

高职高专计算机教材

Visual FoxPro 6.0

应用基础教程

主编 常广炎 杨淑清

辽宁民族出版社

高职高专计算机教材

Visual FoxPro 6.0 应用基础教程

主 编 常广炎 杨淑清
副主编 池 云 杨 彬 姚 晔

辽宁民族出版社

© 常广炎 杨淑清 2006

图书在版编目 (C I P) 数据

Visual FoxPro 6.0 应用基础教程/常广炎, 杨淑清主
编. — 沈阳: 辽宁民族出版社, 2006.2

ISBN 7 - 80722 - 156 - 9

I. V… II. ①常… ②杨… III. 关系数据库—数
据库管理系统, Visual FoxPro 6.0—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2006) 第 005237 号

出版发行者: 辽宁民族出版社

(地址: 沈阳市和平区十一纬路 25 号 邮编: 110003)

印刷者: 沈阳市东实印刷厂

幅面尺寸: 184mm × 260mm

印 张: 23 $\frac{1}{4}$

字 数: 500 千字

印 数: 1—2000

出版时间: 2006 年 2 月第 1 版

印刷时间: 2006 年 2 月第 1 次印刷

责任编辑: 文忠实

封面设计: 戴 钊

责任校对: 崔维诚

定 价: 39.00 元

联系电话: 024—23284348

邮购热线: 024—23284335

E-mail: lnmz@mail.lnpgc.com.cn

前 言

Microsoft Visual FoxPro 6.0 关系数据库系统是新一代小型数据库管理系统的杰出代表,它以强大的功能、完整而又丰富的工具、极高的处理速度、友好的界面以及完备的兼容性等特点,深受广大用户的欢迎。

Visual FoxPro 6.0 提供了一个集成化的系统开发环境,它不仅支持结构化编程技术,还支持面向对象的可视化编程技术,并拥有功能强大的可视化程序设计工具。目前 *Visual FoxPro 6.0* 是用户收集信息、查询数据、创建集成数据库系统、进行实用系统开发的较为理想的工具软件。

本书主要介绍如何使用 *Visual FoxPro 6.0* 命令和面向对象程序设计的方法,介绍了关系数据库管理系统的基础理论及系统开发技术。内容包括:*Visual FoxPro 6.0* 系统简介、*Visual FoxPro 6.0* 语言基础、表的设计与操作、结构化程序设计、数据库设计、表单与控件、查询和视图、报表设计与应用、菜单设计、应用系统的开发与发布过程等。书中配有丰富的习题(包括上机练习),更适合教学的需要。

本书是高等职业学校、高等专科学校、成人高等学校 *Visual FoxPro* 课程的教学参考书,供高职、高专各专业学生使用;也可用作全国计算机水平考试中 *Visual FoxPro* 部分及各类短训班的培训教学参考书和广大工程技术人员进行 *Visual FoxPro 6.0* 程序设计及数据库项目开发的参考资料;也可单独作为 *Visual FoxPro 6.0* 爱好者入门和提高的参考书籍。

本书由常广炎、杨淑清主编,由池云、杨彬和姚晔担任副主编,常广炎编写第二章、第三章、第四章、第七章和第十一章,杨淑清编写第一章、第五章和第六章,池云编写第十章和第十二章,姚晔编写第八章和第九章,杨彬编写第十三章和第十四章。

书中存在的错误和不足之处,欢迎广大读者批评指正。

编 者

二〇〇六年一月八日

目 录

第一章 Visual FoxPro 6.0 概述	1
1.1 数据库的基本概念	1
1.2 数据模型	3
1.3 关系数据库	6
1.4 Visual FoxPro 的发展过程	8
1.5 VFP 的功能	10
1.6 VFP 系统的环境与安装	12
1.7 VFP 系统的启动	13
1.8 VFP 环境介绍	14
测试题	16
第二章 Visual FoxPro 基础	18
2.1 VFP 中文版的性能指标	18
2.2 VFP 文件组成	18
2.3 项目管理器	20
2.4 设计器与生成器	27
2.5 建立工作目录与搜索路径	28
习题	30
测试题	30
第三章 Visual FoxPro 数据库及其操作	32
3.1 数据库的概念	32
3.2 数据库的建立	35
3.3 建立自由表	38
3.4 维护数据库表中的记录	42
3.5 筛选表记录	46
3.6 建立索引	48
习题	52
测试题	53
第四章 数据库和表	55
4.1 数据库的设计	55
4.2 数据库中表的操作	61
4.3 字段与记录属性设置	64
4.4 创建和编辑关系	74
4.5 建立参照完整性规则	79

