

关系数据库管理系统
dBASE III dBASE III + FOXBASE +
出错分析处理及
程序设计技巧

罗伟其 编著

● 近百个出错案例
分析及处理方法

● 几十种程序
设计技巧

中山大学出版社

关系数据库管理系统

dBASE III dBASE III + FOXBASE+

出错分析处理及程序设计技巧

罗伟其 编著

中山大学出版社

· 广州 ·

(粤)新登字 11 号

版权所有 不得翻印

图书在版编目(CIP)数据

关系数据库管理系统 dBASE Ⅲ、dBASE Ⅲ+、FOXBASE+ 出错分析处理及程序设计/
罗伟其编著. —广州:中山大学出版社, 1995. 9

ISBN7-306-01033-6/TP·51

I 关…

II 罗…

III ①计算机应用②数据库③出错分析④程序设计

IV TP3

中山大学出版社出版发行

(广州市新港西路 135 号 510275)

普宁怡昌印刷厂印刷 广东省新华书店经销

787×1092 毫米 16 开本 25.625 印张 60 万字

1995 年 9 月第 1 版 1995 年 9 月第 1 次印刷

印数: 0001—3000 册 定价: 29 元

内 容 提 要

本书以专题讨论的方式对汉字 dBASE III、dBASE III + 和 FOXBASE+ 的出错原因及排除错误方法和程序设计技巧作介绍。全书分为两大部分,第一部分是错误原因分析与错误排除方法,第二部分是程序设计技巧。本书中全部例题的程序均已在 IBM 微型计算机上调试通过。

本书内容丰富、具体、实用。适合于学习和使用汉字 dBASE III、dBASE III + 和 FOXBASE+ 的各类人员。

- 本书采用的汉字 dBASE III 是 1.0 版本,汉字 dBASE III + 也是 1.0 版本,汉字 FOXBASE+ 用的是 2.1 版本。
- 本书的全部例题均在微型计算机上调试通过。dBASE III 和 dBASE III + 在 640KB 内存及以上的 IBM 兼容机都可以运行,FOXBASE+ 则至少需要在 1MB 内存的 IBM 兼容机上才可以运行。

前 言

近几年来,在我国流行最广、应用面最大的微型计算机数据库管理系统是汉字 dBASE III、dBASE III+和 FOXBASE+。它们的共同特点是语句精练、易学易用、数据管理功能强。

然而,dBASE III、dBASE III+和 FOXBASE+的学习和编写应用程序并不是很容易的。它的易学易用的特点只是相对于其它编译式的高级语言的学习和使用而言的。作者在多年的教学和应用软件开发实践中,发现许多人在学习和使用过程中,普遍存在下面两大问题:

1. 许多人在学习时,觉得 dBASE III或 dBASE III+或 FOXBASE+的命令很快就能掌握,但上机操作时,总是出现这样那样的由于命令使用不当引起的错误。由于几乎所有的教科书或用户手册都把主要篇幅放在命令的介绍上,对命令使用的许多细节或应注意的问题往往缺少介绍或介绍甚少,尤其是几乎没有提供如何分析出错原因和排除错误的方法。一旦出现错误,读者往往不知所措,无法解决问题,只好频频求助于任课教师或他人(对自学者来说更是加深了难度),降低了学习的效率。这一问题同样也给许多经验不足的编程者带来程序调试的困难。

2. 当开始学习编写程序时,一些简单的程序编写方法,对大多数学习过 dBASE III或 dBASE III+或 FOXBASE+的人来说,是容易掌握的。但是要想使用它来编写可靠性好、速度快、效率高、通用性强的实用程序,还是不容易的。学习和使用计算机的高级语言不多、编程经验不足的编程人员,尤感困难,从而影响了编程效率。这是由于教科书或用户手册都是注重介绍单一命令的语法,给出的例题大多是孤立地为了介绍某命令而设的,对如何综合使用命令进行程序设计缺少介绍或例题甚少,使得学习者或编程经验不多的编程人员难以通过模仿来培养综合编程的能力和提高了编程的效率。

