

一种基于 xBASE 的通用信息系统设计与实现

陈永红 (北京理工大学管理学院)

一、引言

以关系数据库系统 xBASE 为基础的信息系统得到了越来越广泛的开发和应用,但目前这类系统的开发都是由编程人员针对特定的应用需求编写特定的应用程序,这种程序往往是就事论事,直接操作管理对象的记录数据库。用这种方法程序编制简单,最容易实现应用所要求的功能,但由于程序只能对特定数据库的规定结构进行处理,程序与数据相互依赖,使系统的可维护性、灵活性和适应性都很差。一方面,当原管理对象的变化导致数据库结构发生变化时,必须对源程序进行修改;另一方面,当有新的管理对象时,尽管应用功能类似,也要重新编写程序,因此造成大量类似程序的重复编写。

由于信息系统的应用是一个发展的过程,随着用户对系统认识的提高和使用的熟练,用户将不断提出改进意见或激发出新的需求,同时系统的应用环境也在不断发生变化;另一方面,大多数信息系统都是对信息进行建库管理,尽管管理的实体对象有所不同,但管理功能往往类似,多是以信息的增加、修改、删除、查询、统计和打印输出等为主。以上几点都要求系统具有很强的通用性、可维护性和适应性,以满足用户的应用要求。

本文提出了一种以数据字典为基础的通用系统实现方法,以提高基于 xBASE 的系统性能。

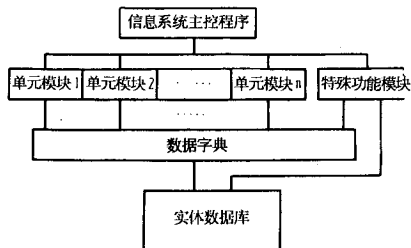
二、通用系统的基本设计思想

信息系统在可维护性和灵活性方面的不良性能主要来源于数据功能操作对实体数据对象的过分依赖。在数据库系统的三级模式体系结构中,是通过引入模式后形成的外模式/模式、模式/内模式二级映象来实现数据的逻辑独立性和物理独立性的,因此按照类似的思路,也应在信息系统的功能操作与操作对象之间引入一个中间环节,将实体数据库的描述信息抽象出来,程序功能并不

直接操作实体数据库,而是通过对这一中间环节的操作,间接地实现对实体数据库的操作。这个中间环节中存着对实体数据库的结构、规范和完整性约束等描述信息,即关于数据的数据——数据字典。基于这种思想,系统中将存在两类数据库文件,一类是实体数据库文件,其内容是现实管理对象的数据记录;另一类是数据字典库文件,其内容是对实体数据库的描述性信息。虽然实体数据库文件的结构各异,但数据字典库文件的结构是相同的,系统中的每个功能模块将通过对数据字典库的操作,来实现对实体数据库文件的操作。并且只要操作不同的数据字典库,就可以实现对不同的实体数据库文件进行操作,从而实现了一套程序可应用于多个不同的管理对象。

对基于 xBASE 的信息系统的功能分析可以看出,系统的基本功能可划分为若干个单元功能模块,系统中大部分应用可通过这些单元功能模块组合实现,单元功能模块划分的程度和大小直接影响系统的通用性和灵活性。当然对于特定系统的不能用单元模块组合的特殊功能要求,仍应通过编写特殊的专用功能模块来实现。

根据上述思想,基于数据字典的系统应具有如图所示的结构:



目前被广泛应用的 xBASE 系列数据库系统,由于对关系操作和关系完整性的支持不够充分,并且不支持数据字典管理功能,因此要完全通过编程来实现基于数据字典的系统。

三、数据字典库的基本结构

对于每一个实体数据库文件,建立一个与之对应的数据字典库文件,数据字典库中的所有字段用于说明实体数据库中字段的各种特征,数据字典库中的每一条记录对应实体数据库中的一个字段,即实体数据库中有多少个字段,数据字典库中就有多少条记录,从而使实体数据库的所有结构信息都存在与之对应的数据字典库中。并通过一定的命名规则,使实体数据库与数据字典库相对应,以便于程序能够自动进行识别和对应,本系统中的命名规则是将实体数据库文件名后再加字符“DD”作为相应的数据字典库文件名。

