数据库与程序设计

马涛
李蕴
王海燕
谭业武

主编

山东大学出版社

数据库与程序设计

主　审　郝兴伟　刘明军
主　编　马　涛　李　蕴
　　王海燕　谭业武
副主编　唐好魁　邢静波

山东大学出版社

图书在版编目(CIP)数据

数据库与程序设计/马涛等主编．—济南：山东大学出版社，2005.8
ISBN 7-5607-3067-1

Ⅰ．数…　Ⅱ．马…　Ⅲ．①数据库-基本知识②程序设计-基本知识　Ⅳ．TP31

中国版本图书馆 CIP 数据核字(2005)第 096635 号

内容简介

本书以 Visual FoxPro 6.0 数据库管理系统为平台，循序渐进地介绍数据库的基本知识、结构化程序设计方法、面向对象程序设计方法以及在此基础上的应用系统开发方法，力求学习者从中掌握这些知识点，能够根据自身的需要开发出实用的小型数据库应用系统。本书分上下两篇共 11 章，上篇主要介绍数据库的基础知识，包括数据库基础、Visual FoxPro 概述、数据库及数据表的操作、查询与视图、SQL 关系数据库查询语言；下篇主要介绍程序设计方法及系统开发，包括结构化程序设计方法、面向对象程序设计方法、表单及其控件设计、菜单及工具栏设计、系统开发实例。

本书内容全面、深入浅出、例题丰富，适合作为高等院校或各类大专院校的学生学习数据库开发和程序设计的教材，同时也适合具有同等文化程序的读者自学或参考之用。

山东大学出版社出版发行
(山东省济南市山大南路 27 号　邮政编码：250100)
山　东　省　新　华　书　店　经　销
安丘市意中印务有限责任公司印刷
787×1092 毫米　1/16　18.5 印张　473 千字
2005 年 8 月第 1 版　2005 年 8 月第 1 次印刷
定价：28.00 元

前　言

随着计算机技术的飞速发展，人类已经进入了信息时代，从一开始的学习打字，编辑文本，到现在的电子商务，人们对于计算机的应用越来越深入。因此，仅仅了解计算机的一些基础知识已远远不能适应当今社会的需要，人们应该了解计算机较深层次的技术，如计算机如何管理数据、处理信息，如何执行各种命令。迄今为止，数据库技术仍是信息处理、数据管理最有效的方法之一，而关系数据库是应用最广泛的数据库。目前，国内使用的小型关系数据库管理系统中，Visual FoxPro 系统占据了重要的地位，该系统系列历史悠久（从 1980 年的 dBASEⅡ到现在的 Visual FoxPro 6.0）、应用范围广泛、功能全面，是学习、应用关系型数据库管理系统以及程序设计方法的较为理想的系统软件。

本书以 Visual FoxPro 6.0 数据库管理系统为平台，循序渐进地介绍数据库的基本知识、结构化程序设计方法、面向对象程序设计方法以及在此基础上的应用系统开发方法，力求学习者从中掌握这些知识点，能够根据自身的需要开发出实用的小型数据库应用系统。全书分上下两篇，共 11 章。第一章主要介绍数据库系统的基本概念，包括数据库系统的组成、数据模型、数据库设计与管理等内容；第二章的内容主要包括 Visual FoxPro 6.0 的基本概念、命令格式和项目管理器；第三章介绍数据库、数据表的操作、索引、关联等数据库的基本知识点；第四章介绍查询与视图；第五章介绍关系数据库标准语言 SQL；第六章从结构化程序设计方法到面向对象程序设计方法都作了详细的介绍，为后面的编程奠定了理论基础；第七章、第八章着重介绍了基于面向对象程序设计方法的表单及其控件的设计和应用，以及类的创建和应用；第九章介绍了应用程序中不可缺少的菜单和工具栏的设计；第十章介绍了非常实用的报表设计；第十一章通过一个完整的数据库应用系统开发实例，介绍了数据库应用系统开发的一般方法。