本书就是为了解决上面的两大问题,根据作者多年的教学和应用软件开发的实际工作经验编写的。全书分为两大篇:第一篇是 dBASE III、dBASE III+和 FOXBASE+的错误原因分析与错误排除方法。这一部分是对比较常见的命令出错作专题讨论。讨论的方法是先给出错误例题,然后对错误原因进行分析,最后给出排除错误的方法。这样的讨论目的,是既为了通过对错误原因的分析,使读者更详细地和真正地掌握好所学的命令,对出现的错误能及时正确地排除,防止或减少以后再出现使用命令的错误,提高上机操作、调试程序的效率。因此本书很适合于各级学习 dBASE III或 dBASE III+或 FOXBASE+的人员使用,可作为这类教材的学习辅助资料。第二篇是 dBASE III、dBASE III+和 FOXBASE+的程序设计技巧。这一部分对编程中经常涉及到的主要技巧作专题介绍。在讨论中力图突出实用性的特点,往往对同一个问题给出了几种不同的程序编写方法,并且对这些方法进行比较、归纳和总结。目的在于提高读者的解决实际问题的能力和效率。每一种方法都给出详细的程序清单,而且这些程序全部在微型计算机上调试通过。虽然书中给出的方法不一定是最佳方法,但都是可行的和实用的方法。作者希望既能帮助经验不多的程序设计者提高编程能力,又能供经验丰富的程序员查阅参考。所以这部分可以视为是学习和使用 dBASE III、dBASE III+或

· FOXBASE+的提高篇。

目前使用的 dBASE III、dBASE III+和 FOXBASE+版本很多,版本之间存在有一些差异,因时间原因作者不可能对所有的版本进行测试。但就本书的内容来说,基本上不受版本的影响,存在不同的仅仅是第一篇的出错信息的显示不同(主要是汉化时翻译不是采用相同的标准引起的)。本书采用的 dBASE III 是 1.0 版本, dBASE III+也是 1.0 版,FOXBASE+用的是 2.1 版。第一部分的各专题的错误信息在给出中文的同时,也给出英文(放在括号内),以便不同版本的对照。

由于作者水平有限,缺点和错误一定是难免的,敬请广大读者和同行专家批评指正。

、 作 者

1994 年 1 月于暨南园

目 录

第一篇 错误原因分析与错误排除方法 (1)

- 1.1 建立数据库 CREATE 命令使用不当的出错应如何处理 (3)
- 1.2 修改数据库文件结构时为什么会引起数据的丢失,如何恢复被
丢失的数据 (6)
- 1.3 为什么在 DIR 命令中与 LIST STRUCTURE 命令中显示出的记录数不
一致 (14)
- 1.4 “Memo 文件不能被打开”(.DBT file cannot be opened)的出错
如何处理 (16)
- 1.5 为什么会出现“文件不存在”(File does not exist)的错误 (17)
- 1.6 “无数据库在使用,请输入数据库的名字:”(No database is in USE,
Enter file name;)的出错如何处理 (19)
- 1.7 与数据库文件有密切相关的文件有哪些,使用时应注意什么 (20)
- 1.8 “命令中出现了不可识别的短语/关键字”(Unrecognized phrase / key-
word in command)的错误如何处理 (21)
- 1.9 “语法错”(Syntax error)的出错如何处理 (23)
- 1.10 如何处理“数据类型不匹配”(Data type mismatch)的错误 (24)
- 1.11 “变量没有找到”(Variable not found)出错原因是什么,如何排除 (29)
- 1.12 “不可识别的命令动词”(Unrecognized command verb)的出错
如何处理 (36)
- 1.13 为什么不能用单个英文字母 A—J 做数据库文件名 (37)
- 1.14 为什么要避免字段名与内存变量名相同的命名做法 (37)
- 1.15 当执行 QUIT 命令退出时,若系统显示错误信息,应如何处理 (40)
- 1.16 “没有足够的内存”(Insufficient memory)的出错如何处理 (41)
- 1.17 “打开的文件太多”(Too many files are open)的出错如何处理 (42)
- 1.18 “不可识别的函数”(Invalid function name)的出错如何处理 (43)
- 1.19 “无效的函数变量”(Invalid function argument)的出错如何处理 (44)
- 1.20 “无效日期”(Invalid date)的出错如何处理 (45)
- 1.21 如何处理“STR() 函数执行错误:超出范围”(* * * Execution error
on STR() ; out of range)的错误 (46)
- 1.22 如何处理“SUBSTR()函数执行错误:开始位置超出范围”(* * * Execution
error on SUBSTR() ; start point out of range)的错误 (47)

