

数据库及其管理系统的 结构 and 设计

苏洪元 编

北京科海培训中心

前 言

数据库是数据信息管理的最新技术,广泛地应用于国民经济、文化教育、情报等领域,为计算机应用开辟了广阔的天地。

我国已经引进了相当数量的大中型计算机,其中许多是IBM公司的计算机,且大多配有数据库管理系统。IBM最成熟的数据库管理系统之一是DL/I(即IMS/DB部分)。它是一种层次结构的数据库管理系统,目前已有许多用户安装了DL/I系统。

DL/I是一个性能好、使用、维护要求较高的数据库管理系统,只有在熟悉该系统的结构、实现逻辑及处理方法之后,才能用好它,充分发挥该系统的作用。本书是根据编写者多年使用、维护DL/I的经验、在解剖IMS/DB系统、分析该系统的源程序之后的总结。重点描述了DL/I系统的内部结构和实现逻辑。前四章介绍DL/I数据库(段、块和数据库记录、数据集)的物理结构,第五章着重介绍DL/I系统中各主要模块间的关系和功能、各主要模块的处理方法和流程,并介绍了数据库系统各主要控制块间的关系和内容。以便读者对该系统有一个较全面、深刻的理解,从而在如何用好DL/I、充分发挥它的作用方面起点参考作用。同时,若读者在详细了解层次结构数据库管理系统(DL/I)之后,能较快地熟悉网状结构和相关(关系)结构的数据库管理系统,那是我的莫大荣幸。

本书可供从事数据库系统设计和数据库应用的技术人员阅读,也可供大专院校的高年级学生或研究生做选修课教材和毕业设计参考。

全书由清华大学计算机科学系杨德元教授审阅,并提出了许多有价值的修改意见。科海培训中心主任、高级工程师董洪皋、付主任华根娣也曾给予了很大的支持和帮助,在此一并表示谢意。

由于编写者的水平不高,经验不足,疏漏谬误之处祈请读者指正。

编者: 苏洪元

1988年2月

目 录

一、数据库物理结构及其描述	(1)
1.1 HDAM数据库的物理结构	(1)
1.1.1 HDAM数据库中段的结构	(1)
1.1.2 HDAM数据库块的结构	(2)
1.1.3 数据库记录的存储形式	(4)
1.2 HIDAM数据库的物理结构	(5)
1.2.1 HIDAM索引数据库中逻辑记录和块的结构	(5)
1.2.2 HIDAM 数据库记录的存储形式	(6)
1.2.3 HIDAM索引数据库记录的存储形式	(9)
1.2.4 HIDAM索引数据库的结构	(9)
1.3 HISAM数据库的物理结构	(17)
1.3.1 HISAM数据库中段的结构	(17)
1.3.2 HISAM数据库中逻辑记录结构	(17)
1.3.3 HISAM数据库中块的结构	(18)
1.3.4 HISAM数据库记录存储形式	(19)
1.4 数据库描述	(21)
二、数据库处理	(28)
2.1 程序说明块	(28)
2.2 数据库处理命令	(32)
2.2.1 数据库管理系统与应用程序之界面	(33)
2.2.2 段查找变量	(37)
2.3 数据库处理	(38)
2.3.1 数据库初始装入	(38)
2.3.2 检索 (RETRIEVE) 处理	(42)
2.3.3 插入处理	(43)
2.3.4 删除处理	(44)
2.3.5 置换处理	(46)
2.3.6 HDAM (也适用于HIDAM) 数据库的空间管理	(46)
三、多数据集组及第二索引	(51)
3.1 多数据集组	(51)
3.1.1 在HDAM和HIDAM数据库中使用多数据集组功能	(53)
3.1.2 数据库记录在多数据集组中的存储形式	(54)
3.1.3 定义多数据集组	(55)
3.1.4 使用多数据集组后的空间管理	(57)
3.2 第二索引	(57)

3.2.1	第二索引用到的段型	(59)
3.2.2	层次的改组	(61)
3.2.3	第二索引数据库的存储形式	(62)
3.2.4	指针段结构及段内各字段的用途	(63)
3.2.5	利用与系统相关的字段使指针段的键值唯一	(64)
3.2.6	在DB中说明第二索引的应用	(65)
3.2.7	索引省略	(67)
3.2.8	第二索引的维护	(68)
四、	逻辑数据库	(69)
4.1	逻辑关系	(69)
4.2	建立逻辑关系的方法	(72)
4.2.1	单向逻辑关系	(73)
4.2.2	双向物理配对逻辑关系	(75)
4.2.3	双向虚拟配对逻辑关系	(76)
4.3	逻辑关系中使用的指针	(76)
4.3.1	逻辑父段指针	(78)
4.3.2	逻辑子段指针	(79)
4.3.3	物理父段指针	(79)
4.3.4	逻辑孪生指针	(80)
4.4	交点数据	(81)
4.5	逻辑关系中使用的通路	(83)
4.5.1	逻辑子段	(84)
4.6	为使用逻辑关系的那些物理数据库中的段型定义定序字段	(84)
4.6.1	为真逻辑子段定义定序字段	(84)
4.6.2	使用逻辑关系后控制块之间的关系	(84)
4.7	在物理数据库描述中说明逻辑关系	(85)
4.8	逻辑数据库描述	(89)
4.8.1	定义逻辑数据库的规则	(93)
4.9	针对逻辑关系选择置换、插入和删除规则	(97)
4.9.1	置换规则	(98)
4.9.2	插入规则	(98)
4.9.3	删除规则	(98)
4.9.4	删除段的可达性	(100)
五、	数据库管理系统结构及设计	(102)
5.1	系统概述	(102)
5.1.1	数据库管理系统的设计需求	(102)
5.2	IMS/VS中数据库管理系统结构及设计	(104)
5.2.1	系统公共服务控制	(104)
5.2.2	数据库应用/调度界面	(104)