数据字典库具有如下的数据库结构:

序号	字段变量名	字段类型	字段宽度	字段中文含义
1	FIELD NAME	C	10	字段变量名
2	FIELD TYPE	C	1	字段类型
3	FIELD LEN	N	6	字段长度
4	FIELD DEC	N	3	字段小数位数
5	FIELD MEAN	C	16	字段中文含义
6	FIELD FLAG	C	4	字段标志
7	FIELD DBF	C	8	字段相关的数据库名
8	FIELD INQ	C	1	字段是否可查询
9	FIELD REL	C	40	字段编辑格式限制

数据字典库的前4个字段与用 COPY STRUCTURE EXTENDED 语句生成的数据库文件的结构库相同,其它字段为实体数据库的其他特征信息,其含义和取值说明如下:

1. 字段变量名

命名规则按所使用的语言如 Foxpro 的要求,一般由字母、数字和下划线等字符构成,长度以 10 位为限;

2. 字段类型

取所使用的语言如 Foxpro 所支持的类型,以 1 位大写字母表示,如 C、N、D、L、M 分别表示字符、数值、日期、逻辑和备注型;

3. 字段长度

字段变量所允许的字符总个数;

4. 字段小数位

数值型字段变量小数点右边的位数,如不是数值型字

段,可保持为空;

5. 字段中文含义

当字段需要屏幕显示或打印输出时,用此项作为数据的内容提示或表头,应为用户易于理解的应用性词汇;

6. 字段标志

是程序通过数据字典操作实体数据库的一个重要分支判断标记,有如下的取值含义:

(1)“ZM”:表示该字段是实体数据库的主码,一个数据字典库中只能有一个主码;

(2)“ZHM”:说明实体数据库要通过组合码进行标识,该字段是实体数据库的组合码之一

(3)“WLM”:表示该字段是实体数据库的外来码,其来源实体数据库名由下面的第 7 项“字段相关的数据库名”说明;

(4)“0”:表示该字段是直接由键盘输入的;

(5)“1”:表示该字段的值是从一组规定的应用词汇中选择出来的,并将直接将选择的词汇作为字段的值,如“职称”字段的取值可从“技术员、助理工程师、工程师、高级工程师”中选择出来,直接取值为“工程师”,从而减少用户的汉字输入,提高输入的准确性。这一组规定的应用词汇存在一个代码库中,库名由下面的第 7 项“字段相关的数据库名”说明;

(6)“11”:表示该字段的值是从一组规定的应用词汇中选择出来的,但字段中存的并不是选择出来的词汇,而是该词汇所对应的代码,如以 01,02,03,04 分别表示“技术员、助理工程师、工程师、高级工程师”,字段的取值为 03,从而减少存储空间占用,且便于编码标准化。同样也要有一个代码库,其中不仅有词汇的名称,还应有相对应的编码;

(7)“2”:表示该字段(必须为数值型)是通过计算得出的,计算公式可借助下面的第 9 项“字段编辑格式限制”写出;

(8)“3”:表示该字段是本实体数据库的一个虚拟字段,记载的是与本实体数据库存在一对多关系的附属实体数据库的相关内容。例如,一个职工的档案有多条履历信息,则档案记录与履历记录是一对多的关系,则在实体数据库人事档案对应的数据字典中有一个虚拟字段“个人履历”,其标志为“3”,表示有一个与人事档案实体库相关的附属实体库——个人履历库,相应也有一个附属数据字典。附属实体数据库和附属数据字典的库名分别由第 1 项“字段变量名”和下面的第 7 项“字段相关的数据库名”指出;

(9)“4”:表示该字段也是本实体数据库的一个虚拟字段,记载的是对本实体数据库进行的一种特殊处理,该特殊处理是用一个专用程序实现,程序名由第 1 项“字段变量

名”标明,用户可在此挂接单元功能模块之外的特殊处理程序;

7. 字段相关的数据库名

本字段的取值含义随第6项“字段标志”字段取值的不同而有所不同,详见“字段标志”的有关说明;

8. 字段是否可查询

表示在对实体数据库进行条件组合查询时,是否可对该字段设置查询条件,取“1”为可查询,取“0”为不可查询,如备注字段、虚拟字段和其它应用上有特殊要求的字段就可设为不可查询字段。

