

高级数据库开发系统

db_VISTA III 参考手册

(共两册)

另一册为《db_VISTA III 用户指南》

王点依 编译

- ☐ 网状数据库模型
- ☐ 快速 B-树索引
- ☐ 数据库定义语言 DDL
- ☐ 高速缓冲虚拟磁盘技术
- ☐ 一百多个数据库操作函数
- ☐ 自动转换 DBASE 型的数据
- ☐ 支持局域网和多用户计算机
- ☐ 支持 MS DOS、UNIX 和 VAX VMS

前 言

db_VISTA 是一个数据库管理系统(DBMS),它为 C 语言应用程序的开发提供了强大的、高性能的数据库管理功能。db_VISTA 采用了网络数据库模型和 B-树索引方法,它能为 C 程序员提供一套高效的组织和访问简单及复杂信息结构的方法,同时保持最小的数据冗余。

db_VISTA 支持多种机型和操作系统,包括:MS DOS、UNIX 和 VAX VMS。它是用 C 语言编写的,并可以得到它的全部源代码,这就能保证了 C 应用程序的可移植性,使您的应用程序能在从单用户微机到局域网、小型机甚至于大型机的广阔市场领域中销售。

db_VISTA 系统的主要特性如下:

- ☐ 事物处理和注册
- ☐ 自动数据库恢复
- ☐ 支持局域网和多用户计算机上的多用户操作
- ☐ 基于网络数据库模型和快速 B-树索引的存取方法
- ☐ 多数据库存取,即一个应用程序可以同时打开和存取多个数据库
- ☐ 交互式数据库存取工具,便于控制存储在 db_VISTA 数据库中的任何信息
- ☐ 文件传送工具,可在 ascii 码(或 dBASE II / III)与 db_VISTA 数据库之间传输文件
- ☐ 数据存储使用多个卷
- ☐ 用于定义 db_VISTA 数据库内容和结构的、类似于 C 语言的 DDL 语言
- ☐ 为检测数据库完善性而提供的数据库相容性检查工具
- ☐ 为实现数据库快速存取而设置的高速缓冲虚拟磁盘
- ☐ 包含一百多个库函数的运行库,可以完全控制数据库
- ☐ 还提供了两个与 db_VISTA 相关的产品,来组成一套完整的 db_VISTA III 数据库

开发系统,介绍如下:

☐ db_QUERY,是基于 SQL 语言并可用 C 语言调用的数据库查询系统,专用于 db_VISTA 数据库的查询,它为 db_VISTA 网络模型和易于使用的关系数据库的关系查询提供了网络性能和效率

☐ db_REVISE,可以把数据库的修订版直接在已有的数据库上实现,而不需要数据库的转换程序或数据库的传入、传出。它是最高级的数据库重组工具之一,允许对数据库做任何修正。

db_VISTA 的文档分为两册。第一册是《db_VISTA 用户指南》,讲述了系统使用的方法和例子。第二册是《db_VISTA 参考手册》,详细地介绍了 db_VISTA 的每个实用程序、库函数和状态码。

译者于北京

1991 年 10 月

目 录

第一章 引 言

1.1 概 述	1
1.2 文档说明	1
1.3 新的特性	2
1.4 记号约定	3

第二章 人 门

2.1 软件包内容	4
2.2 版本注释	4
2.3 安装	4

第三章 数据库概念

3.1 引 言	5
3.2 定 义	5
3.3 网络数据库模型	6
3.4 其它数据库模型	8
3.4.1 网状数据库模型	8
3.4.2 关系数据库模型	9
3.5 网络模型的优点	9
3.6 db_VISTA 数据库的组成	10

第四章 操作概述

4.1 引 言	11
4.2 系统组成	11
4.3 操作流程	13
4.4 范例介绍	15
4.4.1 需求分析	15
4.4.2 数据库设计	15
4.4.3 程序设计	18
4.4.4 程序描述	19

第五章 数据库设计

5.1 引 言	25
5.2 数据库定义语义	25
5.2.1 DDL 入门	25

5.2.2 Database 语句	26
5.2.3 Timestamp 语句	27
5.2.4 文件声明	27
5.2.5 记录声明	28
5.2.6 字段声明	30
5.2.7 集合声明	33
5.3 DDL 处理器操作	35
5.4 数据库设计考虑	38
5.4.1 逻辑设计	38
5.4.2 物理数据库设计	46
5.5 数据库设计范例	49
5.5.1 引 言	49
5.5.2 需求分析	49
5.5.3 数据库设计	50

