



数据库 基础教程

(实验指导)

主 编 田 青 刘 畅
主 审 李洪心

高等院校
经济与管理类专业
公共基础课
标准教材



东北财经大学出版社

高等院校经济与管理类专业公共基础课标准教材

数据库基础教程

FoxPro for Windows

(实验指导)

主编 田青 刘畅

主审 李洪心

2010年10月

东北财经大学出版社

前 言

为配合普通高等院校经济与管理类专业数据库课程的课堂教学，使学生更好地理解和消化课堂教学内容，我们组织编写了这本《数据库基础教程实验指导》。

《数据库基础教程》是由东北财经大学出版社出版的，由李洪心主编的高等学校经济与管理类专业公共基础课标准教材，本实验指导与之配套，共设置了 13 个实验单元。

其中，前 12 个单元与教材的前 11 章内容相一致，第 13 单元给出了一个综合性的实验例题，使学生对使用 FoxPro 进行编程有一个整体上的认识，同时，要求学生自己至少编写一个有一定实用价值的应用程序。

由于大部分学校的计算机相关课程已经实现了计算机教学，使课堂教学和上机实验成为一体，因此，本书可在课堂上进行，待教师讲解后即可进行练习。同时，本实验指导以实验内容为单元，而不以上机时间为单元，全部上机时间为 30 学时，可分多次进行。

本书由田青、刘畅主编。具体参编人员如下：

尹征杰（第 1 单元）

刘树安（第 2 单元）

刘 畅（第 3、4 单元）

廖明怡（第 5 单元）

田 青（第 6 单元）

卢永艳（第 7、8 单元）

孙 平（第 9 单元）

倪亚东（第 10 单元）

谢兰云（第 11 单元）

刘继山（第 12 单元）

田青、刘畅，（第 13 单元）

李洪心教授作为《数据库基础教程》（教科书）的主编，担任了本书的主审，对本书的内容进行了全面、认真的审阅，在此我们表示由衷的感谢。

由于时间仓促，加之我们的水平有限，书中存在的错误和不当之处，请读者批评指正。

编者

2002 年 1 月

目 录

第 1 单元	FoxPro 的基本操作	1
第 2 单元	FoxPro 的基本元素	4
第 3 单元	数据库的建立与管理	16
第 4 单元	排序、索引与查询	22
第 5 单元	结构化查询语言 SQL 与 RQBE 窗口查询	31
第 6 单元	多数据库操作	34
第 7 单元	程序操作与简单程序的编写	41
第 8 单元	循环结构程序的编写与数组的应用	47
第 9 单元	过程与自定义函数	55
第 10 单元	屏幕与菜单设计	60
第 11 单元	目录管理器及其导航器	64
第 12 单元	应用程序用户界面设计	73
第 13 单元	综合练习	82

第1单元 FoxPro 的基本操作

1.1 实验目的

- (1) 熟练掌握 FoxPro 的安装、启动和退出
- (2) 熟悉 FoxPro 的菜单

1.2 实验内容

(1) FoxPro 的安装

FoxPro 2.6 for Windows 可以用软盘安装,也可以用光盘安装(常用)。不论是软盘还是光盘,安装时我们只要找到软盘或光盘上的 install.exe 或 setup.exe 文件,运行它就可以了(用鼠标双击 install.exe 或 setup.exe 文件的图标)。

(2) FoxPro 的启动

进入 Windows 后,点击开始菜单,移动鼠标到程序项 FoxPro for Windows,点击鼠标左键,进入 FoxPro 系统。

(3) FoxPro 的退出

若要退出 FoxPro,可在 Command 窗口中键入 quit 或在 File 菜单中选 Exit,即可退出 FoxPro,此时自动关闭 FoxPro 所有已打开的文件。

(4) FoxPro 的菜单

在 FoxPro 的主菜单条上共有 9 个菜单项,其主要功能如下:

①File: 文件操作

- New、Open、Save、Close: 文件的创建、打开、保存和关闭。
- Print Setup、Print: 打印机设置、文件打印。
- Catalog Manager: 目录管理器。
- Exit: 退出 FoxPro。

