法等丰富的对象机制, 并能提供高度集成的、可支持客户机/服务器应用的用户接口, 从而使在多媒体中的应用更加完善。

1.2.5 关系数据库系统简介

关系数据库系统是数据库技术发展史上重要的里程碑。

早在20世纪60年代后期, 数据处理就进入了数据库管理阶段。1970年6月, IBM公司的E.F.Codd发表论文《A Relational Model of Data for Large Shared Databanks》, 由此奠定了关系数据库的理论。20世纪70年代后期, 关系数据库理论的研究基本上进入了成熟阶段。到20世纪80年代初, 微型计算机的推广, 使得各种关系数据库产品迅速进入计算机应用市场。20世纪90年代, 各公司研制的各种关系数据库产品百花齐放, 百家争鸣, 关系数据库系统产品的更新换代进入了一个繁荣的阶段。

从20世纪70年代末的大众数据库dBASE II, 到今天可视化的数据库语言VFP, 历经了时代的荡涤演变过程。表1-4从一个侧面简要地反映了这一演变过程。

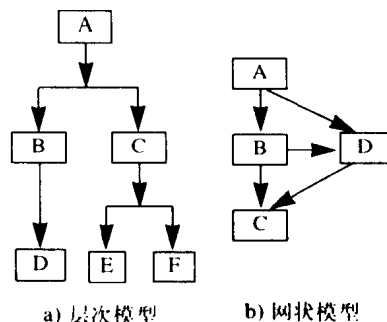


图1-3 非关系数据库模型

表1-4 微机数据库系统产品一览表

年 代	数据库管理系统产品	公 司
20世纪70年代	dBASE II、III、IV	美国Ashto-Tate公司
20世纪80年代	FoxBase	美国Fox Software公司
20世纪90年代初	FoxPro 2.0、2.5	美国Microsoft公司
20世纪90年代末	Visual FoxPro 3.0、5.0、6.0	美国Microsoft公司

1.3 数据模型

数据库中的数据是有组织、有秩序地存放的，而数据模型就用来描述数据的组织方式。因此，数据库的核心是数据模型，而关系模型是使用最广泛的一种数据模型。

现实世界的数据是多种多样的，怎样搜集需要的信息，然后进一步转换为计算机及数据库所允许的形式，并反映到数据库中去呢？现实世界到计算机世界的演变如何实现？对三个世界的认识是解决这些问题的基础。

1.3.1 三个世界

我们已经知道，现实世界中的数据与计算机世界中的数据是两个不同的概念，那么我们究竟怎样才能把现实世界中各种复杂的、相互关联的事物（各种数据）以计算机及数据库允许的形式反映到数据库中去呢？我们可借助如下“三个世界”，沿着逻辑思维的常规顺序，对事物逐层递进地进行理解：

- 现实世界
- 信息世界
- 计算机世界

1. 现实世界

现实世界（realistic world）是独立于人们意识之外的客观存在的事物及其相互联系。其基本的要素是事物，它们彼此联系、彼此区别，具有独特的特征，这些事物与特征就是人们所要加工处理的对象，即原始世界的原始产物。

2. 信息世界

信息世界（information world）即观念世界，是现实世界中的事物在人们头脑中的反映，它是对客观事物的抽象，它介于现实世界与计算机世界中间。信息世界的基本单位是实体，现实世界中事物的特征在信息世界中抽象为实体的属性。

3. 计算机世界

计算机世界（computer world）即数据世界，它以数据形式反映信息世界中的信息，是对信息世界中的实体和属性的数据表示，是现实世界中的客观事物在计算机中的最终表示形式。在计算机世界中，描述对象的数据称为记录（record），描述属性的数据称为字段（field），描述实体模型的数据称为数据模型，同一类记录的集合称为文件。