5.2.3	数据库调用分析	(106)
5.2.4	数据库调用处理	(106)
5.2.5	数据库调用资源管理	(106)
5.2.6	数据库存取方法界面	(107)
5.2.7	数据库公用程序	(107)
5.3	程序结构	(109)
5.3.1	数据库应用/ 调度界面	(109)
5.3.1.1	语言界面和程序请求处理	(110)
5.3.1.2	控制块建造	(112)
5.3.1.3	控制块装入	(113)
5.3.2	数据库调用分析	(114)
5.3.3	数据库调用处理	(114)
5.3.3.1	检索处理模块 (DFSDLR00)	(114)
5.3.3.2	装入/ 插入处理模块	(115)
5.3.3.3	删除/ 置换处理模块	(116)
5.3.4	数据库调用资源管理	(117)
5.3.4.1	缓冲区管理	(117)
5.3.4.2	HIDAM和HDAM数据库空间管理	(118)
5.3.4.3	打开/关闭处理	(119)
5.3.5	数据库存取方法界面	(118)
5.3.6	数据库公用程序	(122)
5.3.6.1	数据库重组/ 装入公用程序	(122)
5.3.6.2	数据库初始装入之考虑	(124)
5.3.6.3	物理数据库重组之考虑	(126)
5.4	数据保护措施	(126)
5.4.1	并发存取控制	(126)
5.4.2	程序异常结束时的复原	(128)
5.4.3	数据库复原公用程序	(128)
5.5	数据库管理系统各主要模块的执行流程	(130)
5.5.1	系统初始化	(130)
5.5.2	控制块建造模块执行流程	(136)
5.5.3	调用分析模块执行流程	(145)
5.5.4	装入/ 插入处理模块执行流程	(150)
5.5.5	数据库装入执行流程 (插入处理)	(168)
5.5.6	置换处理执行流程	(173)
5.5.7	删除处理执行流程	(175)
5.5.8	检索处理执行流程	(177)
5.5.9	索引维护程序执行流程	(195)
5.5.10	处理第二索引的执行流程	(205)

5.5.11	解决逻辑关系的执行流程	(206)
5.5.12	HD空间管理执行流程	(206)
5.6	数据库管理系统使用的控制块	(212)

第一章 数据库物理结构及其描述

目前，国际上普遍使用的数据库管理系统有三种：第一种是层次结构（HIERARCHICAL STRUCTURE）数据库管理系统；第二种是相关结构（RELATIONAL STRUCTURE）数据库管理系统；第三种是网状结构（NETWORK STRUCTURE）数据库管理系统。IBM公司的信息管理系统（INFORMATION MANAGEMENT SYSTEM WITH VIRTUAL STORAGE——以下简称IMS/VS）所提供的数据库管理系统是层次结构数据库的管理系统（IMS/VS由数据通讯系统和数据库管理系统两大部分构成）。它是一个很成功的数据库管理系统。根据寻找数据库记录的方法和处理数据库的方式，此系统把它所支持（管理的）的数据库分成不同类型，最常用的有HDAM（HIERARCHICAL DIRECT ACCESS METHOD）数据库、HIDAM（HIERARCHICAL INDEXED DIRECT ACCESS METHOD）数据库和HISAM（HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD）数据库。为适应寻找数据库记录所使用的方法和处理数据库及数据库记录时所用的方式，这三类数据库中段的结构和数据库在磁盘上的存储形式和物理块的结构都各有特殊的形式。由于任何数据（无论是档案中的或数据库中的）在磁盘中都是按块（物理记录）存放的，因而，数据库管理系统通常都把若干个数据库记录放在一个块中。因此，本章描述的数据库物理结构实际上是指段的结构、块的结构和数据库在磁盘中的存放形式。

1.1 HDAM数据库的物理结构

在实际应用中，许多用户都希望能快速、随机地存取数据库中各记录，又希望能快速、直接地存取数据库记录中的段，适应这种需求，数据库管理系统提供了HDAM数据库。HDAM数据库是通过散列程序（由系统提供或由用户编写）把数据库记录根段键值转换成地址而直接取得指定的数据库记录。数据库记录中的段都用指针连接。

1.1.1 HDAM数据库中段的结构

HDAM数据库中段的结构如图1—1所示

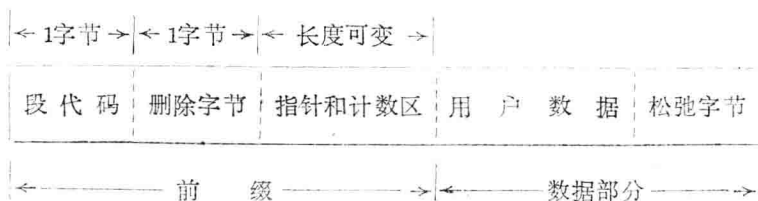


图 1-1 HDAM 数据库中段的结构

其中，指针和计数区由一个计数器（当需要时）和若干个指针组成，其格式如图1-2所示：

计 算 器	层或物理 次或子		物理 父段 指针	逻辑 子		逻辑 父段 指针	逻辑 子段		物理 子段
	前向 指针	后向 指针		前向 指针	后向 指针		第一 指针	最末 指针	

图 1—2 指针和计数区格式

图 1-2 展示了计数器和各类指针在前缀中的次序。是否需要计数器、需要多少类指针完全取决于用户在向数据库管理系统描述所设计的数据库时提供的参数。在实际应用中，每类指针可能不止一个，这时，总是按次序先放完前一类所有指针后才放紧接的那类指针。

