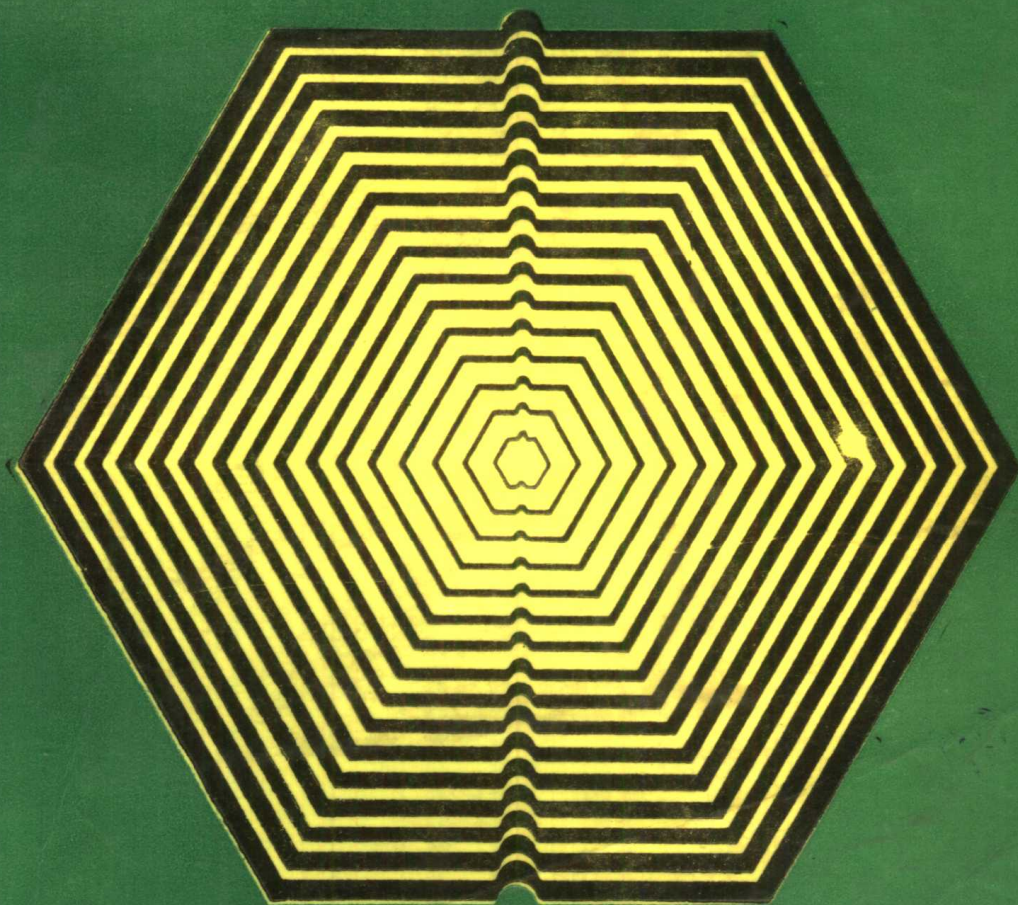


高等学校试用教材

实用 数据库 管理系统汇编 (一)

萨师煊 王珊 编



高等教育出版社

高等学校试用教材

实用数据库管理系统汇编(一)

萨师炫 王 珊 编

高等教育出版社

内 容 提 要

实用数据库管理系统汇编是《数据库系统概论》的补充,是为高等院校计算机及有关专业数据库课程编写的实习教材。

本汇编将介绍各种主要的数据库系统产品,本书作为汇编的第一分册,首先讲述其中较常用的六个产品的基本功能、特点和使用方法。这六个系统是:HP 机器上的 IMAGE 网状数据库系统,关系数据库系统 ORACLE, Rbase 5000, INFORMIX, dBASE III 和 UNIFY。本书最后还介绍了评价关系型数据库系统的基本准则。至于其它主要产品,将在后续分册介绍。

本书既可以作为大专院校计算机有关专业的数据库实习教材和上机指南,也可供从事计算机应用的科研人员、工程技术人员等使用。

高等学校试用教材
实用数据库管理系统汇编(一)

萨师焯 王 珊 编

*

高等教育出版社出版
高等教育出版社照排中心照排
新华书店北京发行所发行
北京市制本总厂印刷

*

开本 787×1092 1/16 印张 19 字数 470 000

1990 年 4 月 第 1 版 1990 年 4 月第 1 次印刷

印数 0001—21301

ISBN7-04-002733-X/TP · 61

定价 4.20 元

前 言

在我们编写《数据库系统概论》(以下简称《概论》)第一版的时候,数据库的应用刚刚开始,拥有商品化的数据库系统产品的单位(包括高等院校)还很少。为了帮助学生数据库实习任务的完成,我们在第一版的附录Ⅱ中介绍了一个网状数据库系统——IMAGE的系统结构、功能和使用方法,使学生初步了解如何在一个实际的数据库管理系统(以下简称DBMS)上建立一个数据库及其应用的步骤和方法。

