

INFORMIX

关系数据库管理系统

程瑜荣 赵奎元等 编译

清华大学出版社

INFORMIX

关系数据库管理系统

程瑜荣 赵奎元 金声 沈金发 译

清华大学出版社

内 容 简 介

本书为“INFORMIX”关系数据库管理系统的用户手册，主要根据《INFORMIX relational database system for UNIX》(1983年3.00版)翻译而成。内容包括：导论、DBBUILD 数据描述语言编译程序、面向屏幕的数据输入程序、INFORMER 查询语言、DBSTATUS 数据库监控程序、C 程序设计语言应用语言库、ACE 关系报表书写程序、PERFORM 用户屏幕产生和管理处理软件包、C-ISAM 索引顺序存取法。

可供计算机专业的师生、应用计算机的工程技术人员及一般高档微机用户阅读、参考。

JB628/17

INFORMIX 关系数据库管理系统

程瑜荣 赵奎元等 译

☆

清华大学出版社出版

(北京清华园)

北京市昌平环球印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

☆

开本：787×1092 $\frac{1}{16}$ 印张：17 $\frac{3}{4}$ 字数：443 千字

1987年2月第1版 1987年2月第1次印刷

印数：0001~7000

统一书号：15235·263 定价：3.65元

目 录

第一章 导论	1
I、什么是数据库管理系统.....	1
II、什么是关系数据库系统.....	1
III、INFORMIX 概述	3
IV、系统一般命令.....	8
V、附录 I ——与 COBOL, C, BASIC 或其它程序设计语言编写的 C-ISAM 文件结合的问题.....	8
VI、附录 I ——操作系统中的 INFORMIX	9
VII、附录 II ——INFORMIX 保留字	11
第二章 DBBUILD 数据描述语言编译程序	14
I、引言.....	14
II、DBBUILD 语言的定义	15
III、运行实例.....	20
IV、数据库结构的更改.....	21
V、附录——出错信息.....	22
第三章 ENTER 1 数据输入程序	26
I、引言.....	26
II、过程介绍.....	27
III、重要特性.....	28
IV、命令格式及使用.....	29
V、附录——出错信息及解释.....	39
第四章 ENTER 2 面向屏幕的数据输入程序	44
I、引言.....	44
II、对话描述.....	45
III、重要特性.....	47
IV、命令及其用法.....	48
V、附录——出错信息及注释.....	55
第五章 INFORMER 查询语言	57
I、引言.....	57
II、语法和结构.....	58
III、命令的语法和定义.....	59
IV、语法概要.....	70
V、高级特性.....	71
VI、附录——出错信息.....	79

第六章 DBSTATUS 数据库监控程序	84
I、引言	84
II、重要特点	85
III、DBSTATUS 命令定义	86
IV、硬件和软件发生故障后的恢复	96
V、修改模式	97
VI、附录——出错信息及解释	97
第七章 C 程序设计语言应用语言库	102
I、引言	102
II、INFORMIX 应用语言库例行程序	102
III、应用语言库接口	111
IV、应用语言库程序举例	111
V、附录——出错信息及解释	116
第八章 ACE 关系报表书写程序	120
I、引言	120
II、ACE 语言的描述	124
III、怎样编制和运行 ACE 程序	137
IV、高级特征的描述	140
V、ACE 语言的完整定义	148
VI、例子	163
VII、适用于专家的 ACE 功能	185
VIII、错误信息	193
第九章 PERFORM 用户屏幕产生和管理处理软件包	197
I、引言	197
II、用户屏幕格式的定义	198
III、现有数据的浏览	207
IV、数据库内容的改变	211
V、高级功能	212
VI、实例	221
VII、附录——出错信息表	224
第十章 C-ISAM 索引顺序存取法	231
I、导言	231
II、C-ISAM 命令要览	231
III、C-ISAM 命令格式及其说明	233
IV、数据类型	241
V、索引的定义与使用	242
VI、索引压缩	246
VII、锁定	248
VIII、C-ISAM 文件的建立和使用	251

