

中西文兼容INFORMIX

数据库管理系统实用参考

黎连业 魏 军 解 文 编著

北京科海培训中心

中西文兼容INFORMIX 数据库 管理系统实用参考

黎连业 魏 军 解 文 编著

北京科海培训中心

前 言

中西文INFORMIX数据库管理系统，越来越被广大计算机应用工作者所接受，这主要是被它的优越性所吸引。这些优越性具体表现为：

(1) 多用户。在多用户操作系统下，用该数据库编制的程序和建立的文件，可供多个用户同时上机使用，而不必担心建立的数据文件被一个用户打开使用时，别的用户不能再打开的问题。

(2) 字段长。在一个记录中，它所属的字段占用字节长达800多个（合400多个汉字），能够满足用户的特殊需求。

(3) 简易。用户在使用计算机时，总希望简易输入（即代号输入），输字输出具体含义，为打印报表提供方便，而不希望把汉字输出具体含义内容以数组或参数方式定义在程序内部。该数据库提供了这一方便。

(4) 多文件。在一个数据库中可建立多个数据文件，方便用户使用。

(5) 查询内容丰富。该数据库为用户设置了丰富的查询功能，数值内容可查区间值内的内容（即从多少开始到多少结束），在文字范围内可查任意一个字符或多个字符。

(6) 组合查询。若用户在一个记录众多的内容中，只记得其中几项了，那么可根据用户所知的项目查出所需内容。

从上述优点来看，用INFORMIX编制明显优于其他语言编制的程序系统。同时，还要强调说明一点，用该数据库管理语言ACE编制程序时，比用其他语言编制程序快3~5倍。

本书是作者开发、应用INFORMIX数据库的体会，目的是抛砖引玉，引起读者对INFORMIX数据库的重视。书中列举的例子及源程序从编制技巧上来说，是不精练的，但为了说明问题，我们还是把这些源程序附上。书中亦难免有缺点和错误，请读者批评指正。

黎连业 魏 军 解 文

于中科院计算所

目 录

第一篇 INFORMIX数据库基本内容

第一章 绪 论

1.1 数据库系统.....	(1)
1.2 关系型数据管理系统.....	(1)
1.3 INFORMIX概述.....	(2)

第二章 数据描述语言编译程序DBBUILD

2.1 引言.....	(4)
2.2 DBBUILD语言的定义.....	(4)
2.3 实例.....	(7)
2.4 用DBBUILD更改数据库结构.....	(9)

第三章 面向行的数据录入程序 (ENTER1)

3.1 引言.....	(11)
3.2 命令.....	(13)
3.2.1 SELECT命令.....	(13)
3.2.2 FIND命令.....	(14)
3.2.3 NEXT命令.....	(16)
3.2.4 PREVIOUS命令.....	(17)
3.2.5 SAME命令.....	(17)
3.2.6 ADD命令.....	(18)
3.2.7 DELETE命令.....	(19)
3.2.8 UPDATE命令.....	(19)
3.2.9 PRINT命令.....	(20)
3.2.10 BYE命令.....	(22)
3.2.11 HELP命令.....	(22)

第四章 面向屏幕的数据录入程序 (ENTER2)

4.1 引言.....	(24)
4.2 命令.....	(25)
4.2.1 SELECT (S-选择) 命令.....	(25)
4.2.2 FIND (F-查找) 命令.....	(27)
4.2.3 NEXT (N-向下) 命令.....	(28)
4.2.4 PREVIOUS (P-向上) 命令.....	(29)
4.2.5 UPDATE (U-修改) 命令.....	(29)
4.2.6 ADD (A-追加) 命令.....	(30)