今天,数据库技术已经从高等院校走向社会,微机上关系数据库系统迅速发展,功能不断增强,效率日益提高,使数据库的应用越来越广泛,学生上机实习条件已根本改善。但是,新的问题又产生了,关系数据库系统产品种类很多,学生上机实习缺少一本比较全面的实习指南。同时,应用部门在选购数据库系统产品时也遇到了应如何选择比较适合自己应用需要的系统。这就需要在选购标准。许多同志向我们提出了新的要求,希望能介绍更多的数据库系统,给出评价这些系统的标准。这就是我们编写这本汇编的动机。它既可以作为《概论》的分册,供学生实习用,也可以作为一本独立的教材或参考书,供从事计算机工作的科研人员、工程技术人员等使用或选购产品时的参考。

目前,在我国使用的网状型数据库系统大都是CODASYL型或CODASYL型的简化。在《概论》中我们已详细讲述了CODASYL的基本概念、方法和技术,在此就不再介绍了。原《概论》中的IMAGE系统作为非CODASYL型的网状数据库系统的实例和学生实习指南仍保留,把它列为本书的第一章。

我国使用的层次型数据库系统大多是IBM机器上的DL/1或IMS,我们已在《概论》中作为层次模型的代表介绍了,这里也不再复述。

在我国,大型关系数据库系统有IBM的SQL/DS,ORACLE公司的ORACLE系统等。SQL/DS和BD2十分相近,是在SYSTEMR基础上开发的商品化系统。它们的概念、所采用的SQL语言等在《概论》中已详细讨论。

大型机上的数据库系统一般不适合学生上机实习用。当前,微机关系型数据库系统产品十分丰富,因此学生上机实习比较适宜用微机关系数据库系统。我们的重点自然就是介绍这些产品系统。如ORACLE系统,我们介绍它的微机版本而不介绍它在大型机上的版本。

本书中,我们选择了功能比较齐全,使用比较方便以及在国外和我国使用较多的一些系统。它们是:ORACLE,Rbase 5000,INFORMIX,dBASEⅢ,UNIFY,分别为本书的第二、三、四、五、六章。本书的第七章介绍了关系模型的奠基人E. F. Codd提出的关系型数据库产品的十二条准则。这是我们评价和购买关系型数据库系统的标准。

本书第一、二、七章由王珊编写,第三至六章分别由刘怡、罗力、刘伶、杨明伟编写。

全书由萨师煊、王珊修改、定稿。

清华大学俞磐祥同志和中国科技大学研究生院罗晓沛同志阅读了全书，并提出许多有益的意见，我们对于他们表示诚挚的感谢！

由于编者水平有限，书中缺点和错误肯定不少，希望读者批评指正。

编 者

1989年 6月

于中国人民大学

目 录

第一章 IMAGE 数据库系统(1)

1.1 概述(1)

1.2 IMAGE 的数据结构(2)

1.2.1 术语介绍(2)

1.2.2 数据集的类型和关系(3)

1.3 IMAGE 数据库描述语言(7)

1.3.1 模式结构(7)

1.3.2 口令部分(8)

1.3.3 数据项部分(8)

1.3.4 数据集部分(10)

1.3.5 建立IMAGE数据库根文件 ... (12)

1.3.6 建立IMAGE数据库、数据集文件和装入数据(16)

1.4 用宿主语言调用IMAGE库过程

程存取数据库(17)

1.4.1 IMAGE库过程(17)

1.4.2 调用库过程的方式(18)

1.4.3 存取路径和当前值的概念.....(18)

1.4.4 读取方式(19)

1.4.5 库过程解释(19)

1.5 用自含语言QUERY存取

数据库(26)

1.5.1 概述(26)

1.5.2 QUERY命令(27)

1.6 IMAGE的存储结构(34)

1.6.1 主集的结构(34)

1.6.2 明细集的结构(35)

1.7 IMAGE数据库的维护(35)

1.7.1 数据库的重组织(35)

1.7.2 数据库的恢复(37)

1.7.3 IMAGE公用程序(42)

1.8 实例(47)

第二章 ORACLE 关系数据

库系统(53)

2.1 概述(53)

2.1.1 ORACLE系统概貌(53)

2.1.2 ORACLE产品系列(54)

2.1.3 ORACLE可运行的系统(56)

2.1.4 ORACLE特点(57)

2.1.5 ORACLE的安装和运行环境(58)

2.2 SQL/UF1.....(59)

2.2.1 进入和退出UF1(59)

2.2.2 SQL和UF1(60)

2.2.3 数据定义(66)

2.2.4 索引和聚集(67)

2.2.5 建立树结构并进行查询(69)

2.2.6 空值处理(72)

2.2.7 数据控制命令——数据的共享和保密(72)

2.2.8 报表格式输出(74)

2.2.9 UF1命令(77)

2.3 交互式应用工具——IAF(79)

2.3.1 IAG概况(79)

2.3.2 应用结构和应用设计(80)

2.3.3 IAG对话(81)

2.3.4 IAP的使用(86)

2.4 报表生成工具——Report Writer(91)

2.4.1 报表生成工具简介(91)

2.4.2 RPT——报表生成程序(92)

2.4.3 报表正文格式化程序RPF(94)

2.4.4 一个简单的报表实例(96)