Ⅸ、标题文件	267
X、状态字节和出错代码	270
Ⅺ、文件格式	272
附录 B 树检验器	275

第一章 导 论

I. 什么是数据库管理系统

从计算机用户的观点来看,数据库管理系统为存贮信息提供了简便、交互式的存取方法。此外,系统按照预先定义的、便于生成打印报表的格式处理这些信息。

对于面向计算机化的应用来讲,每个计算机用户都有各自的要求。有关人事、仓库管理、销售管理等信息存贮检索软件系统,按其所存贮的信息而各不相同。即使几个用户组存贮的信息相同,但为适应各自不同的需要,每个组往往以不同的方式来组织这些信息。另外,一个公司为了提供一既定功能,希望存贮的信息很少与其它公司需要的信息相同。

然而,所有用于存贮和检索信息的软件系统都具有很多共同之处。这些系统都采用交互式搜集信息的方式,搜集需要添加到计算机磁盘存贮系统中的数据库中的信息。一旦需要,就可以按照各种可读格式调用这些信息。同时,要求作好删除无效信息的准备,并且随时可在计算机内修改这些信息。

这类系统的最普通的例子就是飞机售票系统,电话公司管理的电话查号系统。要建立这样的两个系统,费用是很高的,但它们能够高效地满足用户的需要。

总之,所有事务工作都是管理信息。一旦一个信息实体变得太大,以致不宜再由人工处理,或者对信息查询要求有一定的速度或一定的复杂性,这时只能由机器来进行处理,这就应该将信息进行计算机化。

以往要确立一个执行所要求的数据管理的软件系统,往往行不通,而且价格昂贵。然而,如果将通用数据库管理系统作为软件的基础来建立应用系统,这就不仅使应用系统的价格显著下降,而且也会缩短新系统投入使用的时间。对于所有面向数据的应用系统来说,不必重复编写可共用的大型软件,而可以购买现成的。

根据同样的理由,用数据库管理系统来建立应用系统是十分便利的。这样的软件容易按用户的要求,进行相应的变更。

II. 什么是关系数据库系统

通常将存贮在计算机中的信息分成若干组。下一页中给出了数据分组的实际例子。在例子中,假设了某零售商店正使用的一个数据库。数据库中包含商店的顾客及顾客订购的商品信息。

这些信息被划分成两组,一般称作“文件”。CUSTOMERS(顾客)文件包括顾客的姓名(cname),地址(address),结存金额(balance)。ORDERS(订单)文件中包括顾客订的货,对于每一项已填写的记录来说,记录了顾客的姓名(online),订购的项目(item)及订购的数量(quantity)。

这种分组方式是对数据进行自然分组。在CUSTOMERS文件中为每个顾客设立一项。

然而，只把那些已经订购了某些货物的顾客姓名记入 ORDERS 文件中。此外，如果一个顾客有几项悬而未决的订货，则也把这些信息存贮到该顾客的 ORDERS 文件中，来记录每一项订货。

商店数据库

customers

cname	address	balance
Brooks,B.	7Apple Rd.	10.50
Field,W.	43Cherry Ln.	0
Robin,R.	12Heather Ct.	23.45
Hart,W.	65Lark Rd.	43.00
Court,S.	56Blossom Rd.	0
English,D.	82Alpine Rd.	0

orders

oname	item	quantity
Brooks,B.	Work Bench	5
Brooks,B.	Saw	1
Robin,R.	Work Bench	3
Hart,W.	File	3
Robin,R.	Hammer	8
Court,S.	Saw	3
Court,S.	File	5
English,D.	File	1
English,D.	Hammer	2

如何对某一给定应用的信息进行分组和如何组织这些信息，一般取决于用户进行信息检索的要求。也就是说，信息本身仅仅是被存贮起来，以便检索这些信息。应该按一定的格式存贮这些信息，这种格式允许迅速的回答查询，并且允许用尽量少的程序生成报表。

这类数据的用户可能提出一些典型问题，比如：“谁的帐已结清？”，“哪些人订购了哪些商品？”。如果某个顾客的收付差额（balance）项为零，则表示他的帐已结清。几乎所有的数据库管理系统都提供一种交互式查询语言，来列出收付差额项为某一个值（比如为零）的顾客的记录。