②Edit: 编辑操作

- Undo、Redo: 取消或恢复上一次操作。
- Cut、Copy、Paste: 剪切、复制、粘贴。
- Find、Replace: 搜索、替换等。

③Database: 数据库操作

• Setup、Browse、Append From、Copy to、Sort、Total: 数据库设置、浏览、追加、复制、排序、统计。

• Average、Count、Sum、Calculate: 计算记录的平均值、记录个数、记录求和、综合统计。

• Report、Label、Reindex: 输出报表、标签、重建索引。

④Record: 记录操作

• Append、Change、Delete、Recall: 记录的增加、修改、删除和恢复。

• Seek、Replace: 记录的搜索和替换。

• Goto、Locate、Continue、Seek: 记录的定位、查找。

⑤Program: 程序管理

• Do、Trace、Debug、Compile: 程序的执行、跟踪、调试和编译。

• Macro: 宏命令。

• FoxDoc: 生成 FoxDoc 文档、程序格式化命令。

⑥Run: 运行

• New Query: 建立新的查询文件。

• Query、Screen、Report、Label、Application: 执行查询、屏幕、报表、标签和应用程序。

• Convert dBASE: 转换 dBASE 文件。

• Wizard: 进行各种 Wizard 简化操作。

⑦Text: 文本

• Font、Enlarge、Reduce: 字体选择和缩放。

• Single Space、1.5 Single Space、Double Space: 单倍、1.5 倍、双倍行距。

• Indent、Undent: 文件缩进格式。

• Spelling: 拼写检查。

⑧Windows: 窗口管理

• Hide、Clear、Cycle: 窗口的隐藏、显示和切换。

• Command: 命令窗口的打开与关闭。

⑨Help: 在线帮助

1.3 实验步骤

(1) 进行 FoxPro 的安装, 并注意安装时的屏幕显示的变化, 进行适当的选择。

(2) 进行 FoxPro 的启动, 并观察屏幕显示的变化。

(3) 使用 File 菜单中的 Exit 项进行 FoxPro 的退出。

(4) 在命令窗口使用 quit 命令进行 FoxPro 的退出。

(5) 练习 FoxPro 的菜单操作。

1.4 注意事项

(1) 安装时 FoxPro 提供 3 种安装方式：

① Complete Installation (完整安装)

② Custom Installation (用户安装)

③ Minimum Installation (最小安装)

用户可以根据实际情况选择其中之一，一般采用第一种方式安装。

(2) 每个菜单基都有字母热键，按 ALT 键和菜单基的字母热键可随时访问相应的下拉菜单。如按 ALT + F 即出现下拉的 FILE 菜单。

(3) 每个菜单选项都有一个高亮度字母是该选项的热键，按热键可直接访问相应的菜单项。如菜单选项后标有控制符号如 F1 或 ^U 等，则在命令窗口键入此键，可直接访问该菜单选项。

(4) 按 ESC 键可以退出菜单系统。

第2单元 FoxPro 的基本元素

2.1 实验目的

- (1) 理解 FoxPro 数据类型、常量、变量、表达式的概念
- (2) 理解 FoxPro 文件的操作命令、FoxPro 命令的格式
- (3) 掌握内存变量的建立、输出、显示、释放、保存与恢复的有关命令
- (4) 掌握数值处理函数的有关命令
- (5) 掌握字符处理函数的有关命令
- (6) 掌握日期和时间函数的有关命令

2.2 实验内容简介

- (1) 文件操作的基本命令与 FoxPro 的命令格式

① copy file <文件1> to <文件2>

该命令可以复制任意类型的文件

② copy memo <备注字段名> to file <文件名> [additive]

该命令可以将备注字段中的文本生成一个新的文件

③ dir [路径] [通配符] [to print | to file <文件名>]

该命令可以显示磁盘文件目录

④ list | display files on [盘符路径] [like <通配符>] [to print | to file <文件名>]

该命令可以显示任意类型的文件列表

⑤ erase <文件名> | ?