2.5 ODL——数据库装入实用程序(99)

2.5.1 ODL装入程序的工作过程(99)

2.5.2 ODL控制语言(99)

2.5.3 原始数据文件(101)

2.5.4 登录文件和出错处理(101)

2.6 EXPort/IMPort实用程序(102)

2.6.1 EXP 卸出实用程序的功能 和使用方法	(102)
2.6.2 IMP 重装入实用程序的功能 和使用方法	(104)
2.7 ORACLE LINK 通信程序	(104)
2.7.1 概述	(104)
2.7.2 ORACLE LINK 的使用方法	(105)
2.8 应用编程接口	(106)
2.8.1 概述	(106)
2.8.2 程序环境下的ORACLE	(106)
2.8.3 SQL 语句	(107)
2.8.4 C 样本程序	(108)

第三章 R: base 5000 数据库系统 ... (112)

3.1 概述	(112)
3.2 R: base 5000 数据库结构和限制	(113)
3.3 R: base 命令语言	(113)
3.3.1 R: base 基本命令	(113)
3.3.2 命令子句	(116)
3.3.3 数据库结构的定义、 修改和显示	(118)
3.3.4 数据输入	(123)
3.3.5 数据查询	(128)
3.3.6 数据更新	(134)
3.3.7 报表处理	(137)
3.3.8 应用程序设计	(146)
3.4 应用程序快速生成软件 EXPRESS	(153)
3.4.1 进入EXPRESS 子系统	(153)
3.4.2 定义一个新的数据库	(154)
3.4.3 修改现有数据库定义	(155)
3.4.4 定义一个应用程序	(156)
3.4.5 修改一个已定义的应用程序	(159)

第四章 INFORMIX 数据库管理

系统及其使用	(165)
4.1 概述	(165)
4.1.1 INFORMIX 的功能与特点	(165)
4.1.2 INFORMIX 的文件类型	(166)
4.1.3 INFORMIX 的组成、运行、 HELP 功能及退出方式	(168)
4.2 数据库结构的定义和修改	(168)

4.2.1 数据库结构的定义	(168)
4.2.2 数据库结构的修改	(170)
4.3 数据输入、更新、删除及 屏幕输出	(171)
4.3.1 基本命令的格式和使用	(171)
4.3.2 用户屏幕的生成及修改	(173)
4.3.3 用户屏幕的操作	(179)
4.4 查询语言INFORMER	(179)
4.4.1 基本语法规则	(180)
4.4.2 命令格式和使用	(180)
4.4.3 高级功能	(182)
4.5 报表书写器ACE	(185)
4.5.1 ACE 源程序的生成和执行	(185)
4.5.2 ACE 语言的命令及其 语法规则	(187)
4.5.3 范例	(192)
4.6 C 语言调用 INFORMIX 库 过程的方式	(194)
4.6.1 基本配置及程序运行	(195)
4.6.2 ALL 库过程	(195)
4.6.3 调用库过程的错误信息表	(199)
4.6.4 范例	(200)
4.7 数据库监控程序DBSTATUS	(203)
4.7.1 命令格式和使用	(203)
4.7.2 B- 树检查程序	(205)
4.8 INFORMIX 的UNIX 版本与 PC-DOS 版本的主要差别	(205)

第五章 关系数据库管理系统

dbase III	(207)
5.1 概述	(207)
5.1.1 dBASE III 的运行环境	(207)
5.1.2 dBASE III 功能简介	(207)
5.1.3 dBASE III 文件类型	(208)
5.1.4 启动和退出dBASE III 系统	(209)
5.2 dBASE III 的基本语法和规定	(209)
5.2.1 命令	(209)
5.2.2 表达式	(210)
5.2.3 内存变量	(211)
5.2.4 函数	(212)
5.3 数据库文件的建立和修改	(214)
5.3.1 文件结构的定义和修改	(214)

5.3.2 数据库文件的数据输入	(216)	6.2.2 建立数据库	(258)
5.3.3 数据库文件的修改	(219)	6.3 数据登记屏的形成及	
5.3.4 数据库文件的编辑	(220)	数据的录入	(259)
5.4 数据库文件的使用	(221)	6.3.1 屏幕维护	(259)
5.4.1 数据排序	(222)	6.3.2 建立系统约定的屏幕表格.....	(259)
5.4.2 数据索引排序	(223)	6.3.3 建立自定格式的屏幕表格.....	(260)
5.4.3 数据检索	(225)	6.3.4 用ENTER输入数据	(264)
5.4.4 数据计算	(231)	6.4 构造树形菜单系统	(265)
5.4.5 报表生成	(233)	6.4.1 菜单的维护	(265)
5.4.6 打印标签	(235)	6.4.2 菜单系统的保护措施	(267)
5.4.7 输入输出格式设计	(237)	6.5 使用SQL语言进行数据操纵	(272)
5.4.8 数据库辅助操作命令	(238)	6.5.1 查询语句(SELECT).....	(272)
5.5 dBASE III 程序设计	(239)	6.5.2 数据处理功能	(274)
5.5.1 命令文件的建立和执行	(239)	6.5.3 其它语句	(275)
5.5.2 命令文件的语句	(240)	6.6 数据库模式的修改	(276)
5.5.3 专用于子程序设计的		6.7 输出报表	(279)
几条语句	(241)	第七章 关系型 DBMS 的判别	
5.5.4 应用程序举例	(243)	准则	(284)
5.6 HELP 及 ASSIST 命令介绍	(246)	7.1 关系型DBMS产品的十二条	
5.6.1 HELP 命令	(246)	基本准则	(284)
5.6.2 ASSIST 命令	(248)	7.2 关系模型的九个结构特性	(288)
第六章 UNIFY 系统	(250)	7.3 关系模型的操作特性	(289)
6.1 系统概况	(250)	7.4 关系模型的三类完整性	(290)
6.1.1 UNIFY 的操作环境	(250)	7.5 一种简单评价方法	(290)
6.1.2 文件类型和目录结构	(251)	附录：五个关系数据库系统 (微机版)	
6.1.3 环境变量	(252)	静态特性比较	(293)
6.1.4 进入UNIFY环境	(253)	参考文献	(295)
6.2 建立数据库	(254)		
6.2.1 输入一个数据库设计方案	(255)		