4.2.7 DELETE (D-删除) 命令	(30)
4.2.8 REDRAW (R-重显) 命令	(30)
4.2.9 BYE (B-退出) 命令	(31)
第五章 查询语言 (INFORMER)	
5.1 引言	(32)
5.2 命令	(33)
5.2.1 SELECT DATABASE命令	(34)
5.2.2 PRINT命令	(35)
5.2.3 READ命令	(38)
5.2.4 SORT命令	(38)
5.2.5 UNLOAD命令	(39)
5.2.6 JOINING命令	(40)
5.2.7 算术运算	(42)
5.2.8 聚合运算	(43)
5.2.9 日期函数	(44)
5.2.10 UPDATE命令	(45)
5.2.11 DELETE命令	(45)
5.2.12 ADD命令	(46)
5.2.13 EXECUTE命令	(47)
5.2.14 RUN命令	(47)
5.2.15 PROMPT命令	(47)
5.2.16 LOCK与UNLOCK命令	(48)
5.2.17 ADD INDEX命令	(48)
5.2.18 DELETE INDEX命令	(48)
5.2.19 DBTEMP命令	(49)
5.2.20 HELP命令	(49)
5.2.21 BEY命令	(51)
第六章 关系报表书写程序 (ACE)	
6.1 引言	(53)
6.2 ACE程序设计	(54)
6.2.1 编辑ACE程序	(54)
6.2.2 编译ACE程序	(54)
6.2.3 纠错ACE程序	(55)
6.2.4 运行ACE程序	(55)
6.3 ACE语言的描述	(57)
6.3.1 ACE程序的一般说明	(57)
6.3.2 ACE程序的命令及语句	(58)
6.4 C程序与ACE程序互相调用	(72)
6.4.1 C程序调用ACE程序	(72)
6.4.2 ACE程序调用C函数	(76)
第七章 用户屏幕生成和数据处理软件包 (PERFORM)	
7.1 引言	(78)
7.2 用户屏幕格式的设计与调用	(79)
7.2.1 编辑用户屏幕格式文件	(79)
7.2.2 编译用户屏幕格式文件	(83)

7.2.3 为用户屏幕格式纠错	(84)
7.2.4 调用用户屏幕格式	(85)
7.3 数据处理	(86)
7.3.1 QUERY命令	(87)
7.3.2 NEXT命令	(89)
7.3.3 PREVIOUS命令	(89)
7.3.4 ADD命令	(89)
7.3.5 UPDATE命令	(90)
7.3.6 REMOVE命令	(91)
7.3.7 FILE命令	(91)
7.3.8 SCREEN命令	(93)
7.3.9 CURRENT命令	(95)
7.3.10 MASTER命令	(95)
7.3.11 DETAIL命令	(96)
7.3.12 OUTPUT命令	(97)
7.3.13 BYE命令	(97)
7.4 C程序与PERFORM程序互相调用	(97)
7.4.1 C程序调用PERFORM程序	(97)
7.4.2 PERFORM程序调用C函数	(100)

第八章 菜单建立程序DBMENU

8.1 引言	(102)
8.2 设置一个菜单系统	(102)
8.3 设计菜单系统	(104)
8.4 建立菜单系统	(105)
8.5 菜单系统的使用	(105)
8.6 修改菜单数据库	(106)

第九章 数据库监控程序 (DBSTATUS)

9.1 引言	(107)
9.2 DBSTATUS功能详解	(107)
9.2.1 打印 (显示) 文件的模式或状态	(107)
9.2.2 增加和删除索引	(109)
9.2.3 删除文件和整个数据库以及指定文件的跟踪内容	(110)
9.2.4 启动或中止跟踪控制	(111)
9.2.5 在数据库文件与操作系统文件之间进行数据加载或卸载	(112)
9.2.6 改变字段名	(113)
9.2.7 恢复文件	(114)
9.2.8 选择数据库	(114)
9.2.9 帮助	(115)
9.2.10 退出	(115)
9.3 硬件和软件故障后的恢复	(115)
9.3.1 何时使用跟踪控制	(115)
9.3.2 跟踪控制的约定	(115)
9.3.3 跟踪控制文件的格式	(115)
9.3.4 恢复过程	(116)

