

● 全国师范高等专科学校

计算机系列教材

数据库 技术与应用

王景才 申时凯 陈玉国 连志春



高等教育出版社

全国师范高等专科学校计算机系列教材

数据库技术与应用

王景才 申时凯 陈玉国 连志春

高等教育出版社

内容提要

本书是全国师范高等专科学校计算机系列教材之一。本书在全面介绍 Visual FoxPro 6.0 系统概貌的基础上,着重介绍了利用 Visual FoxPro 6.0 提供的系统功能进行程序设计的方法和技巧,并针对典型实例给出多种实现方法。全书共分 12 章:第一章介绍有关数据库的知识和概念;第二~十章重点介绍 Visual FoxPro 6.0 的安装与配置,菜单和工具栏及项目管理器的使用和操作,表、索引、查询、视图、表单、报表的创建,记录的操作和菜单的设计;第十一、十二章重点介绍 Visual FoxPro 的程序设计基础和面向对象的程序设计方法。

本书内容全面,概念清楚,实用性强,可作为高等师范专科学校相关专业的计算机教材,亦可作为高等院校学生和广大计算机爱好者学习掌握数据库技术与应用的参考教材。

图书在版编目(CIP)数据

数据库技术与应用 / 王景才等编. —北京:高等教育出版社, 2000
ISBN 7-04-008652-2

I. 数... II. 王... III. 数据库系统
IV. TP311.13

中国版本图书馆 CIP 数据核字 (2000) 第 23096 号

数据库技术与应用

王景才 申时凯 陈玉国 连志春

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

经 销 新华书店北京发行所

印 刷 高等教育出版社印刷厂

开 本 787×1092 1/16

版 次 2000 年 7 月第 1 版

印 张 14

印 次 2000 年 7 月第 1 次印刷

字 数 340 000

定 价 18.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

数据库技术是 20 世纪 60 年代开始兴起的一门综合性的软件技术。数据库技术是数据管理的最新方法,也是信息管理中的一项非常重要的技术。进入 20 世纪 90 年代后,随着电脑和网络技术的普及以及计算机硬件的更新,计算机可视化技术在数据库技术中得到了广泛的应用。

数据库管理系统是帮助人们处理大量信息、实现管理科学化和现代化的强有力的工具。Visual FoxPro 是 xBASE 数据库家族中继 dBASE、FoxBASE 和 FoxPro 之后又一广泛使用的微机关系数据库管理系统,是在 Windows 平台上广泛使用的数据库开发工具,它以简单易学、方便灵活、适用面宽等优点深受广大计算机用户的青睐。

本书主要介绍 Microsoft Visual FoxPro 6.0 版。本书编写大纲征求了部分高等师范院校专家、教授的意见,编写过程中既兼顾了已经熟悉 FoxBASE+和 FoxPro 的读者,同时又在详细介绍 FoxPro 语言语法的基础上,深入浅出地介绍了使用编程方法实现对 Visual FoxPro 数据库的操作,是高等师范院校学生学习掌握数据库技术与应用的一本比较理想的教材,亦可作为其他高等院校学生和广大计算机爱好者学习掌握数据库技术与应用的参考教材。

全书共分十二章,第一章介绍有关数据库的相关知识和概念,第二、三、四、五、六、七、八、九、十章重点介绍 Visual FoxPro 6.0 版的基本操作,第十一、十二章重点介绍 Visual FoxPro 的程序设计。第一章由王景才编写,第二、三、四、五章由连志春编写,第六、七、八、九、十章由申时凯编写,第十一、十二章由陈玉国编写。王景才负责全书的策划和修改定稿。

由于时间仓促,内容较多,再加上作者本身的水平有限,书中错误在所难免,敬请广大读者批评指正。

编 者

2000 年 3 月

目 录

第一章 数据库系统概述	1	3.2.8 “帮助”菜单	31
1.1 引 言	1	3.3 使用工具栏	32
1.1.1 人工管理阶段	1	3.3.1 “常用”工具栏	33
1.1.2 文件系统阶段	2	3.3.2 “数据库设计器”工具栏	34
1.1.3 数据库系统阶段	2	3.3.3 “表单控件”工具栏	34
1.2 信息、数据与实体	3	3.3.4 “表单设计器”工具栏	34
1.2.1 现实世界	3	3.3.5 “报表控件”工具栏	34
1.2.2 信息世界	3	3.3.6 “报表设计器”工具栏	34
1.2.3 数据世界	4	3.3.7 “布局”工具栏	34
1.3 数据模型	4	3.3.8 “查询设计器”工具栏	35
1.3.1 实体间的联系	4	3.3.9 “其他”工具栏	35
1.3.2 数据模型	6	3.4 使用命令窗口	35
1.4 数据库系统的体系结构	8	3.5 使用帮助	36
1.4.1 数据库系统的抽象层次	8	习题三	36
1.4.2 数据库系统的组成	10	第四章 项目管理器	37
1.5 Visual FoxPro 概述	12	4.1 创建项目	37
习题一	13	4.2 使用项目管理器	39
第二章 Visual FoxPro 6.0 安装与配置	14	4.2.1 “项目管理器”窗口	40
2.1 安装 Visual FoxPro 6.0	14	4.2.2 使用项目管理器	42
2.2 安装后定制系统	18	4.2.3 自定义项目管理器	44
2.3 配置 Visual FoxPro 6.0	19	习题四	45
2.4 系统的优化	21	第五章 创建表	46
习题二	21	5.1 创建数据库	46
第三章 进入 Visual FoxPro 6.0	22	5.1.1 创建数据库	46
3.1 进入 Visual FoxPro 6.0	22	5.1.2 打开数据库	48
3.2 使用菜单	24	5.1.3 关闭数据库	49
3.2.1 “文件”菜单	25	5.1.4 删除数据库	49
3.2.2 “编辑”菜单	27	5.2 创建表	50
3.2.3 “显示”菜单	28	5.2.1 字段	50
3.2.4 “格式”菜单	28	5.2.2 创建表	52
3.2.5 “工具”菜单	29	5.2.3 操作表	64
3.2.6 “程序”菜单	30	5.3 添加记录	68
3.2.7 “窗口”菜单	31	5.3.1 创建表同时添加记录	68