9. 字段编辑格式限制

一般情况下表示对该字段进行编辑时所应满足的格式限制,可按 FoxPRO 的 SAY/GET 语句中 picture、function、range、valid 等子句的要求书写格式说明;当第6项“字段标志”为“2”时(即字段通过计算求得),此项用于书写计算公式。

这些是数据字典的基本结构构成,在系统开发实践中,可根据实际需要不断进行扩充和完善。

四、单元功能模块的实现

系统中通用性单元功能模块都将按照“通过对数据字典间接操作实体数据库”的原则进行实现,在基本程序中不出现实体数据库名和具体的字段名。数据库名和数据字典名是在主控程序调用单元功能模块时,用“DO < 程序名 > WITH < 参数 >”的方式进行参数传递的,而所操作的具体字段名是从数据字典库的记录中取出来的。

1. 数据编辑模块

将对实体数据库记录进行追加和修改的功能统一在数据编辑模块中。首先在数据字典中找到主码或组合码,按其中文含义提示用户输入要进行编辑的数据记录的主码或组合码,按用户的输入内容在实体数据库中进行记录定位,如已有该记录,则对其进行修改操作,如无有记录,则提示用户进行记录追加,然后转化为对追加的空记录进行修改;按照数据字典中的结构描述信息,动态地计算和构造数据编辑的屏幕格式,将要编辑的数据记录的所有非虚拟字段显示在所构造的屏幕格式中,如数据字典中记录很多,使编辑的数据量超过一屏,应提供翻页操作;然后对数据字典中的所有非虚拟字段记录(即对实体数据库的每个字段)进行循环,根据其“字段标志”的不同设置主码检查、外来码参照、键盘输入、代码选择、按公式计算等多个

分支,进行不同的编辑操作;待所有非虚拟字段编辑完毕后,在屏幕上开窗口列出所有虚拟字段所指的一对多处理或特殊处理,由用户选择操作;对特殊处理只要直接调用数据字典中所指的程序即可;一对多处理,按本数据字典所指,打开附属数据字典和附属实体数据库,按主码或组合码进行记录定位,追加或修改记录,按与一对一实体库类似的操作构造屏幕格式,进行编辑操作,并按一对多的原则,提供针对主实体数据库的一条记录编辑多条附属实体数据库的记录。

2. 数据记录删除模块

在数据字典中定位主码或组合码,提示用户输入要删除数据记录的主码或组合码;按用户的输入内容在实体数据库中进行记录定位,如无相应记录,则提示用户并返回,如找到相应记录,按照数据字典中的结构描述信息,动态地计算和构造数据编辑的屏幕格式,将要删除的数据记录显示在所构造的屏幕格式中,请用户进行确认;经用户确认后,在实体数据库中对该记录执行删除操作;对数据字典中的每一个具有一对多关系的虚拟字段,打开附属数据字典,按关联主码或组合码在附属实体数据库中删除与之对应的多条记录。

3. 组合条件查询模块

在屏幕上开窗口列出数据字典中的所有可查询字段,由用户用箭头键选择要设置条件的字段;根据所选字段的类型,由用户选择条件运算符,对字符型字段应提供包含、等于、不等于等运算符,对数值型和日期型字段应提供=、< >、>、>=、<、<=等运算符;根据“字段标志”的设置,由用户输入或选择具体条件值;一个条件输入完以后,可由用户选择继续输入其它条件的逻辑关系运算符(与、或、非等)或开始查询,在用户选择了逻辑关系运算符以后,重复上述过程,选择其他字段进行查询条件设置;按数据字典的设置将用户输入的所有条件组合,在实体数据库中进行查询操作,计算并显示满足条件的记录个数,将满足条件的所有记录的关键信息在屏幕上逐行列出,用户可用上下箭头和翻页键进行浏览查询;如用户想进一步了解某条记录的详细情况,只要在光标所在记录上按一个功能键,系统按照数据字典中的记录情况,动态地计算和构造数据记录的屏幕格式,将相应的数据记录的所有非虚拟字段显示在所构造的屏幕格式中,并开窗口将数据字典中的所有虚拟字段所指的一对多处理或特殊处理显示出来,可由用户进行选择操作,或执行特殊程序观察结果,或查询