第一章 IMAGE 数据库系统

1.1 概 述

IMAGE 3000 简称 IMAGE, 是美国 HP 公司(HEWLETTPACKARD)生产的 HP3000 计算机上的数据库管理系统. 它属于网状模型, 但加上了若干限制. 例如, 限制了网状的层次只能是二层, 所以确切地讲其数据模型是简化了的网状模型.

HP3000 的操作系统是 Multiprogramming Executive Operating 简称 MPE. 它允许多道程序并发执行, 兼有分时和批处理能力, 具有独立于设备的文件管理, 所有的磁盘文件和 I/O 设备都统一地称为 MPE 文件. 用户在 HP3000 上建立和使用 IMAGE 需要熟悉有关的 MPE 命令和文件命令. 这里我们不单独介绍这些命令, 而在使用 IMAGE 过程中需要用到什么命令就介绍什么命令.

IMAGE 系统由三大部分组成: 数据库定义、数据库存取和数据库维护. 图 1-0 是 IMAGE 系统的一个概貌.

① 在系统分析的基础上进行数据库设计, 把数据组织成符合 IMAGE 系统的数据结构.

② 使用 IMAGE 数据库描述语言来描述设计好的数据结构, 这种描述称为模式. 然后调用 IMAGE 模式处理程序 DBSCHEMA 来处理模式, 生成目标模式, 即数据库的内部描述——根文件. 在 1.3 节中将介绍数据库描述语言的语法规则以及建立根文件的操作过程.

③ 数据库建立者调用公用程序 DBUTIL 中的 CREATE 命令, 系统根据根文件中的结构说明建立数据库的数据集文件, 并把文件初始化.

④ IMAGE 提供了一组能由 COBOL, FORTRAN, SPL, BASIC 和 RPG 等宿主语言调用的库过程来存取数据库的数据. 应用程序员可选择其中任何一种语言来设计应用程序调用库过程以存储、检索和修改数据库数据. 在 1.4 节中将详细介绍这十六个库过程及调用方式.

⑤ IMAGE 还提供了自含式的 QUERY 语言, 用户使用 QUERY 命令不必编写应用程序就可实现对数据库的检索和存取. 在 1.5 节中将介绍 QUERY 命令及其使用方法.

⑥ 数据库一经建立就需要进行维护, IMAGE 提供了一组公用程序实现对数据库的重组织、转储和恢复等维护管理功能. 在 1.7 节中将讨论 IMAGE 数据库的维护方法, 以及这六个公用程序的功能和使用方法.