5.3.2 用 INSERT 命令添加记录	68	7.3.1 使用向导创建视图	112
5.3.3 用 APPEND 命令添加记录	70	7.3.2 使用视图设计器创建视图	115
5.3.4 从其他表中追加记录	70	7.4 删除视图	118
5.4 设置字段的高级属性	71	7.5 更新视图	118
5.4.1 设置字段的标题	71	习题七	121
5.4.2 设置字段的显示格式	73	第八章 创建表单	123
5.4.3 设置字段的缺省值	74	8.1 创建表单	123
5.4.4 设置字段的注释	74	8.1.1 用表单向导创建表单	123
5.4.5 设置字段的有效性规则	75	8.1.2 用表单设计器创建表单	123
5.5 Visual FoxPro 6.0 的命令工作方式	75	8.2 修改表单	125
5.5.1 创建和查看库或表的命令	76	8.2.1 使用表单设计器工具栏	125
5.5.2 操作库或表的命令	78	8.2.2 选择、移动和缩放控件	125
5.5.3 Visual FoxPro 6.0 命令中的常用子句	85	8.2.3 复制和删除表单控件	126
5.6 创建索引	86	8.2.4 对齐控件	126
5.6.1 用表设计器创建索引	87	8.2.5 调整控件的位置	126
5.6.2 用命令创建索引	93	8.2.6 设置控件的 Tab 键次序	127
5.7 修改表	93	8.3 向表单中添加控件	128
5.7.1 用项目管理器修改表	94	8.3.1 用生成器向表单中添加控件	128
5.7.2 用 MODIFY STRUCTURE 命令修 改表	95	8.3.2 向一个用向导生成器生成的表单中 添加控件	129
习题五	95	8.3.3 添加控件和设置控件属性	129
第六章 操作记录	97	8.4 定制表单	130
6.1 编辑记录	97	8.4.1 改变文本的字体和大小	131
6.2 删除记录	98	8.4.2 向表单中添加“形状”和“线 条”	131
6.2.1 标记删除记录	98	8.4.3 向表单中添加图形	131
6.2.2 恢复记录	101	8.4.4 设置表单的前景和背景颜色	131
6.2.3 完全删除记录	102	8.5 保存和运行表单	132
6.3 查看特定记录	102	8.5.1 保存表单	132
6.3.1 使用“转到记录”菜单栏	102	8.5.2 运行表单	132
6.3.2 使用“查找”菜单栏	103	习题八	132
习题六	104	第九章 创建报表	133
第七章 创建查询与视图	105	9.1 报表介绍	133
7.1 创建查询	105	9.2 创建报表	133
7.1.1 使用向导创建查询	105	9.2.1 创建简单报表	135
7.1.2 使用查询设计器创建查询	105	9.2.2 创建分组/总计报表	138
7.2 运行查询	108	9.2.3 创建一对多报表	141
7.2.1 指定查询去向	109	9.3 修改报表	145
7.2.2 运行查询	112	9.4 定制报表	148
7.3 创建视图	112		