要回答第二个问题，需要从两个文件中获取信息。item 字段是在 ORDERS 文件中，但是有关谁已付清了帐单的信息，即收付差额字段，是在 CUSTOMER 文件中。对这个问题的回答，是因为这两个文件都包含有顾客姓名。一旦从 CUSTOMERS 文件中挑选出了那些已付清帐单的顾客，就可以按他们的姓名从 ORDERS 文件中查找他们所订购的

货物。

在关系数据库系统中,设计这种交互式查询语言来处理跨文件的检索,要比以计算机中模型化数据的陈旧方法为基础的数据库系统具有更大的灵活性。

在非关系式的系统中,如果要完全回答跨文件的查询,那么必须要事先对系统描述文件之间的联系。这些关系在初始定义和建立数据库的时候,必须象定义信息分组一样进行定义。

随着数据库的发展,会得到具有新的关系的新文件。一般来说,在数据库文件之间存在着大量有用的关系。当使用关系式数据库时,查询语言可以在任何时候使用存在于文件之间的任何关系。在初始设计数据库的时候不必预先描述任何一种关系。

在这些手册中,按照关系模式的词汇,所谓“文件”信息结构,被认为是一个“关系”,但这种信息结构具有前面所介绍的相似的类表结构。这类表的行,一般都称之为“记录”(record),但现在按关系数据库的术语称之为“元组”(tuples)。最后,“数据字段”(field)的名字,如 CNAME、ADDRESS 等,现在用关系数据库的术语称之为“属性”。

但是在全部 INFORMIX 手册和其它文本中,我们不用这些新的关系数据库术语。为了便于理解,INFORMIX 的用户可以使用更广泛采用的术语“文件”(file)、“数据字段”(field)及“记录”(record)等。然而,当有必要区分 INFORMIX 数据库的文件与典型的操作系统文本(text)的文件时,将使用短语“数据库文件”(database file)来表示作为组成数据库的“文件”。

数据库文件是作为两个操作系统文件来实现的,其中一个文件是关于数据的数据文件,另一个文件是关于索引的索引文件,它被用来加速检索和提供关于索引特性的一些特殊控制信息。对于每一个数据库还有一个操作系统文件,这个文件包含有被称之为数据词典的效能。为数据库文件保持数据的那个操作系统文件以符号“.dat"结尾。保持索引信息的文件以“.idx"结尾。保持数据词典的文件以“.dbd"结尾。

撇开数据库系统的术语不谈,根据七十年代 IBM 公司的 E.F.Codd 对“关系”所下的定义,INFORMIX 的用户仍然可以确信 INFORMIX 是一个真正的“关系”数据库管理系统。INFORMIX 是以称之为“关系代数”的算法为基础的,并且可以证明是“关系完全”的 (relationally complete)。

III. INFORMIX 概述

INFORMIX 数据库管理系统是一种允许用户生成完整的数据管理系统的一系列计算机程序的集合,通常不需要任何用户编程序。当需要用户编程序时,通常可用类似于英语的报表书写语言来编写这些程序。这样使得程序的编写既快又简单。即便这些工具还不够灵活,那么可以使用常规程序设计语言与 INFORMIX 存贮、查询子程序之间的接口。

一旦分析了应用对象,并且设计了数据库的格式,那么就可以采用 DBBUILD 数据描述语言的编译程序来生成想要的数据库及其文件、数据字段。DBBUILD 起初不但生成用来保持数据的空的操作系统文件,而且同时生成用来存放加速查询处理的数据索引文件。

DBBUILD 也生成数据库的数据词典,这是一个包含有关数据库结构的所有信息的操作系统文件。然后,所有访问数据库的其它 INFORMIX 程序都使用该词典。构造一个数

数据库时可用的文件数不受 INFORMIX 软件限制。对于使用 INFORMIX 的大多数计算机用户来说,文件的数目只受存贮器容量的限制。

对 INFORMIX 数据库的文件,操作系统提供口令保护。所生成的文件为生成它们的用户及用户组提供全部使用许可权,但对其他用户则不提供这些许可。未经许可通常导致执行的程序不能读数据词典。可以构造一个数据库,以使它完全驻留在一直录之中(它应驻留在不同设备上的目录之中,但检查跟踪文件除外)。然而,当在 DBBUILD 中使用了 LOCATION 选择项时,有可能把数据库扩展成跨越一定数量的目录和设备。如果没有置 DBPATH,当访问数据库时,用户应处在包含数据库词典的目录中。如果对所需求的数据库设置了 DBPATH,那么 INFORMIX 就可以在任何目录中运行。