第六章 数据库操纵

6.1 引 言	54
6.2 数据库控制	54
6.2.1 打开和关闭数据库	55
6.2.2 操作环境	55
6.2.3 数据库动态初始化	58
6.2.4 运行控制	59
6.3 当前表	60
6.4 数据检索	62
6.4.1 用关键字检索数据	63
6.4.2 用集合检索数据	64
6.4.3 直接存取检索	72
6.5 数据创建	73
6.6 数据修改	76
6.7 数据删除	77
6.8 数据库出错处理	79
6.9 多数据库存取	83
6.9.1 打开多个数据库	84
6.9.2 存取多个数据库	84

第七章 多用户数据库控制

7.1 引 言	86
7.2 操作环境	87
7.3 事务处理	88

7.4 文件锁定.....	89
7.4.1 引言.....	89
7.4.2 读锁定.....	90
7.4.3 写锁定.....	91
7.4.4 独享锁定.....	91
7.4.5 文件锁定的升级/降级	91
7.4.6 分组的锁定请求.....	92
7.4.7 锁定状态函数.....	93
7.4.8 静态记录.....	93
7.5 时 戳.....	93
7.5.1 引 言.....	93
7.5.2 实 现.....	94
7.6 记录锁定.....	97
7.7 程序设计考虑.....	98
7.7.1 基本规则.....	98
7.7.2 数据输入.....	99
7.7.3 数据删除	100
7.7.4 数据修改	102
7.7.5 数据检索	103
7.7.6 其它考虑	106
7.8 锁定管理	106
7.8.1 引 言	106
7.8.2 锁定管理操作	106
7.8.3 MS_DOS/NETBIOS	107
7.8.4 UNIX 系统 V 实现	108
7.9 事务注册和恢复	109
7.9.1 版本 2 的实现	109
7.9.2 事务溢出	109
7.9.3 自动数据库恢复	109
7.9.4 档案注册	110
7.10 单用户使用.....	111

第八章 交互式数据库存取实用程序

8.1 简 介	112
8.2 运行 IDA	112
8.3 IDA 命令菜单的使用方法	112
8.3.1 菜单命令选择	112
8.3.2 选择表	113
8.3.3 IDA 信息	113

8.4 键入与修改记录	113
8.4.1 Display/Edit Record 菜单	114
8.4.2 数据域的编辑	115
8.5 DB_VISTA 函数/IDA 交叉访问	117
8.6 实用程序命令目录	120
8.7 存取命令目录	122

第九章 文件传送实用程序

9.1 简介	134
9.2 输出数据	135
9.3 输入数据	136
9.3.1 数据输入程序的用法	137
9.3.2 输入说明语言	137
9.3.3 数据转换	150
9.4 两种实用程序的混合使用	150

第十章 其他数据库实用程序

10.1 介绍	152
10.2 数据库初始化实用程序	152
10.3 数据库一致性检查实用程序	152
10.4 建立关键字文件实用程序	153
10.5 删除链排序实用程序	153
10.6 数据库字典打印实用程序	153
10.7 格式化转储关键字文件	153
10.8 格式化转储数据文件实用程序	154

第十一章 高级专题

11.1 介绍	155
11.2 数据库文件的组织	155
11.2.1 通用的文件结构	155
11.2.2 数据文件的组织	156
11.2.3 关键字文件组织	161
11.3 数据库字典表结构	163
11.3.1 字典文件的内容	164
11.3.2 文件表	166
11.3.3 记录表	167
11.3.4 字段表	168
11.3.5 集合表	169
11.3.6 属记录表	170

11.3.7 排序表.....	171
11.3.8 组合关键字表.....	171
11.3.9 字典表的实例.....	172
11.3.10 数据库表	175
11.3.11 头文件的内容	176
11.4 当前表.....	177
附录一 版本 2 的升级	178
附录二 数据库有关的术语	179
附录三 db_CISTA3.1 的新内容	183
1. 介绍.....	183
2. 对函数的改进.....	183
3. 对实用程序的修改.....	187