第十章 C语言例行程序库ALL-II

10.1 引言	(117)
10.2 INFORMIX ALL-II例行程序	(117)
10.2.1 标题文件dbio.h	(117)
10.2.2 DBSELECT	(119)
10.2.3 DBSETFILEVIEW	(120)
10.2.4 DBSTRUCTVIEW	(121)
10.2.5 DBFIND	(122)
10.2.6 DBSELFELD	(123)
10.2.7 DBUPDATE	(124)
10.2.8 DBADD	(124)
10.2.9 DBDELETE	(125)
10.2.10 DBLOCK	(125)
10.2.11 DBUNLOCK	(126)
10.2.12 DBVERSION	(126)
10.2.13 DBFILE	(126)
10.2.14 DBNFILE	(127)
10.2.15 DBFIELD	(128)
10.2.16 DBNFIELD	(128)
10.2.17 DBNCOMPOSITE	(129)
10.2.18 DBALIAS	(131)
10.2.19 DBINDEX	(131)
10.2.20 DBADDINDEX	(131)
10.2.21 DBDELINDEX	(132)
10.2.22 DBPRUSING	(132)
10.3 ALL-II 接口	(133)
10.3.1 错误捕获	(133)
10.3.2 参数格式	(133)
10.3.3 中断信号	(133)
10.3.4 连接ALL例行程序	(133)
10.3.5 光标库courses和终端功能库 termcap 支持	(134)
10.3.6 RDS程序库扩充	(134)

附录 语法出错表

第二篇 一个小型会计帐务处理系统

引言

第十一章 会计记帐凭证的设计

11.1 记帐凭证的设计	(157)
11.1.1 记帐凭证的内容	(157)
11.2 记帐凭证库结构设计	(158)
11.3 记帐凭证库屏幕录入数据格式设计	(158)

第十二章 会议帐务程序设计与程序编译步骤

12.1 会计帐务处理系统框图	(159)
12.2 数据库建立和编译	(159)

12.2.1 建立数据库结构 (名称: khlib1)	(159)
12.2.2 编译.....	(159)
12.2.3 三级科目库建立结构 (名称: khlib7)	(159)
12.2.4 编译.....	(159)
12.3 数据录入格式建立和编译.....	(159)
12.3.1 数据库文件录入数据屏幕格式建立 (名称: khlib2)	(159)
12.3.2 编译.....	(161)
12.3.3 三级科目对照库录入数据屏幕格式建立 (名称: khlib 8)	(161)
12.3.4 编译.....	(162)
12.4 ACE关系报表书写器编写程序与编译	(159)
12.4.1 ACE 关系报表书写器编写源程序录入	(162)
12.4.2 ACE 语言程序编译	(162)
12.4.3 编译后的运行语句.....	(163)
第十三章 用关系数据库语言编制应用程序的要点	
13.1 用关系数据库语言编制程序的基本步骤.....	(164)
13.2 关系数据库语言所编程序的连接方式.....	(164)
第十四章 源程序表	
14.1 各模块名称.....	(164)
14.2 源程序表.....	(164)
第十五章 三级科目对照库	
15.1 三级科目对照库数据.....	(208)

第一篇 INFORMIX数据库基本内容

第一章 绪 论

1.1 数据库系统

在现实生活中，我们常常要处理大量的数据信息。例如：库存管理人员每天要处理大量的进出货单；银行人员每天要处理大量的存取款项。此外，库存管理人员还经常查点某种物资的库存量；银行人员则经常查看某人的存入/支出款数等等。对于这些不同种类的大量信息的处理，计算机化是必由之路。然而，如果我们直接去设计和编制处理这些不同种类信息的专用软件系统，而不借助于一些公用的工具和手段，那是极为困难、极不经济的。实际上，就许多种信息来说，对它们的处理在许多方面是共同的。如果我们将这些共同的处理抽象出来，编制成一些可以公用的应用程序，再在此基础上来编制面向不同应用的软件，则更容易、方便、经济得多。