数据库中每个段都由前缀和数据两部分构成。前缀部分是由数据库管理系统建立的，数据部分存放用户数据。松弛字节的作用是保证每个段都对准在半字边界上。

1.1.2 HDAM数据库块的结构

HDAM数据库从第二块开始，各块通常具有如图1—3所示的格式：

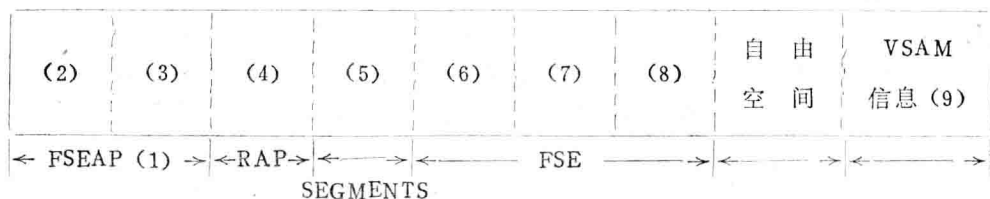


图 1-3 HDAM数据库块的结构

(1) FSEAP (FREE SPACE ELEMENT ANCHOR POINTER)：是自由空间描述段 (FREE SPACE ELEMENT) 的指针驻留点 (ANCHOR POINTER)，由两部分构成；

(2) 占据两个字节，用来存放到达第一个自由空间描述段 (FSE) 的位移量。但是，在存储单元映象块中，这两个字节置零；

(3) 是两个字节长的标志。如果这个块是存储单元映象块，则这两个字节具有非零值；若本块用于存放用户数据（也包括前缀部分），则将这两个字节置零。

(4) 根段指针驻留点 (ROOT ANCHOR POINTER，缩写为RAP) 区。本区域的长度完全依赖于根段指针驻留点的个数。每个块中应保留多少根段指针驻留点，由用户在向数据库管理系统描述数据库时指定。数据库记录的开始地址就存放在根段指针驻留点中。

(5) 此区域用来存放数据库记录中的各段（前缀和用户数据）。若是存储单元映象块，则存储单元映象就从此区域开始，一直延伸到本块的末端或VSAM信息始端。

(6) 占据两个字节，用来存放从本块开始到达下一个自由空间描述段的位移量。

(7) 占据两个字节，用来存放自由空间字节数（其中包括自由空间描述段本身八个字节）。

(8) 是四个字节长的任务标识符，指明释放这些空间的任务。

(9) 七个字节长的VSAM控制信息（VSAM是操作系统提供的一种存取方法）。

以上 (6)、(7)、(8) 三项组成一个自由空间描述段 (FREE SPACE ELEMENT)，一共占据八个字节，它的主要作用是描述本块中某一连续的自由空间。一个块中若干段自由空间就用若干个自由空间描述段描述，各个FSE (在一块中的) 建成一条链，链头地址存放在FSEAP中 (本块开始)。数据库管理系统自动地建立、维护自由空间描述段。

HDAM数据库中第一个块 (当用OSAM作为存取方法时) 是存储单元映象块 (BIT MAP BLOCK)，该块的结构如图1-4所示：

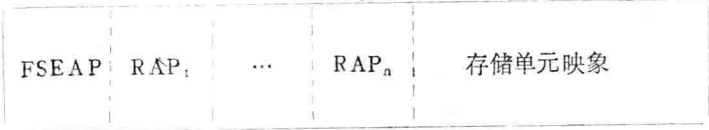


图 1-4 存储单元映象块格式

存储单元映象由一连串的二进制位构成。每一位顺次与HDAM数据库中的块相对应，它指出所对应的块内是否有足够的自由空间容纳数据库记录中最长的一个段。第一位对应于存储单元映象块本身，若该位为零，指示存储单元映象块中已无自由空间。其后的每一位表示每一后续块中是否有自由空间：若该位为 ‘1’ 时，表示该位所对应的块中有自由空间