该命令可以删除文件，如果用?代替文件名，则出现选择文件名的对话框。

⑥ rename <文件1> to <文件2>

该命令可以更改文件名

⑦ FoxPro 的命令格式

命令动词 [<Scope>] [Fields <field list>] [For <exp1>] [While <exp2>]

其中：命令动词指出计算机要执行的操作。<Scope>子句用来指示该命令所操作的数据库的记录范围，即指定一个二维表中的某一行、或部分行、或全部行进行操作，有关范围选择子句有四种形式：

all: 对全部记录进行操作;

next n: 对从当前记录开始, 往下共 n 个记录进行操作;

record n: 仅对第 n 号记录进行操作;

rest: 从当前记录开始, 一直到数据库最后一个记录进行操作。

Fields < field list > 子句实际上是选取一个二维表的某一列或某些列进行操作, 即 < field list > 为数据库文件的一个或多个字段, 字段间以逗号分隔。

For < exp1 > 与 While < exp2 > 子句仍是一个二维表中“行”的条件选择操作, 两个子句的区别是: While 子句的作用是只有当条件为真, 才对当前记录进行所要求的处理, 并把指针指向下一个记录, 一旦遇到使条件不成立的记录, 就停止搜索并结束该命令的执行。For 子句的作用是对条件为真的记录进行所要求的操作, 否则不进行操作, 直到将范围子句规定的记录搜索完为止。两个子句可单独或同时使用, 同时使用时, While 子句优先 For 子句。

(2) 有关内存变量的操作命令

① store <表达式> to <内存变量> | <数组表>

该命令可以为内存变量或数组赋值

② <内存变量> | <数组> = <表达式>

该命令可以把表达式的值赋给内存变量或数组

③ 用 “??,???” 命令输出变量与表达式的值

该命令分别是在屏幕上不换行输出和输出到打印机

④ display memory [like <通配符>] [to print | to file <文件名>]

该命令是显示内存中已定义的内存变量

⑤ release <内存变量表>

release all like [<通配符> | except <通配符>]

clear memory

该命令的功能是释放内存变量

⑥ save to <文件名> | to memo <备注字段> [all like | except <通配符>]

该命令的功能是把当前内存变量和数组存贮到内存变量文件中或数据库文件的备注字段中

⑦ restore from <文件名> | from memo <备注字段> [additive]

该命令的功能是从指定的内存变量文件或数据库文件的备注字段中将存储的内存变量和数组重新调入内存

(3) 有关数值处理函数的操作命令

① exp (<数值表达式>)

该函数的功能是求 e 的以表达式为幂的乘方数

② log (<数值表达式>)

该函数的功能是求数值表达式的自然对数

③ abs (<数值表达式>)

该函数的功能是求数值表达式的绝对值

④ int (<数值表达式>)

该函数的功能是给出数值表达式计算结果的整数部分

⑤ mod (<数值表达式 1>, <数值表达式 2>)

该函数的功能是给出<数值表达式 1>除以<数值表达式 2>的余数,结果值的符号与<数值表达式 2>相同

⑥ round (<数值表达式 1>, <数值表达式 2>)

该函数的功能是根据<数值表达式 2>给定的小数位数对<数值表达式 1>进行四舍五入

⑦ val (<字符表达式>)

该函数的功能是把<字符表达式>转换成数值型数据

⑧ isdigit (<字符表达式>)

该函数的功能是测试字符表达式是否以数字字符开头

⑨ max (<表达式 1>, <表达式 2>, <表达式 3>...)

该函数的功能是返回表达式的最大值,结果是日期型或数值型的数据

⑩ min (<表达式 1>, <表达式 2>, <表达式 3>...)