INFORMIX 被设计成可以通过带有视频显示的计算机终端来使用。然而,大部分 INFORMIX 的程序也可以被一种打印时采用“硬拷贝”的终端使用,这种终端是不带视频的显示器,也可被看作是一种打印机的终端。

三个 INFORMIX 程序,MASTER MENU (主菜单)、ENTER2 数据输入程序和 PERFORM 数据输入及查询处理器,在很大的程度上是使用面向屏幕的视频显示终端。这些程序不能使用硬拷贝终端运行。这些程序选用了视频终端能理解特殊代码的优越性能,如清除屏幕,用某些信息描绘屏幕,移动终端的游标到不同的位置等等。由于这些特殊代码对不同终端是不同的,因此 INFORMIX 使用称作 TERM CAP 的一个系统,此系统作为操作系统的一部分,可用来找出正在使用的终端类型。

INFORMIX 可以通过下面所示的与操作系统间交互式会话的方式进行运行:

```
% informix
```

或者

```
% informix 数据库名
```

如果使用第一种方法,则屏幕显示下列信息:

```
INFORMIX Master Menu                INFORMIX version 3.00
```

```
Copyright(C) 1981, 1982, 1983 Relational Database Systems, Inc.
```

```
Software Serial Number RDS-000001
```

- | | |
|-------------|-------------------------------|
| 1. Perform | Data entry and query by forms |
| 2. Acego | Run Ace report |
| 3. Informer | Query language |
| 4. Select | Database selection |
| 5. Dbstatus | Database administration |
| 6. Schema | Print database schema |
| 7. Enter2 | Screen-oriented data entry |
| 8. Enter1 | Data entry |
| 9. System | Operating system command |

Use space bar, arrow keys, or type number to make selection.

Enter 'b' to exit.

Enter carriage return to execute selection; 1

当运行 INFORMIX 时,如果没有指定数据库名,则可以通过打一个 4 来指定数据库

名, 选择数据库的功能, 并用数据库要处理的编号来响应“提示符”。在当前目录中的所有数据库以及用 DBPATH 所指定的那些数据库都可使用。如果只有一个数据库是可用的, 那么将以默认方式作为选择。

INFORMIX Master Menu

INFORMIX version 3.00

Copyright (C)1981, 1982, 1983 Relational Database Systems, Inc.

Software Serial Number RDS-000001

- | | |
|-------------|-------------------------------|
| 1. Perform | Data entry and query by forms |
| 2. Acego | Run Ace report |
| 3. Informer | Query language |
| 4. Select | Database selection |
| 5. Dbstatus | Database administration |
| 6. Schema | Print database schema |
| 7. Enter2 | Screen-oriented data entry |
| 8. Enter1 | Data entry |
| 9. System | Operating system command |

Use space bar, arrow keys, or type number to make selection

Enter 'b' to exit.

Enter carriage return to execute selection; 4

The following databases are available;

1. bank
2. realty
3. test

Use space bar, arrow keys, or type number to make selection.

Enter 'b' to exit.

Enter carriage return to execute selection; 1

如果运行 INFORMIX 时打入了数据库名, 那么当 Master Menu (主菜单) 被第一次显示出来时, 就表明当前的数据库已建立起来了, 并且显示出信息, “Database database-name selected”。在下面说明的这种情况中, 所选择的数据库被称作“realty”(不动产)。在屏幕底部的“信息区”中显示有关信息。MASTER MENU、ENTER2、PERFORM 用屏幕底部的行作为信息区。

INFORMIX Master Menu

INFORMIX version 3.00

Copyright (C) 1981, 1982, 1983 Relational Database Systems, Inc.

Software Serial Number RDS-000001

- | | |
|-------------|-------------------------------|
| 1. Perform | Data entry and query by forms |
| 2. Acego | Run Ace report |
| 3. Informer | Query language |
| 4. Select | Database selection |
| 5. Dbstatus | Database administration |
| 6. Schema | Print database schema |

- | | |
|-----------|----------------------------|
| 7. Enter2 | Screen-oriented data entry |
| 8. Enter1 | Data entry |
| 9. System | Operating system command |

Use space bar, arrow keys, or type number to make selection.

