

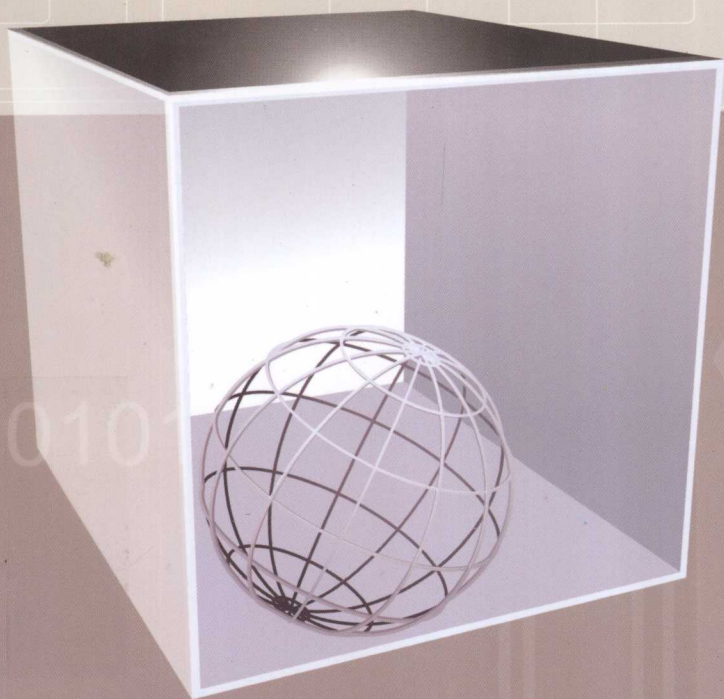
Visual FoxPro

数据库应用与实训

主编 李希敏

参编 崔尚勇 张沛强 杨军莉

白祎花 袁小玲 李 瑞



01 110 101 011 010110 101 01011101 010 001 011010



天津大学出版社
TIANJIN UNIVERSITY PRESS

1 110 10 011 010110 01 010 001 011010

Visual FoxPro

数据库应用与实训

主编 李希敏

参编 崔尚勇 张沛强 杨军莉 白祎花

袁小玲 李 瑞



天津大学出版社

TIANJIN UNIVERSITY PRESS

图书在版编目(CIP)数据

Visual FoxPro 数据库应用与实训/李希敏主编. —
天津:天津大学出版社,2016.3
ISBN 978-7-5618-5544-7

I. ①V… II. ①李… III. ①关系数据库系统—程序
设计 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2016)第 047147 号

出版发行	天津大学出版社
地 址	天津市卫津路 92 号天津大学内(邮编:300072)
电 话	发行部:022-27403647
网 址	publish.tju.edu.cn
印 刷	天津泰宇印务有限公司
经 销	全国各地新华书店
开 本	185mm×260mm
印 张	15.25
字 数	387 千
版 次	2016 年 3 月第 1 版
印 次	2016 年 3 月第 1 次
定 价	32.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换。

版权所有 侵权必究

前 言

根据“Visual FoxPro 数据库应用基础”课程教学的需要,我们组织相关教师编写了本教材,而突出实践性是本教材的主要特点。

本书共分 8 章,构成了 Visual FoxPro 6.0 的主要知识体系,其中主要包括 Visual FoxPro 数据库入门、数据库和表、查询和视图、结构化程序设计、表单设计、报表设计、菜单设计、应用程序的连编和生成等内容。全书以培养学生的应用能力为主要目标,以开发“学生成绩管理系统”为项目案例,将其融入各章实训中,当完成本书中所有实训后,学生不仅掌握了 Visual FoxPro 6.0 的基本内容,还具备了应用 Visual FoxPro 6.0 设计和开发完整数据库应用系统的基本能力。

本书内容全面、实例完整、步骤详细。在内容的选取上,力求由浅入深、循序渐进、通俗易懂。对每个实训,首先给出明确的实训目标,然后简要阐述所需要的基本理论和知识点,再通过讲解多个相关实例的基本操作,将理论与实践融会贯通;大部分实训还安排了若干个课外练习,用以巩固所学知识及操作技能。

本书力求使学生熟悉 Visual FoxPro 6.0 的基本知识和操作技能以及使用它设计和开发数据库应用系统的过程,同时涵盖全国计算机等级考试二级 Visual FoxPro 的全部内容。作为我院“Visual FoxPro 数据库应用基础”精品课程的配套教材,不但为会计电算化专业后续专业课程“AIS 的分析与设计”奠定了基础,还满足了所有专业学生的二级 Visual FoxPro 取证要求,从而使专业课程教学实施“理实一体”“课证融合”的改革思想落到实处。