9.4.1 设置报表数据源	148	11.4.2 选择结构	170
9.4.2 定义报表的页面	149	11.4.3 循环结构	172
9.4.3 增添报表控件	150	11.4.4 过程和函数	177
9.4.4 设置域控件格式	150	习题十一	181
9.4.5 添加线条、矩形和圆形	151	第十二章 面向对象程序设计	185
9.4.6 添加图片	152	12.1 面向对象程序设计的基本概念	185
9.4.7 更改控件颜色	153	12.1.1 面向对象程序设计的发展历史	185
9.4.8 为报表控件添加注释	153	12.1.2 面向过程的程序设计和面向对象的程序设计	186
9.5 预览和打印报表	153	12.1.3 面向对象程序设计的基本概念	187
习题九	154	12.2 Visual FoxPro 中的类	189
第十章 设计菜单	155	12.2.1 Visual FoxPro 中的基类	189
10.1 菜单简介	155	12.2.2 Visual FoxPro 类的分层结构	190
10.2 创建菜单	155	12.2.3 Visual FoxPro 基类的事件和属性	191
10.2.1 使用菜单设计器创建菜单	156	12.2.4 访问对象的属性	191
10.2.2 使用命令创建菜单	161	12.2.5 相对引用	193
习题十	161	12.3 创建类	194
第十一章 Visual FoxPro 程序设计基础	162	12.3.1 使用类设计器创建类	195
11.1 数据类型与数据容器	162	12.3.2 通过编程创建类	199
11.1.1 数据类型	162	12.4 修改类	203
11.1.2 数据容器	162	12.4.1 定义新属性	203
11.2 操作符及表达式	164	12.4.2 编辑新属性	204
11.3 如何编写 Visual FoxPro 程序	166	12.4.3 类的删除、更名、复制	204
11.3.1 创建 Visual FoxPro 程序	167	12.5 操作对象	205
11.3.2 保存 Visual FoxPro 程序	167	12.5.1 添加可视化对象	205
11.3.3 编辑 Visual FoxPro 程序	168	12.5.2 创建对象	212
11.3.4 运行 Visual FoxPro 程序	169	习题十二	217
11.4 程序的控制结构	170		
11.4.1 顺序结构	170		

第一章 数据库系统概述

1.1 引言

数据库技术是 20 世纪 60 年代开始兴起的一门信息管理自动化的新兴学科，是计算机科学中的一个重要分支。随着计算机应用的不断发展，在广泛应用计算机的领域中，数据处理越来越占主导地位，数据库技术应用也越来越广泛。它已在科研、管理、商业、军事、教育和文化等方面受到人们的普遍重视。

数据库是数据管理的产物，随着计算机软硬件的不断发展，30 多年来，数据管理经历了人工管理、文件系统和数据库系统 3 个重要阶段。

1.1.1 人工管理阶段

在计算机用于数据处理的初级阶段，数据的管理是由程序员个人考虑和安排的。程序员把数据处理纳入程序设计的过程中，除了编制程序之外，还要考虑到数据的逻辑定义和物理组织，以及数据在计算机存储设备内的物理存储方式。程序和数据混为一体，在需要引用数据时，直接按地址存取。

这个时期管理的特点是：

1. 不保存数据

计算机主要用于科学计算，每次在算题之前将数据输入，计算结束后，将结果数据输出，计算机中不保存数据。

2. 数据与程序不具有独立性

由于没有软件系统对数据进行管理，程序的存储结构、存储方法和输入输出方式等都随着存储的改变而改变，数据在存储上稍有改变，就要修改程序。

3. 数据是面向应用的

一般地，一组数据对应一个程序，程序与程序之间出现数据重复。同时应用程序各自组织和使用各自的数据，系统对于用户的数据基本上处于放任自流的状态。

人工管理阶段应用程序与数据之间的关系见图 1.1。

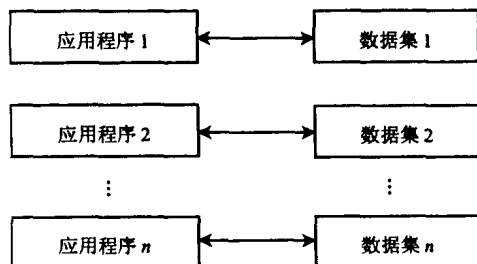


图 1.1 人工管理阶段的数据处理

1.1.2 文件系统阶段

当计算机操作系统包含有文件系统后,把数据组成文件的形式就使得计算机数据管理进入了一个新阶段。这一阶段的数据管理的特点是:

1. 数据以文件的形式长期保存

由于计算机大量用于数据处理,所以数据以文件的形式长期保存在外存上被反复处置。

2. 程序与数据之间有一定的独立性

由于有软件进行数据管理,所以程序与数据之间需要有某种存取方法进行转换(如图 1.2 所示),需要有共同的数据查询修改的例行程序存放在程序库中。文件的逻辑结构与存储结构有一定区别,程序与数据有一定的独立性,数据不能共享。由于文件系统数据的冗余度很高,文件中可能有很多数据是重复的,重复的数据不仅浪费了存储单元,而且在操作时很容易造成一个或同一组数据在不同的文件中不一致。

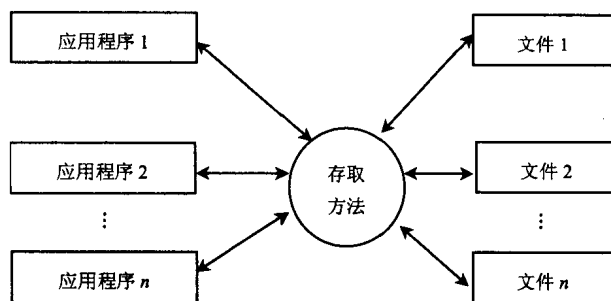


图 1.2 文件系统阶段的数据处理

3. 文件多样化

由于有了直接存取存储设备,也就有了索引文件、链接文件、直接存取文件等,而且使用倒排文件进行多码检索。

尽管有了上述几个优点,但它仍然是一个不具有弹性的无序的信息集合,存在着冗余度大、空间浪费、文件不易扩充、修改费时、有可能引起不相容等缺点,反映不出现实世界事物之间广泛的内在联系。文件系统基本上是一个文件对应于一个或几个应用程序,数据是面向应用的。