1.23	如何处理“STUFF()函数执行错误:字符串太大”(*** Execution error on STUFF(): String too large)的错误	(50)
1.24	如何处理“+”执行错误:被连接的字符串太大”(*** Execution error on +: concatenated string too large)的错误	(52)
1.25	如何处理“REPLICATE()执行错误:字符串太长”(*** Execution error on REPLICATE(): String too large)的错误	(53)
1.26	如何处理“↑”或“*”执行错误:底为负数,指数为小数”(*** Execution error on ↑ or *: negative base, fractional exponent)的错误	(53)
1.27	为什么会出现明明相等的两个数却往往被判断为不等的情况, 如何解决	(54)
1.28	明明两个字符串不相同,为什么比较这两个字符串时的值为真(.T.), 如何解决	(58)
1.29	如何避免数据库文件的数据丢失	(59)
1.30	“遇到了文件结束标志”(End of file encountered)的出错如何处理	(61)
1.31	“遇到了文件开始标志”(Beginning of file encountered)的出错 如何处理	(63)
1.32	使用记忆型(MEMO)字段应注意些什么	(64)
1.33	“在 MEMO 类型字段上的操作无效”(Operation with Memo field invalid) 的出错如何处理	(68)
1.34	为什么会出现“进行排序的记录个数不够”(Not enough records to sort) 的错误	(69)
1.35	“用于分类的磁盘空间不够”(Not enough disk space for SORT)的出错 如何处理	(70)
1.36	“太多的分类关键字段”(Too many sort key fields)的出错如何处理	(71)
1.37	“用于索引的磁盘空间不够”(Not enough disk space for INDEX)的出错 如何处理	(74)
1.38	“索引表达式太大(最大值为 220 个字符)”(Index expression is too big (220 char maximum))的出错如何处理	(74)
1.39	“索引太大(最多 100 个字符)”(Index is too big (100 char maximum) 的出错如何处理	(75)
1.40	“索引文件与数据库不匹配”(Index file does not match database)的出错 如何处理	(75)
1.41	“记录不是索引的”(Record is not in index)的出错如何处理	(76)
1.42	为什么对含有日期型字段的复合表达式作索引关键字建立索引文件时, 日期型字段的值不是按用户所需顺序排列,应如何解决	(77)
1.43	为什么在当前数据库文件存在的记录,用 FIND 或 SEEK 命令 查找不到	(81)
1.44	“记录超出范围”(Record is out of range)的出错如何处理	(86)

