

通用学生成绩管理系统的设计与实现

李 政 (四平师范学院计算机系)

摘要:本文介绍一个通用的学生成绩管理系统的设计方法,给出主要清单。该系统全部用 FoxBASE 编程实现。其中在数据录入、新库结构的形成、专业课的标识、成绩的统计、特别是数据库的横向统计等方面都采用了比较先进的技术,使程序具有很强的通用性和技巧性。

一、系统总体设计

数据库管理系统,一般可分为程序 and 数据库两大部分。

本系统的程序部分包括一个主程序和若干个子程序,为减少系统文件数目,提高文件查找速度,增加软件清晰度,可将所有的子程序放到一个过程文件里。

数据库部分的设计思想是,以学生组为单位分别建库。在这里,高等学校同系同年级同层次的全体学生可做为一个学生组,中等学校同校同年级的全体学生可做为一个学生组。每个学生组对应于三个数据库:成绩库、结构描述文件(库)和统计结果库。其中成绩库是主体,它与结构描述文件对应并通过结构描述文件建立,用来存放学生各门课成绩和一些统计信息(如,总平均分、专业课平均分,名次等)。结构描述文件还用于数据的输入和修改,成绩的统计以及查询打印等。成绩库和结构描述文件要根据不同层次、不同类别学生(即不同学生组)的实际情况来确立。统计结果库用来存放成绩库中各门课程的统计信息(如优秀人数、良好人数、不及格人数、及格率等),它的结构不需要改变。

各数据库文件可按下面规则命名:

```

|_|_|_| 类别(空--成绩库、T--统计结果库、j--结构描述文件)
|_|_|_| 层次(空--不分层次、Y--研究生、B--本科类 Z--专科)
|_|_|_| 年级(如 89、90、91...)
|_|_|_| 系列、校别(如: SX--数学系、JSJ--计算机系、SZ--四中)
  例如,数学系 82 级本科生成绩库名为 SX82B,统计结果库为 SX82B T,结构描述文件为 SX82B J。计算机 93 级专生成成绩库名为 JSJ93Z,统计结果库为
  
```

JSJ93Z-T,结构描述文件为 JSJ93Z-J。

系统包括如下功能:

1.建新库结构

根据不同学生组的实际情况建立相应的成绩库结构,确定各门课的课程名称、课程类别等信息,产生对应的结构描述文件。

2.修改结构

修改已存在的成绩库中课程名、课程类别等信息,并重新产生对应的结构描述文件。

3.输入或修改数据

按多种方式输入、修改成绩库中各种数据。

4.整理/统计

对成绩库进行排序、按学生或按课程统计各种结果。

5.查询/打印

按各种要求查询任意成绩库及统计结果。

下面介绍几个主要功能模块。假定在执行各功能模块之前,成绩库和对应的结构描述文件名已经确定,并分别用变量 CJK 和 CJK J 表示。

二、建新库结构

本模块的技术关键是利用一个已有的结构描述文件样板库 JG.DBF。通过对它的修改得到所需要的成绩库结构,并产生对应的结构描述文件。这里所用的结构描述文件除了有一般结构描述文件的四个字段 FIELD__NAME、FIELD__TYPE、FIELD__LEN 和 FIELD__DEC 外,还增加了一个字段“课程类别”、数值型、长度为 1,其目的是用它来标识各门课类别(如 1 表示专业

必修课, 2 表示选修课, 3 表示公共课), 这一字段的具体应用在第四部分将会看到。

下面是结构描述文件样板库 JG.DBF 的部分内容:

Record#	FIELD__	NAME	FIELD__	TYPE
FIELD__DEC	课程类别			
1	学号	C	5	
2	班级	C	1	
3	姓名	C	6	
4	总平均分	N	6	2
5	总名次	N	3	
6	专必平均分	N	6	2
7	专必名次	N	3	
8	>80门数	N	2	
9	<70门数	N	2	
10	缺考数	N	2	
11	补考门数	N	1	
12	数学分析	N	5	1 1
13	高等代数	N	5	1 1
14	解析几何	N	5	1 1
15	外语	N	5	1 3
.....
37	计算方法	N	5	1 2
38	泛函分析	N	5	1 2