该函数的功能是返回表达式的最小值,结果是日期型或数值型的数据

(4) 有关字符处理函数的操作命令

① 有关字符串测试函数的操作命令

- at (<字符表达式 1>, <字符表达式 2>), <数值 (第 n 次出现)>)

该函数的功能是查找<字符表达式 1>在<字符表达式 2>中第<数值>次出现的起始位置,省略<数值>,则查找第一次出现的位置。若查到,则返回一数值表示其位置,否则返回 0。

- atline (<字符表达式 1>, <字符表达式 2>)

该函数的功能是测定 (<字符表达式 1>在<字符表达式 2>中首次出现的行号。

- len (<字符表达式>)

该函数的功能是测定字符串的长度,返回一数值

- empty (<表达式>)

该函数的功能是测试给定的表达式是否为空值

- isalpha (<字符表达式>)

该函数的功能是测试字符串是否以字母开头,返回逻辑值

- islower (<字符表达式>)

该函数的功能是测试字符串是否以小写字母开头,返回逻辑值

- type (<表达式>)

该函数的功能是测试“表达式”的数据类型，表达式一定用引号括起，输出值为字符型，可以为 C、N、L、D、M、U 等

②有关字符串处理函数的操作命令

- substr (<字符表达式>, <起始位置>, <子串长度>)

该函数的功能是取出<字符表达式>所表示的字符串中的一个子串

- left (<字符表达式>, <N 字符个数>)

该函数的功能是从<字符表达式>的左边开始取 N 个字符

- trim (<字符表达式>)

该函数的功能是截掉<字符表达式>尾部的空格，返回保留的字符串

- space (<数值表达式>)

该函数的功能是产生一个长度为<数值表达式>的空格字符串

- replicate (<字符>, <个数>)

该函数的功能是产生一个由相同字符组成的字符串

- chrtran (<字符表达式>, <原字符>, <替换字符>)

该函数的功能是在<字符表达式>中查找欲被替换的<原字符>，如果找到，就用<替换字符>代替

- strtran (<字符串表达式>, <原字符表达式>, <替换字符表达式>, <替换起始位置>, <替换次数>)

该函数的功能是在<字符串表达式>中查找指定的<原字符表达式>，找到之后，用<替换字符表达式>替换<原字符表达式>的内容

③有关字符转换处理函数的操作命令

- str (<数值表达式1>, <数值表达式2>, <数值表达式3>)

该函数的功能是将<数值表达式1>的值按<数值表达式3>截取其小数位后转换为长度等于<数值表达式2>的字符串

- asc (<字符表达式>)

该函数的功能是将<字符表达式>中最左边字符转换为相对应的 ASCII 码

- chr (<数值表达式>)

该函数的功能是将<数值表达式>的值转换成相对应的字符

- upper (<字符表达式>)

该函数的功能是将<字符表达式>中的所有小写字母转换成大写字母

- lower (<字符表达式>)

该函数的功能是将<字符表达式>中的所有大写字母转换成小写字母

- & (<内存变量>, [<字符表达式>])

该函数的功能是对字符型内存变量进行宏替换

(5) 有关日期和时间函数的操作命令

① date (<日期表达式>)

该函数的功能是给出系统日期

② year (<日期表达式>)

该函数的功能是给出<日期表达式>的由四位数表示的年份

③ month (<日期表达式>)

该函数的功能是给出<日期表达式>相对应的月份数

④ cmonth (<日期表达式>)

该函数的功能是返回<日期表达式>月份值的英文名

⑤ day (<日期表达式>)

该函数的功能是给出<日期表达式>对应的日期号

⑥ cday (<日期表达式>)

该函数的功能是给出<日期表达式>对应的星期几

⑦ ctod (<字符表达式>)

该函数的功能是把<字符表达式>转换成日期型数据

⑧ dtoc (<日期表达式>)

该函数的功能是把<日期表达式>转换成字符型数据

⑨ time (<数值表达式>)

该函数的功能是返回当前系统时间

2.3 实验例题

(1) 将 10 赋给变量 x1 和 x2

Store 10 to x1, x2

(2) 将逻辑真 .t. 赋给 x1

Store .T. to x1

(3) 将“大连”赋给 x3, x4

Store "大连" to x3, x4

(4) 将 100 赋给 A1

A1 = 100

(5) 将“大连”赋给 B1

B1 = "大连"