1.1.3 数据库系统阶段

随着数据管理规模的不断扩大,为了满足大量数据集中存储并提供给众多用户使用的要求,20 世纪 60 年代末期,数据库技术被广泛用于数据处理。

数据库系统的特点是:

1. 采用复杂结构化的数据模型

数据库系统不仅要描述数据本身,还要描述数据之间的联系。这种联系是通过存取路径来实现的,通过存取路径来表示自然的数据联系是数据库与传统文件系统的根本区别。数据库中的数据是公用的、综合的,以最优的方式去适应多个应用程序的需要。

2. 最低的冗余度

数据库系统中的重复数据被减少到最低程度,这样,在有限的存储空间内可以存放更多的数据。

3. 较高的数据独立性

数据和程序彼此独立，数据存储结构的变化尽量不影响用户程序的使用。

4. 数据控制功能

数据库系统具有数据的安全性，以防止数据的丢失和被非法使用；具有数据的完整性，以保护数据的正确、有效和相容；具有数据并发控制，避免并发程序之间的相互干扰；具有数据的恢复功能，在数据库被破坏或数据不可靠时，系统有能力把数据库恢复到最近某个时刻的正确状态。

上述 4 个方面构成了数据库系统的主要特征，数据库系统阶段的程序和数据之间的关系可用图 1.3 表示。

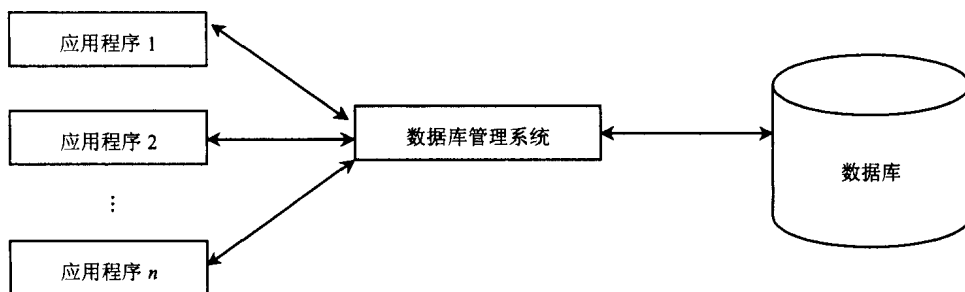


图 1.3 数据库系统阶段的数据处理

综上所述，数据库可以被定义为：一个存储起来为某个特定组织提供多种应用服务并具有尽可能小冗余度的互相关联的数据集合。其数据结构独立于使用数据的程序，对数据的增添、删除、修改和检索，由系统进行统一的控制。

1.2 信息、数据与实体

数据是描述现实世界、载荷信息的物理符号，信息是反映事物物理状态的。现实世界中的一切信息都可以用数据来表示。从现实世界中提取什么来描述以及如何描述，反映了使用者的意图和要求。信息来源于现实世界。经过人们的收集、选择、命名和分类，又作用于现实世界。信息的这一循环跨越了 3 个不同的世界，即现实世界、信息世界和数据世界。

1.2.1 现实世界

存在人们头脑之外的客观世界称为现实世界。现实世界中的事物可以分成“对象”和“属性”两大类。在这个世界里，事物都是千差万别且又相互联系的。例如，仓库管理中首先涉及到的管理，包括货物的存放、进出和检查等。这里就可能有许多报表、图表，都是数据库系统接触到的最原始的数据，对这些数据就要进行“去粗取精、去伪存真、由此及彼、由表及里”的工作，取出数据库系统所需要的数据。譬如在仓库管理中要把输入报表、输出报表、查询格式等的数据收集起来，进行分类，综合出系统所要的数据。

1.2.2 信息世界

信息世界是现实世界的事物在人们头脑中的反映。客观事物在信息世界中称为实体，实体是彼此可以明确识别的对象。实体可分成“对象”与“属性”两大类，例如，人、学校、

商店等属于对象。而表示对象“人”的属性有姓名、性别、年龄、出生日期、籍贯等多方面的特征，属性是客观事物中性质的抽象描述。

1.2.3 数据世界

数据世界是关于所有的事物形成的信息世界，是实体模型数据化。在计算机世界里，通常用下列术语描述数据。

1. 字段

把描述实体属性的数据称为字段或数据项。它是可以命名的最小信息单位，所以又叫数据元素或初等项。字段的命名往往和属性相同。例如职工有工号、姓名、性别、年龄、工作日期、职务和工资等字段。