1.45	“数据库不能被索引”(Database is not indexed)的出错如何处理	(86)
1.46	“没有 Locate 就用 CONTINUE”(Continue without locate)的出错 如何处理	(88)
1.47	“太多的索引”(Too many indices)的出错如何处理	(89)
1.48	为什么会出现数据库文件中的一些记录应该被修改的而漏修改	(90)
1.49	“无效的索引号”(Invalid index number)的出错如何处理	(99)
1.50	“数值溢出(数据被丢失)”(Numeric overflow(data was lost)是怎样引起 的,应如何处理	(99)
1.51	“文件已经打开”(File in already open)是怎么回事	(103)
1.52	“别名已经在使用”(ALIAS name already in use) 错误原因是什么,应 如何处理	(105)
1.53	“不能删除打开的文件”(Cannot erase a file which is open)的出错 如何处理	(106)
1.54	使用宏替换函数 & 应该注意什么	(106)
1.55	建立和使用报表文件(.FRM)时经常会引起那些错误,应如何处理	(109)
1.56	建立和使用标签文件(.LBL)时经常会引起那些错误,应如何处理	(110)
1.57	建立和使用屏幕格式文件(.FMT)时经常会引起那些错误, 应如何处理	(110)
1.58	“不是 DBASE III 文件”的错误原因是什么,应如何处理	(112)
1.59	“无效的逻辑字段操作”(Operation with logical field invalid)的错误原因 是什么,应如何处理	(120)
1.60	“在 MEMO 类型字段上的操作无效”(Operation with Memo field invalid) 的错误应如何处理	(122)
1.61	“无效运算符”(Invalid Operator)的出错如何处理	(122)
1.62	“非逻辑表达式”(Not a logical expression)的出错如何处理	(123)
1.63	“非数字表达式”(Not a numeric expression)的出错如何处理	(125)
1.64	“未终结的字符串”(Unterminated String)的出错如何处理	(127)
1.65	“字符串太长而不适合”(String too long to fit)的出错如何处理	(127)
1.66	“@ ... SAY / GET 语句的 PICTURE 子句错误”(PICTURE error in GET statement)的错误原因是什么,应如何处理	(128)
1.67	“SAY/GET 位置超出屏幕”(Position is off the screen)的错误原因是什 么,应如何处理	(130)
1.68	“打印机未准备好,重试吗?(Y/N)”(Printer not ready, retry?(Y/N)) 的出错如何处理	(132)
1.69	“打印机未连接或被关闭”(Printer is either not connected or turned off) 的出错如何处理	(133)
1.70	如何解决打印机空走纸问题	(133)
1.71	为什么会出现用打印机输出时表格的最后一行不打印,如何解决	(135)

1.72	为什么使用内存变量的数据输出的表格会不整齐,应如何处理	(136)
1.73	为什么在用@语句设计表格时,掺杂使用 SUM 语句会出现打印的表格中间有空行,如何解决	(139)
1.74	为什么 FOR<条件表达式>与 WHILE<条件表达式>不能等同使用	(143)
1.75	为什么用 APPEND FROM <文件名> FOR <条件> 转移数据时,有时是不对的,如何解决	(144)
1.76	“不能用一个无效的过滤条件来显示,离开 DISPLAY 选择项”(Cannot display using an invalid filter,leave DISPLAY option)的出错如何处理	(150)
1.77	“不能保存无效的过滤条件”(Cannot save an invalid filter) 的出错如何处理	(150)
1.78	“无效的查询文件”(Not a valid QUERY FILE)的出错如何处理	(150)
1.79	“在这种环境下查询无效”(QUERY not valid for this environment)的出错如何处理	(151)
1.80	用 SET FILTER 语句设置的过滤条件为什么有时会不起作用,如何解决	(151)
1.81	“错误的参数个数”(Wrong number of parameters)错误的原因是什么,应如何处理	(154)
1.82	“参数语句未找到”(No parameter statement found)错误的原因是什么,应如何处理	(155)
1.83	出现“文件太长,要丢失一些数据”(File too large, some data may be lost)的错误时怎么办	(156)
1.84	出现“写文件时磁盘空间满:<文件名>”(Disk full when writing file : <文件名>)的错误时怎么办	(157)
1.85	当出现“* * 警告 * * 数据可能会被丢失.确定(Y/N)”(* * WARNING * * Data will probably be lost, Confirm (Y/N))错误信息时应怎么办 ..	(158)
1.86	为什么不能准确执行一个多行命令	(158)
1.87	出现死循环怎么办,如何避免死循环	(159)
1.88	结构命令不配对将引起什么样的错误,如何避免结构命令不配对	(160)
1.89	“必须是一个数组定义”(Must be an array definition)的错误如何处理	(164)
1.90	“数组元素的下标出界”(Subscript out of bounds)的错误如何处理	(165)
1.91	为什么利用 RESTORE FROM 命令把内存文件中的内存变量调入内存时,原有的内存变量不见了	(166)
1.92	用 TOTAL 命令得到的结果不正确是怎么回事,应如何解决	(168)
1.93	为什么有时在使用 .NOT.EOF() 作为循环条件时,一次循环都不执行,如何解决	(172)
1.94	“00% Sorted * * * 堆栈溢出 * * * ”的出错原因是什么,应如何处理	(175)