Enter 'b' to exit.

Enter carriage return to execute selection; 1

Database "realty" selected.

当运行 INFORMIX 时, 如果在操作系统命令行打入当前数据库名而未能建立当前数据库, 以及在 MASTER MENU (主菜单) 中用功能 4, 也未能建立当前数据库, 则运行的第一个程序将要求一个数据库名。使用主菜单的功能 5, 总是可以把当前数据库改变到该操作系统目录中的另一数据库。

如果 MASTER MENU、ENTER2 或者 PERFORM 总是显得执行不正常, 在这几种情况中, 屏幕上的字符不在预期的位置, 那么请查阅附在 INFORMIX 上的一起发送的安装说明书。这很可能是由于没有告诉操作系统目前正在使用的终端的型号, 或者将终端的型号弄错了。当用户注册时, 一般应将正在用的终端的有关信息告诉操作系统。称为 ".profile" 的操作系统文件被用来传送这些信息给操作系统, 进一步的讨论, 在随 INFORMIX 发送的安装说明中可见到。

MASTER MENU、ENTER2 及 PERFORM 采用与用户间交互式会话是为了减少用户必须键击的次数。这些程序一旦接收了一个命令就立即响应, 不必打回车键。

然而, INFORMIX 系统的大多数程序用会话方法同终端用户相互对话。例如, INFORMIX 查询语言允许用户打入包含很多字和很多行的命令。用字 "end" (结束) 或分号来结束一条命令。DBSTATUS、交互式数据库监控程序, 是以单行命令为基础进行操作的。

如果需要, 对交互式 INFORMIX 程序发出的任何命令, 都可以通过打入 RUBOUT 键中止其执行。在有些计算机终端上, 这个键被标为 RUB。在另一些计算机终端上, 称这个键为 DELETE 键, 也可能被缩写成 DEL。

数据库用户通常知道哪些数据字段是重要的关键所在。在 STORES 文件中, CNAME 和 ONAME 字段是大多数检索的目标, 它们当然应该是索引数据字段。当用 DBBUILD 模式 (或称数据库布局) 来构成数据库时, 可以规定原始索引和审计追踪。对于关键数据字段, 用户可以用 DBSTATUS 交互式数据库监控程序给文件附加一个索引。被检索的数据字段的数目没有限制。

当产生一个关键字时, 用户可以规定这个关键字是含有重复值还是一个单值的关键字。在 STORES 数据库中, CNAME 应该是单值关键字, 因为在 CUSTOMERS 文件中对每一顾客只有一项。相反, ONAME 文件不应该是单一值关键字, 因为在 ORDERS 文件中对每一给定的用户可以存在几个项。

DBSTATUS 也可以用来生成、启动和停止审计追踪。标准的数据处理操作过程需要定期地将磁盘设备上的全部内容拷贝到磁带或某些可装卸式介质上。一个典型的后备程序应包含使用操作系统的转贮公用程序, 以每周执行一次制作完全备份程序 (complete backup) 的工作, 并且每天要执行一次制作增量备份程序 (incremental backup) 的工作。如果数据项的卷数很多, 那么一天或许要进行两次制作备份程序的工作。

如果硬件或软件系统发生故障，那么结合使用最近用转贮制作的完全备份程序和最近制作的增量备份程序，可从最近转贮的那一点开始，把全部内容重新存贮到磁盘上。然而从最近转贮到发生故障这段时间所执行的全部处理的事情都会丢掉。

进行转贮时，可以使用 DBSTATUS 程序对每个文件进行审计追踪。如果审计追踪正起作用，则在文件中进行一次处理时，这项处理的付本就被写到该审计追踪文件的末尾。这些审计追踪文件可以存放在任何设备上，但应该将审计追踪文件存贮在非保持数据库的设备上。

如果发生了一个系统故障，可以使用后备带将磁盘恢复到最近转贮的位置上。由于审计追踪恰在转贮之后开始，它们包含了从转贮到发生故障期间进行的所有处理的记录。可以使用 DBSTATUS 的 RECOVER（修复）命令来应用这种处理记录重新构成这些文件。