习题	81
测试题	81
第五章 查询与视图	83
5.1 查询与视图	83
5.2 结构化查询语言 SQL 简介	84
5.3 查询数据	86
5.4 视图查询	96
习题	113
测试题	113
第六章 VFP 程序设计	114
6.1 数据及其运算	114
6.2 数据库的操作命令	124
6.3 文件操作	131
6.4 表的基本操作	134
6.5 表的排序与索引	136
6.6 统计命令	142
6.7 使用多个表	145
6.8 命令文件的建立与运行	152
6.9 顺序结构程序设计	154
6.10 分支结构程序设计	155
6.11 循环结构程序设计	158
6.12 过程及其调用	164
习题	173
测试题	175
第七章 报表和标签设计	178
7.1 报表的布局	178
7.2 报表文件的建立	179
7.3 修改报表布局	186
7.4 标签文件的建立	193
7.5 预览和打印报表与标签	195
测试题	196
第八章 面向对象程序设计	197
8.1 基本概念	197
8.2 基本特性	198
8.3 设计方法	199
8.4 可视化编程	201
测试题	208
第九章 表单与控件	210
9.1 表单设计方式	210
9.2 表单管理	216

9.3 常用表单控件	219
9.4 多表表单设计	238
9.5 表单布局设计	240
测试题	241
第十章 菜单	244
10.1 下拉式菜单设计	244
10.2 弹出式菜单设计	252
第十一章 结构化查询语言 SQL	256
11.1 SQL 的数据定义功能	256
11.2 SQL 的数据查询功能	258
11.3 SQL 的数据更新功能	274
测试题	276
第十二章 数据库应用系统设计实例	279
12.1 应用系统开发的基本过程	279
12.2 应用程序的基本结构	281
12.3 应用程序开发实例	283
12.4 应用程序的发布	292
第十三章 常用函数	298
13.1 数值函数	298
13.2 字符函数	300
13.3 日期和时间函数	303
13.4 数据类型转换函数	304
13.5 测试函数	306
测试题	309
基础练习	312
综合练习	330
附录 A 基础练习题参考答案	356
附录 B 综合练习笔试模拟题参考答案	359
附录 C 2005 年全国计算机等级考试二级 VFP 考试大纲	361
参考文献	364

第一章 Visual FoxPro 6.0 概述

Visual FoxPro 6.0(中文版)是 Microsoft 公司 1998 年发布的可视化编程语言集成 Visual Studio 6.0 系列开发产品之一。Visual FoxPro 6.0 是可运行于 Windows 平台的 32 位数据库开发系统,能充分发挥 32 位处理器的强大功能,是一种用于数据库结构设计和应用程序开发的功能强大的面向对象的微机数据库软件。它采用了可视化的、面向对象的程序设计方法,大大简化了应用系统的开发过程,提高了系统的模块性和紧凑性。

Visual FoxPro 数据库是一个关系型数据库。由于 Visual FoxPro 只需很少编程就可以建立数据库应用程序,因而成为一种通用的数据库软件。利用 Visual FoxPro 6.0 可以设计出丰富多彩的用户界面,在用户界面中可以放置各种控制部件,如命令按钮、图形、图片、图表等,从而设计出完全图形化的界面,方便用户的操作和使用。

1.1 数据库的基本概念

数据库是存储在计算机存储设备上,结构化的相关数据集合。它不仅包括描述事物的数据本身,而且还包括相关事物之间的联系。

1.1.1 数据与数据处理

数据是将现实世界中的各种信息记录下来的、可以识别的符号,是信息的载体,是信息的具体表示形式。数据的概念有两个方面的涵义:描述事物特性的数据内容以及存储在媒体上的数据形式。数据形式可以是多样的,例如:某人的出生日期是“1989 年 11 月 4 日”,也可以表示为“89/11/04”,其含义并没有改变。

信息是有用的数据,数据是信息的表现形式。信息是通过数据符号来传播的,数据如不具有知识性和有用性,则不能称其为信息。

数据的概念在数据处理领域中已经大大地拓宽了,数据不仅包括各种文字或字符组成的文本形式的数据,而且包括图形、图像、动画、影像、声音等多媒体数据。

数据处理是指将数据转换成信息的过程,通过数据处理可以获得信息。如通过公司的进货量和销售量,就可以知道库存量,从而为进货提供依据。

1.1.2 数据库管理技术的发展阶段

计算机管理数据技术随着计算机的发展而不断发展,利用计算机对数据进行处理经历了 4 个阶段。

1. 人工管理阶段

计算机诞生之初,外存储器只有纸带、磁带、卡片等,没有像磁盘这样的速度快、存储容量大、可随机访问、直接存储的外存储器。软件方面,没有专门管理数据的软件,数据包含在计算或处理它的程序之中。这一阶段的数据管理任务,包括存储结构、存取方法、输入输出方式等完全由程序员通过编程实现。这一阶段的数据管理称为人工管理阶段。

