

VAX DBMS

网状数据库设计与维护

周胜奎 严蒿芬 张紫夏 编著

地震出版社

87221

354

VAX DBMS
网状数据库设计与维护

周胜奎 严蔼芬 张紫夏 编著

地震出版社

1992

(京) 新登字 095 号

内 容 提 要

本书在全面介绍网状数据库基本概念及如何进行系统分析和概念设计的基础上,详细叙述了如何用 VAX DBMS 进行数据库的实现和维护问题。同时还介绍了应用开发问题和数据操作语言等。

本书对于从事信息系统、数据库实际工作的专业人员是一本很好的参考书,对于 VAX DBMS 用户是一本简明指南,本书还可作为高等学校有关专业数据库课程的参考书。

VAX DBMS

网状数据库设计与维护

周胜奎 严满芬 张紫夏 编著

责任编辑: 傅苏华

*

地质出版社出版

北京海淀区民族学院南路9号

中国地质大学轻印刷厂印刷

新华书店北京发行所发行

全国各地新华书店经售

*

787×1092 1/16 14.5 印张 371 千字

1992 年 4 月第一版 1992 年 4 月第一次印刷

印数 0001—2000

ISBN 7-5028-0558-3/TP·1

(947) 定价: 8.70 元

目 录

绪言	(1)
第一章 基本概念	(3)
1.1 实体及其属性	(3)
1.2 数据项、记录及文件	(3)
1.3 记录之间的逻辑关系与系	(5)
1.4 域和页面	(9)
1.5 记录存放方式	(10)
1.6 系的特性	(10)
1.7 数据库的结构图	(13)
1.8 VAX DBMS 数据库的主要构成	(13)
第二章 概念设计	(17)
2.1 用户需求的收集	(17)
2.2 用户要求的分析	(18)
2.2.1 确认事务处理的任务	(18)
2.2.2 确认实体	(20)
2.2.3 确定实体的属性	(20)
2.2.4 确立实体之间的关系	(21)
2.3 数据分析	(22)
2.3.1 收集与业务有关的资料	(22)
2.3.2 规范实体及其属性	(23)
2.4 事务处理分析	(28)
2.4.1 跟踪处理路径	(29)
2.4.2 模拟事务处理	(30)
2.4.3 合并所有处理路径图	(31)
2.4.4 归档处理	(32)
2.4.5 实体记录值数量估计	(33)
第三章 数据库的实现	(34)
3.1 数据库管理系统的选择	(34)
3.2 数据模型的转换与修改	(35)
3.3 数据库模式的定义 (逻辑设计)	(38)
3.3.1 模式和域的命名	(38)

3.3.2 记录的定义	(39)
3.3.3 系的定义	(41)
3.3.4 数据项值的有效性检验	(42)
3.4 数据库存贮模式的定义 (物理设计)	(43)
3.4.1 缺省存贮模式	(43)
3.4.2 存贮记录	(44)
3.4.3 优化存贮系	(46)
3.4.4 记录存放方式和系的存贮方式	(50)
3.5 子模式的定义 (用户窗口设计)	(51)
3.5.1 重新命名模式中的名称	(51)
3.5.2 定义子模式的区域	(52)
3.5.3 定义子模式的记录	(52)
3.5.4 定义子模式的系	(54)
3.5.5 汽车出租数据库的一个子模式	(54)
3.6 安全模式的定义	(54)
3.6.1 安全模式的结构	(55)
3.6.2 缺省安全模式	(56)
3.6.3 编写基准安全模式	(58)
3.6.4 定义域的许可	(59)
3.6.5 定义记录的许可	(59)
3.6.6 定义系的许可	(60)
3.7 模式编译与公共数据字典 (CDD)	(61)
3.7.1 模式编译	(61)
3.7.2 CDD 的结构	(62)
3.7.3 CDD 的安全和保护	(63)
3.7.4 CDD 的管理	(65)
3.7.5 CDD 的检验和修补	(68)
3.8 数据库的建立和删除	(70)
3.8.1 建立数据库	(70)
3.8.2 指定保密模式	(75)
3.8.3 删除数据库	(76)
3.8.4 清除数据库的内容	(77)
3.9 数据的装入和卸出	(77)
3.9.1 装入数据	(77)
3.9.2 卸出数据	(83)
第四章 应用设计	(87)
4.1 应用开发简述	(87)
4.1.1 应用开发周期	(87)