(6) 将 xs.dbf 复制一份, 命名为 xs01.dbf

Copy file xs.dbf to xs01.dbf

(7) 将当前目录下所有扩展名为 .dbf 文件名清单存入文件 dbf01.txt 中

Dir like * .dbf to file dbf01.txt

- (8) 将当前数据库备注字段 bz 的内容添加到文件 aa.txt 的尾部

Copy memo bz to file aa.txt additive

- (9) 显示 D 盘 foxprow 子目录下扩展名为 .prg 的文件清单

list files on d: \ foxprow like * .prg

- (10) 删除 C 盘 user 子目录下的文件 cj.dbf

Erase c: \ usre \ cj.dbf

- (11) 将当前目录下 xs.dbf 文件改名为 xs88.dbf

Rename xs.dbf to xs88.dbf

- (12) 不换行显示 $46 * 90 + 60$ 的值

?? $46 * 90 + 60$

- (13) 显示已经定义的内存变量，并将结果存入 a1.txt 文件中

Display memo like * to file a1.txt

- (14) 释放已经定义的内存变量 a, b

Release a, b

- (15) 释放已经定义的全局内存变量

Clear memory

- (16) 把当前内存变量和数组存储到数据库 xs.dbf 备注字段中

Save to memo bz all like * (bz 是数据库文件 xs.dbf 的备注字段)

- (17) 从数据库文件 xs.dbf 的备注字段中将存储的内存变量和数组调入内存

Restore from memo bz

- (18) 求 e 的平方

? exp (2)

- (19) 求 26 的自然对数

? log (26)

- (20) 求 -46 的绝对值

? abs (-46)

- (21) 对 $36 * 12.234$ 取整

? int ($36 * 12.234$)

- (22) 求 24 除以 5、24 除以 -5、-24 除以 5、-24 除以 -5 的余数，并比较结果是否有规律。

? mod (24, 5)

? mod (24, -5)

? mod (-24, 5)

? mod (-24, -5)

- (23) 对 $23 * 2.563$ 的值进行四舍五入
? round (23 * 2.563, 2)
- (24) 将字符串 “30” 赋给变量 a, 然后将 a 转换成数值型数据再加上 45
Store “30” to a
? val (a) + 45
- (25) 测试表达式 “ajdjj54646” 是否以数字字符开头
? isdigit (“ajdjj54646”)
- (26) 求数值 455, 584, 789 的最大值
? max (455, 584, 789)
- (27) 求数值 455, 584, 789 的最小值
? min (455, 584, 789)
- (28) 查找字符 “ahb” 在字符 “xvabhbdjahbgfhhbahbh” 第三次出现的位置
? at (“ahb”, “xvabhbdjahbgfhhbahbh”, 3)
- (29) 测试 b1 = “aa” 在 b2 = “gfhjhgjdjhghghdjhghghgdghjghjdfhhhhhdjdiddhfhjfhfjfhfh-hfjdjfaahffj aa” 中首次出现的行号
b1 = “aa”
b2 = “gfhjhgjdjhghghdjhghghgdghjghjdfhhhhhdjdiddhfhjfhfjfhfhfhfjdjfaahffj aa”
? atline (b1, b2)
- (30) 测试字符串 “my name” 的长度
? len (“my name”)
- (31) 测试 “ ” 是否为空
? empty (“ ”)
- (32) 测试 date、date () 与 .t. 数据类型
? type (“date”)
? type (“date ()”)
? type (“.t.”)
- (33) 从字符串 “abcdefg” 的第三个字母开始取出 3 个字母
? substr (“abcdefg”, 3, 3)
- (34) 从字符串 “english” 的左边取 3 个字母
? left (“english”, 3)
- (35) 截掉字符串 “english ” 尾部的空格
? trim (“english”)
- (36) 产生 10 个由 “ab” 组成的字符串
? replicate (“ab”, 10)
- (37) 将字符串 “dddxffsddd” 中的 ‘d’ 用 ‘p’ 来代替