本书由陕西财经职业技术学院李希敏副教授担任主编,崔尚勇、张沛强、杨军莉、白祎花、袁小玲和李瑞等老师参与了编写工作。具体分工如下:李希敏负责全书的总体结构和总纂,并编写了实训 1.3、实训 2.1、实训 3.3、实训 3.4、实训 5.1、实训 6.1、实训 6.2、实训 7.1、实训 7.2 和附录;崔尚勇编写了实训 5.5、实训 5.6 和实训 8.1;张沛强编写了实训 1.1 和实训 1.2;杨军莉编写了实训 3.1 和实训 3.2;白祎花编写了实训 5.2、实训 5.3 和实训 5.4;袁小玲编写了实训 4.1 和实训 4.2;李瑞编写了实训 2.2 和实训 2.3。曹耀辉副教授审阅了全书,并对部分内容做了修订。在编写过程中,我们还参阅了许多专家和学者的相关教材、论著和资料,在此一并深表谢意。

尽管我们对教材中的每个实训都是以认真谨慎的态度完成的,但难免会有疏漏之处,敬请各位读者批评指正。

编者

2015 年 5 月 5 日

目 录

第1章 Visual FoxPro 数据库入门	(1)
实训 1.1 初识 Visual FoxPro 6.0	(1)
实训 1.2 Visual FoxPro 中的数据形式	(9)
实训 1.3 学生成绩管理系统设计	(22)
第2章 数据库和表	(32)
实训 2.1 自由表的创建及其操作	(32)
实训 2.2 数据库基本操作	(43)
实训 2.3 多工作区及表间关联	(52)
第3章 查询和视图	(58)
实训 3.1 SQL 查询	(58)
实训 3.2 SQL 的数据操纵和数据定义功能	(71)
实训 3.3 查询设计器及使用	(79)
实训 3.4 视图的创建及使用	(90)
第4章 结构化程序设计	(97)
实训 4.1 程序设计基础知识	(97)
实训 4.2 多模块程序	(109)
第5章 表单设计	(116)
实训 5.1 类和对象的创建及操作	(116)
实训 5.2 表单向导和表单设计器	(124)
实训 5.3 常用表单控件的使用(一)	(136)
实训 5.4 常用表单控件的使用(二)	(147)
实训 5.5 常用表单控件的使用(三)	(158)
实训 5.6 表单控件的综合应用	(170)
第6章 报表设计	(181)
实训 6.1 报表向导和报表设计器	(181)
实训 6.2 分组报表和多栏报表设计	(193)
第7章 菜单设计	(202)
实训 7.1 下拉式菜单设计	(202)
实训 7.2 顶层表单和快捷菜单设计	(210)
第8章 应用程序的连编和生成	(218)
附录	(224)
参考文献	(236)

第1章 Visual FoxPro 数据库入门

Visual FoxPro 6.0(以下简称 Visual FoxPro)是优秀的桌面数据库管理系统之一,它以开发成本低、简单易学、方便用户等优点迅速发展起来。要想用它开发出适用的数据库应用系统,首先要掌握数据库系统的基础知识,熟悉 Visual FoxPro 的特点、数据形式和使用方法。

实训 1.1 初识 Visual FoxPro 6.0

一、实训目标

通过本次实训,使学生理解数据库的基本概念,掌握 Visual FoxPro 6.0 的打开/关闭的方法;熟悉 Visual FoxPro 6.0 的主界面使用及其系统环境配置;掌握项目的创建及项目管理器的使用。

二、知识要点

1. 数据库的基本概念

(1)数据。数据是描述客观世界中事物的物理符号,其表现形式多种多样,有数字、字母、文字和其他特殊字符组成的文本形式,还有图形、图像、动画、影像和声音等多媒体形式。

(2)数据库(DataBase, DB)。数据库是以一定的组织方式将相关数据组织起来存储在外存储器上的,能被多个应用程序所共享的,与应用程序彼此独立的数据的集合。简单地说,数据库就是有条理、有组织、合理存储大量数据的“仓库”。

数据库具有“一少三性”的特点。其中,“一少”是指冗余数据少,即基本上没有或很少有重复的数据和无用的数据,也没有相互矛盾的数据,从而节约了大量的存储空间。“三性”是指以下几个特性:①数据的共享性,即数据库中的数据能为多个用户服务;②数据的独立性,即全部数据以一定的数据结构单独、永久存储,与应用程序无关;③数据的安全性,即实施数据保护,防止不合法使用引起的数据泄密和破坏,使每个用户只能按规定对数据进行访问和处理。

(3)数据模型。为了反映现实世界中事物本身及事物之间的各种联系,数据库中的数据必须要有一定的结构,这种结构就用数据模型来表示。数据模型应满足3个方面的要求:一是能真实地模拟现实世界,二是容易被人所理解,三是便于在计算机上实现。常用的数据模型有3种:层次模型、网状模型和关系模型。前两者使用链接指针来存储和体现联系,而关系模型是用二维表结构来表示实体及实体之间的联系,这种二维表就是关系。

根据数据模型的不同,数据库可分为层次数据库、网状数据库和关系数据库。其中,关系数据库的应用最为广泛。关系数据库以关系模型为基础,用关系来描述现实世界,关系不仅可以用来描述实体及其属性,也可以用来描述实体间的联系。