本书由郝兴伟、刘明军负责统稿。

具体分工：第一、二章由郝兴伟编写；第三章由王海燕、谭业武编写；第四、五、十章由唐好魁、刑静波编写；第六、十一章由马涛编写；第七、八、九章由李蕴编写。

书中例题均在 VFP6.0 中文版环境下运行通过。但鉴于本书覆盖面广、知识点多、软件版本升级很快，加上作者水平有限，书中不当之处在所难免，敬请广大读者批评指正。

作　者

2005.7

目 录

第一章　数据库基础

数据管理是计算机应用领域中一类重要的技术和研究课题。数据管理是指对各种形式的数据进行收集、储存、加工和传播的一系列活动的总和，其目的是借助于计算机从大量的原始数据中抽取、推导并组织出对人们有价值的信息，作为行动和决策的依据。

从20世纪50年代末，数据管理的研究极大地促进了计算机应用向各行各业的渗透，管理信息系统、办公信息系统、银行信息系统、民航订票系统、情报检索系统等，都属于这一类的应用。今后它仍将是计算机科学技术领域中一门重要的技术和研究课题。数据库技术正是这类应用的直接结果。

1.1　数据库技术的产生与发展

数据库技术是应数据管理任务的需要而产生的，是随着数据管理功能需求不断增加而发展的。数据管理经历了人工管理、文件系统和数据库系统三个阶段。

1. 人工管理阶段

该阶段处在20世纪50年代中期以前，当时计算机主要用于科学计算，其数据管理呈以下特点：

(1)数据不保存：当时计算机主要用于科学计算，一般不需要将数据长期保存，当要计算某一题目时，将需要的数据输入，用完消除。

(2)数据没有相应的软件系统管理：数据由应用程序管理。应用程序既要设计数据的逻辑结构，还要设计物理结构，包括存储结构、存取方法以及输入方式等。

(3)数据不共享：一般一组数据附属于一个应用程序，当多个应用程序涉及某些相同数据时，程序与程序之间会有大量的冗余数据。

(4)数据不具独立性。数据的逻辑结构或物理结构发生变化后，需对应用程序作相应修改。

2. 文件系统阶段

20世纪50年代后期到60年代中期，随着计算机硬件的发展，操作系统中有了专门进行数据管理的软件，称为文件系统。把数据组织成文件的形式可以随机进行查询、增删改等处理，并且数据可以长期保存，实现了以文件为单位的数据共享。

文件系统也明显地存在以下缺点：

(1)编程不方便：操作系统只提供低级文件操作命令，对数据的查询、修改、排序等操作都必须由编程解决。

(2)数据冗余量大：为了兼顾各种应用程序的要求，在设计文件系统时，不得不按最大的需求定义数据格式，造成数据冗余。数据冗余不仅浪费存储空间，且会带来数据的不一致性。

(3)数据独立性差：即文件与应用程序间缺乏独立性，文件结构的每一修改都将导致应用

程序的修改。

(4)不支持并发访问:文件系统一般不支持多个应用程序对同一数据文件的并发访问,当一个程序正查询某一些数据,而另一程序正在修改数据时,有可能不一致、甚至错误。

(5)数据缺少统一管理:在数据的结构、编码、表示格式、命名以及输出格式等方面不容易做到规范化、标准化,在数据的安全和保密方面也难以采取有效措施。

3. 数据库系统阶段

20 世纪 60 年代以来,数据管理针对文件系统的不足,逐步发展成为以统一管理和共享数据为主要特征的数据库系统。在数据库系统中,数据不再仅仅服务于某个程序或用户,而成为若干程序或用户的共享资源,由数据库管理系统 DBMS(Data Base Management System)统一管理与控制。在数据库中,由 DBMS 完成诸如打开、关闭、读、写等文件的低级操作,应用程序不必关心数据存储和其他实现的细节,可以在更高的抽象级别上访问数据。文件结构由 DBMS 修改,从而减少应用程序的维护工作量,提高数据的独立性。使用 DBMS 统一管理数据,可以合理组织数据,减少冗余,可以更好地贯彻规范化和标准化,有利于数据的转换和更大范围内的共享。DBMS 具有适合于不同用户的多种界面,保证并发访问时的数据一致性,增进数据安全性和在故障情况下数据的恢复等功能。同时,它也保证了数据在语义上一致性的完整性约束。