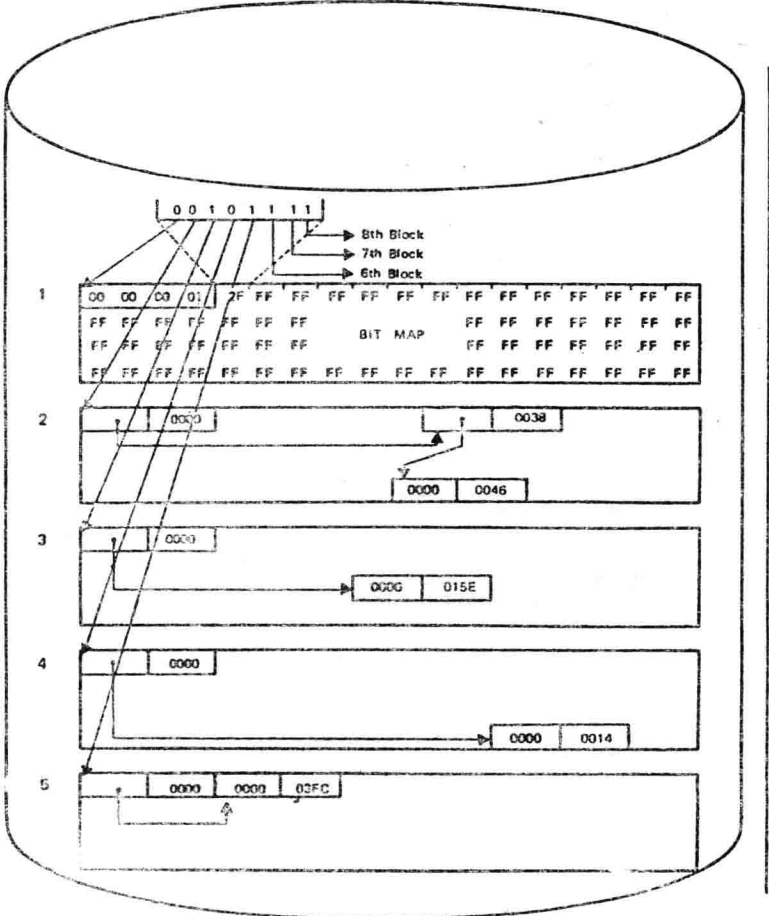


图 1-5 存储单元映象块与后继各块关系示意图

可存放数据库记录中最长的一个段，反之则表示无法容纳最长的一个段（虽有自由空间，但不足以容纳最长的段）。图1-5展示存储单元映象块与其后各块在磁盘上的分布。

1.1.3 数据库记录的存储形式

HDAM数据库由根段寻址区 (ROOT ADDRESSABLE AREA) 和溢出区 (OVERFLOW AREA) 两部分组成。根段寻址区中存放数据库记录中的根段，同时，也是数据库记录中从属段的主要存储区；溢出区用来存放那些不适合存放在根段寻址区中的段。用户在向数据库管理系统描述它所设计的数据库时，要指出根段寻址区的大小，还要指出允许一个数据库记录最多能在根段寻址区中连续使用多少字节。在数据库初始装入时，根段和每一个从属段依次连续地存进根段寻址区，一直到所用的空间量将超过指定的最大值时为止，该数据库记录中剩下的从属段就存放到溢出区中。在HDAM数据库中，数据库记录的装入次序可以任意，因为每一个数据库记录的存放地址都由散列程序根据它的根段键值计算出来的，但是每个根段的从属段必须按层次关系依次装入（在初始装入时）。HDAM数据库中的根段必须有一个定序字段，但不要求定序字段键值唯一。若要把HDAM数据库记录存进数据库，数据库管理系统将该记录的键值交给散列程序（由系统提供或由用

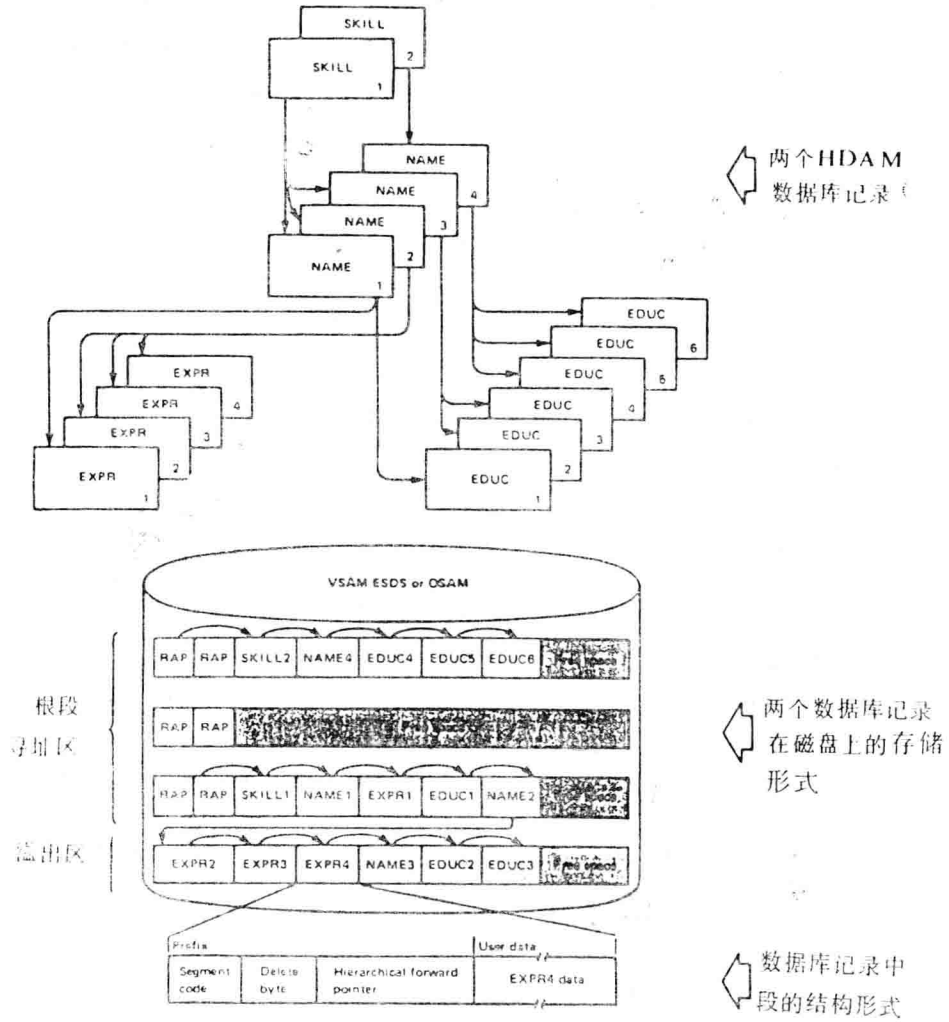


图 1-6 两个HDAM数据库记录在磁盘上的存储形式

户编写),它将此键值经‘杂凑’或其他计算技术转换成一个地址。该地址由相对块号和该块中的指针驻留点号组成。相对块号指出应把该记录存放到根段寻址区中哪一块,根段指针驻留点号指出应把该记录的开始地址存放到该块的哪个指针驻留点内。散列程序把算出的地址交给数据库管理系统,它按这个地址,把根段和尽可能多的从属段(不超过指定的最大值)存进根段寻址区的相应块内。把剩下的从属段(若有)存进溢出区。然后,把根段的开始地址放进该块内指定的指针驻留点中。图1-6展示两个数据库记录在磁盘上的分布情形。

这里要注意的是:根段指针驻留点只存在于根段寻址区的块内,溢出区中的块内没有根段指针驻留点,每块的开始四个字节是自由空间描述段指针驻留点。

若在整个根段寻址区中已没有可用空间能存放根段时,数据库管理系统就把那些根段放进溢出区,但这些根段的开始地址总是放在根段寻址区中指定块的指针驻留点内(即算出的块号或指针驻留点号)。

若在散列程序指定的块内没有足够的空间存放根段时,数据库管理系统将使用HDAM和HIDAM空间寻找算法寻找最合适存储这个根段的空间(这个算法在稍后描述,该算法的核心是寻找离指定块最近的可用空间存储该根段),但这个根段的起始地址总是放在由散列程序指定的那块中的指定指针驻留点内。

在实际使用中,散列程序很难把一个数据库中所有根段的键值都转换成唯一性的地址(唯一的块号与该块中唯一的指针驻留点号之合成),因而,常会把几个根段散列到同一块中的同一个指针驻留点,此时,那个根段指针驻留点中只存放一个根段的地址,其他几个根段经‘物理孪生’指针与这个根段连结起来,形成一条链。这几个根段之间通常都按键值从小到大排列。