(4)数据库管理系统(DataBase Management System,DBMS)。数据库管理系统是为数据库的建立、使用和维护而配置的系统软件。它提供了处理数据的手段,同时也提供了组织数据的方法。按照其管理的数据库的不同,数据库管理系统可分为层次型、网状型和关系型。

(5)数据库系统(DataBase System,DBS)。数据库系统指引进数据库技术的计算机系统,包括硬件系统、数据库、数据库管理系统及相关软件、各种人员(包括开发人员、数据库管理员、最终用户),其中数据库管理系统是核心。

2. 关系的特点

关系的特点主要有以下几个。

(1)关系中的每一个数据项是最基本的单位,不可再分。

(2)关系中的每一行称为一个元组,每一列称为一个属性。每一列数据项的数据类型相同,列数根据需要而设。

(3)关系中行和列的顺序可任意调整。

(4)关系中不允许有完全相同的元组,属性名不能重复。

3. 关系数据库中的关系运算

关系的基本运算有两类,一类是传统的集合运算(交、并、差),另一类是专门的关系运算(选择、投影、连接)。有些运算需要几个基本运算的组合来完成。关系运算的结果还是关系,包括以下几方面的内容。

(1)笛卡尔集。把两个关系的所有属性合并到一起,形成一个新的关系,且新关系中的所有元组是由两个关系的所有元组进行完全连接构成的。

(2)选择。指在关系中提取满足给定条件的元组,或者说选择满足条件的行。经过选择运算得到的仍然是一个关系,但关系模式不变。

(3)投影。指在关系中指定若干个属性组成新的关系。经过投影运算得到一个新关系,其关系模式中的属性往往比原关系模式中的属性少。

(4)连接。连接是关系的横向结合,它将两个关系模式拼接成一个更宽的关系模式,生成的新关系中包含满足连接条件的元组。

4. 数据管理技术的产生和发展

数据管理是数据处理的中心问题,指对数据进行采集、整理、分类、组织、编码、存储、检索和维护的过程。数据管理技术的发展主要经历了人工管理阶段、文件系统阶段和数据库系统阶段3个阶段。

(1)人工管理阶段(20世纪50年代)。在人工管理阶段,计算机的软、硬件均不完善,存储设备只有磁带、卡片和纸带,软件方面没有操作系统;程序员在程序中不仅要规定数据的逻辑结构,还要设计其物理结构。当数据的物理组织或存储设备改变时,用户程序必须重新编制;数据面向应用,不同程序间不能共享数据,数据冗余度大,数据的一致性很难保

证。应用程序与数据之间的关系如图 1-1-1 所示。

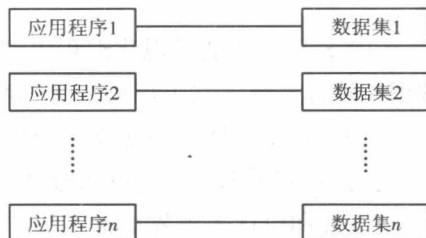


图 1-1-1 人工管理阶段应用程序与数据之间的关系

(2) 文件系统阶段(20 世纪 60 年代)。在文件系统阶段,大容量存储设备和操作系统的出现使数据管理进入了一个新的阶段。数据以文件的形式存在,程序和数据分开存储,使数据与程序有了一定的独立性;各个应用程序可以共享一组数据,实现了以文件为单位的数据共享;数据的组织仍面向程序,故存在大量的数据冗余;文件之间互相独立,不能反映现实世界中事物之间的联系。此阶段应用程序与数据之间的关系如图 1-1-2 所示。

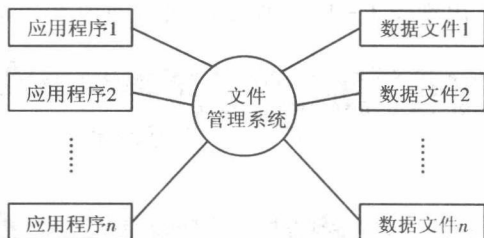


图 1-1-2 文件系统阶段应用程序与数据之间的关系

(3) 数据库系统阶段(20 世纪 60 年代后期)。在数据库系统阶段,以数据为中心组织数据,实现更高的数据共享,减少了冗余数据;程序和数据具有较高的独立性,当数据的逻辑结构改变时,不涉及数据的物理结构,也不影响应用程序,从而降低了应用程序的研制与维护费用。此阶段应用程序与数据之间的关系如图 1-1-3 所示。