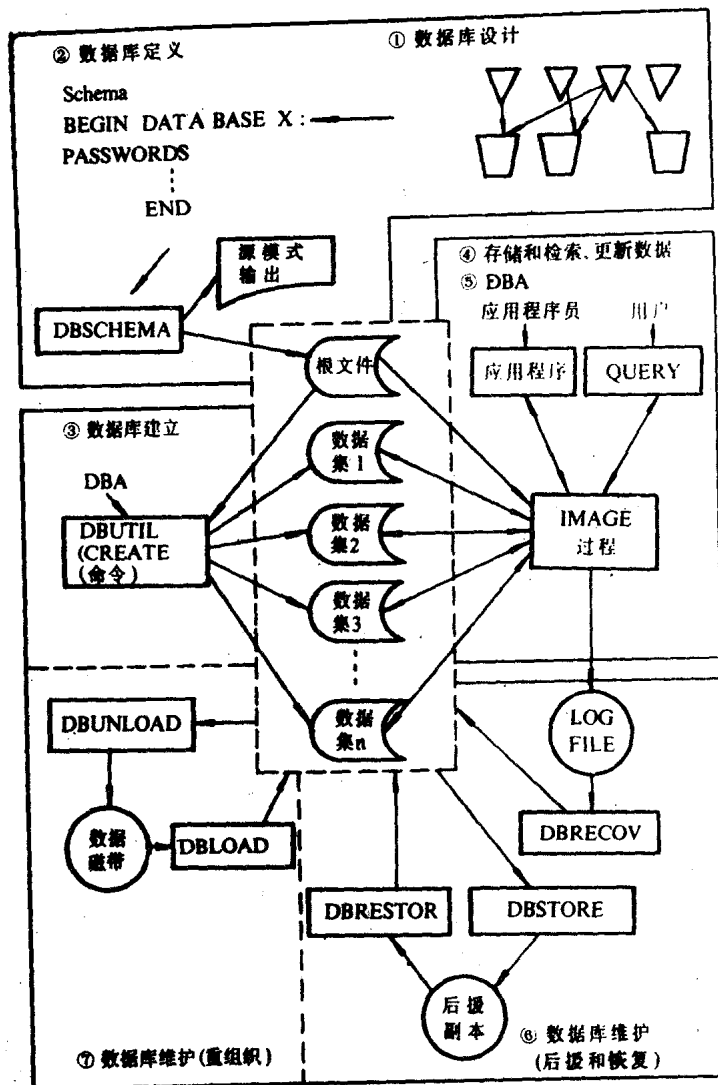


图 1-0

1.2 IMAGE 的数据结构

1.2.1 术语介绍

在描述 IMAGE 数据库系统的数据结构之前，我们先介绍一下 IMAGE 中使用的主要术语。

一个 IMAGE 数据库是一个命名了的关联数据的集合。

一个 IMAGE 数据库由一个或多个数据集(Data Set)组成,数据集是同类型定长记录的集合。在一个数据集中的每个记录有相同的格式,即包含同样的一些数据项,在 IMAGE 数据库中数据记录称为 Data Entry。

IMAGE 数据项是最小的存取单位,数据项有两种:简单数据项和复合数据项。简单数据项只包括单个子项,复合数据项包含两个或两个以上的子项。所有子项具有相同的数据类型和长度。

IMAGE 中数据库、数据集、数据项是用名字来标识的,即每个数据库,每个数据集和数据项都有一个名字。

数据库、数据集、数据项之间的关系可用下图表示:

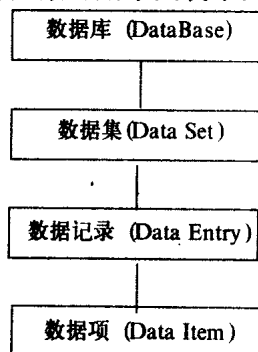


图 1-1

IMAGE 系统限定:一个数据库最多包括255 种数据项,同一种数据项可以包含在不同的数据集里,一个数据集至多包含127 种数据项,一个数据项至多能有255 个子项,此外,还限定一个数据集至多包含 2^{23} (约 8 百万个)记录值。

1.2.2 数据集的类型和关系

IMAGE 数据库系统定义了两种类型的数据集:主数据集(MASTER DATA SET)和明细数据集(DETAIL DATA SET),简称为主集和明细集。在图形中主集用三角形 ∇ 表示,明细集用梯形 $\square$ 表示。

例如学生-课程数据库可以表示如下:

主集 STUDENT (学生)的记录格式为

SNO	SN	AGE	SEX	SDEPT	SADD
-----	----	-----	-----	-------	------

即具有数据项 SNO(学号), SN(姓名), AGE(年龄), SEX(性别), SDEPT(学生所在系), SADD(学生宿舍地址)。

主集 COURSE(课程)的记录格式为

CNO	CN	PCNO	TEAH
-----	----	------	------

即具有数据项 CNO(课程号), CN(课程名), PCNO(先修课程号), TEAH(任课教员)。其

中 PCNO 为复合项,子项数目为5,TEAH 也是复合项,子项数为3,表示一门课程最多可以指出五门主要的先修课和三位任课教员。

主集TERM (学期)的记录格式为[TNO],即学期号。

明细集 SC (学生选课)的记录格式为

SNO	CNO	G	PDATE	TNO
-----	-----	---	-------	-----

即具有数据项为 SNO (学号), CNO (课程号), G (成绩), PDATE (选修日期), TNO (第几学期)。

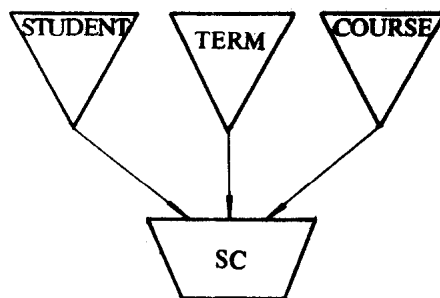


图1-2

1.2.2.1 主数据集

主数据集具有如下特征:

1. 主数据集一般是用来记载与实体有关的唯一可识别的信息的,例如,上例中用 STUDENT 和 COURSE 来存放关于学生和课程的信息。

2. 在主集中必须定义一个,也只能定义一个数据项为它的查找项 (search item),此查找项不能是复合项,只能是简单项。

在主集中查找项值不能重复,即任意两个记录的查找项必须有不同的值,所以查找项的值能用来唯一标别主集中的某个记录值。系统按照查找项的值来存放数据记录,对照 DBTG 系统则主集中记录的存放方式是计算型的。我们可以按照查找项的值迅速地检索到主集中具有此查找项值的记录。