- 1.95 为什么在条件中使用〈逻辑型变量名〉=.T. 的表达式时会显示出
错信息:“Operator/operand type mismatch” (177)

第二篇 程序设计技巧 (181)

- 2.1 如何合理设计数据库文件 (183)
- 2.2 如何提高数据的输入速度 (186)
- 2.3 如何使用 REPLACE 语句来提高命令的执行效率 (192)
- 2.4 如何检查和排除文件中的重复输入记录 (197)
- 2.5 如何应用宏替换函数 & (201)
- 2.6 如何在编程中应用 SUBSTR() 函数 (212)
- 2.7 如何巧用 DISPLAY 和 LIST 命令 (218)
- 2.8 如何提高索引文件的效率 (223)
- 2.9 怎样使数值为“0”的数据不打印输出 (231)
- 2.10 如何提高统计速度 (235)
- 2.11 如何提高分类求和处理速度 (240)
- 2.12 如何实现多字段的分类求和 (241)
- 2.13 如何实现使用 TOTAL 命令求和时,做到既不发生数据溢出又不浪费
存储空间 (244)
- 2.14 如何用程序方式来修改任一数据库结构 (246)
- 2.15 如何提高数据的检索(查找)速度 (252)
- 2.16 如何巧用 COPY 命令来复制用户所需的数据 (255)
- 2.17 如何认识功能菜单设计的一些技巧 (261)
- 2.18 如何使用多记录数据输入及修改的屏幕表格化光标控制技术 (282)
- 2.19 如何掌握数据库文件的数据管理程序设计 (292)
- 2.20 如何掌握数据库文件查询程序设计 (310)
- 2.21 如何掌握 FOXBASE+ 的数组功能 (327)
- 2.22 如何应用 FOXBASE+ 中自定义函数功能 UDF (331)
- 2.23 如何巧妙设置口令字(PASSWORD)(密码) (335)
- 2.24 如何设计和实现数据库文件的保密 (344)
- 2.25 如何在程序中增加容错的功能 (349)
- 2.26 如何掌握程序设计中使用打印机的技巧 (365)
- 2.27 如何实现与高级语言的数据通信 (378)
- 附录 1 函数一览表 (385)
- 附录 2 命令一览表 (388)
- 参考文献 (398)

第一篇

错误原因分析与错误排除方法

1.1 建立数据库 CREATE 命令使用不当的出错应如何处理

一、字段名无定义

例1 键入下面的命令

CREATE ZGJK

DBASE III 系统屏幕显示信息如下:

	字段名	类型	宽度	小数
1		字符/正文		

现输入字段名: 职工号

若按下面的格式输入职工号字段名, 然后按回车键

	字段名	类型	宽度	小数
1	职工号	字符/正文		

DBASE III 则将输入的字段名立即把“工号”两个汉字去掉, 仅留下“职”, 即下面的显示信息:

	字段名	类型	宽度	小数
1	职	字符/正文		

原因 这类错误出错的原因是对 CREATE 命令中字段名的要求了解不深引起的。CREATE 命令对字段名的要求是:

- (1) 字段名必须以字母或汉字开头, 其余可以是汉字、字母、数字或下划线。
- (2) 字段名的字符之间不能有空格。
- (3) 字段名在同一个数据库结构中是唯一的。

根据字段名定义的要求, 我们就可以知道例1的出错是因字段名中含有空格, 从出现空格的地方开始到最后字符全部不接受, 所以不能定义该字段名。出现这种情况时不同的系统在屏幕上将显示不同的出错信息显示:

DBASE III 系统: 无出错信息提示。

DBASE III+ 系统: 不会出现这种情况, 一旦出现插入空格, 立即响铃。

FOXBASE+ 系统: 在屏幕的下方显示

Illegal field name.

Press any key to continue...