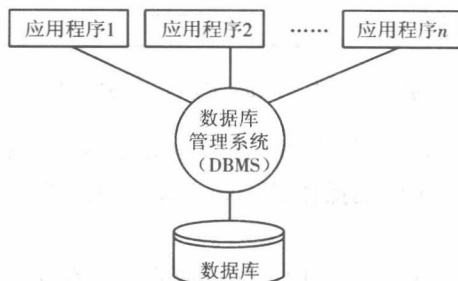


图 1-1-3 数据库系统阶段应用程序和数据之间的关系

网络技术的发展及新型数据库系统的出现带来了数据库技术发展的新高潮。基于关系模型的关系数据库系统功能的扩展与改善,分布式数据库系统、面向对象数据库系统和

数据仓库等数据库技术的出现,构成了新一代数据库系统的发展主流。

5. Visual FoxPro 6.0 概述

Visual FoxPro 6.0 是微软于 1998 年发布的可视化编程语言集成包 Visual Studio 6.0 中的一员,是一个关系型桌面数据库管理系统。它采用可视化、面向对象的程序设计方法,简化了应用系统的开发过程,使数据的组织、数据库的建立及应用系统的开发更为方便快捷;内置 200 余种函数;支持网络;使用“项目管理器”创建和管理应用程序;提供向导、生成器和设计器等工具加快了构建程序框架和设计表单界面的进程等,这些特点使其受到众多用户的青睐。

6. Visual FoxPro 6.0 的启动和退出

执行“开始”菜单中的“Microsoft Visual FoxPro 6.0”命令,或者双击桌面上的 Visual FoxPro 的快捷方式图标,均可启动 Visual FoxPro。

执行“文件”菜单中的“关闭”命令,或单击 Visual FoxPro 窗口上的“关闭”按钮,或在命令窗口执行 QUIT 命令,均可退出 Visual FoxPro。

7. Visual FoxPro 6.0 的主界面

启动 Visual FoxPro 后,出现图 1-1-4 所示的窗口即为 Visual FoxPro 的主界面。

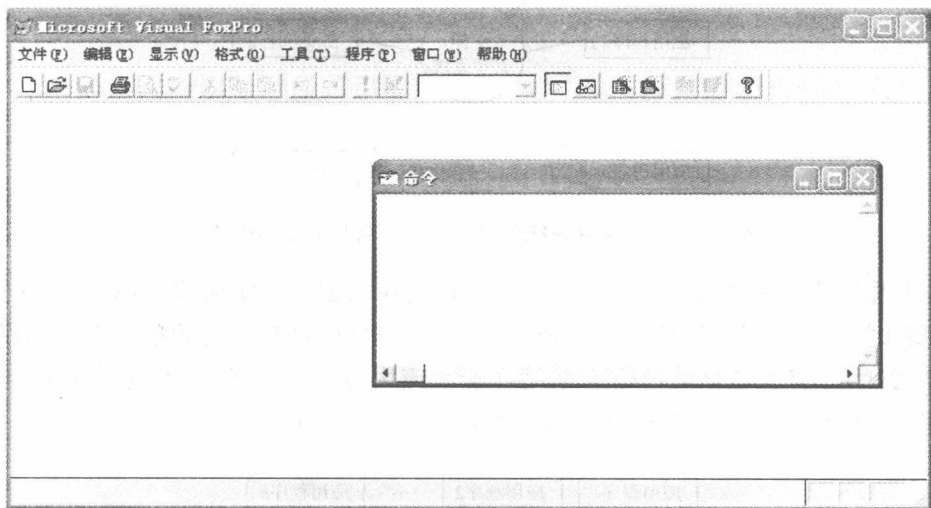


图 1-1-4 Visual FoxPro 的主界面

该主界面由标题栏、菜单栏、工具栏、命令窗口、显示区域和状态栏组成。Visual FoxPro 具有跟踪用户操作的能力,对不同的操作,Visual FoxPro 主菜单栏和各菜单项的子菜单不尽相同,这种情况也称为上下文敏感。

Visual FoxPro 的命令窗口和工具栏均可显示或隐藏。在命令窗口执行过的命令会在其中自动保留,可重复执行以前执行过的命令,还可对命令进行修改、删除和复制等操作。

8. Visual FoxPro 的项目及其项目管理器

在 Visual FoxPro 中,项目指文件、数据、文档和 Visual FoxPro 对象的集合。一个项目对应一个项目文件(.pjx)和一个项目管理器。项目管理器是 Visual FoxPro 处理数据和对象的

主要组织工具,通过它可以简便地创建、组织和处理各种文件和对象,利用它还可进行项目连编和连编应用程序,生成 .app 文件或 .exe 文件。

三、实训环境

每名学生配备一台已安装了 Windows XP 或以上版本及 Visual FoxPro 6.0 的计算机。

四、实训内容

1. 显示/隐藏常用工具栏和命令窗口

操作步骤如下。

(1) Visual FoxPro 默认显示常用工具栏。如果要隐藏该工具栏,可执行“显示”菜单中的“工具栏”命令,在打开的工具栏对话框中,不勾选“常用”复选框,单击“确定”按钮即可,如图 1-1-5 所示。