主集中其它数据项的值是允许重复的,因为不用它们来确定数据记录的位置。

1.2.2.2 明细数据集

明细数据集具有以下特征:

1. 明细集一般是用来记载与主集有关联的某些信息的。例如在学生-课程数据库中 SC 是用来记载某一学生选修各门课程的成绩和日期等信息。

2. 一个主集和一个明细集可以建立一种联系,这种联系是通过两个数据集中相应的查找项来确定的。也就是说,在明细集中也必须具有与相关联主集的查找项相容的数据项。所谓相容是指它们的值是可比的。例如 SC 中具有与 STUDENT 的查找项 SNO 相同的数据项 SNO,当然在明细集中这一查找项的值是允许重复的。这样,在明细集中所有数据项的值均是可重复的。

IMAGE 系统负责建立和维护每一主集记录 (以下简称主记录) 同所有具有相同的查找项值的明细集记录 (以下简称明细记录) 之间的联系。

3. 明细集可具有 0 ~ 16 个查找项,也就是说它可以不与任何主集相关联 (当查找项的个数为 0 时),或至多和 16 个主集相关联。

1.2.2.3 路径和链

主集和明细集之间通过相容的查找项定义的一种联系称为一个路径 (PATH),它相应于 DBTG 中系型 (Set type) 的概念。而与某一路径相联系的主集相应于 DBTG 中的首记录,明细集相应于属记录。

在 IMAGE 数据库系统中路径的实现是用链的方法。

与同一主记录相联系的所有明细记录(它们必须有相同查找项值)构成一条链(CHAIN),它相应于 DBTG 中系值的概念。某一路径可具有多条链,即有多个系值,主集中有多少个记录就有多少条链,虽然有的链允许是空的,即相应于 DBTG 的系值允许属记录数为零的情形。

在 IMAGE 中规定了一条链至多容许包括 65535 个明细记录。

在每条链中,明细记录的逻辑顺序按用户指定的规则维护,这相应于 DBTG 中系序的概念。详细内容将在 1.2.2.6 中介绍。

IMAGE 中允许主集和明细集之间存在多条路径,这些路径对应主集中同一个查找项,而每个路径对应明细集中不同的查找项,但这些查找项必须与主集中的查找项相容。

IMAGE 的链是这样来实现的:系统对每个主记录的一个路径建立和维护两个指引元,一个指向链的第一个明细记录,一个指向最后一个明细记录,在链中除了最后一个记录外,每个记录有指引元指向它的“下一个”记录;除了第一个记录外,每个记录有指引元指向它的“上一个”记录。

IMAGE 支持按链存取明细记录,由主记录找到它的明细记录,由明细记录找到它的上一个或下一个明细记录。

图 1-3 是一个 PATH 及其 CHAIN 的示意图,由(a)中看到,一个路径总是由主集指向明细集。

IMAGE 系统规定:

1. 一个数据集或者被定义为主集,或者为明细集,一个数据库至多可由 99 个数据集组成。

2. 一个明细集可以定义 m 个查找项,通过 m 条路径与 n 个主集相联系,其中 $0 \leq n \leq m \leq 16$ 。

3. 一个主集必须有也只能有一个查找项,它可以通过 r 条路径与 s 个明细集相联系,其中 $0 \leq s \leq r \leq 16$ 。

所以 IMAGE 数据库系统支持如图 1-4 所示的数据结构。

1.2.2.4 自动主集和手工主集

IMAGE 中主集又分为两种类型:自动的和手工的,它们具有以下特征:

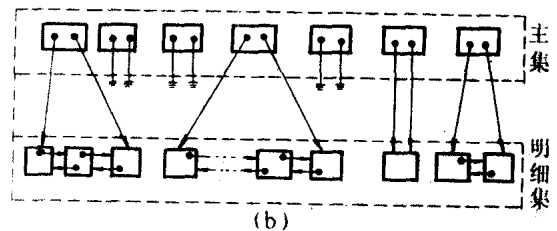
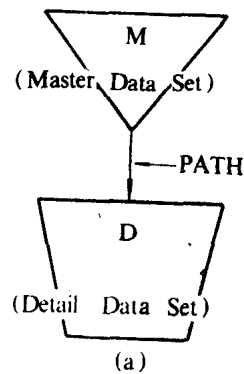


图 1-3 路径和链

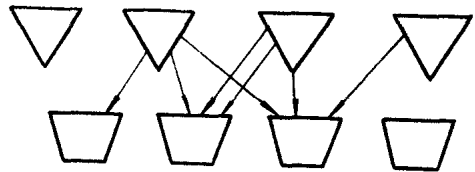


图 1-4

手工主集

1. 可以单独存在而不必与任何明细集相联系
2. 除查找项外, 还可以包含其它数据项.
3. 手工主集中的所有记录, 必须由用户手工地加入或删除, 而系统提供以下的完整性保护: 如果手工主集中的主记录尚未加入, 那么与之相关联的任何明细记录不能加入. 一个主记录仅当它没有任何相关联的明细记录时才允许被删除.

自动主集

- 必须与一个或多个明细集相联系.
- 只有一个数据项, 即查找项.
- 用户不能手工地增加或删除自动主集中的记录.