1.2 HIDAM数据库的物理结构

在实际应用中,有不少用户既想能快速地随机存取数据库记录,又希望能快速地顺序存取数据库记录,适应这种需求,数据库管理系统提供了一种HIDAM数据库,它由两个数据库组成:用于存放全部用户数据的数据库叫做HIDAM数据库;另一个叫做HIDAM索引数据库,这个数据库只有一个段型。这个段型的数据部分中只包含HIDAM数据库中某一根段的键值,前缀部分的指针区中存放该根段在HIDAM数据库中的地址。

HIDAM数据库中段和块的结构与HDAM中段和块的结构相同(但根段指针驻留点区中最多只有一个根段指针驻留点),这里不再描述。

1.2.1 HIDAM索引数据库中逻辑记录 and 块的结构

HIDAM索引数据库的主要作用是使数据库管理系统能迅速地随机或顺序存取数据库记录,因此,HIDAM索引数据库只有一个段型。这个段的结构与普通段的结构稍有差别,而且随所用的存取方法而变化。所以,在HIDAM索引数据库的存储形式中都以逻辑记录形式表达,而不以段的形式表达。

指 针 (1)	段代码	删除字节	第一逻辑子段指针 (2)	根段的键值 (3)
------------	-----	------	-----------------	--------------

图 1-7 HIDAM索引数据库中逻辑记录结构 (ISAM/OSAM)

当用操作系统提供的ISAM/OSAM存取方法时，它的逻辑记录结构如图1-7所示。
这里，需作一些说明：

- (1) 指针：本字段占用三个字节。若使用ISAM（这是操作系统的一种存取方法），这个指针中存放一个OSAM物理记录（即块）的相对记录号，这个块中所包含的根段键值应比本记录中的根段键值稍低。若使用 OSAM，则这个指针中存放一个OSAM物理记录的相对记录号，这个块中所包含的根段键值稍高于本记录的根段键值。
 - (2) 是一个第一逻辑子段指针，它指向HIDAM数据库中具有本记录中所保留键值的根段。
 - (3) 由第一逻辑子段指针所指向的根段所具有的键值（在HIDAM数据库中）。
- 使用ISAM/OSAM作为存取方法时HIDAM索引数据库中块的结构如图1-8所示：

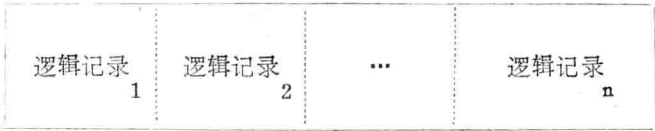


图 1-8 HIDAM索引数据库中块的结构 (ISAM/OSAM)

由图1-8可见，一个块中可存放多个逻辑记录。

使用VSAM作为存取方法时，HIDAM索引数据库中逻辑记录的结构如图1-9所示：

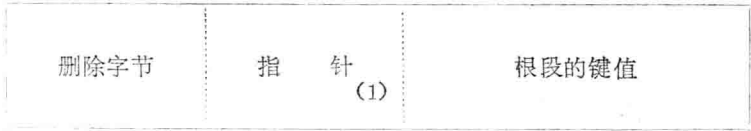


图 1-9 HIDAM索引数据库中逻辑记录结构 (VSAM)

这里：(1) 指针占用四个字节，它指向HIDAM数据库中与本记录中根段键值相同的那个根段。

使用VSAM作为存取方法时HIDAM索引数据库中块的结构如图1-10所示：

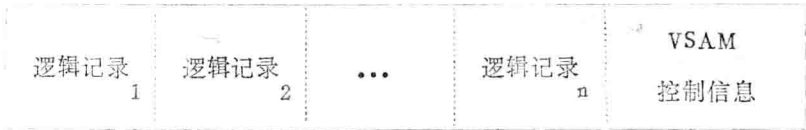


图 1-10 HIDAM索引数据库中块的结构 (VSAM)

1.2.2 HIDAM数据库记录的存储形式

HIDAM类型的数据库是通过一个索引去存取数据库记录，因此，所有根段必须各有唯一的键值，这样，数据库管理系统就可根据每个根段的键值建立HIDAM索引数据库，它的每一个逻辑记录都指向HIDAM数据库中的相应根段。在HIDAM数据库初始装入时，所有根段都应按键值从小到大依次装入，每个根段的全部从属段都应按层次关系随所属根段一起装入。图1-11展示两个HIDAM数据库记录和初始装入后它们在存储器中的存储形式。

由图1-11可见，HIDAM数据库与HIDAM数据库不同：它没有根段寻址区，也没有溢出区，它是由一系列的块所组成（这里省掉了前面的存储单元映象块）。装入数据库时，