综上所述，对于人们而言，有三个认识领域：现实世界、信息世界和计算机世界。它们之间的转化关系如图1-4所示。

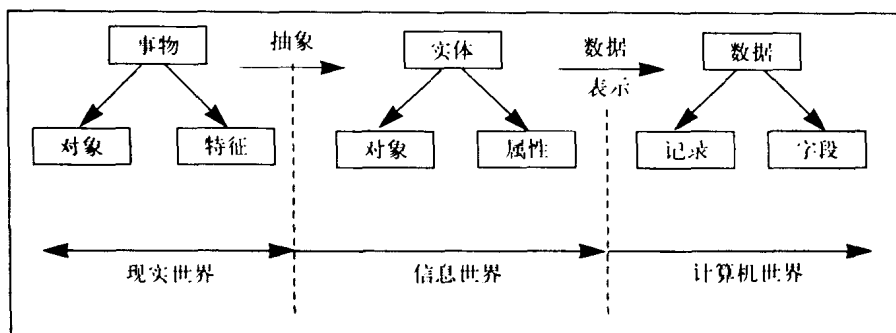


图1-4 信息的三个领域之间的关系

1.3.2 数据模型

通过三个世界的层层演进，我们可将客观世界中的事物通过转化，最后以计算机允许的形式反映到计算机中。但这一过程究竟如何实现？首先需要为研究对象确定模型。

所谓模型（model），是对研究对象抽象化、形式化的一种描述。一般情况下，我们需要建立两种模型，即信息模型和数据模型。

信息模型是在信息世界中为研究的对象建立的一个较为抽象的（脱离任何计算机）的模型。数据模型是在计算机世界的数据库层所建立的能被计算机接受的实际模型。下面我们来具体分析数据模型。

1. 数据模型

数据模型作为数据库的核心部分，是计算机世界中的一种软件模型，它是数据库管理系统研究的一个最主要的问题。数据模型的种类很多，但常见的数据模型有三种：

- 层次模型（hierarchical model）
- 网状模型（network model）
- 关系模型（relational model）

（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，基本结构是树形结构，自顶向下，层次分明。其典型代表是IBM公司的IMS（Information Management System）。

图1-5所示就是典型的“学校”层次模型。

从图1-5中，我们可以看出层次模型的主要特点是：

- 1) 有且仅有一个被称作“根结点”的结点，它没有双亲结点。
- 2) 根结点以外的结点有且仅有一个双亲结点。
- 3) 根结点处于第一层，其直接子结点处于第二层，依次类推。各结点间不存在相互关系，各层相互独立。

（2）网状模型

网状模型实际是一个不加任何条件的图，基本结构是图形结构，关系错综复杂。其典型代表是20世纪70年代CODASYL数据库任务组提出的DBTG系统，也称CODASYL系统。

图1-6所示就是“系”网状模型。

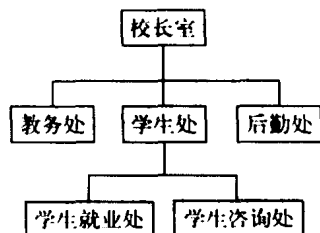


图1-5 “学校”层次模型结构图

从图1-6中,我们可以看出网状模型的主要特点是:

- 1) 允许结点有一个或多个的父结点。
- 2) 可以有一个以上的结点没有父结点。
- 3) 没有层的概念、结点间彼此相互渗透,相互包含。

(3) 关系模型

关系模型是目前最流行的一种数据模型,它用二维表来表示实体和实体之间的联系。其基本结构就是二维表,如表1-2所示的学生表、表1-3所示的课程表。由于二维表在数学上称为关系,所以基于二维表的数据模型称为关系模型,一张二维表对应一个关系。

关系模型的特点是:

- 1) 描述一致性,不仅用关系描述实体本身,而且也用关系描述实体间的联系。
- 2) 具有更好的数据独立性。
- 3) 关系必须是规范化的关系,这使关系模型结构简单、操作简便。
- 4) 关系模型建立在数学概念的基础上,有较强的理论基础。

与层次模型与网状模型相比,关系模型概念简单、清晰,操作直观、易学易用。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,Visual FoxPro由于采用关系模型,因此称为关系数据库管理系统。