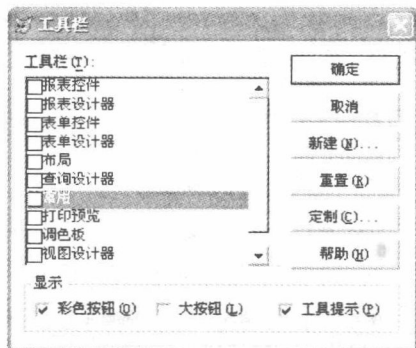


图 1-1-5 隐藏常用工具栏

(2) 要显示该工具栏,只需要在工具栏对话框中勾选“常用”复选框即可。

(3) 显示/隐藏“命令窗口”。Visual FoxPro 的命令窗口默认状态下为显示。此时,常用工具栏上的“命令窗口”按钮处于按下状态。若要隐藏命令窗口,单击该按钮使其弹起即可。

2. 配置 Visual FoxPro 系统的基本环境

在 F 盘上创建成绩管理系统文件夹,将该文件夹设置为文件的默认目录;设置日期的显示格式为“年月日”,显示 4 位年份,日期分隔符为“-”。

操作步骤如下。

(1) 在 F 盘上创建文件夹:成绩管理系统。

(2) 执行“工具”菜单中的“选项”命令,打开“选项”对话框。在“区域”选项卡中按照图 1-1-6 所示设置日期的显示格式。

此设置相当于在命令窗口执行如下命令:

```
SET DATE TO YMD
```

```
SET CENTURY ON
```

SET MARK TO[-]

(3)在“文件位置”选项卡中双击默认目录条目,打开图 1-1-7 所示的“更改文件位置”对话框。

(4)勾选“使用默认目录”复选框,单击文本框右侧的“...”按钮,在选择目录对话框中,选择“f:\成绩管理系统”为默认目录,单击“确定”按钮,返回图 1-1-8 所示的界面,单击“确定”按钮,此设置为本次运行有效。

若单击“设置为默认值”按钮,则设置为永久有效,除非用户再次更改文件目录。

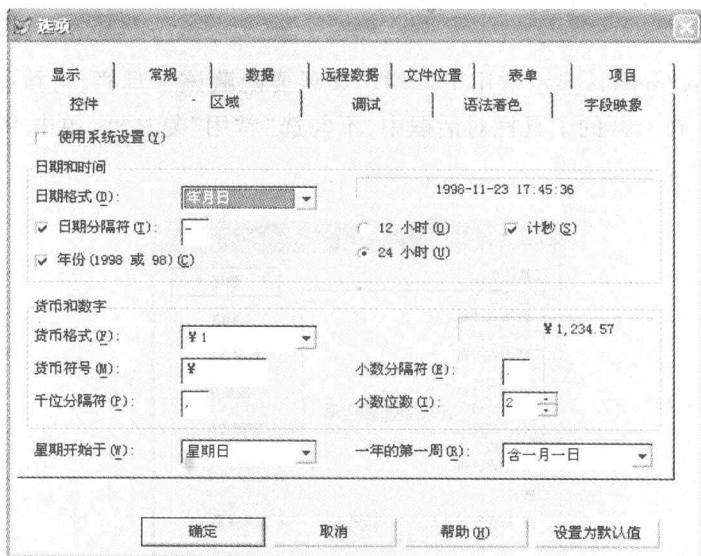


图 1-1-6 设置日期的显示格式

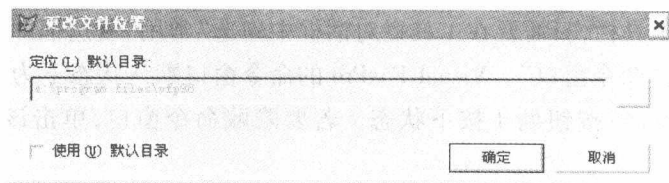


图 1-1-7 “更改文件位置”对话框

还可以通过命令方式设定默认目录。如在命令窗口执行如下命令:

SET DEFAULT TO F:\成绩管理系统

3. 创建项目文件“学生成绩管理. pjt”