第一章 引言

1.1 概述

db_VISTA 是一个数据库管理系统(DBMS),它为 C 语言应用程序的开发提供了强大的、高性能的数据库管理功能。db_VISTA 采用了网络数据库模型和 B-树索引方法,它能为 C 程序员提供一套高效的组织和访问简单及复杂信息结构的方法,同时保持最小的数据冗余。

db_VISTA 支持多种机型和操作系统,包括:MS-DOS、UNIX 和 VAX VMS。它是用 C 语言编写的,并可以得到它的全部源代码,这就能保证了 C 应用程序的可移植性,使您的应用程序能在从单用户微机到局域网、小型机甚至于大型机的广阔的市场领域中销售。

本系统的主要特性如下:

- 支持局域网和多用户计算机上的多用户操作
- 事物处理和注册
- 自动数据库恢复
- 基于网络数据库模型和快速 B-树索引的存取方法
- 多数据库存取,即一个应用程序可以同时打开和存取多个数据库
- 交互式数据库存取工具,提供易于使用的程序来控制任何存储在 db_VISTA 数据库中的信息

●文件传送工具,可以在 ASCII 码(或 DBASE II / III)与 db_VISTA 数据库之间传输文件

- 数据存储使用多个卷
- 用于定义 db_VISTA 数据库内容和结构的类似 C 语言的 DDL 语言
- 为检测数据库完善性而提供的数据库相容性检查工具
- 为实现数据库快速存取而设置的高速缓冲虚拟磁盘
- 包含一百多个库函数的运行库,可以完全控制数据库
- Raima 还提供了两个与 db_VISTA 相关的产品,来组成一套完整的 db_VISTA III 数据库开发系统,介绍如下:

●db_QUERY,是基于 SQL 语言并可用 C 语言调用的数据库查询系统,专用于 db_VISTA 数据库的查询,它为 db_VISTA 网络模型和易于使用的关系数据库的关系查询提供了网络性能和效率

●db_REVISE,可以把数据库的修订版直接在已有的数据库上实现,而不需要数据库的转换程序或数据库的传入、传出。它是最高级的数据库重组工具之一,允许对数据库做任何修正。

1.2 文档说明

db_VISTA 的文档分为两册。第一册是《db_VISTA 用户指南》,讲述了系统使用的方法

和例子。第二册是《db_VISTA 参考手册》，详细地介绍了 db_VISTA 的每个实用程序、库函数和状态码。本书是第一册，各章内容如下：

第一章《引言》，概述 db_VISTA 的主要特征以及本书结构，指出此版本包括的新性能，阐明本书的记号约定。

第二章《入门》，讲述 db_VISTA 的入门知识。本章将说明软件包的内容和安装。

第三章《数据库基本概念》，讲述 db_VISTA 数据库的基本概念，包括：本书中数据库术语定义；网络数据库模型和关系数据库模型的区别；解释 db_VISTA 数据库的基本元素等。不管以前对数据库掌握多少，每个人都应该学习本章。

第四章《操作概述》，概述 db_VISTA DBMS 的每个组件。介绍编写 db_VISTA 应用程序的通用操作流程，并给出一个简单、完整的范例。

第五章《数据库设计》，详细介绍了 DDL 语言及其使用，并讲解如何设计 db_VISTA 数据库，还给出设计实例来介绍常用结构。

第六章《数据库操纵》，讲述 C 应用程序中控制、存取和操纵数据库的 db_VISTA 库函数的使用。

第七章《多用户数据库控制》，讲述多用户和共享的数据库程序的 db_VISTA 应用。介绍基本多用户概念和 db_VISTA 用户实现的细节，并讲解进行高效地同步存取共享数据库的多用户控制函数的用法，以编写高性能的多用户应用程序。

第八章《交互式数据库存取工具 IDA》，详细讲述了 IDA 的用法。

第九章《文件传送工具》，说明在 db_VISTA 数据库和 ASCII 文本文件或 DBASE I / II 文件之间转换数据的工具的用法。

第十章《其它数据库工具》，讲解本书以外没有描述的 db_VISTA 附加工具的使用。本章描述的实用程序有：数据库初始化工具 (initdb)，数据库存取语言 (del)，数据库相容性检查工具 (dbdecl)，关键字文件建立工具 (keybuild)，字典式打印工具 (prdbd)，格式化数据文件转储工具 (datdump)，格式化关键字文件转储工具 (keydump)。