4.1.2 应用设计与数据库设计的关系	(87)
4.1.3 开发环境设置	(88)
4.2 加锁	(91)
4.2.1 VAX DBMS 中的锁	(91)
4.2.2 存贮域加锁	(91)
4.2.3 记录级加锁	(93)
4.2.4 锁的短缺	(95)
4.2.5 锁机制的优化	(95)
4.3 数据库查询实用程序(DBQ)	(96)
4.3.1 启用 DBQ	(96)
4.3.2 访问数据库	(97)
4.3.3 定位和检索记录	(101)
4.3.4 使用当前值指针	(107)
4.3.5 修改数据库中的记录	(112)
4.4 DML 程序设计的有关问题	(118)
4.4.1 信息的获取	(118)
4.4.2 DML 预编译器的使用	(120)
4.4.3 错误和异常处理	(128)
4.4.4 程序优化	(132)
4.4.5 使用程序方式的 DBQ 子程序	(132)
4.5 VAX DATATRIEVE 的主要功能	(133)
4.5.1 访问数据库	(133)
4.5.2 检索记录	(134)
4.5.3 定义过程	(135)
4.5.4 编写报告	(137)
4.5.5 形成图形	(142)
第五章 数据库的维护	(145)
5.1 监视数据库	(145)
5.1.1 监视进程	(145)
5.1.2 活动用户信息列表	(148)
5.1.3 数据库特性的显示	(149)
5.1.4 数据库的统计监视显示	(152)
5.2 打开 / 关闭数据库	(155)
5.3 数据库的后备与重装	(156)
5.3.1 数据库的后备	(156)
5.3.2 重装数据库	(157)
5.4 日志和恢复	(159)
5.4.1 映象前日志及自动恢复	(159)

5.4.2 映象后日志	(160)
5.4.3 用映象后日志文件恢复数据库	(161)
5.5 数据库的内部表示	(162)
5.5.1 数据库存贮页面的结构	(162)
5.5.2 具有瞬象能力的数据库存贮页面	(174)
5.5.3 空间域管理 (SPAM) 页面的结构	(175)
5.6 完整性检验及局部修补	(176)
5.6.1 检验数据库结构的完整性	(177)
5.6.2 数据库页面的修补实用程序 (DBALTER)	(179)
5.7 数据库性能分析	(187)
5.7.1 分析记录的存贮问题	(189)
5.7.2 利用转贮分析数据库的性能	(195)
5.7.3 利用统计监视分析数据库的性能	(196)
5.8 数据库的修改	(199)
5.8.1 修改模式	(200)
5.8.2 修改存贮模式	(202)
5.8.3 加入、删除和修改子模式	(202)
5.8.4 加入、删除和修改安全模式	(203)
5.8.5 增加域	(203)
5.8.6 修改数据库的特性	(205)
附录	(209)
A-1 汽车出租数据库的模式定义	(209)
A-2 汽车出租数据库的存贮模式定义	(211)
A-3 汽车出租数据库的一个子模式	(213)
B-1 部件数据库的 BACHMAN 图	(215)
B-2 部件数据库的模式定义	(215)
B-3 部件数据库的存贮模式	(219)
B-4 部件数据库的缺省子模式	(220)
参考资料	(225)

绪 言

在现代社会中,对信息的管理已变得越来越重要,如银行管理、商店管理、订票系统、情报资料检索等等,都需要处理大量的信息。同时,由于计算机技术的发展,使大量的信息通过计算机进行自动化管理也有了可能。数据库技术就是为了满足对大量而相互又有联系的信息进行有效的管理而发展起来的,而且正在发展和完善之中。到目前为止,数据库管理系统主要有三大类:一类是层次型数据库管理系统;另一类是网状数据库管理系统;第三类是关系数据库管理系统。这三类数据库管理系统各有其特点。一般说来,层次型数据库管理系统使用方便,数据结构直观,但是它难以构成复杂的数据模型,或者冗余度较大。网状数据库管理系统能够处理复杂的数据模型,并且冗余度较小,但是由于实体之间的联系用系来建立,数据结构比较复杂,建立数据库和对数据库进行管理都比较困难。数据库建成后,由于网状数据库中的记录之间有指针相连,所以检索速度比较快。因此,对于那些结构比较复杂而又不经常改变结构的数据模型,用网状数据库管理系统比较合适。关系型数据库中实体之间的联系是通过键值建立的,每个实体是相对独立的,所以建库和对库的修改都比较方便、简单,但是有关数据必须通过键值查找,不如网状数据库用指针相连检索快。因此,关系型数据库管理系统适用于那些检索速度并不重要,或者数据库规模较小而结构又比较简单的数据模型。到底采用什么样的数据库管理系统取决于实际问题的需要及本单位的技术力量,如果用网状数据库管理系统建库,一般需要专门的数据库管理员。