这个阶段的特点是:数据量大,客观上没有长期存储大量数据的硬件设备;数据与其应用程序一一对应,也就是说一组数据只面向一个应用,不能被共享;没有形成文件的概念。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代后期,计算机开始大量地用于各种管理中的数据处理工作。大量数据的存储、检索和维护成为紧迫的需求。此时,在硬件方面,可直接存取的磁盘成为外存储器的主流;软件方面,出现了高级语言和操作系统。

这一阶段的数据处理采取程序与数据分离的方式,有了程序文件与数据文件的区别。数据文件可以长期保存在外存储器上被多次存取,在操作系统的文件系统的支持下,程序使用文件名访问数据文件,程序员只需关注数据处理的算法,而不必关心数据在存储器上如何存取。这一阶段的数据管理称为文件(系统)管理阶段。

文件系统中的数据文件是为了满足特定的需要而专门设计的,为某一特定的程序而使用,数据与程序相互依赖。同一数据可能出现在多个文件中,这不仅浪费空间,而且由于不能统一更新,容易造成数据的不一致性和数据冗余。

3. 数据库系统阶段

随着社会信息量的迅猛增长,计算机处理的数据量也相应增大,文件系统存在的问题阻碍了数据处理技术的发展,于是数据库管理系统便应运而生。

数据库技术的主要目的是有效地管理和存取数据资源,提高数据的共享性、一致性和完整性,减少数据的冗余度,使数据与应用程序独立,从而减少应用程序的开发和维护费用。

数据库管理系统从 20 世纪 60 年代末问世以来,一直是计算机管理数据的主要方式。

4. 分布式数据库系统阶段

20 世纪 70 年代以前,数据库多数是集中式的,网络技术的发展为数据库提供了良好的运行环境,使数据库从集中式发展到分布式,从主机/终端系统结构发展到客户/服务器系统结构。

1.1.3 数据库系统

1. 基本概念

(1) 数据库

数据库(DB)是存储在计算机存储器中,结构化的相关数据的集合。它不仅存放数据,而且还存放数据之间的联系。

数据库中的数据面向多种应用,可以被多个应用程序共享。其数据结构独立于使用数据的程序,对于数据的增加、删除、修改和检索由系统软件进行统一的控制。

(2) 数据库管理系统

数据库管理系统(DBMS)是指帮助用户建立、使用和管理数据库的软件系统,主要包括三部分:数据定义语言(DDL)、数据操作语言(DML)以及其他管理和控制程序。

(3) 数据库应用系统

数据库应用系统(DBAS)是利用数据库管理系统资源开发的面向某一类实际应用的应用软件系统。一个 DBAS 通常由数据库和应用程序两部分构成,它们都需要在数据库管理系统 DBMS 支持下开发和工作。

(4) 数据库系统

数据库系统(DBS)是指引进数据库技术后的计算机系统,实现有组织地、动态地存储大量相关数据,提供数据处理和信息资源共享的便利手段。数据库系统由五部分组成:硬件系统、

数据库集合、数据库管理系统及相关软件、数据库管理员和用户。

2. 数据库系统的特点

与文件系统相比,数据库系统具有以下特点:

(1) 数据共享减少冗余

在数据库系统中,对数据的定义和描述已经从应用程序中分离出来,通过数据管理系统来统一管理。数据的最小访问单位是字段,既可以按字段的名称存取库中某一个或某一组字段,也可以存取一条记录或一组记录。

(2) 采用特定的数据模型

数据库中的数据是有结构的,这种结构由数据库管理系统所支持的数据模型表现出来。数据库系统不仅可以表示事物内部各数据之间的联系,而且可以表示事物与事物之间的联系,从而反映出现实世界事物之间的联系。因此,任何数据库管理系统都支持一种抽象的数据模型,关于数据模型将在 1.2 中具体介绍。

(3) 具有较高的数据独立性

在数据库系统中,数据库管理系统提供映像功能,实现了应用程序对数据的总体逻辑结构、物理存储结构之间较高的独立性。用户只以简单的逻辑结构来操作数据,无需考虑数据在存储器上的物理位置与结构。