2. 一些基本概念

(1) 实体

实体(entity)是客观存在并可以相互区别的事物,如学生、课程、成绩等。

(2) 属性

属性(attribute)是实体所具有的某一特性,包括属性的型和属性的值,一个实体可以用若干个属性来描述。学生表中的学号、姓名、性别就是学生实体的属性。

(3) 字段

字段(field)也称为数据项,是数据结构中的最小单位,它是不可分割的。学生名李林、刘枫都是字段。每个字段的名称叫字段名。

(4) 记录

记录(record)也称为元组,指关系中的一行数据,用它来描述实体。它由若干数据项组成。学生表中的一行:02011101 李林 男 1981/02/12 0512-58221363 020111就是一条记录。

(5) 关系模式

关系模式是对关系结构的描述,描述关系由哪些属性组成、这些属性来自哪些域、属性到域的映像关系以及关系的完整性约束条件。其格式为:关系名(属性名1,属性名2,……,属性名n)。以表1-1的学生表所示的关系为例,其关系模式可表示为:

学生(学号,姓名,性别,出生日期,联系电话,班级代号)

一个关系模式对应一个关系文件的结构。

(6) 关键字

关键字(key)是关系中唯一标识记录的一个或一组属性,这样在不同记录中,由于相应的属性值不同,可以把不同的记录区分开来。

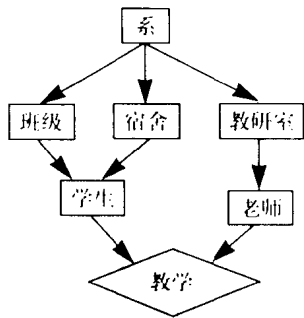


图1-6 “系”网状模型结构图

1) 候选关键字

候选关键字 (candidate key) 是具有关键字特性的一个或多个属性的统称。候选关键字既能唯一地确定记录, 它包含的字段又是最精练的。在一张二维表中, 必存在候选关键字。在表1-1的学生表中, 我们可以用“学号”作为候选关键字, 也可用“姓名”、“联系电话”联合组成候选关键字。

2) 主关键字

主关键字 (primary key) 是从多个候选关键字中选出的一个关键字, 用来唯一标识实体。在任何一个二维表中, 主关键字必须包含一个不同于其他记录的唯一的值, 且包含的字段个数是最少的。在表1-1的学生表中, “学号”值肯定是唯一的, 因此可以用“学号”作为学生表的主关键字。

3) 外部关键字

外部关键字 (foreign key) 是关系的一个属性, 它不是这个关系的关键字, 但却是另外一个关系的关键字, 此时称这个属性为关系的外部关键字。例如, 若存在学生表与成绩表, 此两张表中都存在“学号”字段, 且以“学号”作为学生表的主关键字, 那么该“学号”就是成绩表的外部关键字。

1.3.3 关系数据库

VFP中的表数据均以关系模型的方式组织, 放置这样的表的数据库就称为关系数据库。在关系数据库中, 最基本的元素是表。一个表就相当于现实生活中的一张二维表格。二维表格中的一行, 相当于表中的一条记录; 二维表格的一列, 相当于表中的一个字段。表格中的每一列都有名称, 这些名称即为表的字段名。

表及其相关概念在三个世界中的对应关系如表1-5所示。

表1-5 从三个世界解析关系模型

现实世界	信息世界	计算机世界
二维表	关系	表文件 (.dbf)
行	实体	记录
列	属性	字段

根据二维表的特点, 关系数据库中的表具有如下特性:

- 表的列确定了表的结构, 字段的个数、名称、类型、宽度等要素决定了二维表的结构。结构相对固定, 记录可以增删。
- 表的每一列均有唯一的字段名, 彼此无重复。
- 表中不允许有完全相同的记录。
- 表中行的顺序与列的顺序可以任意交换。

1.4 Visual FoxPro的安装和启动

Visual FoxPro (简称VFP) 是Microsoft公司第一个带有Windows 95标志的应用软件。它起源于xBASE编程语言系列。VFP是目前微机上使用的最优秀的数据库管理系统软件, 它采用了可视化的、面向对象的程序设计方法。利用增加的向导功能, 在不需掌握VFP的各种命令