第十一章《高级专题》，细述了 db_VISTA 数据字典和数据库文件的结构和组织。

附录《版本 2 的转换》，讲解把 db_VISTA 版本 2 数据库和应用程序转变为版本 3 的必要步骤。

1.3 新的特性

新版 db_VISTA 增加的新功能如下：

- 提供了 MS-DOS 下的基于具有 NETBIOS 接口的网络、UNIX 系统
- 自动数据库恢复
- 增强文件锁定控制性能
- 记录锁定
- 多数据库存取能力
- 任选的复合关键字字段
- 用于检测其他用户所做修改的时戳
- 指定数据库文件的页大小

另外,新版本还有一些小的功能,如:DDL,列出 DDL 交叉访问表;虚存页快速查询;删除链记录的重用;数据库文件的动态重命名;DBCHECK,检查集合相容性。

最后,本书重收和扩充了使用 db_VISTA 的范例和用法。

1.4 记号约定

完整的 db_VISTA 系统包括的语言有

- 数据库定义语言(DDL)
- 输入规范语言(ISL)
- db_QUERY 结构查询语言(SQL)
- db_REVERSE 修正规范语言(RSL)

用于描述语言语法的特殊符号意义如下:

- [] 表示选项是任选的。
- / 用来间隔选项,每次只能用一项。
- ... 表示前面选项被重复零次或多次。当在一行之中使用时,表示符号前的选项(通常在括号中)被重复。当...占一行时,表示上一行被重复零次或多次。在程序范例中使用表示省略的代码,与上述意义无关。

第二章 入门

2.1 软件包内容

先检查软件包内容是否完整。软件包应包含：

- db_VISTA 版本信息
- db_VISTA 用户指南
- db_VISTA 参考手册
- 装有适用于特定操作系统的 db_VISTA 的磁盘

2.2 版本注释

软件包中的分类文档包含了特定操作环境下 db_VISTA 的版本注释。此文档详细描述了磁盘的内容,本书中未说明的当时系统的最新改动,关于操作系统和编译器的信息,以及安装说明。一定要仔细阅读此文档。1号盘上的 README 文件描述了对 db_VISTA 的最后修改。

2.3 安装

安装说明见上节及 README 文件。

第三章 数据库概念

3.1 引言

本章所述的数据库基本概念对理解 db_VISTA 系统非常重要。3.2 节中定义了本书所使用的数据库术语。3.3 节将介绍 db_VISTA 系统的基础——网络数据库模型。3.4 节将介绍两种在其它数据库中使用的数据库模型。3.5 节将介绍网络数据库模型相对于关系型数据库模型的优点。3.6 节将介绍 db_VISTA 数据库的组成。

虽然本章提供的大量知识能使初学者有效地使用 db_VISTA,但是,数据库技术非常复杂,需要做进一步的学习才能透彻理解。理解本章中的概念对于灵活使用 db_VISTA 是必不可少的。

3.2 定义

数据库中信息的基本单位是“字段”(field)。字段(或数据字段)是具有名字、类型(例:字符型或整型)和长度等属性的数据项。例如:雇员姓名、生日、社会保险号码、财产清单号码和编号。其它数据库系统或著作中可能将字段称为 attribute, entity 或 column。

“记录”(record)是带有名称的相关字段集合,它是存储和访问的基本单位。例如,在一个查帐数据库中,名为“check”的记录可能包含以下字段:

- 日期
- 帐号
- 帐目用途
- 数量

数据库中帐目记录的每一事务都包含了上述字段的值。对“记录”的其它叫法有“表”(table)或“文件”(file)。被定义记录(由字段构成)称为“记录类型”,类似于 Pascal 或 C 语言中的记录以及 C 与 PL/I 语言中的结构。

记录类型中的所有信息都存储在文件中。文件是组成数据库的基本物理存储单元。一个数据库,实际上就是一些相关文件的集合。

要快速或顺序访问记录需要通过一个被称为“关键字”的字段。例如在上述的 check 记录中“帐号”即是关键字,因为指定一个帐号,就能马上找到相应的 check 记录。

本书的目录即是一个典型的例子。目录的内容相当于“关键字”,而相应书中的内容就是“记录”。通过目录查找要看的页比一页一页翻要快得多。另外,因为关键字在“索引”中也是排序的,所以关键字也很快就能找到。关键字文件也是类似的,只不过排序和查找工作由计算机完成。一种效率最高的索引查询技术叫做 B 树方法,db_VISTA 对关键字文件即采用了此法。