数据库管理系统就是这样一些可以公用的应用程序的集会。它为我们集中统一合理地存储和管理数据信息提供了一种手段和工具。利用它，我们可以方便地对数据库中的信息进行输入、查询以及统计等处理，它也能使我们快速而方便地编制出面向各种应用的程序来。

数据库管理系统主要由下列三类软件组成：

- (1) 用以建立、查询、修改、维护数据库等的语言；
- (2) 对数据库进行管理、调度的数据库管理程序；
- (3) 数据库服务程序，用于数据库生成、统计、故障恢复等等。

1.2 关系型数据管理系统

关系型数据管理系统是常见的几种数据库管理系统中的一种。关系型数据库管理系统特别擅长于处理信息结构类似表格的各种信息。比如，利用关系型数据库管理系统的查询语言，不但可以查询一个数据库文件中的信息，还可以方便地跨文件查询。

1.3 INFORMIX概述

INFORMIX是关系型数据库管理系统中的一种。与其他关系型数据库管理系统(如dBASE III)相比较,它具有存取速度快和存储开销小等优点。因为INFORMIX是以二进制机器码形式存储数据的。此外,它不但功能十分强,也有很大的灵活性。在通常情况下,用户根本无需编程,就可以利用INFORMIX提供的语言来查询、输入信息及进行其他方面的处理。如果需要编程,报表书写语言是一个有力的工具。利用这一工具可快速方便地编写相应的程序。在上述手段都无能为力时,用户可以利用各种高级语言(如C)与INFORMIX之间的接口来编制满意的程序。

INFORMIX有如下一些常用的程序:

- (1) 主菜单 从中可以调用其他INFORMIX程序和命令。
- (2) DBBUILD 一种数据描述语言,允许用户建立或修改数据文件。
- (3) FORMBUILD 用户可以使用该程序设计自己所习惯的数据输入屏幕格式。
- (4) PERFORM 使用用户设计的表格进行数据输入和查询。
- (5) INFORMER 一种用于数据检索的查询语言。
- (6) ACE (包括ACEPREP和ACEGO) 建立和打印格式化报表。
- (7) DBSTATUS 数据库监控程序。
- (8) ENTER1 用于硬拷贝终端(它是一个打印机,并且不具备屏幕显示)的数据输入程序。
- (9) ENTER2 数据输入程序(较PERFORM简单)。
- (10) ALL-II 应用语言程序库,它提供常规程序设计语言与INFORMIX例行程序之间的接口。
- (11) DBMEMU 提供用户设计自己应用系统中菜单的功能。
- (12) BCHECK 用于检验索引。

下面,我们将叙述如何利用INFORMIX的各种程序来开发应用软件。

第一步,首先要对应用对象进行分析,并据此设计数据库模式。分析的方法很多,由于这超出了本书的范围,就不在此赘述了。

数据库模式用符合DBBUILD规范的文本文件来描述,这些文本文件可用任何编辑工具(如XENIX中的vi)来编辑。

第二步,用DBBUILD数据描述语言的编译程序来生成所需要的数据库及其文件和数据字段。每编译一个模式描述文件,将至少生成两个文件。如果与该文件有关的数据库事先不存在,则还将产生一个相应的数据词典。数据词典是一个包含有关数据库结构的所有信息的操作系统文件。在生成数据词典之后,所有访问数据的其他INFORMIX程序都将使用该词典。

第三步,根据用户的要求,设计数据录入、查询的方式,以及打印的报表的格式。如果需要,则编制相应的程序。

(1) 若想设计符合用户对数据的录入和查询要求的屏幕格式,以进行输入和简单查询,则可以利用FORMBUILD和PEFORM。具体做法是:首先要用文本编辑软件建立一个格