程序清单的主要内容如下:

```

PROC CJGLI
WJM = TRIM(CJK) + ".DBF" && 成绩库名
==>WJM
IF FILE(WJM)
WAIT "该文件已经存在,请检查或修改库结构..."
TO NNN
RETU
ENDIF
USE JG &&打开结构描述文件样板库
BROW &&进行修改
PACK &&删除做了标记的记录
CREA &CJK FROM JG &&建立成绩库结构
USE JG
COPY TO &CJK J &&建立对应的结构描述文件
RETU
    
```

三、数据的录入及修改

本模块提供以下几种输入及修改方式: 1. 输入、修改学生名册, 2. 输入、修改补考数, 3. 输入、修改各门课成绩, 4. 修改满足条件的记录和字段, 5. 浏览修改全部数据。

其中输入、修改学生名册可用命令

BROW FIEL 学号, 班级, 姓名, 实现, 输入、修改补考数可用命令

BROW FIEL 学号, 班级, 姓名, 补考门数 FREE 补考门数 NOAPP 实现, 浏览修改全部数据, 可用命令

BROW LOCK 3 NOAPP 实现。

下面重点介绍“输入、修改各门课成绩”及“修改满足条件的记录和字段”程序。

为了能够任意指定某门课程进行输入或修改成绩, 可首先用程序段

```

ZDS=0
USE &CJK_j &&打开结构描述文件
CO 12 &&课程名从第 12 号记录开始
DO WHILE .NOT. EOF()
ZDS=ZDS+1
XX(ZDS)=FIELD NAME
SKIP
ENDDO
    
```

求出课程门数送变量 ZDS, 并将各课程名送给下标变量 XX(1), XX(2), ...。这里用到了结构描述文件, 以使程序具有通用性。然后用程序段

```

DO WHILE .T.
DO XSZD &&显示字段
ZDH=0
21.1SAY "请输入程序号: "GET ZDH PICT"99"
RANG 0.ZDS
READ
IF ZDH=0
EXIT &&出口
ENDIF
KCM=TRIM(XX(ZDH))
USE &CJK
BROW FIEL 学号, 班级, 姓名, &KCM FREE
    
```

&KCM NOAPP

ENDD0

进行输入或修改指定课程的成绩。这里,子程序 XSZD 将下标变量 XX(1),XX(2),...的值(这里都是课程名)及其序号按一定格式在屏幕上显出来供选择(程序清单略),最后用命令

BROW FIEL 学号,班级,姓名,&CM FREE

&KCM NOAPP

显示学号、班级、姓名和指定的课程名并使输入和修改固定在指定的课程名字段。

修改满足条件的记录和字段,可通过下面程序段实现。在这段程序中,用了另外两个子程序 OZD 和 SRTJ,其中 OZD 用来取所有字段名送下标变量 XX(1),X(2),...,字段个数送变量 ZDS,SRTJ 用来指定条件表达式,确定的条件表达式字符串用变量 JSTJ 表示。程序以对话方式通过输入字段的序号来确定要修改的字段名,用变量 SCZD 表示。最后通过命令

EDIT ALL &SCZD FOR &JSTJ

实现对满足条件的记录和字段的修改。

USE &CJK_J &&打开对应的结构描述文件

描述文件

DO OZD &&取各字段名 == > XX

(1),XX(2),...字段数 == > ZDS

DO XSZD &&显示各字段名及序号

DO SRTJ &&指定条件

SCZD="FIELDS 学号,班级,姓名"