“关键字扫描”就是对索引中每个关键字进行顺序读取的操作,可以用来产生记录的顺序表以及对指定的大量记录(如两个日期之间输入的帐单)进行快速检索。

记录类型之间通常存在着数据联系。例如,上述的查帐数据库可能包含“预算”(budget)记录类型,可以被定义由以下字段组成:

- 预算码(关键字字段)
- 预算描述
- 月报
- 差额

这时需要把预算记录用 check 记录联系起来。通过在 check 记录类型中加一个预算码字段,就能在两种记录类型间建立某种联系。这样只需输入一个 check 记录,就能根据“预算码”立即找出相应的 budget 记录,并根据 check 中的“数量”对“差额”情况做适当修改。

数据库的内容和组成的概念性定义称为“模式”(schema)。模式应包括所有的记录类型及其字段和关键字的定义等。DBMS 把这个模式称为“数据字典”(dictionary)。在 db_VISTA 及其它许多 DBMS 中,“模式”用数据库定义语言即 DDL 来定义。

下面是 db_VISTA DDL 定义的查帐数据库。实际的 DDL 语句将在第五章中讲述。

```
database ckngacct{
    data file "chkng.dat" contains budget,check;
    key file "chkng.key" contains code,check__no;

    record budget{
        key char code[6];
        char cat__desc[48];
        float allocation;
        float balance;
    }

    record check{
        key int check__no;
        int check__date;
        char bud__code[6];
        char paid__to[48];
        float amount;
    }
}
```

数据模型或数据库模型是记录间关系的概念性表述。在查帐数据库的例子中,budget 记录与 check 记录的联系是通过“预算码”这一共同字段来建立的。通过共同字段来建立和维持记录间关系的方法,称为“关系型数据库模型”。其他的数据库模型,特别是 db_VISTA 支持的“网络数据库模型”可以不通过共同数据字段,而是通过相关记录的物理连接,来支持建立记录间的联系。下一节将详细讲解这些概念。

3.3 网络数据库模型

网络数据库模型就是用集合来清楚地定义并直接维护记录类型之间的联系。一个“集合”(set)定义了两个记录类型间的一对多关系。例如:

一个棒球联合会有多支棒球队，

一支棒球队有多名球员。

集合由连接的指针表构成，指针指向网络中相关记录的位置。

图 3.1 讲解上述棒球例子中的集合关系，其中方框代表联合会、球队和球员记录类型，箭头代表相关记录的连接。

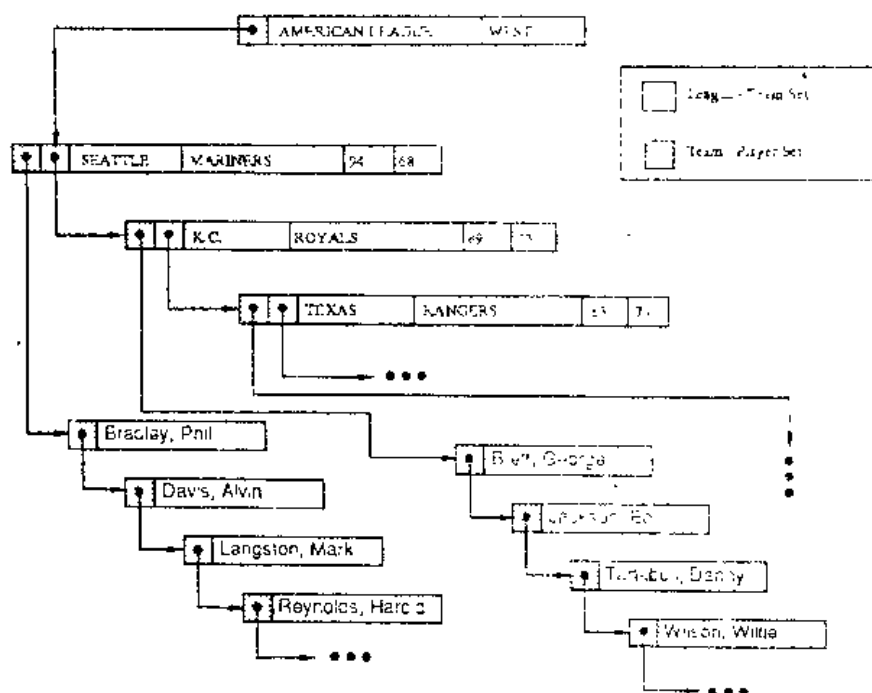


图 3.1 集合关系的例子

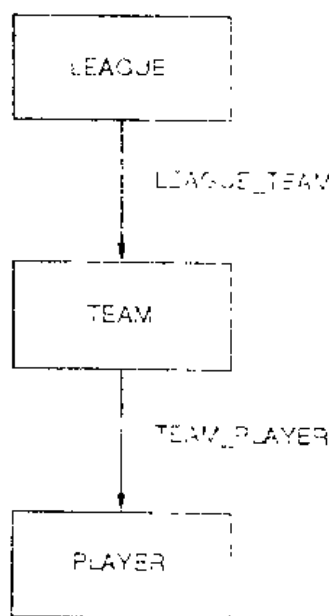


图 3.2 模式图的示例

数据库设计中用“模式图”来阐明相关数据记录的关系。图 3.2 即为网球例子的模式图。在此图(及本书中所有模式图)中，方框代表记录类型，箭头代表集合。league_team 集合

构成了 league 记录(称为 league_team 集的“首记录”)和 team 记录(称为 league_team 集的属记录)间的一对多联系。team_player 集构成了 team 记录和 player 记录间的一对多联系。

在查帐的例子中,可以用一个 transactions 集合来描述 budget 记录与 check 记录的联系,对特定的预算描述定义了 budget 记录(首记录)和 check 记录(属记录)之间的一对多联系。这时,check 记录中的“预算码”字段就是多余的了,可以不在 check 记录类型中定义。可以把 db_VISTA DDL 修改如下:

```
database ckngacct{
    data file "chkng.dat" contains budget,check;
    key file "chkng.key" contains code,check_no;

    record budget{
        key char code[6];
        char cat_desc[48];
        float allocation;
        float balance;
    }

    record check{
        key int check_no;
        int check_date;
        char paid_to[48];
        float amount;
    }

    set tranmotions{
        onder last;
        owner budget;
        member check;
    }
}
```

一个记录类型可以是多个集合中的首记录,也可以同时是其它多个集合中的属记录。这样,图 3.3 模式图中的网络结构是正确的。

注意,记录 E 指向它本身。记录 C 是两个集合中的属记录,同时也是另两个集合中的首记录。记录 B 在多个集合中作为 E 的首记录出现,所有这些都是 db_VISTA 的合法结构。

3.4 其它数据库模型

还有两个主要的数据库模型是网状模型和关系模型,将在下面介绍。

3.4.1 网状数据库模型

网状数据库模型是网络模型的特例,其中一个记录只能是一个集合的属记录,但仍可以

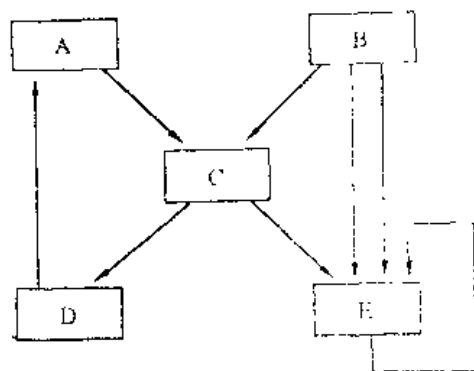


图 3.3 网络结构的例子

是多个集合的首记录。首记录被称为“父”，属记录被称为“子”。

象网络模型一样，集合也是用连接表来实现。图 3.4 给出了一个网状模式结构的例子。

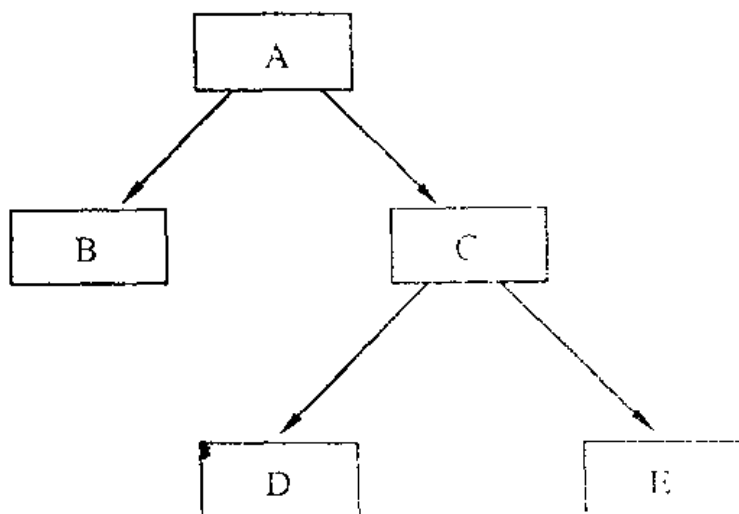


图 3.4 网状结构的例子

3.4.2 关系数据库模型

关系数据库模型把数据库表示成二维表。它的列(或称属性)相当于数据字段，行相当于记录。这种用表格来表示的数据库很容易来使用基于集合代数理论的标准关系数据库操作来操纵。