DBSTATUS 也提供 LOAD 和 UNLOAD 命令，将数据从数据库文件中移进或移出。这对于使用其它计算机文件的数据，甚至使用其它计算机的数据对数据库进行初始化是很有用的。用 INFORMIX 查询语言也可按可装载格式生产文件，这样可以将 INFORMIX 和 DBSTATUS 组合，这在数据库中用某些取自其它文件的信息来初始化新文件是非常有用的。这些问题在手册中的 DBSTATUS 部分将作进一步的讨论。

DBBUILD 数据描述编译程序用于首次生成数据库，也可通过增加、删除、扩展或压缩等方式用 DBBUILD 数据描述编译程序来改变数据库的结构。在 DBBUILD 手册中将作进一步的讨论。

INFORMER 查询语言是一种交互式语言，它是用来打印任何文件或打印任意文件中选定的部份的一种类似英语的语言。由于它的句法是以关系式数据库模式为基础，因此对于前面讨论过的跨文件的问题，它也可以回答。

在 INFORMER 被使用了一次之后，同样的查询开始变为相当规范化。为了节省键击次数，可将任何查询记入一个操作系统编辑文件，然后用 INFORMER 的执行命令交互式地执行。一个 EXECUTE（执行）文件可以包括一条 INFORMER 命令，或者一条任意长的命令序列。

进行查询时，每次可改变查询的某些部分。要做到这一点，可以通过查询期间构成的对用户输入作提示的执行文件来实现。

ACE 关系式报告书写程序是一种新的计算机语言的编译程序，它具有 INFORMER 相同的选择功能，但对数据格式化的能力方面具有更大的灵活性。可以用 ACE 执行所有标准数据处理的算术运算，包括某些统计运算。如：TOTAL（总计）、PERCENT（百分率）、COUNT（计数）以及 AVERAGE（求平均值）。

如果用户的要求需专门定制，超出了通过改变数据库模式，或者进一步发展查询语言或 ACE 报告书写程序所能做的范围，则可以将传统的程序设计语言同面向数据库数据结构的应用语言库一起使用。

如果用户的要求是冲突的，那么可以用某种方法通过扩展 INFORMIX 系统来满足，为此请写一个系统维护要求，通知 RDS。如果系统运行发生意外，并且怀疑系统程序出错，也可以采用上述办法。

IV. 系统一般命令

INFORMIX 组成程序 ENTER1、ENTER2、DBSTATUS、INFORMER、PERFORM 及 ACE 都执行专门的功能。然而还有一些性能,无论运行什么程序,用起来都是很方便的。例如,无论用户正在运行什么程序,在屏幕上显示出数据库的模式往往是有用的。

当要改变数据库的结构时,用户希望显示出数据库的模式。因此,DBSTATUS 允许把模式打印到终端屏幕上、行式打印机上,或者到一个文件上。然而,为了便利起见,在 ENTER1 和 INFORMER 中,也可以使用简单的 PRINT SCHEMA (打印模式) 命令形式;同样也可以使用主菜单上的一个选择项。

PRINT STATUS 命令展示了有关数据库文件的大小和现有的索引等有用的信息。PRINT STATUS 命令在 DBSTATUS 中可以按完整形式使用,在 INFORMIX 交互式程序中也可按简单形式使用。无论何处,都可使用 PRINT SCHEMA 命令的简单形式。

即使当前正在运行数据库软件,如果还能够运行一条操作系统命令,那也是很有用的。作为主菜单的一项功能,这是可以做到的,但是这也可以从 ENTER1、ENTER2、DBSTATUS 和 INFORMER 中得到。这可以按两种方式使用。第一种方式是很清晰的,如下所示:

```
>run "ls-l"
```

在这个例子中,在当前目录中列出文件的命令,被传送给操作系统,而结果在屏幕上显示出来。自数据库程序执行命令后,数据库程序准备接受另一命令,无论是 RUN (运行) 命令,还是任何其它命令,程序都可接受。