本书介绍如何使用网状数据库管理系统。大家知道,计算机软件的使用是一门实践科学,只有通过具体例子或通过具体使用才能容易地理解各种概念及各种命令和语句的使用方法。所以本书以 VAX (DBMS) 网状数据库管理系统为对象,比较全面地介绍如何进行概念设计、逻辑设计、物理设计,以及数据库建立后如何对数据库进行存取、维护和性能分析等等。

VAX DBMS 是遵从 CODASYL (Conference on Data System Language) 规范的通用数据库管理系统,它建立在 ANSI (美国国家标准协会) 数据库定义委员会 1981 年 3 月工作文件的基础上。可以说,它是标准的网状数据库管理系统,掌握了它,其他有关网状数据库管理系统的使用也就迎刃而解。

网状数据库的结构与传统的文件结构有较大差异,因此有一些新的概念。对于那些不熟悉这些概念,即使已能熟练地用文件编写程序的程序员,往往也会感到困难。所以本书的第一章首先以通俗的形式介绍文件与网状数据库在结构上的差异,并介绍网状数据库中的一些基本概念。

数据库的建设是一项复杂而严密的工作,它的目标是用计算机完成某个系统或某个部门的各种处理任务,不能有丝毫的差错。所以首先要仔细考虑把哪些信息送到计算机中去,这些信息又如何进行组织,使它尽量不重复,所谓尽量减少数据的冗余度,而调用起来又十分方便,能充分满足各种处理的要求。这些都是数据库的概念设计要做的工作。概

念设计结果得到了一个数据模型。本书的第二章详细地介绍如何进行概念设计。

有了数据模型以后，就要在计算机上具体实现数据库的建设，即数据库的逻辑结构设计、物理设计（存贮模式）、用户窗口设计（子模式）及安全模式的设计，然后建立数据库。第三章中介绍如何利用 VAX DBMS 网状数据库管理系统进行这些设计和建立数据库。

建立数据库的目的就是让计算机自动完成日常的各种业务处理，所以数据库建立后就要编写各种应用程序，利用这些程序完成各种处理任务。程序设计中数据的存取是通过数据库操纵语言（DML）进行的。DML 与其他高级语言（如 BASIC、FORTRAN、PASCAL 或 C 语言等）一起使用，这些高级语言被称为宿主语言。本书第四章介绍如何借助这些高级语言对数据库进行存取。VAX DBMS 还提供了一种对话式数据库查询实用程序 DBQ，它可以作为查看和调试数据库的一种手段。另外，在第四章的一开始还介绍了应用开发的环境及给数据库加锁等内容。

本书的最后一章，即第五章介绍如何进行数据库的日常维护及性能分析。数据库投入使用以后，由于电源故障、软件错误或误操作等，都会造成数据库的破坏，所以必须对数据库进行后备和检验，以便发现错误及时进行恢复。另外，由于对数据库进行不断的存取，其性能逐步退化，所以还必须经常对数据库进行性能分析，找出退化的原因，并进行改进，也就是对数据库进行修改、卸出并重装数据或重新构造数据库等。

本书力求语言通俗易懂、简炼，而且比较全面地介绍使用网状数据库管理系统的全部过程。因此，对于有志学习网状数据库的同志将是一本较系统的入门书，对于 VAX DBMS 的用户将是一本简明的指南。

第一章 基本概念

1.1 实体及其属性

客观世界中存在着各种各样的物体、事件、人物、地点、组织等等，我们把它们统称为“事物”。对于某一个部门或某一个单位来说，也存在着各种各样的“事物”。例如，汽车出租公司内有职工、汽车、订车单、发票及各地的出租汽车站等等。信息是对“事物”的具体描述和反映。在数据库所讨论的信息范畴内，实体(ENTITY)与某一类“事物”相对应。例如，汽车出租公司中的职工可以用一个实体来表示，汽车也可以用一个实体来表示，发票、订车单及出租汽车站都可以用实体来表示。

每一个“事物”都有一定的特征。例如汽车，它有型号、颜色、出厂日期、生产厂家、许可证号等等。我们用实体的属性(ATTRIBUTE)来描述“事物”的特征。如职工这个实体，它的属性有姓名、年龄、性别、职务、文化程度等等。