相应记录的一对多数据记录;从一个记录的细节查询退出后,又返回到浏览查询状态,用户可继续进行其他查询,或打印查询结果,或退出查询。

基于类似的思路,还可以编制其它一些单元功能模块,如基本统计功能模块可应用数据库系统提供的统计命令,按组合条件查询模块中类似的条件设置方式,对数据字典中的数值型字段进行条件统计和汇总计算;打印输出模块可应用数据字典对实体数据库中字段和记录进行选择打印,打印格式完全可以从数据字典的结构信息中计算和生成出来;代码维护模块可对数据字典中所有“字段标志”为“1”或“11”所涉及到的代码库的内容进行增、删、改操作。还有许多单元功能模块可以基于数据字典编写出来,这里就不一一赘述了。

五、系统应用与发展

1. 系统的建立

应用基于数据字典的方法构造信息系统原型的步骤为:

(1)根据所进行的系统分析与设计结果,编写管理对象的数据字典;

(2)应用数据字典中非虚拟字段记录的前4个字段,以“COPY <数据库文件名> FROM <结构库>”的命令方式生成实体数据库结构;

(3)针对数据字典中包含一对多关系的虚拟字段记录,编写附属数据字典,并生成相应的附属实体数据库结构;

(4)用“DO <单元功能模块> WITH <[数据字典名],[实体数据库名]>”命令方式调试信息系统的中通用性单元功能模块;

(5)编制系统所需的不能以通用单元功能模块实现的特殊功能程序;

(6)编写系统主控程序:将所有菜单构成项和程序模块调用方式存入一个菜单项目字典库,主控程序通过操作这个菜单项目字典库来实现系统控制;也可直接用数据库系统的菜单生成器来形成主控程序。

2. 系统的维护与扩展

当系统原型建立起来以后,可根据用户的要求,在不改动源程序的情况下对系统进行下列方式的维护和功能扩充:

(1)数据结构调整:可根据用户的数据库结构调整要求,直接调整数据字典库,再以数据字典库为基础重新生成新的实体数据库结构,从而完成数据结构的修改。

(2)屏幕显示和打印输出格式的调整:通过调整数据字典中记录的顺序和相应的“字段中文含义”实现格式的调整。

(3)特殊处理功能的增加:编写特殊处理专用程序模块,直接挂接在主控程序上。

(4)数据代码的维护:通过代码维护功能模块,可选择对数据字典中所有“字段标志”为“1”或“11”的字段所涉及到的代码库内容进行增、删、改。

(5)新管理对象的增加:编制新的数据字典,并依此生成新的对象实体数据库,在主控程序中调用带新参数的单元功能模块。

3. 系统功能的进一步发展

随着系统中单元模块功能的改善以及模块个数的增加,系统的通用性、灵活性和可维护性将会得到不断提高。

从系统的建立和维护过程可以看到,要求系统开发和维护人员必须对系统的数据字典结构和单元功能模块相当理解和熟悉,特别是“字段标志”、字段之间的对应关系、虚拟字段等的设置,一般人员很难直接应用。为此,应在系统的数据字典和单元功能模块基础之上,编写一个用于系统生成的外壳程序,通过一个友好的用户界面,专门用于数据字典、单元模块的控制和调用关系的建立和维护。这样,系统开发人员通过这个界面来建立和维护信息系统,而最终用户通过系统的主控菜单界面应用信息系统,从而实现了一个基于数据字典的信息系统生成器。

笔者用小型关系数据库系统 FoxPRO for DOS V2.5 进行了系统程序编制,完全实现了上述基于数据字典的通用信息系统的思想和功能,并以此系统为工具进行了若干个信息系统的建立和维护工作。基于数据字典的信息系统与传统方法生成的信息系统相比,系统程序的技术复杂性和编写难度都有所增加,但使得系统的通用性、灵活性、可维护性等性能都有了很大提高,所付出的代价是由于通过数据字典间接地操作实体数据库,使程序的运行效率降低,对运行环境和空间的要求较高。但随着信息系统需求的不断上升,计算机系统性能价格比的不断提高,这些将不再成为问题。