式说明文件，然后用FORMBUILD对其进行编译。最后，启动PERFORM程序，调用编译好的屏幕文件。在格式说明文件里，程序员可以随心所欲地把要显示的字段放在用户要求的位置上，还可以使用其他能够更清楚地说明要显示的字段含义的标题。此外，一个格式不仅可以设计成多个屏幕，还可以用多个文件的字段建立格式。

一般情况下，利用FORMBUILD和PEFORM就可设计和实现满足用户要求的实用而美观的屏幕格式。特殊情况下，可以用高级语言与INFORMIX之间的接口。

如果用户不要求自行设计屏幕格式，他可以利用面向正文的交互式数据录入和更新程序ENTER1对数据库逐个记录地进行添加、更新和删除；用户可以利用面向屏幕的交互式数据录入和查询程序，即通过运行ENTER2，根据数据词典中的信息所构造的一个简单屏幕，来查找文件或逐个地查询、修改文件中的记录；此外，用户也可以用交互式查询语言INFORMER来查询。

(2) 对于报表生成，一般情况下，需要借助于关系报表书写语言ACE。

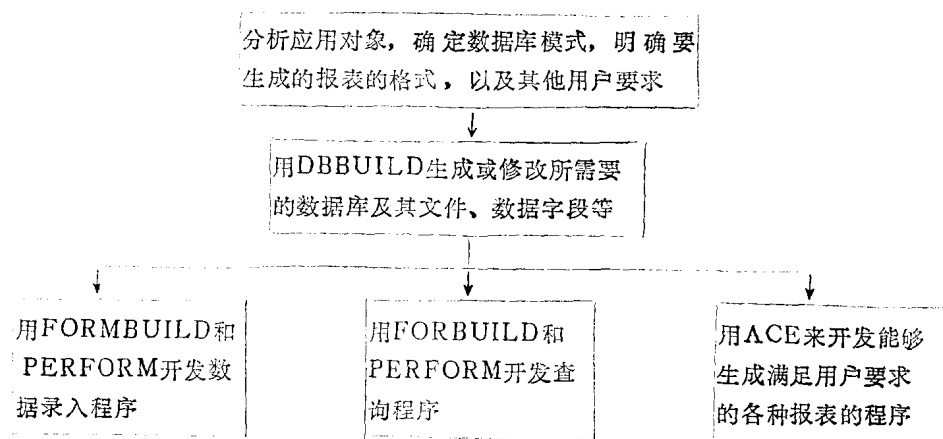
ACE是一种从数据库中产生报表的程序设计语言。利用它可以非常简单方便地编制能生成各种报表的实用程序。

ACE允许用户把多个文件的信息综合到一个报表之中，文件的个数可以为5~8个。为了便于指出哪些信息将要在报表中打印，ACE设计了三类操作：“选择”(selection)，“投影”(project)和“连接”(join)。

“选择”和“投影”作用于同一个文件，前者可用来挑选出一部分要汇集到报表中的记录，后者则只选出要输出到报表中的字段。“连接”是将任何两个以上的文件按照一些共同的字段(同类型、同长度、同数值)连接起来，以备进行选择 and 投影操作之用。

一旦信息收集好，就可以用简单而强有力的报表格式化语句来生成报表格式。需要注意的是，为收集信息所用的选择、投影和连接操作，以及生成表格所用的报表格式化语句都是报表书写程序的一部分。它们的这种密切结合使含有数据库任何子集的报表可以很快生成出来。

下面，我们用一个简单的图示来表示上述步骤：



第二章 数据描述语言编译程序DBBUILD

2.1 引言

DBBUILD可用于生成一个新的数据库，在数据库中增加文件，以及更改原有文件的结构。对于每个要生成或修改的数据库文件，都要先用文本编辑工具编辑一个称为“模式”(SKEMA)的符合DBBUILD语言规范的文件。该文件中应指定数据库文件所属的数据库名、该数据库文件的文件名，及其各字段的名称、长度、类型及其他信息。在编辑该文件之后，就可以用编译程序DBBUILD对该文件进行编译。