DO WHILE .T. &&确定要修改的字段

21.0 CLEAR

22.1 SAY SCZD

ZDH=0

21.1 SAY "请键入要修改的字段号:"

GET ZDH PICT "99" RANG 0, ZDS

READ

IF ZDH=0 &&直接回车,结束

EXIT

ENDIF

ZDM = TRIM(XX(ZDH)) &&根据字段号求出字段名

SCZD=SCAD+" "+ZDM &&字段名

连接

ENDD0

USE &CJK &&打开指定的成绩库

EDIT ALL & SCZD FOR

&JSTJ &&修改满足条件的记录、字段

四、数据统计

数据统计主要包括:(1)求每个学生的总平均分和专业必修课平均分,(2)求每个学生的总名次和专业必修课名次,(3)对每门课程进行统计分析。

1.求每个学生的总平均分和专业必修课平均分

在学生成绩数据库中,每一门课做为一个字段,由于不同学生组所学的课程不同,对应的成绩库字段名就不同。要使程序具有通用性,则面临两个难题:①如何实现通用的数据库横向统计。②如何标识课程类别。解决这两个问题的关键,一是使用了结构描述文件,二是在程序中定义了两个数组 KC(50)和 ZVK(50),分别用来存放成绩库中所有课程字段名和专业课字段名。因而使程序简练、通用性强。

下面是程序清单:

DIME KC(50),ZVK(50) &&定义两个数组,分别存放所有课名和专业课名

STORE 0 TO KCS,ZVKS&&KCS—课程总门数,ZVKS—专业课门数

USE &CJK_J &&打开结构描述文件

GO 12 &&课程名从第 12 号记录

开始

DO WHILE .NOT. EOF() &&求课程总门数、专业课门数,课名送给两个数组

KCS=KCS+1

KC(KCS)=FIELE NAME

IF 课程类别=1

ZVKS=ZVKS+1

ZVK(ZVKS)=FIELD NAME

ENDIF

SKIP

ENDDO

USE &CJK&&打开成绩库

DO WHILE .NOT. EOR() &&求总平均分、

专业必修课平均分

```
STORE "" TO KCM,ZYKM
```

```
I=1
```

```
DO WHILE I <= KCS  &&产生求所有课总分的算术表达式 == > KCM
```

```
KCM = KCM + TRIM(KC(I)) + "
```

```
I = I + 1
```

```
ENDDO
```

```
I=1
```

```
DO WHILE I <= XVKS  &&产生求专业必修课总分的算术表达式 == > ZVKM
```

```
ZYKM = ZYKM + TRIM(ZYK(I)) + "
```

```
I = I + 1
```

```
ENDDO
```

```
CCD1 = LEN(KCM)
```

```
CCD2 = LEN(ZVKM)
```

```
KCM = SUNS(KCM, 1, CCD1-1)  &&
```

去掉尾部的“+”号

```
ZYKM = SUNS(ZYKM, 1, CCD2-1)
```

```
REPL 总平均分 WITH (& KCM)
```

/KCS &&将结果存入数据库

```
REPL 专必平均分 WITH (& ZYKM)
```

/ZVKS

```
SKIP
```

```
ENDDO
```

2.求每个学生的总名次和专业必修课名次

根据成绩库中已有的总平均分求出每个学生的总名次,并将结果存入数据库,可用下面程序段实现。

```
USE &CJD  &&打开成绩库
```

```
INDEX ON -1★总平均分 TO
```

KKK &&按总平均分降序建立索引

```
MC=1
```

```
DO WHILE .NOT. EOF()
```

```
REPL 总名次 WITH MC
```

```
MC=MC+1
```

```
SKIP
```

```
ENDDO
```

根据成绩库中已有的专业必修课平均分求出每个学生的专业必修课名次,可用类似的方法,程序不再列出。

3.对每门课程进行统计分析

根据成绩库中各门课每个学生的成绩,对课程进行统计分析,要求统计出各门课的:实考人数,该课的平均分,优秀、良好、中等、及格、不及格人数及其所占的比例。统计结果要放到数据库(统计结果库)中,以备查询和打印。统计结果库的结构如下:

Field	Field	Base	Type	Width	Dec
1	课程名称		Character	10	
2	实考人数		Numeric	3	
3	课平均分		Numeric	7	2
4	优秀人数		Numeric	3	
5	优秀率%		Numeric	7	2
6	良好人数		Numeric	3	
7	良好率%		Numeric	7	2
8	中等人数		Numeric	3	
9	中等率%		Numeric	7	2
10	及格人数		Numeric	3	
11	及格率%		Numeric	7	2
12	不及格人数		Numeric	3	
13	不及格率%		Numeric	7	2

对课程进行统计虽然是一个数据库纵向求和(求记录个数)问题,但要对每一门课进行统计,由于不同学生组对应的课程名不同,给程序的通用性带来困难。解决的关键一是用结构描述文件,二是定义了一个二维数组 TT(40,13),每一行表示一门课,每一列表示一个统计项。实现程序如下:(程序清单转第 11 页)

数据库的查询、打印可用通常方法实现。为节省篇幅,这里不再讨论。

```

DIME TT(40,13)  &&定义最多40门课,每门课13项统计结果
USE &CJK.J      &&打开结构描述文件
GO 12           &&课程名从第12号记录开始
KCS=0
DO WHILE (.NOT.EOF) &&求出课程门数KCS,课名送TT(1,1),TT(2,1),...
  KCS=KCS+1
  TT(KCS,1)=FIELD.NAME
  SKIP
ENDDO
USE &CJK        &&打开成绩单
I=1
DO WHILE I=KCS  &&对每一门课进行统计
  LJS=TRIM(TT(1,1))+".0.1"  &&缺考成绩用".0.1"表示
  COUNT ALL FOR &LJS TO TT(1,2)  &&实考人数
  SJX=TT(1,1)
  AVER ALL &SJX FOR &LJS TO TT(1,3)  &&总平均
  LJS=TRIM(TT(1,1))+".90"
  COUNT ALL FOR &LJS TO TT(1,4)      &&优秀人数
  STORE 100*TT(1,4)/TT(1,2) TO TT(1,5)  &&优秀率
  LJS=TRIM(TT(1,1))+".80 .AND. "+TRIM(TT(1,1))+"<.90"
  COUNT ALL FOR &LJS TO TT(1,6)      &&良好人数
  STORE 100*TT(1,6)/TT(1,2) TO TT(1,7)  &&良好率
  LJS=TRIM(TT(1,1))+".70 .AND. "+TRIM(TT(1,1))+"<.80"
  COUNT ALL FOR &LJS TO TT(1,8)      &&中等人数
  STORE 100*TT(1,8)/TT(1,2) TO TT(1,9)  &&中等率
  LJS=TRIM(TT(1,1))+".60 .AND. "+TRIM(TT(1,1))+"<.70"

```

```

COUNT ALL FOR &LJS TO TT(1,10)      &&及格人数
STORE 100*TT(1,10)/TT(1,2) TO TT(1,11)  &&及格率
LJS=TRIM(TT(1,1))+".<60 .AND. "+TRIM(TT(1,1))+">.0.1"
COUNT ALL FOR &LJS TO TT(1,12)      &&不及格人数
STORE 100*TT(1,12)/TT(1,2) TO TT(1,13)  &&不及格率
I=I+1
ENDDO

**将统计结果送统计结果数据库
CJK_T=TRIM(CJK)+".T"  &&定义统计结果库名
USE TJ                &&打开统计结果库样板
COPY STRU TO &CJK.T  &&建立统计结果库结构
USE &CJK T
I=1
DO WHILE I=KCS      &&将统计结果存入数据库
  APPPE BLANK
  REPL 课程名称 WITH TT(1,1),实考人数 WITH TT(1,2)
  REPL 课平均分 WITH TT(1,3),优秀人数 WITH TT(1,4)
  REPL 优秀率% WITH TT(1,5),良好人数 WITH TT(1,6)
  REPL 良好率% WITH TT(1,7),中等人数 WITH TT(1,8)
  REPL 中等率% WITH TT(1,9),及格人数 WITH TT(1,10)
  REPL 及格率% WITH TT(1,11),不及格人数 WITH TT(1,12)
  REPL 不及格率% WITH TT(1,13)
  I=I+1
ENDDO
RETURN

```