关系模型表中的联系通常是用共同的数据字段来建立。回想查帐的例子中 budget 记录与 check 记录间的联系，就是通过 check 记录中用来标识预算描述的“预算码”来建立的。

关系模型与网络模型的主要区别是关系模型通过共同字段来建立记录间联系，而网络模型可以直接定义这些联系。

注意，可以把关系型数据库转变成网络型，也可以把网络型数据库转变成关系型。

3.5 网络模型的优点

关系型数据库系统非常流行主要是由于它对基于简单数据模型的使用非常容易，而网络模型的最大优点在于它的高性能。考虑图 3.5 中的图。

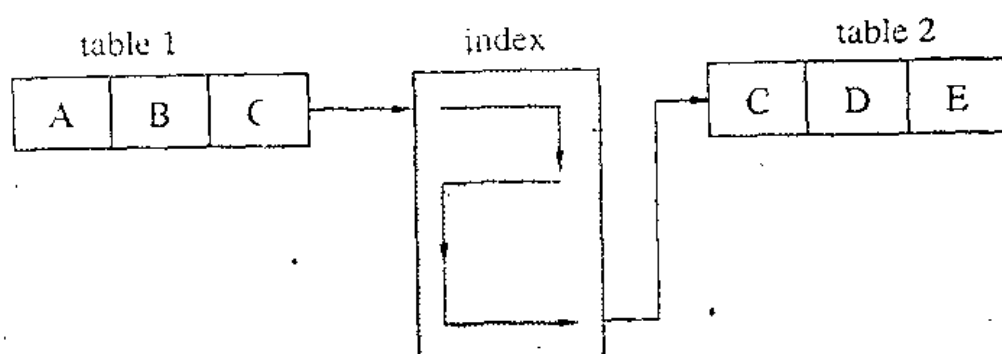


图 3.5 关系型 DBMS

此图说明两个表用共同数据字段 C 来建立联系。注意 C 必须在两个表中都定义,并且为访问表 2 必须有一个索引。把此图与图 3.6 中有关网络模型的图比较。

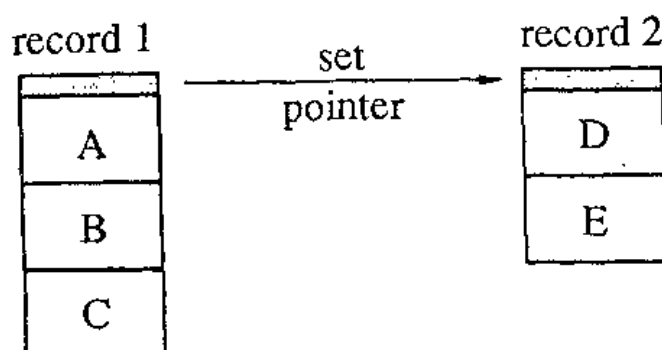


图 3.6 网络 DBMS

网络模型直接定义两个记录的联系而无需重复字段和索引文件,从而消除了数据冗余。并且相关的记录可以在数据库中直接读取,不用先读索引再读记录。

但是,如果一个数据库设计使用索引更方便时,db_VISTA 也能够实现。网络存取和索引存取是 db_VISTA 提供的两种独立方法,两者共同使用将给数据库设计带来最大的灵活性。

3.6 db_VISTA 数据库的组成

下面定义的数据库元素构成了 db_VISTA 数据库。db_VISTA 数据库包括:

- ☐“数据字典”,存储了描述数据库内容和组织的信息;
- ☐“数据文件”,包含了一个或多个记录类型;
- ☐“关键字文件”,包含了一个索引及若干个关键字字段。

记录包括数据字段、关键字字段和集合连接(对用户透明)。它们可以用集合导航方式(即通过指定集合的相应连接表)或关键字段(索引快速查找)来存取,也可两者都用。

第五章和第六章将详尽阐述这些元素。