RUN 命令的简写形式如下:

```
>!ls-l
```

执行这条命令的结果和上述完全一样。这里用!号来表示这一行中剩下的部分应传给操作系统去执行,而正处于运行状态的数据库程序将要等待到操作系统完全响应这条命令为止。由于面向屏幕的数据输入程序 ENTER2 的单一符号命令的取向,以及 PERFORM 都不能把它们自身提供给诸如 RUN 这样的命令,因此对于执行 ENTER2 中的一个操作系统命令,使用!符的简写方式是唯一的方法。

V. 附录 I ——与 COBOL,C,BASIC 或其它程序设计语言编写的 C-ISAM 文件结合的问题

INFORMIX 3.00 版的一系列特性容易将 INFORMIX 与由 COBOL、C、BASIC 或其它程序设计语言构成的 C-ISAM 文件结合起来。通过指定主关键字的办法,用户可以不必考虑要求 INFORMIX 赋予唯一的标识符。如果没有指定主关键字,在数据库文件中类型为 SERIAL 的数据字段将作为默认的主关键字。如果既不存在 SERIAL 字段,又未说明主关键字,INFORMIX 将把一个四字节的唯一的标识符放到数据库文件每个记录的前面。

虽然这种标识符对多数应用来说不会出现问题,但会产生与用 COBOL 或其它程序设计语言构成的 ISAM 文件相兼容的问题。选择主关键字可以使程序员避免这些潜在问题。

有很多情况需要主关键字,但在数据库文件中不存在单个数据字段的唯一性。在这类情

况中, COMPOSITE 数据字段可用来作为主关键字。orders 文件可以使用部门和项目代码的组合(其中项目代码跨部门复制)来决定唯一的 COMPOSITE 数据字段,它可以被指定为主关键字。

在某些应用中,使用 DBBUILD 中的 LOCATION 选择项也是有益的。ISAM 的应用可以以不同于 DBBUILD 的方法来指明数据库文件。在这类情况下,用户可以变换数据库文件名,使其同 ISAM 的标志一致。

VI. 附录 II——操作系统中的 INFORMIX

读、写许可

在操作系统中生成的任何文件或目录都被规定了一套许可权限,这种权限用来决定用户对相应的文件或目录的存取权。可以用“ls-l”命令来打印出这些许可权。我们看到在每行的前十个符号是许可权。许可权表格后面跟着一个数字,然后是文件或目录的属主名。列出的剩余部分是由目录或文件按字节表示的大小、文件最近被修改的日期和时间、以及文件名或目录名等组成。列出的典型清单如下:

```
% ls -l
-rw-rw----- 1 susan      871 Apr 21 16:04 bank.dbd
-rw-rw-rw- 1 susan      465 Apr 14 11:18 emp.sch
-rw-rw----- 1 susan    2464 Apr 15 11:54 employee.dat
-rw-rw----- 1 susan    2560 May 11 11:00 employee.idx
-rwxrwxrwx 1 john       962 May 16 14:45 form2
-rw----- 1 john     1278 May 16 14:45 form2.frm
drwxrwxrwx 2 john        48 Apr 10 10:10 scripts
```

许可权表格的第一个符号将是“d”(目录)或者是“-”(文件)。剩下的九个符号可看成每三个符号为一组的三组。三组中的第一组定义了文件属主的存取特权(在上述例子中,属主是 susan 或 john)。第二组是为文件属主所在的操作系统用户组提供对该文件的存取权。最后一组是用来给系统中所有其他用户提供对文件的存取权(公用)。字符 r、w、x 反映出读、写和执行许可处于“开的状态”;出现在这些符号所处的位置上的连字符“-”,表示相应的特权受到禁止。

在上述例子中,文件 emp.sch 属于 susan,它可以被属主、属主所处的小组、或者系统中的其他用户读、写。文件 bank.dbd 只可以被 susan 和 susan 的用户组中的用户读、写。文件 form2 可以被系统中的任何用户进行读、写或执行,而文件 form2.frm 只可以被 john 读、写。scripts 是一个允许系统的任何用户进行读、写或执行的目录。