实体是某一类“事物”的总称，对于实体中某一个特定的“事物”称为实例 (INSTANCE)，实例的特性称为属性值 (ATTRIBUTE VALUE)。

1.2 数据项、记录及文件

把实体及属性在计算机中进行存贮时，我们使用了如下术语：

数据项 (ITEM) ——它是计算机数据存贮中被命名的最小单元，与实体的某一属性相对应，数据项值与属性值相对应，它可以用一个变量(实型、整型或字符型等等)进行存贮。例如，出租汽车公司中职工这个实体，每个职工登记了如下内容 (表 1-1)：

表 1-1 职工人事记录

姓 名	年 龄	性 别	职 务	文化程度
张 三	52	男	经理	大学毕业
李 四	24	女	秘书	高中毕业
王 五	38	男	司机	高中毕业
...	...	...	...	...

表 1-1 中职工实体具有五个属性。每一个实例可用五个变量进行存贮，姓名用一个字符变量存贮，年龄用一个整型变量存贮，性别、职务及文化程度等都分别用一个字符型变量存贮。

记录 (RECORD) ——它与实体相对应，是某一实体所有属性相对应的数据项的集合。上例职工实体中，五个数据项构成了“职工”记录。记录中所有数据项的名称及其类型构成的集合，称记录型 (RECORD TYPE)。因此，“职工”记录的记录型是：

(姓名，年龄，性别，职务，文化程度)

记录中某一个特定内容，称为记录值 (ACCURRENCE)，如“职工”记录中的：

(张三，52，男，经理，大学毕业)

是一个记录值。对于某一个记录来说，记录型只有一个，而记录值可以有任意多个，本书中用矩形表示一个记录（它与某一实体相对应），用圆来表示记录值，如图 1-1 所示。

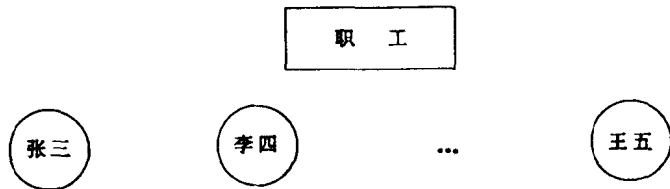


图 1-1 记录型和记录值

文件 (FILE) ——对于某一个记录型，所有（或者部分）记录值的集合，构成了文件。每一个文件都有一个文件名，文件名及与此文件有关的信息都登录在计算机的管理系统中。使用时，根据文件名确定有关信息，然后再找到有关记录值。

不同记录型的记录值也能构成一个文件，但是本书中所提到的文件(指数据文件)，其记录值都是同一个记录型的。也就是说，同一个文件中的记录值都具有相同的记录格式。

由于文件在计算机存储器中存放的组织方式不同，形成了不同类型的文件。不同类型的文件其检索方法完全不同。下面我们介绍几种最常用的文件类型：

- 顺序文件 (SEQUENTIAL FILE)：记录值先后顺序地存放在某一物理存储器中，每个记录之间用特定的符号分割，如图 1-2 所示。它的结构比较简单。检索某一个记录值时，必须从头找，直到你所需要的记录为止。要加入记录，只能放在最后。如果想中间加入或删除一些记录，就得重新构成新的文件。顺序文件的优点是结构简单，不需要附加的存贮开销，连续、大量读取时速度也比较快。

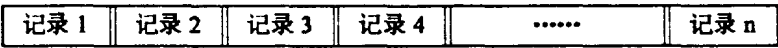


图 1-2 顺序文件的存放方式

- 相对文件 (RELATIVE FILE)：记录值按某一顺序存放在某一物理存储器中，记录值的长度是固定的，根据顺序号可以计算出某一记录在存储器中的位置。因此通过顺序号可以随机地进行存取。

- 杂凑文件 (HASHED KEYED FILE)：按某一计算方法（杂凑法）对记录值中的键值进行计算，根据计算结果确定它在存储器内的存放位置。对于计算出同一地址的多个记录按顺序存放，或放在不同的地方，再用指针把它们连在一起。键 (KEY) 是记录中的一个数据项，它能在文件中唯一地标识这个记录值。键也称关键字，对于杂凑文件可以通过键值进行存取。

- 索引文件 (INDEXED FILE)：不仅把记录值存放在计算机的物理存储器内，另外还建立一个索引表，如图 1-3 所示。索引表中的 $K_1, K_2, \dots, K_n$ 为按某一顺序排列的键值， $R_1, R_2, \dots, R_n$ 为与这些键值相对应的记录值在存储器中的位置， $R_1, R_2, \dots, R_n$ 也就是指针 (POINTER)，它在索引区中指出了记录值在存贮区中的地址。由此