当一个明细记录加入明细集中时, 如果这个记录的查找项值不是相关联的自动主集中的某一个记录, 则系统自动地在自动主集中建立这个记录的查找项值. 当一个自动主集记录的所有相关联的明细记录已被删除, 即与此主记录相关联的每条链均为空时, 该主记录被自动删除.

注意: 在一般情况下, 一个明细记录总是与它的手工主记录一起存于数据库中, 但一个手工主记录的某一条链允许是空的. 已有手工主记录的查找项值用于进行完整性检查, 以防止不合法的明细记录加入数据库. 当查找的值是不可预言的或者比较多时, 不妨采用自动主集, 以节省操作时间. 当一个数据项可构成一个主记录时, 数据库设计者就要在手工和自动两种类型的主集中作出选择. 采用手工方式, 可执行数据的完整性检查, 选用自动方式, 可以节省时间.

1.2.2.5 主路径

前面已经指出, 一个明细集可以具有多个路径与多个主集相联系. IMAGE 数据库系统容许 DBA 指定其中一个为主路径 (PRIMARY PATH). 主路径和其它的路径的区别是: 主路径的各条链上的明细记录在物理上是最邻近的, 当数据最初装入数据库时, 连续的存储位置就分配给主路径中同一链的诸明细记录. 随着新记录的不断插入和老记录的删除, 这种好的性能又会慢慢地削弱. 于是使用一段时间后, 数据库就要重新组织. DBA 可以用 IMAGE 的公用程序来完成数据库的重组, 详见 1.7 节.

这里, 路径与主路径的概念对应于 DBTG 中每个属记录 (型) 可以参加多个系, 而属记录的存放只能通过其中一个系的概念. 所以 IMAGE 中所有的明细记录的存放方式相应于 DBTG 中经由系的方式.

数据库设计者应该指定最经常使用的路径作为主路径, 以提高检索效率.

1.2.2.6 排序项

对于任何一个路径, 可以指定某数据项 (除查找项外) 作为排序项 (SORT ITEMS), 其类型可以是整型或字符串型. 若定义了排序项, 则链中明细记录的逻辑顺序按排序项值的升序排列. 所指定的排序项实际上是主排序项, 当链中有多个记录具有相同的排序项值时, 它们的相对顺序由扩充字段的值决定. 扩充字段包括排序项以及随后的 (按数据集定义中数据项的列举次序) 数据项. 所以 DBA 为一路径指定的排序项以及它随后的所有数据项组成了该路径的实际排序项.

如果多个记录实际排序项的值仍相同时, 则按记录加入到数据库中的先后次序确定. 这个“先来在前”的原则也适用于未指定排序项的情形.

对照 DBTG 这相当于系序的概念. 这里的系序有两种. 若定义了排序项, 则是按排序项排序的

系同时用排队的方式加以补充。若没有定义排序项，则系是排队的方式。IMAGE 中明细记录的次序是由逻辑指引元维护的逻辑顺序，不是指物理上的位置。下图表示当一个记录加到明细集中时它将嵌入到链的什么位置。

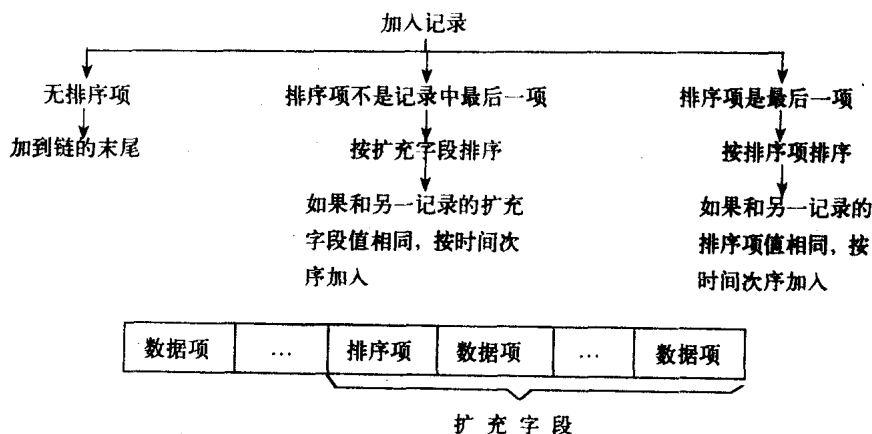


图 1-5 把记录加入排序链中的情形

不同的路径可以有不同的排序项，某个路径的排序项可以是另一路径的查找项。以上介绍了 IMAGE 的数据结构，附带说明一点，IMAGE 没有提供子模式的概念。

1.3 IMAGE 数据库描述语言

建立一个数据库，首先要在系统分析的基础上进行数据库设计，把所要研究的现实世界的数据库组织成 IMAGE 所支持的数据结构，然后利用 IMAGE 系统所提供的数据库描述语言 (DBDL) 来定义数据库。用 DBDL 写出的数据库描述程序称为数据库模式。它可以按照卡片或磁带上的 ASCII 代码文件的形式存放在 MPE 系统中，也可以作为编目磁盘文件存放在 MPE 系统中。下面介绍 IMAGE DBDL 的语言规则。