(4) 有统一的数据控制功能

数据库可以被多个用户或应用程序共享,数据的存取往往是并发的,即多个用户同时使用同一个数据库。数据库管理系统必须提供必要的保护措施,包括并发访问控制功能、数据的安全性控制功能和数据的完整性控制功能。

1.2 数据模型

在现实世界中,事物之间是存在联系的,这种联系是客观存在的,是由事件本身的性质决定的。例如,学校教学系统中的教师、学生、课程、成绩等都是相互关联的。通常把表示客观事物及其联系的数据及结构称为数据模型。

1.2.1 信息世界中的基本概念

1. 实体(Entity)

客观存在并且可以相互区别的事物称为实体。实体可以是实际的事物,如教师、职工、部门、单位等;也可以是抽象的事件,如订货、选修课程等。

2. 实体集(Entity Set)

实体集是具有相同类型及相同性质(或属性)的实体集合,例如,某个学校的所有学生的集合可以被定义为实体集 Students。

3. 属性(Attribute)

实体所具有的某一特性为属性。一个实体可以由若干个属性来刻画。例如学生可以用学号、姓名、性别、出生日期等属性描述。但对每个属性来说,各实体有自己的属性,即属性被用来描述不同实体间的区别。

4. 联系(Relationship)

实体之间的对应关系称为联系,它反映了现实事物之间的相互联系,例如,一位学生可以选择多门课程;一个部门中可以有多个职工。

1.2.2 实体之间的联系

在现实世界中,事务内部以及事务之间是有联系的,例如,学生和课程之间有选课的联系,读者和图书之间有借阅的联系。联系(也称关系)可以归纳为:一对一的联系、一对多的联系和多对多的联系三类。

1. 一对一的联系(1:1)

若对于实体集 A 中的每一个实体,在实体集 B 中都有唯一的一个实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一的联系。例如,一个生产车间有一个车间主任,反过来,每个车间主任又对应一个生产车间,则生产车间和车间主任之间具有一对一的联系。

2. 一对多的联系(1:n)

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n>0)$ 个实体与之联系,反之,对于实体集 B 中的每个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体集 B 具有一对多的联系。例如,一个部门有若干个职工,而每个职工只在一个部门工作,则部门与职工之间是一对多的联系。

3. 多对多的联系(m:n)

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n>0)$ 个实体与之联系,反之,对于实体集 B 中的每个实体,实体集 A 中也有 $m(m>0)$ 个实体与之联系,则称实体集 A 与实体集 B 具有多对多的联系。例如,学生和选修课程的联系,某个学生可以选修多门课程,某选修课程也可以被多名学生选修。

1.2.3 实体及实体关系的表示

信息世界所对应的模型为概念模型,概念模型能够方便、准确地表示出上述信息世界中的常用概念。概念模型的表示方法很多,其中最常用的是 P. P. S. Chen 于 1976 年提出的实体_联系方法(Entity_Relationship Approach)。该方法用 E_R 图来描述现实世界的概念模型。

E_R 图提供了表示实体、属性和联系的方法。

实体型:用矩形表示,矩形框内写明实体名。

属性:用椭圆的长方形表示,实体框和属性框之间用直线连接。

联系:用菱形框表示,框内填上联系名。

图 1-1 用 E_R 图描述了实体及其属性联系图。

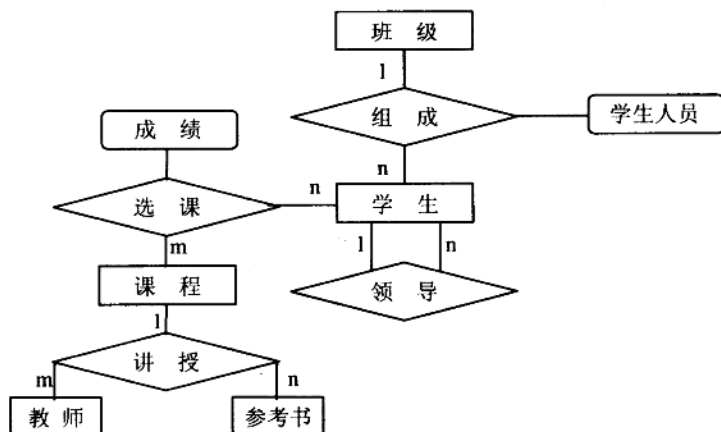


图 1-1 实体及其属性联系图

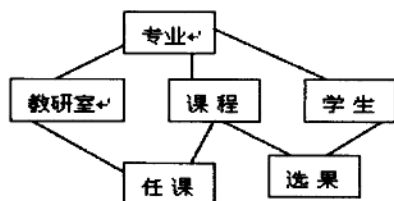
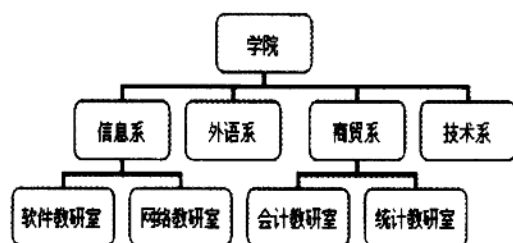
1.2.4 数据模型简介

