

## 如何提高微机数据库的 程序运行速度

牛红兵

在较大型的数据库应用系统中,程序的运行速度是系统的关键问题。如果没有较快的运行速度,将会导致整个系统的失败。究竟怎样才能提高程序的运行速度呢?

当前,我国使用的诸种数据库管理系统中 dBASE 和 FoxBASE PLUS 占有重要的地位。对于 dBASE 和 FoxBASE PLUS,要提高其程序运行速度,就需要动词准确、条件明晰、优化算法、适宜环境。具体做法如下:

### 一、提高系统建库速度

提高数据库的建库速度是开发数据库管理系统的重要问题。对于中文信息的录入工作来说,不仅先用优秀的汉字操作系统,配备熟练的录入人员,可以提高建库速度,还应当挖掘数据库管理系统的潜力。在程序设计方面下功夫。

#### 1.使用 SET CARRY ON 命令

该命令特点是可免去输入相邻两个记录内容相同的数据。键入该命令后,执行 APPEND 命令时将携带前面记录中的数据到新的记录中,输入记录时,若与前一个记录段内容相同的就只需回车,或做少量的修改。

#### 2.使用 COPY 和 JOIN 命令

在一个大的程序中,库文件可能会很多,且一些库文件中字段相同。这样就可用 COPY TO <库文件名> Structure [FIELDS <字段名>], COPY TO <库文件名> Structure Extend 和 Create [<库文件名>] FROM <结构描述文件名>, JOIN WITH <别名> TO <文件名> FOR <条件> [FIELDS <字段名表>]等命令,利用已有的库文件建立新文件,节约建库时间。

#### 3.字段一部分内容完全相同的快速建库法

数据库中,往往在一定范围内,某一字段的内容完全相同。在输入数据时,先不输入这些有规律的内容,待其他字段内容输完后,再用 GO 和 REPLACE 命令整理这

些字段内容,符合条件的字段的内容就可以由计算机添补完整。

#### 4.APPEND 命令的使用

当数据库每个记录的字段内容多,而重复率较大的字段较少,或者一部分字段内容均为有重复的汉字时,可先建立一个只含部分字段的结构文件,对其输入结构后,用 APPEND FROM <库结构文件名> 命令,将这部分数据加到已打开的含有全部字段结构的数据库中,最后 GO 和 REPLACE 命令将其他有重复的字段内容加到库中,完成建库工作。

#### 5.从实际出发设置功能键

在一个程序使用频率高的命令,不必千篇一律地利用菜单,而可用命令 SET Function <数值表达式> TO <字符表达式> 设置 F2—F10 功能键的功能,以达到一键输入的目的。注意功能键 F1 的定义是不能改变的。

#### 6.字符串提示输入法

数据库中一些字段的内容是汉字或字母,且其种类是不太多的情况下,将数据以代码的形式存储到代码库中,要输入其中内容时,将其调出显示在屏幕上,选择相应的代码即将实际内容输入到其他库中,缩短键盘输入时间。

#### 7.减少清屏读盘操作

由于利用系统的输入输出格式或自编的屏幕格式时,每输入一个记录都要清屏、读盘、显示屏幕格式和提示信息,增加了循环的频度,浪费了很多时间,应将屏幕格式文件作为过程文件,在输入程序中调用。当数据输入时只显示一次屏幕格式和提示信息,而在连续输入过程中屏幕格式和提示信息将不再发生变化,只有输入的内容反复更新。

## 二、在执行程序中准确应用命令

数据库管理系统中,各个动词都有各自的特点,在程序设计中要尽量用其优点,避其缺点。

#### 1.提高记录排序速度

对于记录排序,用 SORT 建立一个排序文件,要比用 INDEX 命令建立索引文件慢得多。在程序中可用:

```
INDEX ON <字段名> TO <文件名>
SET INDEX TO <文件名>
```