2. 记录

字段的有序集合称为记录。一般用一个记录描述一个实体，所以记录又可定义为能完整地描述一个实体的字段集。例如一个职工的记录由工号、姓名、性别、年龄、工作日期、职务和工资等有序字段组成。

3. 文件

同一类记录的集合称为文件。文件是描述实体集的，所以它又可以定义为描述一个实体集的所有记录集。所有的职工记录组成了一个职工文件。

4. 关键字

能唯一标识文件中每一个记录的字段或字段集称为关键字。例如职工的工号可以作为职工记录的关键字。

在数据库系统中，每一个概念都有型和值之分，例如职工实体集中，职工是一个实体的型，而具体的人张红、李梅等是实体的值。又如工号、姓名、性别和年龄是型，1004、张红、男和20是值。对于记录和字段也是这样，也有型和值之分。

1.3 数据模型

1.3.1 实体间的联系

现实世界中的事物都是彼此关联的，任何一个实体都不是独立存在的，因此描述实体的数据也是互相关联的。联系有两种：一种是实体内部的联系，反映在数据上是记录内部即字段间的联系；另一种是实体与实体之间的联系，反映在数据上是记录之间的联系。

数据库系统不但要考虑记录内部的联系，还必须考虑记录之间的联系。这种联系是比较复杂的，根据这种联系设计的数据结构也是复杂的，这就使得数据库系统也变得比较复杂了。

实体间的关系虽然复杂，但经抽象化后，可把它们归结为3类：一对一联系、一对多联系和多对多联系。

1. 一对一联系

若对于实体集A中每一个实体，实体集B中至多有一个实体与之联系，反之亦然，则称实体集A与实体集B具有一对一的联系，记为1:1。如图1.4(a)所示。

2. 一对多联系

若对于实体集 A 中的每一个实体, 实体集 B 中至少有一个以上实体与之联系; 反之, 对于实体集 B 中的每一个实体, 实体集 A 中至多有一个实体与之联系, 则称实体集 A 与实体集 B 具有一对多的联系, 记为 $1:n$ 。如图 1.4(b) 所示。

3. 多对多联系

若对于实体集 A 中的每一个实体, 实体集 B 中至少有一个以上实体与之联系, 反之亦然, 则称实体集 A 与实体集 B 具有多对多的联系, 记为 $m:n$ 。如图 1.4(c) 所示。

从图 1.4 中可以看出 $1:1$ 联系是 $1:n$ 联系的特例, 而 $1:n$ 又是 $m:n$ 联系的特例。三者之间的关系是包含关系。

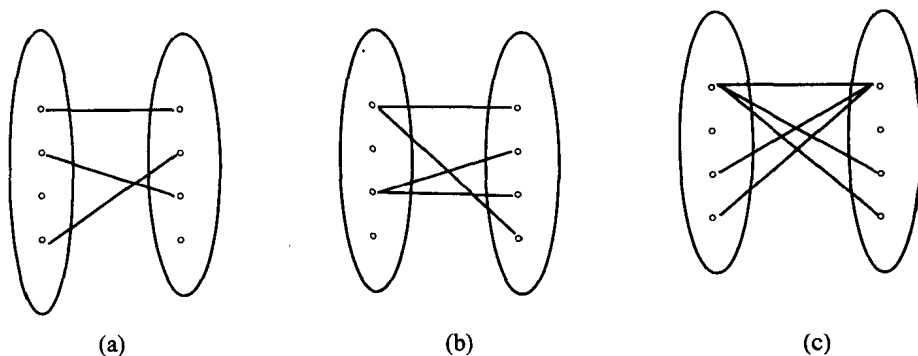


图 1.4 两个实体集之间的联系

实体间的联系可用实体联系模型(E-R 模型)来表示, 这种模型直接从现实世界中抽象出实体类型及实体间联系, 然后用实体联系(E-R)图表示数据模型, 见图 1.5。

E-R 模型有三种基本成分: 实体、联系和属性。

(1) 用矩形框表示实体型, 并在框内写上实体名。

(2) 用椭圆形表示实体的属性, 并用无向边把实体与其属性连接起来。

(3) 用菱形表示实体间的联系, 并在菱形框内写上联系名。

(4) 用无向边把菱形分别与有关实体相连接, 在无向边旁注上联系类型。如实体之间联系也具有属性, 则把属性和菱形也用无向边连接上。

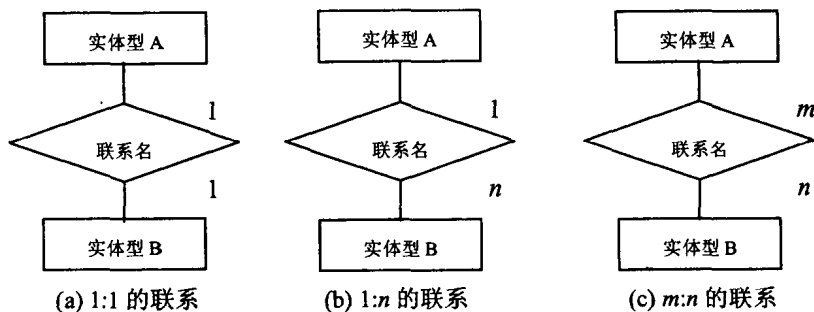


图 1.5 实体集联系 E-R 图

例如, 在实体联系中, 用图 1.6 表示一个公司只有一个经理。用图 1.7 表示一个经理可以领导多名职工(因为经理也是职工的一员, 所以这属于同一实体集内的一对多联系)。用图 1.8 表示每个学生可以选修多门课程, 同时一门课可以有多个学生选修。