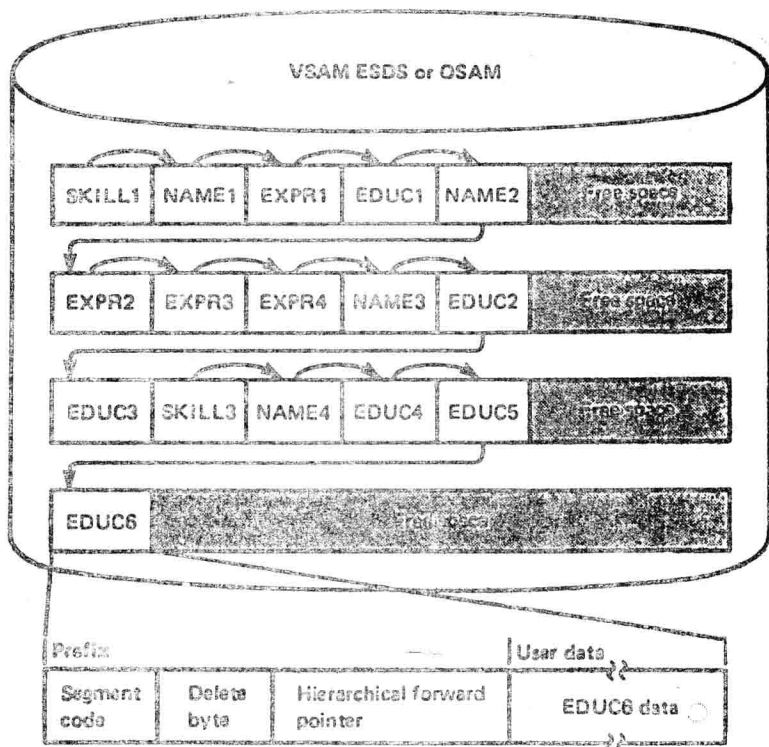
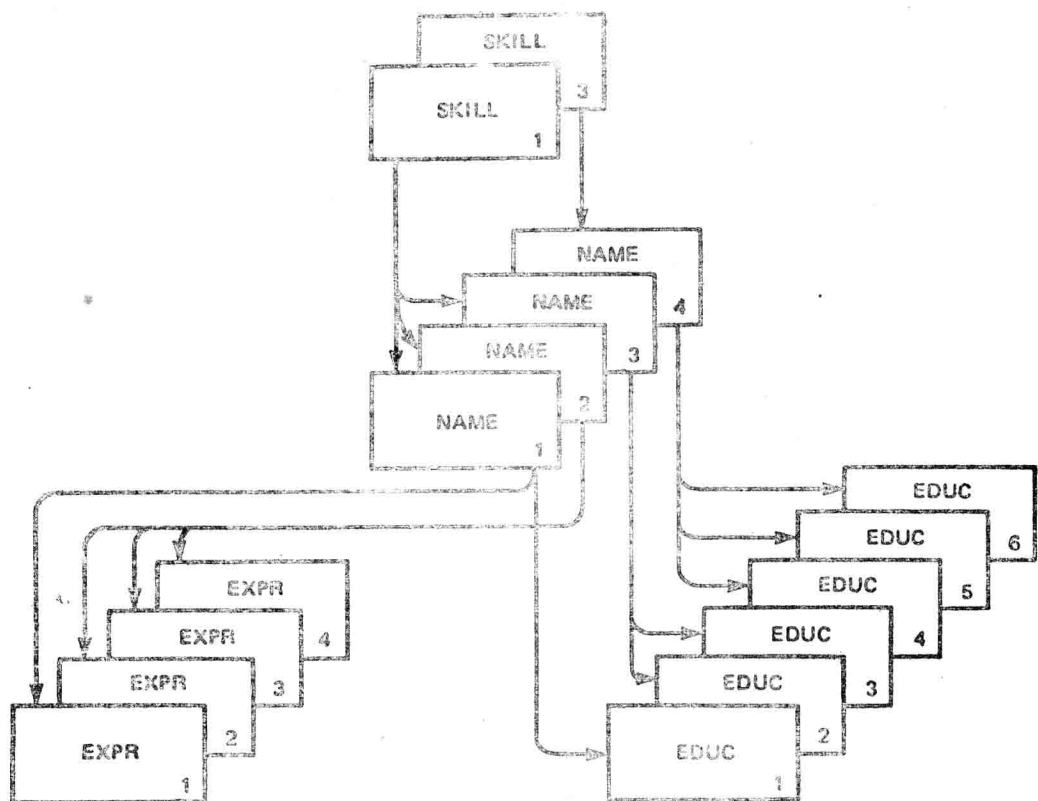