创建项目文件的操作步骤如下。

(1)设置文件存储的默认目录为 F:\成绩管理系统。

(2)新建一个名为“学生成绩管理. pjt”的项目文件,即可打开该项目对应的项目管理器,如图 1-1-9 所示。

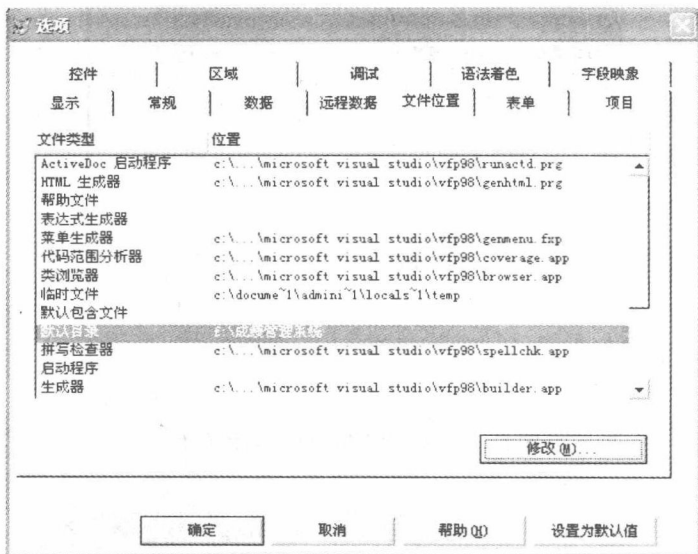


图 1-1-8 设置默认目录

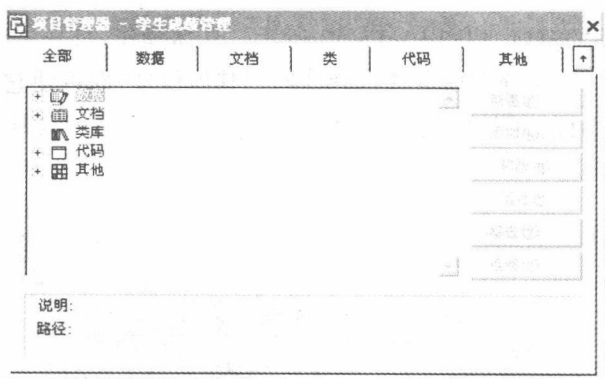


图 1-1-9 项目管理器

4. 打开项目文件“学生成绩管理.pjx”

打开项目文件的具体操作为执行“文件”菜单中的“打开”命令,在“打开”对话框中,选择文件类型为“项目”,再选中“学生成绩管理.pjx”,单击“确定”按钮即可,如图 1-1-10 所示。

5. 设置项目管理器的外观

(1) 移动、缩放、折叠和展开项目管理器。改变“学生成绩管理. pjt”的项目管理器的位置和大小,进行展开和折叠。具体操作如下:将指针指向项目管理器的标题栏,按住鼠标左键拖动即可改变其位置;将指针移到项目管理器的4个边框或4个对角上,按住鼠标左键拖动即可改变其大小;单击项目管理器右上角的向上箭头,可将其折叠起来,如图1-1-11所示;单击项目管理器右上角的向下箭头,即可展开项目管理器。

(2) 拆分项目管理器。在“学生成绩管理.pjx”对应的项目管理器中,使“数据”“文档”

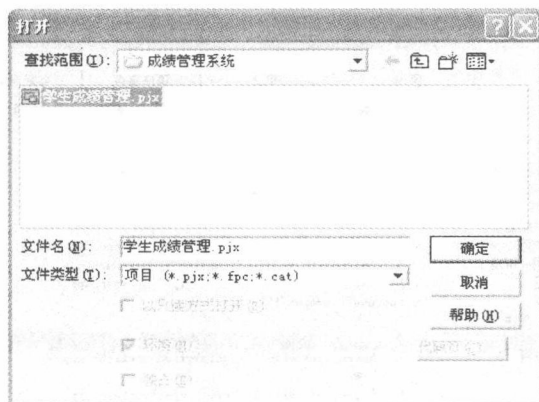


图 1-1-10 打开已有的项目文件

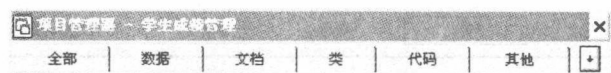


图 1-1-11 折叠项目管理器

选项卡成为独立、浮动的窗口,如图 1-1-12 所示。具体操作:打开该项目文件的项目管理器,将“数据”“文档”选项卡分别拖离项目管理器到其他位置,即可使它们成为独立的窗口。单击选项卡上的图钉图标,还可使其始终显示在屏幕的最顶层。需要还原时,单击该选项卡上的“关闭”按钮即可。

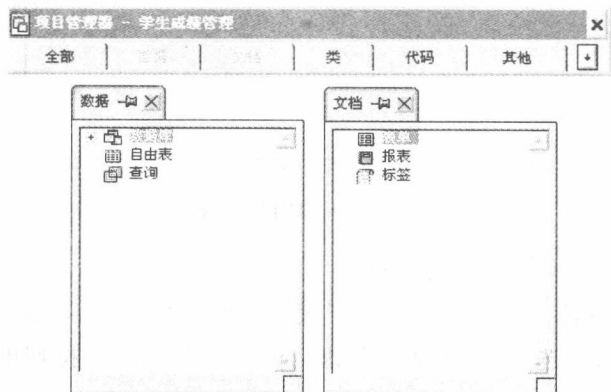


图 1-1-12 拆分项目管理器

6. 项目管理器的使用

要在项目管理器中进行文件的创建、添加、修改、删除以及其他操作,详见后面的相关实训。

五、课外练习

1. 单选题

(1) 关系运算中的选择运算是()。

- A. 从关系中找出满足给定条件的元组的操作
- B. 从关系中选择若干个属性组成新的关系的操作
- C. 从关系中选择满足给定条件的属性的操作
- D. A 和 B 都对