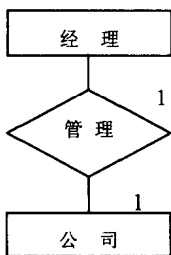


图 1.6 1:1 型联系

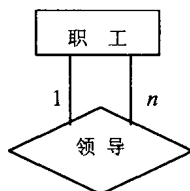


图 1.7 1:n 型联系

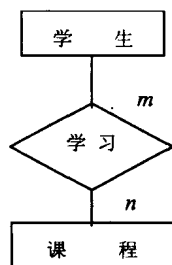


图 1.8 m:n 型联系

对于某一个实体本身具有多个属性的，或 3 个及更多个实体之间的联系也可用 E-R 图表示。

E-R 图方法是抽象和描述现实世界的有力工具，是数据模型的基础，因而比数据模型更一般、更接近现实世界。

1.3.2 数据模型

实体模型是客观实体的运动规律和实体之间相互关系的抽象和表达。数据模型则是实体模型的数据表示，它仍然间接地反映了客观实体之间的相互联系与依存关系。

由实体联系模型到数据模型的转换需要做以下几件事情：

- (1) 给数据模型命名，使不同模型得以区别。
- (2) 给每个记录型命名，以表示和说明同一模型中具有的记录类型。
- (3) 给每个数据项命名，以说明和区分每个记录类型所具有的数据项，并确定记录的主关键字。
- (4) 说明各个记录型之间的联系，必要时给这种联系命名。

1. 层次模型

数据的层次模型是以记录为结点的有向树。它的主要特征是：

- (1) 有且仅有一个结点无父结点，这个结点为树的根，称为根结点。
- (2) 其余的结点有且仅有一个父结点。

图 1.9 是层次模型的结构示例。图中无父结点的结点 A 为根结点。同一父结点的结点称为兄弟结点(或枝结点)，如图中的 B 和 C 是兄弟，D 和 E 是兄弟，F、G 和 H 是兄弟，I 和 J 是兄弟。没有后代结点的结点称为叶结点，如图中的 D、I、J、F、H 和 G 都是叶结点。

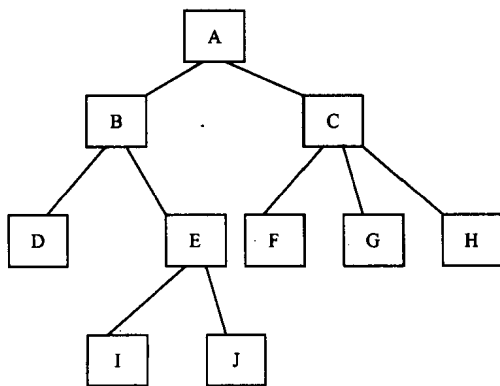


图 1.9 层次型数据模型

层次模型的优点是联系简单、层次分明。缺点是每层之间的联系要遵循既定的路径。当查询某个下层数据时，只能从根到枝再到叶，不能倒查，也不能从中间插入。同时，更主要的是层次模型难于表示多对多的联系。