可见，索引文件除存放记录值本身以外，还得存放索引表，因此系统的存贮空间开销就大大增加。但是由于有了索引表，检索速度大大提高，增加或删除记录也非常方便，不必重新构造记录值的存贮区。在有的文件管理系统中，可以对不同的数据项建立多种索引途径。另外，为了提高索引速度，可以建立多级索引，也就是对索引表再建立索引表。

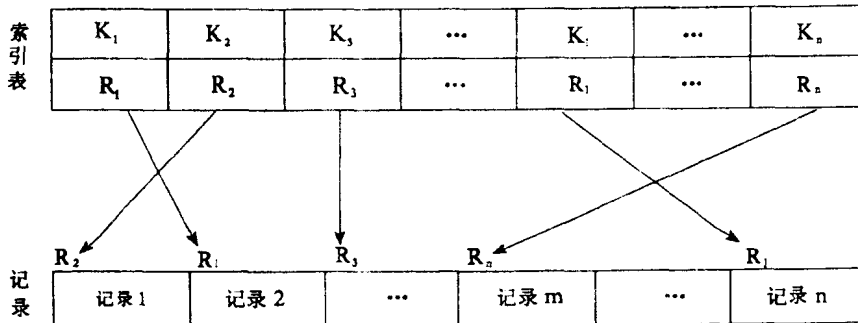


图 1-3 索引文件结构

从以上的文件结构中可以看出，不同文件的记录值之间是没有联系的。但是在处理问题时，往往要根据一个文件中的某个记录值，到别的文件中寻找与此有关的其他记录值。同时，为了表示不同文件中的记录值之间的关系，不得不增设另一个文件记录值中已有的数据项，这样数据的冗余度就大大增加。网状数据库就是要从结构上解决这些问题，把不同文件中的有关记录值用指针联系在一起。这样很容易找到有关的不同记录类型的记录值，也减小了冗余度。当然，由于设立了指针，系统的存贮空间开销也就增加。下面我们介绍实体之间的关系，表达方式及记录值在网状数据库中是如何存放的。

1.3 记录之间的逻辑关系与系

实体与实体之间有各种各样的关系。例如，学校中有教研组，教研组长，教师，班级，学生及课程等实体。每个教研组有一个主任，一个教研组内有多个教师，一个班级中有若干学生。每年学校开设各种课程，如外语、数学、物理等等。不同的学生参加不同课程的学习，每门课程又有不少同学参加学习等等。也就是说不同记录之间的记录值有各种各样的关系。这些属于不同记录类型的记录值之间的关系，有如下三种：

- 一对一关系 (1:1)：一个记录（或实体）中的某一记录值只能与另一记录（或实体）中的一个记录值有关系；
- 一对多关系 (1:M)：一个记录（或实体）中的某一个记录值与另一记录（或实体）中的多个记录值有关系；
- 多对多关系 (M:N) 一个记录（或实体）中的某一个记录值与另一记录（或实体）中的多个记录值有关，而另一记录（或实体）中的某一个记录值又与此记录（或实体）中的多个记录值有关系。

例如，一般来说，教研组与教研组长之间是一对一的关系，因为一教研组只能有一个主任，假定一个教师只能担任一个教研组的主任。班级与学生之间是一对多的关系，因为一个班级有多个学生，一个学生只能属于一个班级。学生与课程之间的关系是多对多的