图 1-11 两个数据库记录装进HIDAM数据库后的存储形式

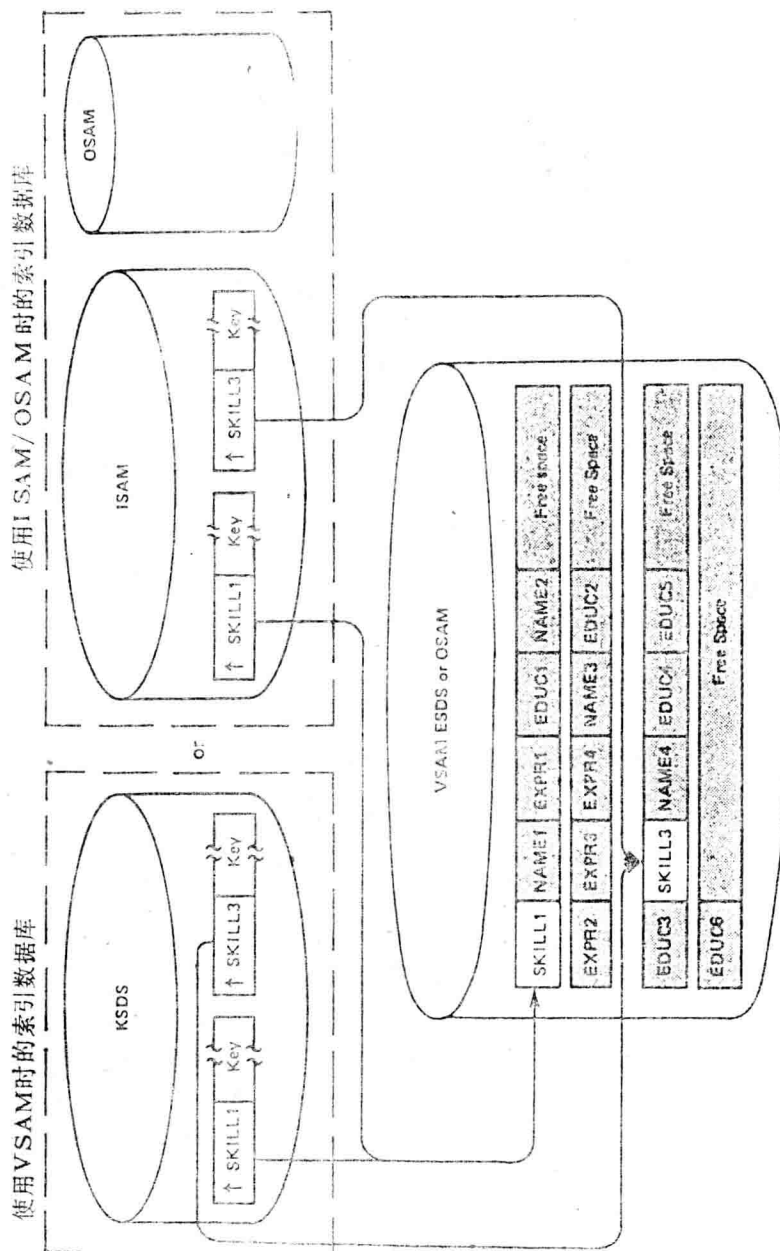


图 1 AM 数据库之间的关系

按照这些记录进入的先后次序一个记录紧接一个记录放在块中（块中要留下用户指定要保留的自由空间）。数据库记录中的段都用指针连接。

1.2.3 HIDAM索引数据库记录的存储形式

当每个根段存储到HIDAM数据库中时，数据库管理系统为那个根段建立一个索引段（由根段键值和根段所在地址组成），并把它存进HIDAM索引数据库。当使用操作系统提供的VSAM存取方法时，HIDAM索引数据库是一个VSAM KSDS (KEY SEQUENCE DATA SET) 数据集；若使用操作系统提供的ISAM/OSAM存取方法时，这个索引数据库由一个ISAM数据集和一个OSAM数据集所组成（初始装入时所建立的索引段存放在ISAM数据集，初始装入完毕后，在以后的处理中所产生的索引段存放在OSAM数据集中）。图1-12展示初始装入两个数据库记录后HIDAM数据库和HIDAM索引数据库之间的索引关系。

这里需注意的是：HIDAM索引数据库是在用户插入数据库记录的根段（无论是在初始装入时或在以后的数据库处理过程中）时由数据库管理系统自动建立和维护的，用户不作任何干预。

1.2.4 HIDAM索引数据库的结构

索引数据库的结构随用户选用的存取方法变化。当用户选定VSAM为存取方法时，索引数据库就是一个VSAM KSDS数据集；若选用ISAM/OSAM作存取方法，索引数据库由ISAM和OSAM两个数据集构成。在实际应用中，大多数都选用VSAM作为存取方法。因此，索引数据库的结构与VSAM KSDS数据集、ISAM数据集和OSAM数据集的结构密切相关。

VSAM KSDS数据集由两部分组成，一个叫数据部分，一个叫索引部分。索引数据库中的全部索引段就存放在KSDS数据集的数据部分，它的存放形式如图1-13所示：

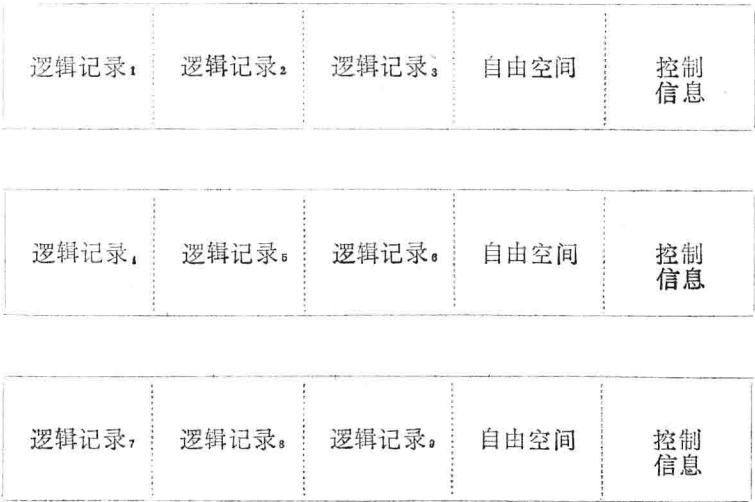


图 1-13 索引段在KSDS数据集的数据部分之存放形式

图1-13中有三个块，在VSAM中通常用“CI”（CONTROL INTERVAL）取代块的称呼，因此，图1-13中有三个CI，每个CI中存放三个逻辑记录（亦即存放了三个索引段）。KSDS数据集的数据部分就是由许多这样的CI组成。由于每个逻辑记录中包含了HIDAM

数据库中各相应根段的键值和地址,因此,只要在索引数据库中找到指定键值的逻辑记录,就能根据指针值(该逻辑记录中的)立即找到HIDAM数据库中的指定根段(亦即是数据库记录的开始)。这样,怎样才能从KSDS数据集的数据部分快速地找出所要的逻辑记录就变得十分重要。KSDS数据集的索引部分就是为了完成这种功能而建立的。索引部分通常由一层或几层所构成,如图1-14所示:

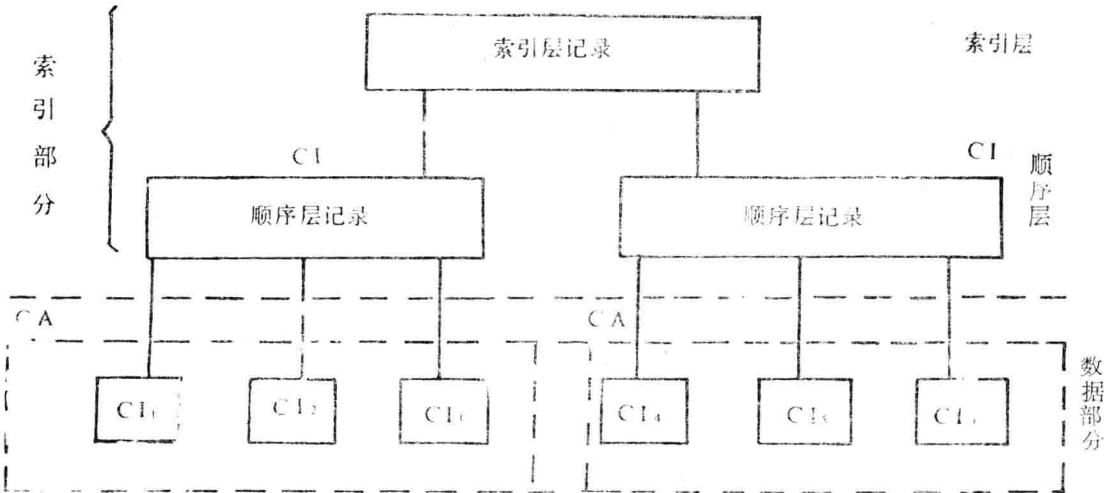


图1 -14 KSDS数据集中索引部分结构示意图

索引部分的最低层称为顺序层,顺序层上面的层(可能有两层)叫索引层。在顺序层中,每个CI中存放一个顺序记录,该记录中有一个指针指向对应的CA (CONTROL AREA的简称,一个CA通常是磁盘中的一个柱体),那个CA中的每一个CI在该顺序记录中都有一个登记项,每个登记项的格式如图1-15所示:

指针 (指向CI)	CI中最高键值
-----------	---------

图 1-15 顺序记录中登记项的格式

图1-15中的 指 针部分指向该CA中的某一CI; 最高键值部分存放由指针所指的那个CI中所存放的全部逻辑记录中的最高键值。这样,只要能找到顺序层中的相应顺序记录,就能立即找到数据部分中的相应CI,在这个CI中就一定能找到具有指定键值的逻辑记录。

索引层中每个CI存放一个索引层记录,每个索引层记录包含若干个登记项,每个登记项由一个指针和一个最高键值两部分构成。指针指向低一层中的某个CI,最高键值指示那个CI中的最大键值。这样,整个VSAM KSDS数据集就可表示成图1-16所示的形式。

实际上,图1-16所示的也就是一个索引数据库(使用VSAM作存取方法时)的结构。

若用户指定要取具有键值‘75’的数据库记录,数据库管理系统首先从HIDAM索引数据库开始寻找:取索引部分的最高层,放进缓冲区A,把键值‘75’与最高层中第一个登记项的最高键值比较,发现‘75≤374’(见图1-16,若指定键值大于登记项中之键值,取下一个登记项比,直到找到指定键值小于或等于登记项中的键值,否则为出错),则从登记项中取出指针值,读出顺序层中第一个CI,放进缓冲区B,用指定键值75与该CI

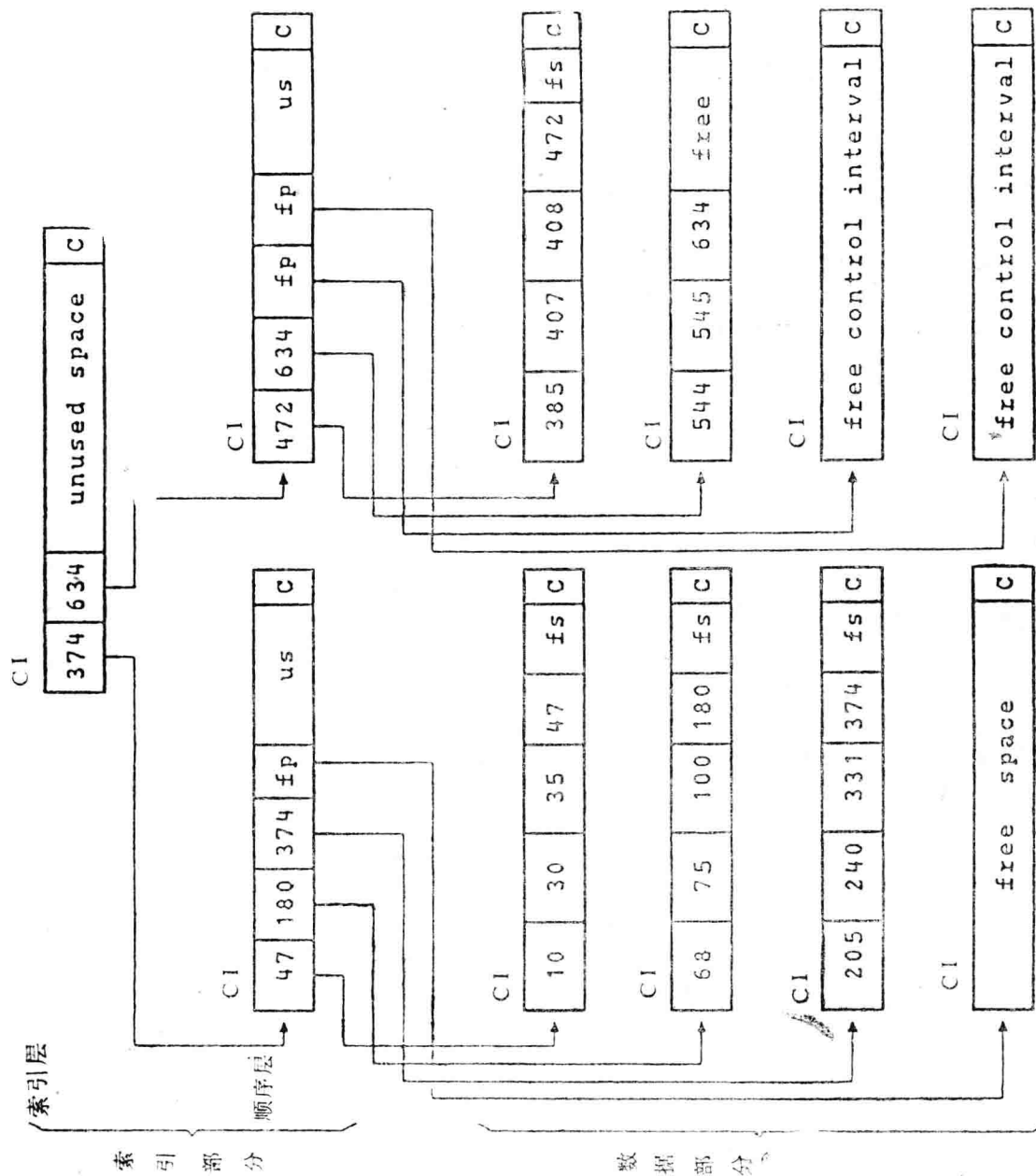


图 1-16 KSDS数据集的结构示意图