2. 网络模型

网络模型亦称网状模型。它是以记录类型为结点的网络结构。这种数据模型与层次模型的显著区别在于：

- (1) 可以有一个以上结点无双亲。
- (2) 至少有一个结点有多于一个的父结点。

层次模型中从子女到双亲的联系是唯一的，而网络模型则允许有至少一个的父结点。图 1.10 是一个教学实体的网络数据模型。

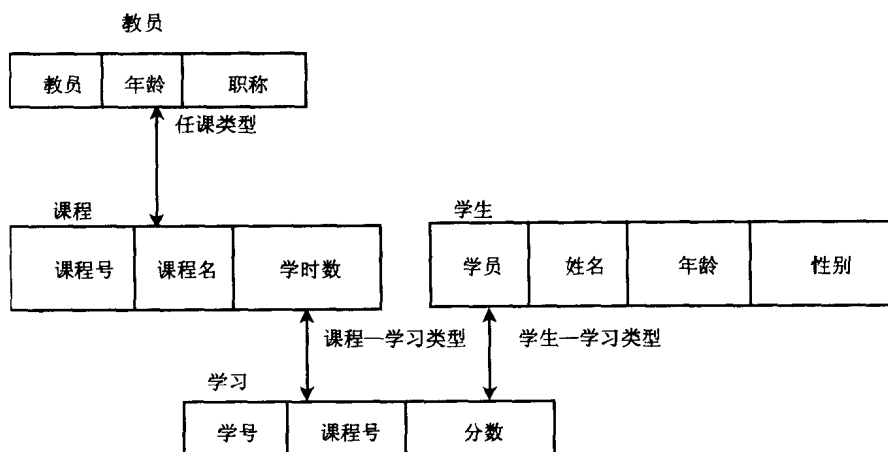


图 1.10 数据的网络模型

3. 关系模型

关系模型以关系代数为理论基础，关系模型是把数据的逻辑结构归结为满足一定条件的二维表的模型。在关系模型中，每一个关系是一个二维表，用来描述实体或实体之间的联系(如表 1.1 所示)。表格中的每一行是一个元组，它相当于一个记录值，用以描述一个个体。在关系模型中，每一列称为一个属性，相当于记录中的一个字段。属性的取值范围称为域。关键字是一个或若干个属性的集合，它能唯一地标识关系中的各元组。

表 1.1 设备表

设 备	编 号	价 格	部 门
车床	016-1	62 044.61	21
车床	016-2	27 132.73	21
磨床	037-2	241 292.12	22
钻床	038-1	5 275.00	23
微机	100-1	8 810.00	12
复印机	101-1	10 305.01	13

关系模型中的关系具有如下的性质:

- (1) 关系中的每一列都是不能再分的数据项。
- (2) 关系中的每一列中所有的数据属于同一类型。
- (3) 关系中各列被指定相异的名字。
- (4) 关系中各行相异, 不允许有重复的行。
- (5) 关系中行和列的次序可以任意调换, 不影响它们的信息内容。

1.4 数据库系统的体系结构

1.4.1 数据库系统的抽象层次

在数据库系统中, 用户看到的数据与存放在计算机中的数据不是一回事。实际上它们之间已经过了两次变换。一次是系统为了减少冗余, 实现数据共享, 把所有用户的数据进行综合, 抽象成一个统一的数据视图。第二次是为了提高存取效率, 把全局视图的数据按照物理组织的最优形式来存放。

整个数据系统分为三级: 外部级(子模式)、概念级(模式)和内部级(存储模式)。尽管计算机系统各种各样, 各种计算机中使用的操作系统也不尽相同, 但是绝大多数系统在总的数据库体系结构上仍具有三级结构的特征(如图 1.11 所示)。

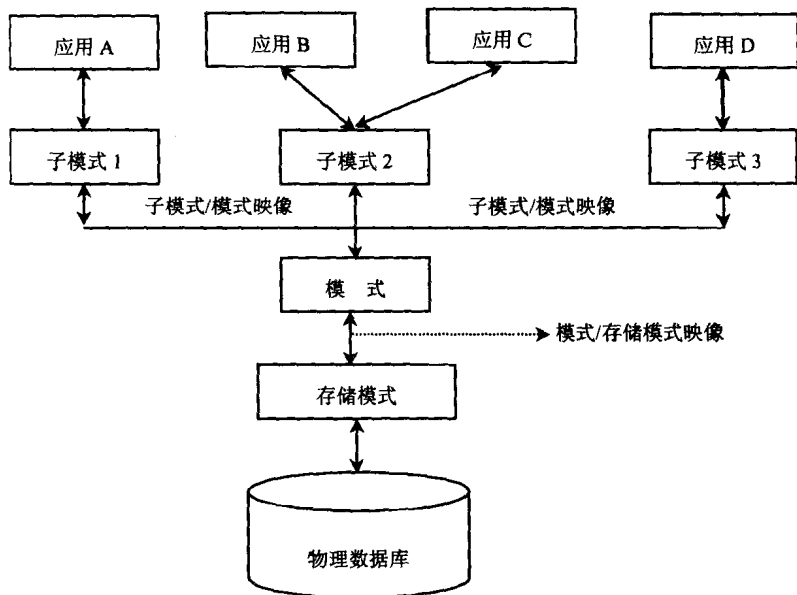


图 1.11 数据库系统的三级模式

从某个角度看到的数据特性称为数据视图。外部级最接近用户, 是单个用户所能看到的数据特性。单个用户使用的数据视图称为外模型。概念级是涉及到所有用户的数据定义, 也就是全局的数据视图, 称之为概念模型, 有时也称数据模型。内部级最接近物理存储设备, 涉及到实际数据存储的方式。物理数据存储的模型称为内模型。这些模型用数据定义语言描述分别得到外模式(又称子模式)、概念模式(又称为模式)和内模式(又称为存储模式)。

1. 外模式

外模式是用户与数据库系统的接口。

单个用户的视图称为外部视图。一个用户往往只用到数据库的一部分。外部视图由若干外部记录类型组成，这些外部记录类型、内部记录类型可能不一样。用户使用数据操纵语言对数据库进行操作，实际上是对外部记录进行操作。

外模式与概念模式的分离，使数据库系统的灵活性大大增加。外模式是从模式导出的，它必须是导出它的模式的逻辑子集。但在外模式里，不仅那些与用户无关的数据可以略去，而且数据项可以按照用户使用的习惯重新命名，记录类型也可以重新组合和命名。应用程序员不必再关心整个数据库的全局逻辑结构，他只与外模式发生直接联系，按照外模式存储和操纵数据。