用DBBUILD编译模式文件，将生成至少以下两个操作系统文件：

- (1) 数据文件；
- (2) 索引文件。

数据文件的名称由在模式文件中给出的文件名加上一个扩展符“.dat”构成；索引文件的名称由在模式文件中给出的文件名加上一个扩展符“.idx”构成。

数据文件用来存放数据，索引文件则包含数据文件的索引，索引文件中的索引数不受限制。数据文件和索引文件共同表示了数据库文件。

如果模式中描述的文件为数据库的第一个文件，那么在用DBBUILD编译模式文件时，还将创建另一个操作系统文件，用于存放数据词典，这就是数据词典文件。数据词典文件的名称由模式文件中给出的数据库名和扩展符“.dbd”构成。

2.2 DBBUILD语言的定义

DBBUILD数据库描述语言模式文件的一般结构如下所示：

```
database 数据库名
{注释}
file 数据库文件名
[location “路径名”]
field 字段名 type 类型标识符[长度] index [索引类型]
[permissions 文件和字段存取权限说明]
end
```

这里，用{...}括起的内容是程序的注释；用[]括起的内容是可选择的，不是一定要存在的；其他部分是一个完整的DBBUILD程序必须具备的内容。此外，字段定义可以有多个，有的字段可以不是索引。

下面，我们详细介绍一下 DBBUILD 的句法和结构。

一、标识符

标识符可以是汉字、英文字母 (A~Z)、数字 (0~9)，以及下划线 “_” 的任意组合，但必须以汉字或字母打头，长度为2~50个字符。数据库名、文件名及字段名都是标识符。由于所有标识符在存放 to 数据词典中时均被转换为小写，因此，不能按大小写的不同来指定标识符，虽然大小写可以混用。

二、常数

在 DBBUILD 语言中唯一的常数是整常数，它被用来描述 “字符型” 字段的长度。

三、注释

注释是为使人能看懂程序而写的说明。

注释可以在模式文件的任何地方出现，也可以跨行。

四、关键字

DBBUILD 语言的关键字可用任意组合的大、小写字母组成。各种关键字及其用法如下：

1. DATABASE 数据库名

关键字 DATABASE 用以命名要创建的数据库。该关键字及其后面的数据库名在模式文件中都是必须的。

2. FILE 数据库文件名

关键字 FILE 用以命名要创建的数据库文件。关键字和 FILE 以及其后的数据库文件名都是必须的，而且是唯一的。

3. FIELD 字段名

关键字 FIELD 为正被加到数据库中的文件定义每一个字段。在模式文件中，至少要用该关键字定义一个字段。但须注意各字段名不仅在数据库文件中是唯一的，在整个数据库中也是唯一的。

4. TYPE 类型标识符

各字段都必须用关键字 TYPE 及其 “类型标识符” 来加以说明。类型标识符必须是下列关键字之一：

CHARACTER, INTEGER, FLOAT, DOUBLE, LONG, SERIAL, DATE, EDATE, YDATE, MONEY 和 COMPOSITE。

(1) CHARACTER (字符型)

用 CHARACTER 说明的字段，其记录值可以是汉字、字母、数字和任意符号。如果使用了 TYPE CHARACTER 作为类型说明，则其后必须有一个 “长度” 说明来指定长度。“长度” 说明的形式为：[LENGTH]××××。其中，“××××” 可以是 1~2048 之间的任何整数。

CHARACTER 和 LENGTH 可分别简写为 CHAR 和 LEN。

(2) INTEGER (整型)

用 INTEGER 说明的字段，其取值范围为 -32 768~32 767 的任何整数，其简写形式为 INT。

(3) LONG (长整型)

用 TYPE LONG 说明的字段，其取值范围为 -2 147 483 648~+2 147 483 647 之间的所有整数。

(4) FLOAT (单精度型)