? chrtran ("dddxffsddd","d","p")

- (38) 将 4657.899 保留两位小数后转换为长度等于 8 的字符串

? str (4657.899, 8, 2)

- (39) 将 "state" 最左边的字符转换为相应的 ASCII 码

? asc ("state")

- (40) 将 ASCII 码 49 转换为相应的字符

? chr (49)

- (41) 将字符串 "hfhFGFsfh" 中的所有的小写字母转换成大写字母

? upper ("hfhFGFsfh")

- (42) 将字符串 "jhGHJfk" 中的所有的大写字母转换成小写字母

? lower ("jhGHJfk")

- (43) 显示系统当前的年份

? year (date ())

- (44) 显示日期表达式 cmonth ({00-08-23} 月份的英文名

? cmonth ({00-08-23})

- (45) 给出系统当前日期的日期号

? day (date ())

- (46) 把系统当前日期转换成日期型数据, 并测试其类型

? cdo (date ())

? type ("cdo (date ())")

- (47) 测试 "foxpro" 是否以字母开头

? isalpha ("foxpro")

- (48) 测试 "foxpro" 是否以小写字母开头

? islower ("foxpro")

- (49) 显示系统的当前日期

? date ()

- (50) 显示系统的当前时间

? time ()

2.4 实验习题

- (1) 将 36 赋给 x1, x2

- (2) 将 "中国大连" 赋给变量 C

- (3) 输出 $3 * 6 + 90$ 的值
- (4) 输出 $6 * 9 > 100$ 的值
- (5) 输出 $\{99 - 10 - 01\} + 46$ 的值
- (6) 将 C 盘 FoxPro 子目录下的文件 cj.dbf 拷贝到 A 盘根目录下
- (7) 列出当前磁盘当前路径下的文件目录
- (8) 将数据库文件 xs.dab 中备注字段 bz 中的内容添加到文件 b1.txt 中
- (9) 删除 D 盘根目录下的 a02.txt 文件
- (10) 将文件 xs.dbf 改名为 xs0011.dbf
- (11) 不换行输出字符串 “ADFGVH”
- (12) 显示内存中已定义的内存变量
- (13) 把当前的内存变量存贮到文件 b1.txt 中
- (14) 将文件 b1.txt 中存储的内存变量和数组重新调入内存
- (15) 求 e 的 9 次方
- (16) 求 18 的自然对数
- (17) 求 34 乘 -3 的绝对值
- (18) 求 785.678 的整数部分
- (19) 求 -24 除以 5 的余数
- (20) 对 765.8936 进行四舍五入, 保留 2 位小数

- (21) 测试字符串 “7899ajdjj54646” 是否以字母开头
- (22) 测试 98 - 12 - 30、89 - 05 - 12 与 96 - 06 - 23 哪个日期为最大
- (23) 测试字符串 “ab” 在字符串 “xvabhdjabgfhhbabh” 中第二次出现的位置。
- (24) 测试 “name” 在 “foxpro system name” 中的行号。
- (25) 测定字符串 “中国大连” 的长度
- (26) 测定字符串 “AA” 是否为空
- (27) 测定字符串 “88foxpro AA” 是否以字母开头
- (28) 测定字符串 “88foxpro AA” 是否以小写字母开头
- (29) 测定表达式 “36 * 9” 的类型
- (30) 从字符串 “abcdefg” 的第四个字母开始截取 2 个字母
- (31) 从字符串 “book” 的左边开始截取 2 个字母
- (32) 截掉字符串 “book” 右边的空格
- (33) 生成一个由 8 个字符 “cd” 组成的字符串
- (34) 将数值 467857.899 转换为保留两位小数，长度为 11 的字符串。
- (35) 将 “my” 最左边字符转换为相对应的 ASCII 码
- (36) 将 80 的值转换成相对应的字符
- (37) 显示系统的当前时间