对于一个数据库系统而言，任何一个应用程序都必须使用而且只能使用一个外模式，才能对数据库中的数据进行操作。一个模式可以支持若干个外模式，但每一个外模式只能属于一个模式。

2. 概念模式(模式)

数据库的全局逻辑视图称为概念视图。它由若干个概念记录类型组成，它是数据库中全部数据逻辑结构的描述，即数据库中所有记录类型的整体描述。

概念模式还要描述记录之间的联系、所允许的操作、数据的一致性、有效验证、安全和其他管理控制方面的要求。数据按外模式的描述提供给用户，按内模式的描述存储在磁盘中。而概念模式提供了一种约束其他两级的相对稳定的中间观点，它使得这两级的任何一级的改变都不受另一级的牵制。

概念模式还必须达到数据独立性。模式描述中必须不涉及到存储结构、访问技术等细节。这样，模式描述中就不会遇到存储字段的表达、记录的物理顺序、索引方式等存储 / 访问细节问题。只有这样，模式才算做到了数据独立，而在模式基础上定义的外模式才能做到数据独立。

3. 内模式

内部视图是数据库结构中最低一级的逻辑表达，它由若干内部记录类型组成。内部记录也称为存储记录。内部视图用 DDL(数据描述语言)描述后得到的是内模式，描述内部视图的 DDL 称为内模式 DDL。内模式要定义所有的内部记录类型，定义一些索引、数据在存储器的安排以及安全性、恢复和其他管理方面的细节。所以内模式是数据在物理存储结构方面的描述。

由于内部记录不涉及物理记录和物理块，因此内部级与物理级是不同的。比内模式更接近于物理存储的那些软件机制是属于操作系统的一部分，例如从磁盘上读一部分数据或写一部分数据到磁盘上等操作。

外模式、模式、内模式都有“源”与“目标”之分。用 DDL 书写的都是源模式，系统不能直接使用，必须由数据管理系统提供编译程序，将源模式编译成二进制的目标码，成为机器可使用的模式，即目标模式，放在目标库(又称为数据字典)中，以供系统调用。

数据库的三级体系结构是数据的三个抽象级别，它把数据的具体组织留给数据库管理系统管理，使用户能抽象地处理数据，而不必关心数据在计算机中的表示和存储。这三级结构之间往往差别很大，为实现这三个抽象级别的转换，数据库管理系统在这三级结构之间提供了两层映像：外模式 / 模式映像、模式 / 内模式映像。

1. 外模式 / 模式映像

这个映像存在于外部级和概念级之间，它定义了模式与物理模式的映射关系。当整个系统要求改变模式时，可以改变映射关系而保持子模式不变。这种用户数据独立于全局的逻辑数据的特征叫逻辑数据独立性。

2. 模式 / 内模式映像

这个映像存在于概念级和内部级之间，它定义了模式与物理的映射关系。当为了某种需要，例如提高某文件的存取效率时，可以改变物理模式，同时改变模式 / 内模式映射而保持模式和子模式不变。这种全局的逻辑数据独立于物理数据的特征叫物理数据独立性。

有了上述两种数据独立性，数据库系统就把用户数据和物理数据完全分开了。使得用户摆脱了繁琐的物理存储细节。由于用户程序不依赖物理数据，这就减少了应用程序的维护开销。

1.4.2 数据库系统的组成

数据库系统是一个复杂的系统。这里要说明的是数据库系统不单是指数据库，也不单是指管理系统，而是指包括计算机系统在内的整个系统。通常数据库系统由以下几部分组成：

- ① 系统硬件
- ② 系统软件
- ③ 数据库管理系统
- ④ 数据库管理员
- ⑤ 数据库

1. 系统硬件资源

数据库系统的硬件资源包括安装、建立和运行数据库所必须的计算机及其外部设备。

2. 系统软件资源

数据库系统的软件资源通常有：

- (1) 操作系统
- (2) 宿主语言，如 C、FORTRAN、COBOL 和 PL / 1 等语言
- (3) 数据库管理系统
- (4) 实用程序

软件资源中，核心部分是数据库管理系统(Data Base Management System,简记为 DBMS)。数据库管理系统包括各类语言(如模式描述语言、子模式描述语言、数据操纵语言、物理数据描述语言和查询语言等)、实用程序(装配程序、日志程序和统计分析程序等)和数据库运行控制程序(数据库管理程序、并发控制程序、存取控制程序、数据更新程序、完整性、安全性保护程序和有效性、合法性检验程序)等。

3. 数据库管理系统

数据库管理系统是数据库系统中对数据进行管理的软件系统，它是数据库系统的核心部分，数据库系统的查询、更新以及各种操作都是通过 DBMS 完成的。

数据库管理系统是基于某种数据模型的，所以可以把它看作某种数据模型在某种计算机系统上的实现。在不同的计算机系统中采用不同的数据模型的数据库管理系统的用户接口、系统功能也是不同的。

DBMS 的主要功能