1.3.1 模式结构

DBDL 语言的若干约定如下：

1. 在语法格式中出现的标点符号不能省略，必须按规定书写。
2. 在语法格式中出现的方括号不是语言成分，它表示括号内的项目是任选的。
3. 可以在模式的任何地方加注释，其形式为《字符串》，但不允许嵌套注释。注释没有语义，模式处理程序不处理它们，但将被包括在源模式输出中。
4. 除了在 <口令> 中的大写、小写英文字母是被区分的之外，其它地方小写字母将被模式处理程序自动转换为大写字母。

模式的总体结构是：

语法: BEGIN DATA BASE < 数据库名>

PASSWORDS: < 口令部分>

ITEMS: < 数据项部分>

SETS: < 数据集部分>

· 数据库描述语言是一种自由格式的语言, 可以在模式的任何处插入空格(符号和保留字中间除外), 以使书写形式美观。空格、保留字、标点符号可用来区分不同的语法元素。

· < 数据库名>是以字母开头的 1 ~ 6 个字母或数字组成的字符串, 它是数据库的唯一标识。

· 模式中各部分描述的次序必须遵循上述的次序, 不能颠倒。

1.3.2 口令部分

语法: < 用户类别号> [< 口令>] ;

⋮

< 用户类别号> [< 口令>] ;

例如: 5 READER;

10 WRITER;

口令部分定义用户类别和口令, 以防止对数据进行未经许可的存取, 提供数据库数据的安全保护。

· < 用户类别号>是 1 到 63 内的整数, 用户类别号必须是唯一的。

· < 口令>是由 1 到 8 个 ASCII 字符组成的符号串, 不允许出现分号和回车控制符, 允许空格, 但无意义。

· 若在< 用户类别号>后未指定口令, 则该用户类别号无意义。

· < 口令>部分标识了一个用户类别号, 即该口令之前的用户类别号。若同一口令出现在两个< 用户类别号>之后, 则该口令标别的是值较大的那个类别号。

1.3.3 数据项部分

数据项部分定义数据库中所有的数据项, 定义包括数据项名, 类型, 最大容许的长度以及指定哪些用户类别号对它有存取权等。

语法: < 数据项描述> ;

⋮

< 数据项描述> ;

其中< 数据项描述>的格式为:

< 项名> , [< 子项数>] < 类型> [< 子项长度>] [< 读写权限表>] ;

例: SNO, X6 (5/10);

PCNO, 5 X4 (5/10);

↑ ↑ ↑ ↑
数据项名 子项数 类型 子项长度 读写权限表

· <项名> 由1至16个ASCII字符组成的字符串, 第一个字符必须是字母, 后继字符必须取自字母A~Z, 数字0~9或+ - * / ? / # % & @.

· <子项数> 1至255的整数, 指明该数据项包含的子项数目. 缺省时隐含为1, 即简单项.

· <类型> 定义数据项的类型, IMAGE中允许的类型是I, J, K, R, U, X, Z, P, 具体描述见图1-6.

类 型	语 义	长 度 单 位
I	以二进制补码表示的带符号的二进制数	字
J	同上(使用QUERY时只允许COBOL COMP形式的数据)	字
K	无符号的二进制数, 绝对二进制数	字
R	实数, 二进制浮点数格式表示	字
U	ASCII字符串, 不包含小写字符, 一个字符占一个字节	字节
X	不受限制的ASCII字符串	字节
Z	带符号的十进制数(每一位十进制数占一个字节, 符号放在最后一位数字的顶区)	字节
P	带符号的压缩十进制数, 每位十进制数占半个字节, 最右边的那个“半字节”用作符号位, “+”表示为1100, “-”表示为1101, 若无符号时表示为1111.	半字节

图1-6 IMAGE数据项类型说明

· <子项长度> 1至255的整数, 其单位可以是字(word), 字节(byte)和半字节, 它由类型决定. 如图1-6所示, 缺省时隐含长度值为1.

数据项长度: 数据项长度为 <子项长度> × <子项数>, 必须保证它是偶数个字节或整数个字长. 其最大长度为2047字长.

IMAGE中一个字节为8bits, 一个字为16 bits.

一个数据项无论是简单项还是复合项, 总是作为一个单元来处理, 是不可分的最小存取单位. 当用户程序存取数据库数据时, 系统不进行数据类型的有效性检验, 不执行数据的转换. 一般地, 当两个数据有相同的存储表示时, IMAGE以相同的方式处理它们. 用户不必按照IMAGE数据项的类型来存储宿主语言程序所定义的数据. 因而用户应当清楚数据库数据项和其它程序设计语言变量的存储表示之间的关系. 图1.7给出与COBOL和FORTRAN中等价的表示.

IMAGE	COBOL	FORTRAN
I	S9 ~ S9(4), COMP	INTEGER
I2	S9(5) ~ S9(9), COMP	INTEGER 4
K		LOGICAL
R2		REAL
R4		DOUBLE PRECISION
Xn	X(n), DISPLAY	CHARACTER n
Zn	9(n), S9(n), DISPLAY	
Pn	9(n), S9(n-1), COMP-3	

图 1-7