单精度型字段表示的是一单精度浮点数，其取值范围为 $-1.7e38 \sim +1.7e38$ 。小数点两边允许有7位有效数字。

(5) DOUBLE (双精度型)

双精度型字段的值是一带符号浮点数，可以是 $-1.7e308 \sim +1.7e308$ 之间的任何数值。小数点两边共允许有7位有效数字。

(6) SERIAL (序列型)

序列型字段是按顺序赋值的连续的数。如果没有指出起始数，那么第1个记录将被赋值为1。为了在不同的起始点开始该序列，可用“start”指明起始值。例如“start 100”表示序列将从100开始。

INFORMIX 在记录指定之后，赋值给其序列号。为了保持顺序性和完整性，INFORMIX不允许用户改变序列字段的值。另外，在同一个数据库中，只能定义一个序列字段。

(7) DATE、EDATE、YDATE (日期型)

有三种不同格式的日期型字段可供选择：

① “mm/dd/yy” (美国格式)

② “dd/mm/yy” (欧洲格式)

③ “yy/mm/dd” (中东格式)

其中，mm表示月份，dd表示日期，yy表示年份。

在 ACE 和 INFORMER中可以对三类字段进行排序，可以对日期型字段进行算术运算，而且两个日期型字段可以按时间顺序进行比较。在财务应用及在需要以日历来标识文件的应用中，这些功能是非常有用的。

(8) MONEY (货币型)

货币型字段是作为双精度浮点数存放的（按分计算），录入格式为 \$000.00，计算精度为 \$0.1。

(9) COMPOSITE (组合型)

组合型字段是2~8个模式文件中前面已定义字段的组合。

五、INDEX [索引类型]

在创建一个数据库文件时，可以根据需要对一个或多个字段建立索引。建立索引的方法如下：

(1) 建立不可重复索引，在相应的字段后面加上“index”。例如：field 姓名 type char 6 index。

(2) 建立可重复索引，除了在相应的字段后面加上“index”外，还要加上“dups”。例如：field 姓名 type char 6 index dups。

(3) 建立主索引，在相应的字段后面加上：“index”和“primary”。例如：field 姓名 type char 6 index primary。

不可重复索引要求建立索引的字段的内容不能重复，必须是唯一的。可重复索引则允许建立索引的字段的内容重复。主索引属于不可重复索引，但要求所建的索引字段必须唯一地标识一个记录。在一个数据库文件中，只能指定一个主索引。

如果未建立主索引，INFORMIX将默认“序列型”字段为主索引；如果也没有“序列

型”字段，INFORMIX则自动在每个记录前加一个4字节的不可见字段，该字段为隐含的主索引。

六、〔LOCATION “路径名”〕

通过使用LOCATION任选项，可将一个数据库中的各文件分布在若干个不同的目录、文件系统或磁盘及磁带中。LOCATION的使用简化了C-ISAM、COBOL或其它语言对INFORMIX文件的存取。

七、〔PERMISSIONS文件和字段存取权限说明〕

在XENIX/UNIX系统下的INFORMIX版本中，可以设置用户权限。而在MS-DOS系统下的INFORMIX版本中，则不支持这种功能。XENIX/UNIX的存取权限控制高于INFORMIX数据库的权限控制。

INFORMIX文件级的权限控制限制用户对记录的读、添加、修改和删除等操作。INFORMIX字段级的权限控制则限制用户对各个字段的读和修改操作。

2.3 实 例

下面，我们将通过一个劳动工资方面的例子，来看看如何利用DBBUILD建立数据库。

正如1.3节所述的那样，在我们对应用对象进行分析，并且完成了数据库模式的设计之后，就可以用DBBUILD数据描述语言编译程序来生成满足用户要求的数据库及其文件和数据字段了。