数据库中的数据从整体来看是有结构的,即所谓数据的结构化。各实体以及实体间存在联系的集合称为数据模型,数据模型的重要任务之一就是指出实体间的联系。按照实体集间的不同联系方式,数据库分为三种数据模型,即层次模型、网状模型和关系模型。

1. 层次模型

层次模型的结构是树型结构,树的节(结)点是实体,树的枝是联系,从上到下为一对多的联系。每个实体由“根”开始,沿着不同的分支放在不同的层次上。如果不再向下分支,则此分支中最后的节点称为“叶”。图1-2为某学院的机构设置,“根”节点是学院,“叶”节点是各教研室。

支持层次模型的数据库管理系统称为层次数据库管理系统,其中的数据库称为层次数据库。



2. 网状模型

用网形结构表示实体及其之间的联系的模型称为网状模型。在网状模型中,每一个节点代表一个实体,并且允许节点有多于一个“父”节点。这样网状模型代表了多对多的联系类型,如图1-3所示。

支持网状模型的数据库系统称为网状数据库管理系统,其中的数据库称为网状数据库。

3. 关系模型

关系模型是以数学理论为基础构造的数据模型,它用二维表格来表示实体集中实体之间的联系。在关系模型中,操作的对象和结果都是二维表(即关系),表格与表格之间通过相同的栏目建立联系,如图1-4所示。

编号	姓 名	性别	出生日期	婚否	职务	工资	工作时间	籍贯
1	王云鹤	女	01/05/69	.T.	科长	828.00	12/11/91	辽宁
2	张晓云	女	12/25/74	.F.	秘书	718.00	01/02/93	长春
3	刘玉龙	男	04/17/55	.T.	处长	935.00	12/11/75	长春
4	刘宗武	男	10/18/75	.F.	科员	717.00	01/01/96	长春
5	孙大鹏	男	06/15/75	.F.	科员	618.00	11/11/97	河北
6	李 莉	女	12/12/70	.T.	副科长	724.00	11/12/97	河南
7	赵永年	男	12/15/63	.T.	科员	680.00	06/06/85	广东
8	王丽娟	女	10/10/72	.T.	科员	678.00	09/07/94	广东
9	李云峰	男	01/11/70	.T.	科长	796.00	11/02/92	广西

图 1-4 关系模型

关系模型有很强的数据表达能力和坚实的数学理论,结构单一,数据操作方便,最易被用户接受,以关系模型建立的关系数据库是目前应用最广泛的数据库。由于关系数据库具有许多优秀功能,层次数据库和网状数据库均已失去其重要性。

1.3 关系数据库

自 20 世纪 80 年代以来,新推出的数据库管理系统几乎都是基于关系模型。Visual FoxPro 就是一种关系数据库管理系统。

1.3.1 基本概念

1. 关系与表

关系的逻辑结构就是一张二维表,如学籍表、课程表等。在 Visual FoxPro 中,一个关系就是一个“表”,每个表对应一个磁盘文件,表文件的扩展名为 .DBF。

2. 属性与字段

一个关系有很多属性(即实体的属性),对应二维表中的列(垂直方向)。每一个属性有一个名字,称为属性名。对于一张二维表格来说,属性就是表格中的栏(列),同栏的数据应具有相同的性质,如“姓名”这一栏就只能填入姓名数据,而不能是其他数据。

在 Visual FoxPro 中,属性表示为表中的“字段”,属性名即为字段名。

3. 关系模式与表结构

对关系的描述称为关系模型,一个关系模式对应一个关系的结构。其格式为:

关系名(属性名 1,属性名 2,⋯,属性名 n)

在 Visual FoxPro 中对应的表结构为:

表名(字段名 1,字段名 2,⋯,字段名 n)