关系，因为一个学生参加多个课程的学习，某一课程又有多个学生学习。这些关系用图 1-4 表示。

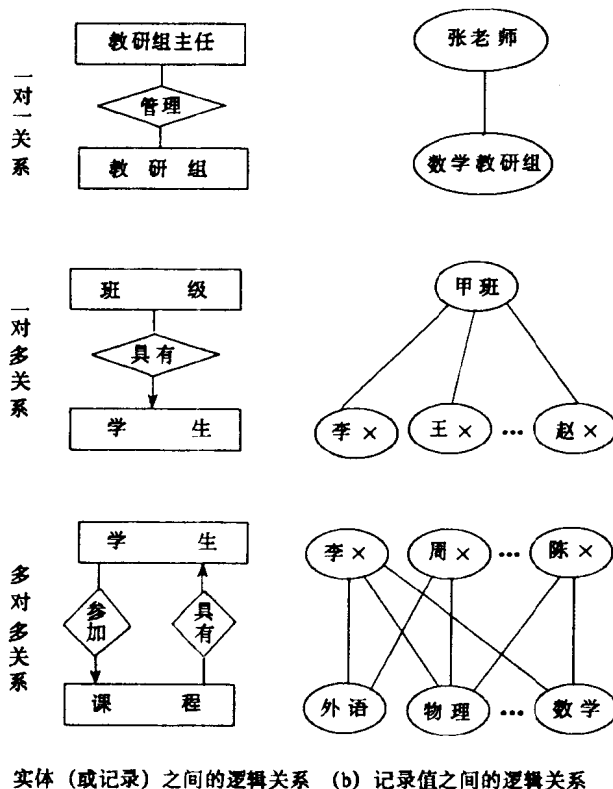


图 1-4

一个实体不仅可以与另一个实体建立关系，也可以与多个实体建立关系；二个实体之间不仅可以建立一个关系，而且可以建立多个关系，这样多个实体之间就形成了复杂的网状关系。这在后面章节的例子中就能看到。

记录值之间的联系靠什么来实现呢？靠指针！用指针把有关的记录值连接在一起，如图 1-5 所示。图中 (a) 表示教研组实体与教师实体之间存在一种一对多的关系。(b) 表示数学教研组记录值如何通过指针把数学教研组的教师记录值连接在一起。这样，当我们要找数学教研组的教师记录值时，先在教研组的记录中找到数学教研组记录值，然后通过数学教研组记录值与教师记录值之间所建立的指针找到数学教研组的老师记录值。

把两个记录型之间的有关记录值用指针连接在一起形成一个队列，这些记录值及指针构成了系值 (SET OCCURRENCE)，两个记录型之间存在的这种关系，它的名称及有关特性（如记录值的连接方式，新记录加入到系中的顺序，记录值的存放方式及记录值的属藉等，这些特性将在后面详细说明）构成了系型 (SET TYPE)，系型和系值统称为系 (SET)。与记录型和记录值一样，两个记录之间的某一关系，系型只有一个，而系值可以有任意多个，系在网状数据库中是最重要的概念。

在一对多的系中，“一”的一边称为系主 (OWNER) 或主记录；“多”的一边称为成员 (MEMBER) 或下属记录。一个记录值为了加入一个系值，必须增加三个指针：

- 指向此系值中后面一个记录值所在位置的指针 (NEXT POINTER);
- 指向此系值中前面一个记录值所在位置的指针 (PRIOR POINTER);
- 指向此系值中主记录所在位置的指针 (OWNER POINTER).

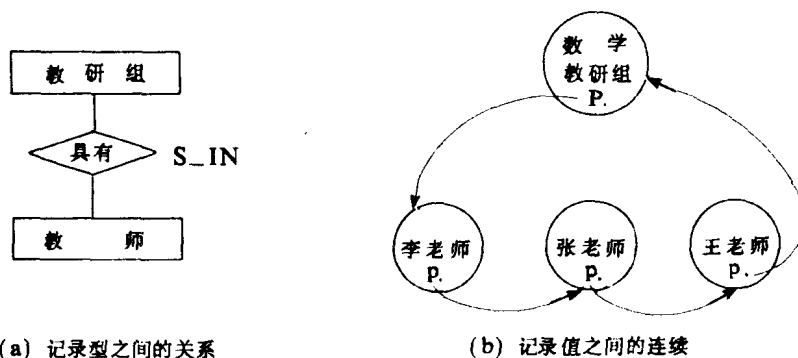


图 1-5

图 1-6 表示系值中三类指针的指向。以后的章节中为了简便起见只画一个指针 (NEXT POINTER)，其实都有三个指针，个别情况下有的指针是空的。

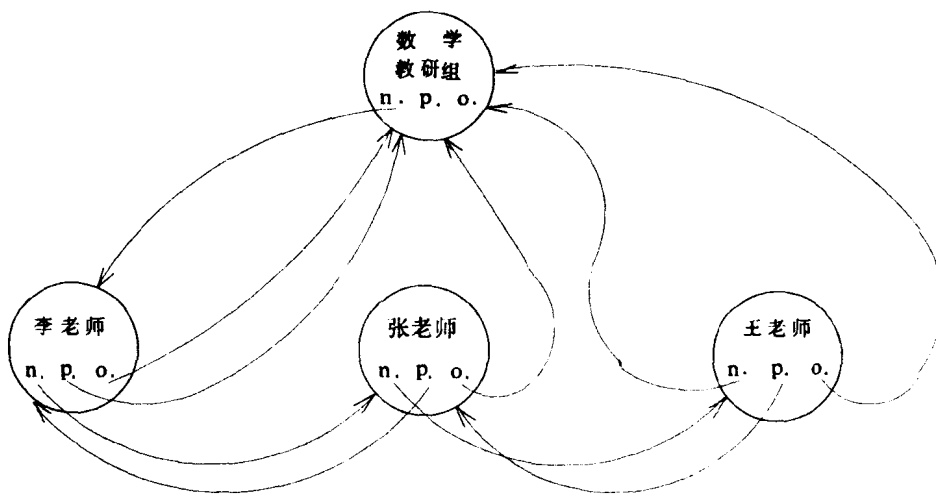


图 1-6 系值之间的指针

系必须遵循如下规则:

- 不管是主记录还是成员记录，在一个系型中只能参加一个系值;
- 每一个系值必定有一个系主，成员记录可以任意多个，也可以是空;
- 一个记录型可以参加多个系型，它可以作多个系型的系主，它也可以是多个系型的成员，它既可以是一些系的系主，同时又是一些系的成员，但是它不能是一个系的系主，同时又是这个系的成员记录;
- 一个系型必须有一个也只能有一个系主，成员至少是一个，也可以有多个。系主可以是系统 (SYSTEM)，这意味着可以通过数据库管理系统直接进入这个系的成员记录。这种

系称为奇异系，亦称无首系，其表示方式如图 1-7 所示，其中 ALL_D 为系名。



图 1-7 奇异系

根据以上规则，一对一实体之间，谁作为系主都没有关系。多对多的实体间关系，可以增加一个连接实体，使它变成二个一对多的系。如学生与课程是多对多的关系，增加一个“选修课程”记录，使学生与“选修课程”记录之间及课程与“选修课程”记录之间都是一对多的关系，见图 1-8。

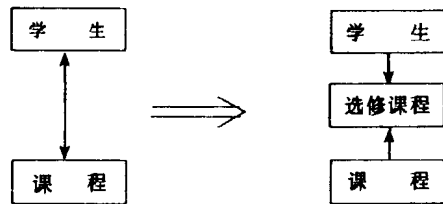


图 1-8

“选修课程”记录可以只包括学生记录和课程记录中的二项键值，尽量减少冗余度。变化后的记录值之间的关系表示在图 1-9 中。

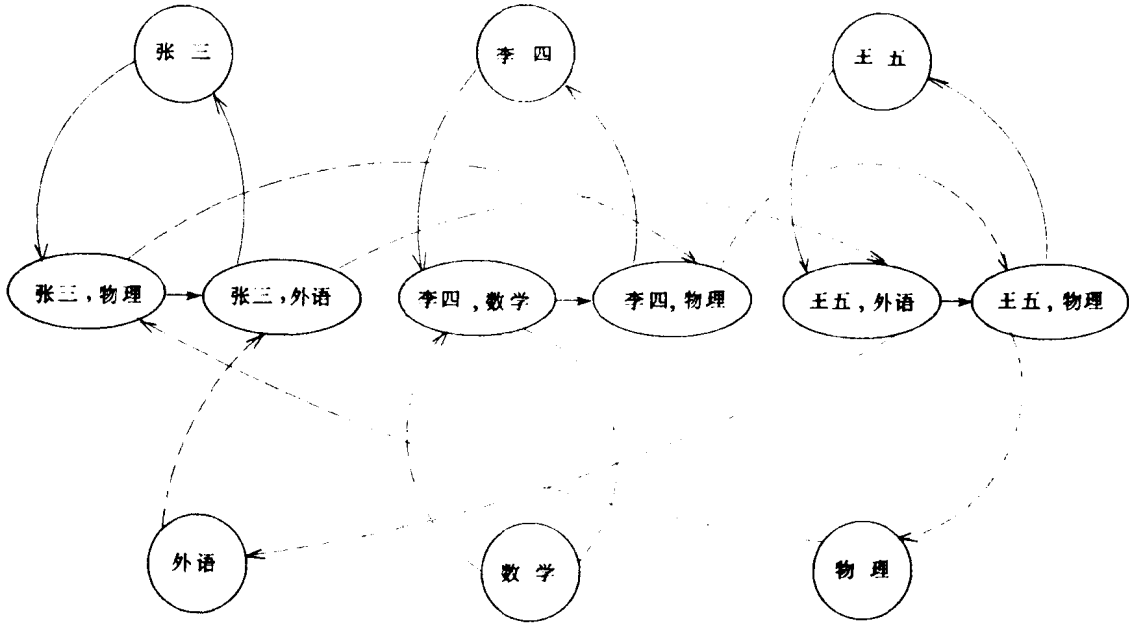


图 1-9

1.4 域和页面

在网状数据库中进行数据存贮时，先把计算机的存贮单元划分成若干个域 (AREA)，每个域分成若干个页面 (PAGE)，每个页面又分成若干行 (LINE)，域和页面的大小由用户在建立数据库时定义，行的长短与存贮在这个行中的记录值有关，一个行放一个记录值。图 1-10 是域、页面及行的示意图。