COPY TO <排序文件名>

三个命令代替 SORT 命令,这三个命令执行的总速度相当于 SORT 命令的  $1/3 \sim 1/2$ 。

## 2.提高检索速度

在数据库管理应用系统中,检索是不可少的基本功能。LOCATE 命令是模糊检索命令,它将指针定位于满足条件的第一条记录,不需要索引文件的配合,与 CONTINUE 命令联用,可实现批量数据的检索。FIND 和 SEEK 命令是快速检索命令,指针定位于满足条件的第一条记录,但需要索引文件的配合使用,也能实现批量检索,检索速度要比 LOCATE 快得多。若 FIND 或 SEEK 与 dBASE 或 FoxBASE PLUS 中提供的范围选择项 REST 和 FOUND()函数联用,能达到目前 dBASE 类数据库管理系统中最快的检索速度,通用程序如下:

```
USE <文件名> INDEX <索引文件名>
FIND <字符串> / <数值> 或 SEEK <表达式>
IF FOUND()
DISPLAY REST WHILE <条件>
ENDIF
USE
RETURN
```

## 3.提高记录的统计速度

在 dBASE 类数据库管理系统中采用 COUNT 或 SUM 命令进行大量数据统计汇总时速度很慢。在一个上千人的数据管理库中,统计一遍几乎需要 40 分钟。如果对数据库进行索引后,用 FIND 命令帮助进行累计,则所用时间为原来的  $1/6$  左右。

## 4.提高批量更新速度

REPLACE 命令的不同运用方法对程序执行速度会产生较大的影响。在循环中根据条件判断结果来执行 REPLACE 命令将比执行 REPLACE ALL 多耗时 2~4 倍。FOR 语句对 REPLACE 命令的执行速度有明显影响,满足条件的记录越少,使用 FOR 语句就越有价值。在程序设计中要对处理对象进行合理估计,如果所有记录都满足条件,当然就不使用 FOR 语句;如果满足条件的记录数低于 90%,可以加上 FOR 语句。满足条件的记录越少,FOR 语句的加速优势就越明显。

## 5.合理组成过程文件

利用过程文件可以减少磁盘操作次数,缩短扫描文件目录的时间。但过程文件调入内存,又要增大程序的渐近空间复杂度。这就需要从实际出发,在内存容量不是很充裕的情况下,将互相调用较多的模块组成过程文件,而联系不大的则不必凑合在一起。

## 6.提高无相同字段两库的记录更新速度

UPDATE 命令在 dBASE III 和 FoxBASE PLUS 中用于更新库记录的,UPDATE 不仅要求有相同的字段名,建立索引文件,而且更新速度较慢。快速更新法是首先打开数据库甲,对甲数据库中的字段 AA 进行索引,然后转到工作区 B,打开数据库乙,再在循环块中用 LOCATE 和 REPLACE 命令,将数据库乙的字段 BB 与数据库甲中字段 AA 相比较,若相同,则更新。

## 7.提高数据库记录的删除效率

当数据库中记录较多,而且有相应索引文件打开,用 DELETE 和 PACK 命令来删除记录和重新整理数据库时,将耗费许多时间。好的程序应该这样:若同时也有要向数据库中追加一部分记录,则可以不用 DELETE 和 PACK 去真正地物理删除,而可将要追加的记录直接写到原来要删除的记录上,若追加的记录多于要删除的记录,再将余下的记录加到数据库的末尾。

以上这种方法对于索引后的数据库记录的删除也是可行的。首先定位记录指针到要删除的记录上,然后使用 REPLACE 命令,将该记录的全部字段的内容替换为空格,使其不具有实际意义。当这种方法删除的记录到一定数量后,再真正物理删除和重新整理数据库及索引文件,以减少反复读盘次数。

## 8.编译 PRG 文件

编译 dBASE 可以对 PRG 文件编译,FoxBASE PLUS 则对 PRG 文件进行伪编译,将源程序转换成内部代码,可使运行时间成为 PRG 文件的  $1/10 \sim 1/5$ 。经过(伪)编译的文件长度一般压缩到原来的  $1/3 \sim 1/2$ ,也提高了程序执行时的装载速度。

## 9.尽量采用多工作区操作

dBASE 和 FoxBASE PLUS 都允许同时打开十个工作区,尽量利用这一优势进行多区操作,避免使用 USE 命令频繁地打开,关闭数据库和减少读盘时间,从而提高运行速度。

## 10.提高运算速度