4. 元组与记录

在一个表格(一个关系)中,行(水平方向)称为“元组”。在 Visual FoxPro 中,元组表示为表中的“记录”。

一个表中可以有多个记录,也可以没有记录,没有记录的表称为“空表”。

5. 域

域是属性取值的范围,不同的属性有不同的取值范围,取不同的域。如成绩的取值范围是 0~100,逻辑型属性的取值只能是 .T.(真)或 .F.(假)。

6. 码与关键字

用来区分不同元组(实体)的属性或属性组合,称为码。在 Visual FoxPro 中对应的概念是关键字,关键字是字段或字段的组合,用于在表中唯一标识记录。如学生成绩表中的学号字段可以是关键字,因为学号不可能重复,可以用来唯一标识一个记录,性别字段就不可以是关键字,因为表中性别可能会在不同记录中出现,即有两个或两个以上的记录该属性是相同的。

码的任意真子集都不能成为码,这样的“最小码”称为“候选码”。候选码可能有多个,被选中用来区别不同元组的候选码称为主码。在 Visual FoxPro 中,对应的概念是:候选关键字和主关键字。

如果表中的某个字段不是本表的关键字,而是另外一个表中的关键字,则称该字段为外部关键字。

7. 关系模型与数据库

从集合论的观点来看,一个关系模型就是若干个有联系的关系模式的集合,一个关系模式是命名的属性集合,另外,关系是元组的集合,元组是属性值的集合。

在 Visual FoxPro 中,把相互之间存在联系的表放到一个数据库中统一管理。例如,在教工管理数据库中包括职工档案表和职工工资表。数据库文件的扩展名为 .DBC。

1.3.2 数据完整性

数据完整性是指数据库中数据的正确性和一致性(或相容性),数据完整性用来防止数据库中不存在不合法的数据,防止错误的数据进入数据库中。

数据完整性可以分为实体完整性、域完整性和参照完整性。

1. 实体完整性

实体完整性是指数据库表的每一行都有一个唯一的标志。实体完整性由实体完整性规则来定义,完整性规则是指表中的每一行在组成码(关键字)的列上不能有空值或重复值,否则就不能起到唯一标识行的作用。

2. 域完整性

域完整性是指数据库中数据取值的正确性。它包括数据类型、精度、取值范围以及是否允许空值等。取值范围又可分为静态和动态两种:静态取值范围是指列数据的取值范围是固定的,如年龄小于 150;动态取值范围是指列数据的取值范围由另一个列或多列的值决定或更新列的新值依赖于它的旧值。

3. 参照完整性

参照完整性是指数据库中表与表之间存在码(关键字)与外码(外部关键字)的约束关系,利用这些约束关系可以维护数据的一致性或相容性,即在数据库的多个表之间存在某种参照关系。要实现这种参照关系,首先创建表的码与外码:

(1)当对含有外码的表进行插入、更新操作时,必须检查新行中外码的值是否在主表中存在,若不存在就不能执行该操作。

(2)当对主表中的行进行删除、更新操作时,必须检查被删除行或被更新行中主码的值是否在被一个或多个外码参照引用,若正被参照引用就不能执行该操作。

1.3.3 对关系数据库的要求

通常生活中的二维表格有多种多样,不是所有二维表格都能被当作“关系”而存放到数据库中。也就是说,在关系模型中对“关系”有一定规范化要求。

(1)关系中的每个属性(列)必须是不可分割的数据单元。

(2)同一关系中不应有完全相同的属性名,即在同一个表格中不能出现相同的列(字段)。

(3)关系中不应用完全相同的元组,即在同一个表格中不能出现相同的行(记录)。

(4)元组(记录)和属性名(字段)与次序无关,即交换两行或两列的位置不影响数据的实际含义。

1.3.4 关系运算

关系运算对应于 Visual FoxPro 中对表的操作,在对关系数据库进行查询时,为了找到用户感兴趣的数据,需要对关系进行一定的运算。这些运算以一个或两个关系作为输入,运算的结果将产生一个新的关系。关系运算有选择、投影、连接三种运算。

1. 选择运算

选择运算是指从关系中找出满足给定条件的元组,又称为筛选运算。选择的条件以逻辑表达式给出,使得逻辑表达式的值为真的元组被选取。选择是从行的角度进行的运算,即选择

部分行,经过选择运算可以得到一个新的关系,其关系模式不变,但其中的元组是原关系的一个子集。

在 Visual FoxPro 中,选择操作使用命令短语 FOR|WHILE <条件> 或设置记录过滤器来实现。

2. 投影运算

从关系模式中指定若干个属性来组成新的关系称为投影。投影是从列的角度进行的运算,经过投影可以得到一个新关系,其关系模式所包含的属性个数往往比原关系少,或者属性的排列顺序不同。投影运算提供了垂直调整关系的手段,体现出关系中列的次序无关的特性。