(2) Visual FoxPro 6.0 创建项目的命令是()。

- A. CREATE PROJECT
- B. CREATE ITEM
- C. NEW ITEM
- D. NEW PROJECT

(3) 在 Visual FoxPro 6.0 中,一个项目可以创建()。

- A. 一个项目文件,集中管理数据和程序
- B. 多个项目文件,根据需要设置
- C. 两个项目文件,分别管理数据和程序
- D. 以上几种说法都不对

(4) 项目管理器中包括的选项卡有()。

- A. 数据、菜单和文档
- B. 数据、文档和其他
- C. 数据、表单和类
- D. 数据、表单和报表

(5) 项目管理器中的“运行”按钮可以运行()。

- A. 查询
- B. 程序
- C. 表单
- D. 以上全部都可以

(6) 项目管理器中的“关闭”按钮用于关闭()。

- A. Visual FoxPro
- B. 设计器
- C. 数据库
- D. 项目管理器

2. 填空题

(1) 想要将日期型或日期时间型数据中的年份用 4 位数字显示,应当使用设置命令_____。

(2) 项目文件及其项目备注文件的扩展名分别为_____和_____。

(3) Visual FoxPro 6.0 中的项目管理器实际上是 Visual FoxPro _____、_____和_____的集合,也可以认为是 Visual FoxPro 系统的资源管理器。

3. 上机操作题

(1) 新建一个文件夹 ABC,将其设置为 Visual FoxPro 的默认目录,该设置只是本次运行有效。

(2) 显示/隐藏命令窗口、数据工作期窗口、数据库设计器工具栏。

(3) 创建项目 xsxm.pjx。

实训 1.2 Visual FoxPro 中的数据形式

一、实训目标

通过本次实训,要求学生熟练掌握常量和变量的使用;熟练掌握常用函数的使用;熟练

掌握各种表达式的使用。

二、知识要点

Visual FoxPro 中有 4 种形式的数据:常量、变量、函数和表达式。

1. 常量

常量用来表示一个具体的、不变的值,有 6 种类型,如表 1-2-1 所示。

表 1-2-1 常量的类型

类 型	说 明	示 例
数值型(N)	表示数值的大小,由数字、小数点和正负号组成。可以用科学记数法表示	157、-45 3.26E12、1.6E-8
货币型(Y)	表示货币值,格式与数值型常量类似,但有前置符号(\$)。不能用科学记数法表示,在存储和计算时保留 4 位小数	\$ 67.372
字符型(C)	也叫作字符串,是用半角单引号(')、双引号("")或方括号([])等定界符把任意的字符括起来形成。定界符必须成对匹配,不作为常量本身的内容。但如果此定界符本身是字符串的内容时,则需要用另一种定界符为该字符串定界	'Abc'、"Thank you" "ab[cd]"、[]、[] [二级 Visual FoxPro]
日期型(D)	表示一个具体日期。这里只讨论严格日期格式({^yyyy/mm/dd}):用定界符大括号({})把年、月、日三部分括起来,年、月、日之间用斜杠(默认)、连字号、句点等分隔符相连。书写时注意大括号内必须以脱字符(^)开头;年份必须是 4 位;年、月、日不能省略、次序不能颠倒	{^2012-09-25}
日期时间型(T)	表示一个时间点,格式为{<日期>,<时间>}。严格的日期时间格式是{^yyyy-mm-dd,hh:mm:ss[alp]}	{^2012-09-25,11:30p}
逻辑型(L)	只有逻辑真和逻辑假两个值。句点定界符不能省略	.T.、.Y.、.F.、.N.

注意:货币型、日期型、日期时间型常量在内存中占 8 个字节,逻辑型常量占 1 个字节。影响日期格式的常用设置命令有如下 4 个。

- (1)SET MARK TO"日期分隔符"。设置显示日期型数据时使用的分隔符。若没有指定分隔符,则采用系统默认的斜杠。
- (2)SET DATE [TO] MDY|DMY|YMD。设置日期的显示格式。
- (3)SET CENTURY OFF|ON。设置日期型数据年份的显示位数。选 OFF 时年份显示两位,选 ON 时年份显示 4 位。该命令的默认短语为 OFF。
- (4)SET CENTURY TO [世纪值] ROLLOVER [年份参照值]。确定用两位数字表示年份时所处的世纪,若该日期的两位数字年份大于或等于[年份参照值],则它所处的世纪即为[世纪值],否则为[世纪值]+1。

如执行命令 SET CENTURY TO 19 ROLLOVER 49 后,函数 CTOD('12/18/53')表示的日期为 1953 年 12 月 18 日;函数 CTOD('12/18/45')表示的日期为 2045 年 12 月 18 日。

2. 变量

变量是指在操作过程中值或类型可以改变的数据。变量有 3 个要素:变量名、数据类型

和变量值,变量名不区分字母大小写。Visual FoxPro 的变量分为字段变量和内存变量两类。

二维表中的字段名即为字段变量。表中的同一个字段名,各个记录的取值不尽相同。其常用的数据类型有字符型(C)、数值型(N)、整型(I)、货币型(Y)、逻辑型(L)、日期型(D)、日期时间型(T)、备注型(M)和通用型(G)。