例如，在对某单位职员的工资进行了分析之后，用编辑工具编辑了如下两个符合DBBUILD规范的模式文件gzk1和gzk2。

```
gzk1
database 工资库
file 工资
field 编号      type int    index
field 年月      type char   5
field 姓名      type char   6    index
field 基本工资  type float
field 工龄工资  type float
field 副食补贴  type float
field 家属补贴  type float
field 主副补贴  type float
field 洗理费    type float
field 独生子女  type float
field 交通费    type float
field 书报费    type float
field 补其他1   type float
field 补其他2   type float
field 房租费    type float
field 国库券    type float
```

```

field 扣其他1    type float
field 扣其他2    type float
field 扣其他3    type float
end
gzk2
database 工资库
file  职级
field 序号          type char    2
field 职称          type char   10
field 级别          type char    8
end

```

一旦编辑好了模式文件，我们就可以进行编译，产生数据库及其文件和字段了。

```
#dbbuild gzk1✓
```

DBBUILD 数据描述语言编译程序

INFORMIX 版本3.30.02

版权所有 (C) 1981, 1982, 1983, 1984 关系数据库系统公司。

软件序列号 RDS-R223174

文件“gzk1”将被编译。

编译成功。数据库“工资库”的字典文件已经建立或更改。

程序结束。

由于在此之前，数据库“工资库”事先并不存在，编译gzh1将产生以下3个操作系统文件：数据文件（工资.dat），索引文件（工资.idx），数据词典文件（工资库.dbd）。

必须注意，这是编译成功的情形。如果模式文件中的dbbuild描述有错误或其他原因，则都将导致编译失败。编译失败时，编译程序将显示出错信息，生成含有错误清单的一个文件，这基本上是一个带出错编号和讨论问题时所需信息的模式文件。此文件由模式文件名和扩展符“.err”来命名。例如，如果我们将gzk1中的file拼成ficc，则在用“dbbuild -r”编译gzk1时，将显示：

```
DBBUILD 数据描述语言编译程序
```

INFORMIX 版本3.30.02

版权所有 (C) 1981, 1982, 1983, 1984 关系数据库系统公司

软件序列号 RDS-R223174

文件“gzk1”将被编译。

编译不成功。发现1个错误。

文件“gzk1.err”已经生成。

程序结束。

gzk1.err的内容如下：

```
database 工资库
```

```
ficc 工资
```

A grammatical error has been found on line 2, character 1.

The construct is not understandable in its context.

见错误号 2018

```

field 编号          type int      index
field 年月          type char    5
field 姓名          type char    6      index

```



```

field 基本工资      type float
field 工龄工资      type float
field 副食补贴      type float
field 家属补贴      type float
field 主副补贴      type float
field 洗理费        type float
field 独生子女      type float
field 交通费        type float
field 书报费        type float
field 补其他1       type float
field 补其他2       type float
field 房租费        type float
field 国库券        type float
field 扣其他1       type float
field 扣其他2       type float
field 扣其他3       type float
end

```

在成功地编译gzk1之后，我们再尝试编译gzk2。

```
# dbbuild gzk2
```

由于前面已编译过gzk1，数据库“工资库”已经存在，故编译gzk2成功的话，也将只产生两个操作系统文件：数据文件（职级.dat）索引文件（职级.idx）。而不重新生成数据库的数据词典文件（工资库.dbd），因为该文件在编译gzk1时就已经生成。

2.4 用DBBUILD更改数据库结构

更改数据库结构也分两步完成。首先要重新编辑一个能生成新结构数据库的模式文件，然后用dbbuild对该文件进行编译。

例如，如果我们想在前面的工资库中增加一个新字段：扣其他4，类型为float，则先编辑好新的模式文件gzk1：

```

gzk1
database 工资库
file 工资
field 编号      type int    index
field 年月      type char   5
field 姓名      type char   6    index
field 基本工资  type float
field 工龄工资  type float
field 副食补贴  type float
field 家属补贴  type float
field 主副补贴  type float
field 洗理费    type float
field 独生子女  type float

```