在 Visual FoxPro 中,投影操作使用命令短语 FIELDS <字段 1>,<字段 2>,...,或设置字段过滤器来实现。

选择和投影经常联合使用,从数据库文件中提取某些记录和某些数据项。

3. 连接运算

从两个关系中选取满足连接条件的元组组成新关系,称为连接(或链接)。连接是关系的横向结合,连接运算将两个关系模式的属性名拼接成一个更宽的关系模式,生成的新关系中包含满足条件的元组。连接过程是通过连接条件来控制的。连接条件中将出现两个关系中的公共属性名,或者具有相同语义、可比的属性。

选择和投影运算都是单目运算,它们的操作对象只是一个关系,相当于对于一个二维表进行切割。连接运算是二目运算,需要两个关系作为操作对象,如果需要连接两个以上的关系,应当两两进行连接。

在 Visual FoxPro 中,连接操作相当于对两个二维表进行拼接。有两种意义下的连接操作,用 JOIN 命令实现两个表的连接将得到一个新的表;关联操作命令 SET RELATION 属于逻辑上的连接操作。

1.3.5 关系运算的优化

1. 自然连接

自然连接是指去掉重复属性的等值连接,它是按照属性对应相等为条件进行的连接操作。自然连接是最常用的连接运算。

系统在执行连接运算时,要进行大量的比较操作,因此执行比较费时。尤其在包括许多元组的关系之间进行连接时,更加突出。

2. 优化方法

优化的一般方法是:

(1)进行选择运算,尽量减少关系中元组的个数,缩小参与连接运算关系的数量,减少访问记录的次数;

(2)能投影的投影,使关系中的属性个数减少。在投影时必须注意保留连接两个关系所需要的公共属性或具有相同语义的属性,否则关系之间就失去了联系;

(3)再进行连接操作。

利用关系的投影、选择和连接运算可以方便地分解或构造新的关系。

1.4 Visual FoxPro 的发展过程

自从 Visual FoxPro 推出以来,不仅使得 Xbase 数据库管理系统搭上了“可视化”的快车,而

且与其他编程语言(如 Visual Basic、Visual C++ 等)并驾齐驱。事实上,Visual FoxPro 已成为微型计算机当今最流行的数据库管理系统之一。它的发展主要经历了三个阶段。

1. dBASE 阶段

美国 Ashton-Tate 公司在 1981 年推出 dBASE II,从此,Xbase 就是建立在原始 dBASE II 语言语法和文件格式基础之上的关系数据库产品。1984 年该公司又推出了 dBASE III,随后又推出它的改进型 dBASE III Plus,这些产品功能一代比一代强。由于使用方便,性能优越,被广泛用于 PC 机进行事务管理和数据处理,赢得了“大众数据库”的美称。

2. FoxBASE 和 FoxPro 阶段

1984 年美国 Fox Software 公司,推出关系数据库 Fox 产品最早的第一个产品是 FOXBASE,在运行速度上大大超过 dBASE III,并且第一次引入了编译器。1987 年相继推出 FoxBASE2.0 和它的最高版本 FoxBASE2.1。

1989 年 Fox Software 公司,推出了 FoxBASE 的升级换代产品 FoxPro 1.0,它正是今天 Visual FoxPro 6.0 的前身。它不仅首次引入了基于 DOS 操作系统的窗口技术,使用户面对的不再是单一的圆点提示符,而且极大地扩充了 Xbase 语言命令,支持鼠标,操作方便。同时,它还是一个全兼容 dBASE 和 FoxBASE 的伪编译型集成环境的数据库开发系统,继而又在 1991 年 1 月推出了 FoxPro 2.0,在性能上有重大的提高。

1993 年 1 月公布了 FoxPro 的两种版本:FoxPro 2.5 for DOS 和 FoxPro 2.5 for Windows。

同年晚些时候再次推出了 FoxPro 2.5b,从此 FoxPro 2.5 就在世界各国微机用户中广泛流行。

1994 年发表的 FoxPro 2.6 比 FoxPro 2.5 增加了多种“向导”工具,从而简化了最终用户的操作,但在程序开发方面未见有明显的改进。

3. Visual FoxPro 阶段

1995 年微软公司推出了 Visual FoxPro 3.0 版,Visual FoxPro 3.0 是一个可运行于 Windows 3.x、Windows 95 和 Windows NT 环境的数据库开发系统。它第一次把 Xbase 产品数据库的概念与关系数据库理论接轨。