第 1 页	第 2 页	...	第 n 页
自由空间	自由空间	...	自由空间
第 i 行	第 j 行		第 k 行
...	...		...
第 1 行	第 1 行		第 1 行
第 0 行	第 0 行		第 0 行

图 1-10

为了管理和存贮方便，一个数据库可以建立多个域，每一个域有一个域名。一个记录型的记录值可以存放在一个域中，也可以存放在不同的域中。一个域可以只放一个记录型的记录值，也可存放多个记录型的不同记录值，到底如何存放合适，这是数据物理设计要解决的问题。有关数据库设计问题请看第三章。

域除域名外，在它的建立过程中按先后顺序还建立了一个域号。所以记录值在网状数据库中的位置被域号、页面号和行号完全确定。这三个数字称为数据库码 (DBKEY，简记为 DBK)。例如，某个记录值的数据库码为 2:1:3，说明这个记录值被存放在第二个域的第一个页面中，其行号为 3。所以记录值的数据库码具有逻辑地址的功能。根据它来直接存取相应的记录值，当对数据库中的记录值进行存取时，数据库管理系统中用指针来保存此记录值所在的位置 (用 DBKEY 表示)，直到下次进行存取为止。这种用来保存最后被存取过的记录值位置的指针称为当前值指针 (CURRENCY INDICATOR)。当前值指针有如下几类：

- 运行单元当前值指针：一个数据库只有一个，保存这个数据库中正在存取或最后存取过的记录值的位置；

- 域的当前值指针：每一个域有一个，保存这个域中正在使用或最后使用过的记录值位置；

- 记录当前值指针：每个记录型有一个，保存这个记录型内最后被使用过的记录值的位置；

- 系的当前值指针：每个系有一个，保存这个系中最后被存放过的记录值的位置。这些指针的值可以是下面三种的一个：

- (1) NULL——空的，说明这个指针尚未被使用过；

- (2) DBK——一个数据库码，它是最后被存取过的记录值的位置；

(3) DBK(P)——一个带标志的数据库码，它也是最后被存取过的记录值的位置，不过这个记录值已被删除掉了，也就是说最后访问这个记录值时，删除了这个记录值。

1.5 记录存放方式

当记录值存放到域中时，可以把它们集中存放，也可以分散放在域内，这决定于记录的存放方式 (RECORD PLACEMENT MODE)。在记录的存贮模式定义中用 PLACEMENT 子句进行说明。

• 群集方式 (CLUSTERED MODE): 把系的成员记录值依次放在主记录值附近，如图 1-11 所示。按这种存放方式，域中某些页面很拥挤，而某些页面往往是空的。这种存放方式的好处是根据主记录检索成员记录时，存取速度比较快，I/O 次数比较少。按这种存放方式，当系值比较大时，再加入一个新的记录值就比较麻烦，因为它要查找多个页面才能确定存放的位置，所以群集存放方式不适用于经常要插入的系型。

页面 1	页面 2	页面 3	页面 4	页面 5	...
	成员 1, 4			成员 2, 4	...
	成员 1, 3			成员 2, 3	
	成员 1, 2			成员 2, 2	
	成员 1, 1			成员 2, 1	
	主记录 1			主记录 2	
(系值 1)			(系值 2) ...		

图 1-11 群集存放

• 分散方式 (SCATTERED MODE): 不管主记录值在什么位置上，把成员记录值随机地分散在域中的各个页面上，如图 1-12 所示。按这种存放方式，记录均匀地分散在域内各页面上，因为每个页面上尚有部分自由空间，所以加入新的成员记录比较方便，但是检索某一系值时需要较多的 I/O 次数。

页面 1	页面 2	页面 3	页面 4	页面 5	...
					...
成员 1,3	成员 2,3			成员 1,2	
成员 2,1	主记录 1	成员 2,2	成员 1,1	主记录 2	
(系值 1 和系值 2)					

图 1-12 分散存放

1.6 系的特性

前面已经讲过，系值中记录值是用指针一个接一个连接在一起的，当系值中记录值很多