文件或目录的属主(或者系统管理员)可以使用“chmod”(改变方式)命令来改变许可权。方式一般可以用 1000 到 777 之间的三个整数来指定。方式中的第一位数,为该文件或目录的属主定义了存取权限。第二位数属主所处的用户组定义了存取权限。最后一位数字为系统中所有其他用户定义存取权限。1 表示允许执行的许可,2 表示允许写的许可,而 4 表示读许可。可以把这些数相加来表示多种许可。7 表示全部三种许可权(1+2+4),6 仅表示读、写许可权,等等。因此,命令

```
% chmod 664 bank. dbd
```

将改变 bank. dbd 文件的方式成为允许属主和属主所处的用户组进行读和写，而对系统的所有其他用户仅允许读操作。那么在执行上述命令之后，“ls—l”此文件，应看到这样的信息：

```
-rw-rw-r-- 1 susan      871 Apr 21 16:04 bank. dbd
```

下表表示允许用户使用专门的 INFORMIX 功能所需要的许可权限。（用“--”表示的列指出有关该文件的许可权限的限制与询问时的操作无关。）INFORMIX 还允许除操作系统许可权限之外使用更多的保密限制。有关更详细的信息，请参看在 DBBUILD 部分的“User Access Privileges”。

	database files					database
	.dbd	.idx	.dat	.irm	.arc	directory
DBBUILD	--	--	--	--	--	rwX
DBSTATUS	rw	rw	rw	--	--	rwX
ENTER1	rw	rw	rw	--	--	rwX
ENTER2	rw	rw	rw	--	--	rwX
PERFORM	rw	rw	rw	r	--	rwX
INFORMER						
queries	rw	rw	r	--	--	rx
writing to						
current dir.	rw	rw	r	--	--	rwX
updates	rw	rw	rw	--	--	rwX
ACE						
run report	rw	rw	r	--	r	rx
output report						
to curr. dir.	rw	rw	r	--	r	rwX

终端类型

为了正常地运行 INFORMIX 中面向屏幕的程序(ENTER2,PERFORM及INFORMIX MENU)，正确地定义你所使用的终端的类型是十分必要的。一般可以在文件/etc/termcap 中找到终端定义。如果在你的系统中没有 termcap 文件，或者在 termcap 文件中没有定义正在使用的终端，那么可以使用随 INFORMIX 介质一起发送的 termcap 文件。如果还是找不到正确的定义，那应询问硬件商人，了解恰当的终端入口。

在 ENTER2、PERFORM、或者 MASTER MANU 中，不正确的或不存在的终端定义可引起显示混乱或者光标处在不正确的位置。如果“vi”文本编辑程序工作正常，那么你的终端类型多半是正确的，而 INFORMIX 应该可以用你的终端工作。如果“vi”工作不正常，则 INFORMIX 也不会正常运行。

要规定终端的类型，可以根据所用的 shell 的版本用“set”或“setenv”命令来规定终端类型。用 C shell，命令为：

```
% setenv TERM d1
```

其中 d1 表示终端类型。如果不是使用 C shell，那么命令是

```
% set term = d1
```

后面跟着命令

```
% export term
```

环境变量

INFORMIX 允许使用两个操作系统环境变量, DBPATH 和 DBTEMP。用设置终端类型大体相同的办法来设置环境变量。在本手册的 DBBUILD 部分比较完整地提供了 DBPATH 的文件资料, DBPATH 允许用户在操作系统文件结构中规定数据库词典的位置。DBTEMP 是在 INFORMER 部分提供了其文件资料, DBTEMP 允许用户规定在 INFORMER 和 ACE 中产生的暂时文件的位置。

文本编辑程序

要生成 DBBUILD 模式和 PERFORM 屏幕规范, 必须使用操作系统文本编辑程序。有关文本编辑程序的信息, 可查阅操作系统手册。通常可以使用的文本编辑程序是 “vi”、“ex”、和 “ed”。

Ⅶ. 附录 Ⅱ——INFORMIX 保留字

	blanks	do
A	bottom	dominant
access	by	double
add	bye	downshift
after		duplicates
alias	C	dups
all	call	
allow	char	E
allowing	character	edate
and	clipped	else
ascending	comments	end
ass	comp	erase
assign	composite	every
att	count	ex
attribute		execute
attributes	D	
audit	database	F
average	date	field
avg	db	file
	default	find
B	define	first
before	delete	float
begin	descending	for