例2 在例1的基础上继续输入字段名。设例1 ZGJK.DBF 的结构已经定义了五个字段如下:

	字段名	类型	宽度	小数
1	职工号	字符/正文	8	
2	姓名	字符/正文	8	
3	出生日期	日期型	8	
4	基本工资	数字型	6	2
5	性别	字符/正文	2	

现设想把基本工资字段放在性别的后面,操作步骤如下:

① 在第 6 个字段上打入基本工资。

字段名	类型	宽度	小数
1 职工号	字符/正文	8	
2 姓名	字符/正文	8	
3 出生日期	日期型	8	
4 基本工资	数字型	6	2
5 性别	字符/正文	2	
6 基本工资	字符/正文		

② 按回车键,此时光标移回第 6 个字段名的第一个字符位置,无法移至类型栏目。

原因 是因为出现了同名字段。已有一个基本工资字段,再定义就出错了。而出错的提示因使用不同的系统而不同。

DBASE III: 屏幕上不会提示字段名重复。

DBASE III + E 和 FOXBASE+ : 在屏幕的下方显示

Fields name is already in use.

Press any key to continue...

排错办法 对例 1 只要重新输入正确的不含空格的字段名即可。

对例 2 中出现字段名重复的,请先删除已定义的重复的字段名,然后在适当的字段位置上重新定义。本例的解决办法的操作步骤是:

- (1) 把光标移至第 4 个字段,即基本工资字段;
- (2) 按 Ctrl+U 将基本工资字段删除;
- (3) 把光标移至性别字段下面的新字段位置;
- (4) 定义基本工资字段。

例 2 的错误是在用户对一些字段名的顺序更改时最容易引起的。为了避免这种错误,用户应该牢记一条原则:先删除一个已定义的字段名,然后在适当的位置插入刚删除的字段名。即先删除后插入的原则。

例 3 字段名前面出现空格。这种情况只有在 FOXBASE+ 系统才会出现。例如:

字段名	类型	宽度	小数
1 职工号	字符/正文	8	
2 姓名	字符/正文	8	
3 出生日期	日期型	8	
4 性别	字符/正文	2	
5 基本工资	数值型	6	2
6 职称	字符/正文		

这时屏幕下方显示:

Illegal field name.

Press any key to continue...

二、字段的宽度出错

例 1 设正在定义的数据库文件结构如下:

字段名	类型	宽度	小数
1 职工号	字符/正文	8	
2 姓名	字符/正文	8	
3 出生日期	日期型	8	
4 性别	字符/正文	2	
5 基本工资	数值型	6	2
6 职称	字符/正文	288	

此时出现的错误是：

DBASE III 系统：无出错信息，但光标不能移出宽度栏

DBASE III + 和 FOXBASE+ 系统：在屏幕的下方显示

Illegal data length.

Press any key to continue...

又设正在定义的数据库文件结构如下：

字段名	类型	宽度	小数
1 职工号	字符/正文	8	
2 姓名	字符/正文	8	
3 出生日期	日期型	8	
4 性别	字符/正文	2	
5 基本工资	数值型	3	2

此时出现的错误是：

DBASE III 系统：无出错信息，但光标不能移出宽度栏。

DBASE III + 系统：在屏幕的下方显示

Illegal Numeric data length.

Press any key to continue...

FOXBASE+ 系统：在屏幕的下方显示

Illegal Decimal length.

Press any key to continue...

出现这类错误的原因是对字段宽度的规定了解不够或错按键引起的。虽然数据类型有五种，但真正需要用户定义宽度的只有字符型和数值型两种，其余类型的宽度由系统自动设定。因此我们记住下面两个有关字符型和数值型宽度的规定就可以了。

(1) 字符型最多为 254 个字符；

(2) 数值型的最大宽度包括小数点、负号在内为 19 位，并限制小数部分宽度最多 15 位，而且小数位的宽度至少比总宽度少 2。

三、屏幕上出现：文件已经存在，重写吗(Y/N)

原因 给出的数据库文件名在当前磁盘已经存在。这是因为在同一个磁盘的同一个子目录里不允许有同名的文件。

若在出现这种提示时，用户键入 Y，将破坏原有的整个数据库文件，数据全部丢失，无法恢复。所以使用时要特别注意。