内存变量是内存中的一个存储区域,变量值为存储在这个区域里的数据,变量的数据类型由变量值的类型决定。可以把不同类型的数据赋给同一个变量。内存变量常用的数据类型有6种:字符型(C)、数值型(N)、货币型(Y)、逻辑型(L)、日期型(D)和日期时间型(T)。

当字段变量和内存变量同名时,系统优先访问字段变量。要访问同名的内存变量,必须在变量名前加上 M. 或 M - >。

(1) 内存变量的定义(赋值),有如下两种格式。

格式1: <内存变量名> = <表达式>

格式2: STORE <表达式> TO <内存变量名表>

说明:使用格式1一次只能给一个内存变量赋值;使用格式2一次可给多个内存变量赋相同的值,注意各内存变量名之间必须用逗号隔开。

(2) 表达式值的输出,有如下用法。

格式: ? | ?? [<表达式表>]

说明:选用?时,若没有指定表达式则会输出1个空行,否则会在下一行的起始处依次输出各表达式的值;若选用??,则会在当前行的光标位置处依次输出各表达式的值。

(3) 内存变量的显示,命令格式如下。

格式: LIST/DISPLAY MEMORY [LIKE <通配符>]

说明:LIST MEMORY 能连续显示与通配符匹配的所有内存变量。若已定义的内存变量显示超过一屏,则只显示最后一屏;DISPLAY MEMORY 能分屏显示所有的内存变量。

(4) 内存变量的清除,常用格式如下。

格式1: CLEAR MEMORY

格式2: RELEASE <内存变量名表>

格式3: RELEASE ALL [LIKE <通配符> | EXCEPT <通配符>]

说明:格式1用来清除所有的内存变量;格式2用来清除指定的内存变量;在格式3中若选用LIKE短语则清除与通配符匹配的内存变量,若选用EXCEPT短语则清除与通配符不匹配的内存变量。

(5) 数组。数组是内存中连续的一片存储区域,由一系列元素组成。每个数组元素相当于一个简单变量,可以通过数组名及相应的下标来访问。在Visual FoxPro中,可以给一个数组中的各个元素分别赋不同的值,各元素的数据类型也可以不同。

数组必须先定义后使用。数组创建后,每个数组元素的初始值为.F.。

创建数组的命令格式如下:

DIMENSION|DECLARE <数组名> (<下标上限1> [, <下标上限2>]) [, ...]

注意：

- ①数组下限的最小值为 1,数组元素和简单内存变量的使用方法相同；
- ②在赋值语句和输入语句中若使用数组名,表示将同一个值同时赋给该数组的全部数组元素,在赋值语句的表达式位置不能出现数组名；
- ③在同一运行环境下,数组名不能与简单内存变量同名；
- ④二维数组可等价为二维数组,即可使用二维数组的形式访问二维数组。

3. 表达式

用运算符及圆括号将常量、变量和函数连接起来的式子称为表达式。单个常量、变量和函数都属于表达式的范畴。根据表达式值的类型,表达式可分为数值表达式、字符表达式、日期时间表达式和逻辑表达式(关系表达式属于简单的逻辑表达式)。

(1)数值表达式(算术表达式)。数值表达式由算术运算符将数值型数据连接起来形成,其运算结果仍为数值型。各算术运算符及其优先级如表 1-2-2 所示。

表 1-2-2 算术运算符及其优先级

优先级	运算符	说明	备注
1	-	一元取负运算	
2	* * 或^	乘方运算	
3	*, /, %	乘、除、求余运算	求余运算%和取余函数 MOD()的作用相同
4	+, -	加、减运算	

(2)字符表达式。字符表达式由字符串运算符将字符型数据连接起来形成,运算结果仍为字符型。字符串运算符有“+”和“-”两个,它们具有相同的优先级。

- “+”前后两个字符串首尾相接。
- “-”将前面字符串尾部的空格移到后面字符串的尾部,再连接两个字符串。

(3)日期时间表达式。日期时间表达式运算符也有“+”和“-”两个,运算优先级相同,但其格式有一定的限制。合法的日期时间表达式格式如表 1-2-3 所示,其中的 <天数>和<秒数>均为数值表达式。

表 1-2-3 合法的日期时间表达式的格式

格式	结果及类型
<日期> + <天数>	日期型(指定日期若干天后的日期)
<天数> + <日期>	日期型(指定日期若干天后的日期)
<日期> - <天数>	日期型(指定日期若干天前的日期)
<日期> - <日期>	数值型(两个指定日期相差的天数)
<日期时间> + <秒数>	日期时间型(指定日期时间若干秒后的日期时间)
<秒数> + <日期时间>	日期时间型(指定日期时间若干秒后的日期时间)