1997 年 5 月,微软公司推出了 Visual FoxPro 5.0 中文版。

1998 年 9 月,微软公司推出了 Visual FoxPro 6.0 中文版。这是一个可运行于 Windows 98/XP、Windows 2000、Windows NT 等平台的 32 位数据库开发系统。

2000 年后,微软公司相继推出了 Visual FoxPro 7.0、Visual FoxPro 8.0 和 Visual FoxPro 9.0,现最高版本是 Visual FoxPro 9.0。

Visual FoxPro 9.0 的主要特点:

数据处理和协同能力:开发人员可以利用不同级别的 XML 和 XML 网站服务来创建兼容 .NET 的解决方案。通过改进的 SQL 和最新支持的数据类型与 SQL 服务器交换数据。

可扩展的强大开发工具:该工具提供一系列的功能来帮助开发人员改进用户界面,利用字体、颜色、定制的编辑器和其他功能来个性化用户的 Window 性能。

灵活地建立各种类型的数据库解决方案:开发者可以建立和配置基于 Windows 桌面电脑的单机或远程应用。创建和访问 .NET 技术支持的 COM 构成与 XML 网站服务。

增强的报表功能:新的输出架构提供对数据输出报告和格式的精确控制,同时还提供尽可能详细的细节报告、文本内容和报告相关连接。数据报告支持 XML、HTML、图片格式和定制化的多页打印预览 window 版本格式。Visual FoxPro 9.0 也支持早期旧版创建的报告格式。

由此可见, Visual FoxPro 是 FoxBASE + 之后又一广泛使用的微机关系数据库管理系统。

本书将向读者介绍使用广泛的 Visual FoxPro 6.0 中文版,为了叙述方便,本书在后面的章节中,将 Visual FoxPro 6.0 简称 VFP6.0 或 VFP。

1.5 VFP 的功能

1.5.1 VFP 的基本功能

作为一种数据库软件,VFP 具有下列基本功能:

(1) 可以为每一种类型的信息创建一个表,利用表存储相应的信息。

(2) 可以定义各个表之间的关系,从而很容易地将各个表中相关的数据有机地联系在一起。

(3) 可以创建查询搜索那些满足指定条件的记录,也可以根据需要对这些记录排序和分组,并根据查询结果创建报表、表及图表。

(4) 使用视图,可以从一个或多个相关联的表中,按一定条件抽取一系列数据,并可以通过视图更新这些表中的数据;还可以使用视图从网上取得数据,从而收集或修改远程数据。

(5) 可以创建表单来直接查看和管理表中的数据。

(6) 可以创建一个报表来分析数据或将数据以特定的方式打印出来。例如,可以打印一份将数据分组并计算数据总和的报表,也可以打印一份带有各种数据格式的邮件标签。

1.5.2 VFP 的特点

与其他数据库不同,VFP 在实现上述功能时提供了各种向导,用户在操作时,只需要按照向导所提供的步骤执行,使用起来非常方便。因此,VFP 数据库深受广大用户的青睐。

1. 易于使用

对于熟悉 Xbase 命令语言的用户,可以在 VFP 系统命令窗口中使用命令和函数,也可以使用系统菜单选项直接操作和管理数据。这比开发应用程序具有更大的灵活性和更高的数据处理效率。对于具备数据库应用开发能力的用户,可以用 VFP 开发出可单独运行的应用系统并可使用系统所提供的功能制作发布应用程序的软磁盘。

对于没有数据库使用经验的用户,可以在中文 Windows 环境中,运行 VFP 支持的或可脱离 VFP 支持单独运行的数据库应用系统。这是一种适合办公管理人员操作管理数据的方式。

VFP 即作为一个关系型数据库系统,不仅可以简化数据管理,使得应用程序的开发流程更为合理,而且它还在前期版本的基础上实现了计算机易于使用的构想。所以,许多早期使用 FoxPro 版本的用户在从事数据库开发时都可以转向使用 Visual FoxPro 6.0。对于刚刚进入数据库领域的新用户来说,使用 VFP 建立数据库应用程序要比使用其他软件容易得多。

2. 可视化开发

在过去,程序员的大部分时间都用在编写代码上。VFP 不是这样,因为 VFP 具有可视化开发环境,所以开发人员在描绘用户界面和设置控制属性上所花的时间与在编码上所花的时间差不多。不仅对于用户界面的开发是这样,而且对于数据库的设计、报表的布局 and 开发过程中的其他方面也是这样。

可视化环境使用方便,可以使开发人员直接看到工作是如何进行的。开发时间被缩短,调试次数也减少,而且维护也更容易。

3. 事件驱动