4. 数据仓库

"数据仓库"技术是目前数据处理中发展十分迅速的一个分支。所谓数据仓库,就是一种长期数据存储,这些数据来自于多个异种数据源。通过数据仓库提供的联机分析处理(OLAP)工具,实现各种粒度的多维数据分析,以便向管理决策提供支持。数据仓库系统允许将各种应用系统集成在一起,为统一的历史数据分析提供坚实的平台,对信息处理进行支持。数据仓库具有以下关键特征:①面向主题的:围绕某一主题建模和分析;②集成的:将多个异种数据源以及事务记录集成在一起;③时变的:数据存储从历史的角度提供信息;④非易失的:总是物理地分离存放数据。目前,数据仓库已经成为数据分析和联机分析处理日趋重要的平台。

数据库系统和数据仓库系统的区别主要有以下几点:

(1)面向的用户不同:数据库系统面向的是企业的低层人员,用于日常数据的分析和处理;数据仓库系统面向的是企业的决策人员,提供决策支持。

(2)数据内容不同:数据库系统中存储和管理的是当前的数据;数据仓库存储的是长期积累的历史数据。

(3)数据来源不同:数据库的数据一般来源于同种数据源,而数据仓库的数据可以来源于多个异种数据源。

(4)数据的操作不同:数据库系统提供了执行联机事务和查询处理(On-line Transaction Processing,OLTP)系统。数据库系统主要提供了数据分析和决策支持(On-line Analytical Processing,OLAP)系统,实现数据挖掘和知识发现。

1.2 数据库系统的组成

数据库系统是由支持数据库运行的硬件、数据库、数据库管理系统、应用软件、数据库管理员和用户组成。如图 1-1 所示。

1. 数据库

数据库是长期存储在计算机存储介质上，有一定组织形式、可共享的数据集合。针对应用所需要的收集并抽取大量数据，经过加工处理后保存在数据库中。数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度、较高的数据独立性和易扩展性，并为各种用户共享。

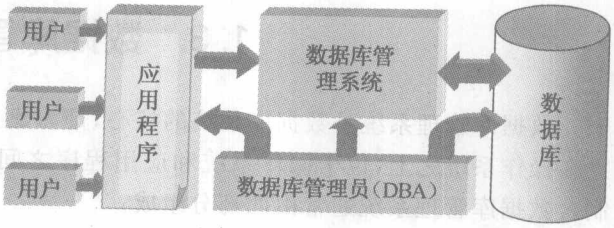


图 1-1　数据库系统

2. 支持数据库运行的硬件

硬件是数据库赖以存在的物理设备，包括 CPU、存储器和其他外部设备等。数据库系统要求要有较大的内存，用以存放系统程序、应用程序和开辟用户工作区及系统缓冲区；而对外部存储器更有特殊要求，一般应配备高速度、大容量的直接存取存储设备(磁盘、光盘等)。

3. 数据库管理系统

数据库管理系统(Database Management System，DBMS)是介于用户和操作系统之间的一层数据管理软件。它由计算机软件生产厂家按商品软件出版。如 ORACLE 公司的 Oracle 系统、SYBASE 公司的 Sybase 系统、Microsoft 公司的 SQL Server 等大型数据管理系统以及 Access，VFP 等小型数据管理系统。

数据库管理系统为数据库的建立、运行和维护提供了统一的管理和控制。用户通过数据库管理系统定义数据和操纵数据，由它保证数据的安全性、完整性、开发使用及发生故障后的系统恢复。

4. 应用程序

程序是对数据的管理，在几乎所有的事务处理中都用到了数据库来管理数据。一个数据库应用可分为客户端应用程序和服务端应用程序两类。服务端应用程序运行在数据库服务器上，是真正存储和操纵数据的，它接受用户程序的请求，对数据进行不同的操作。客户端应用程序运行在客户端计算机上，实现用户的业务逻辑，通过客户端应用程序界面，用户可以发出不同的请求给服务器端，由服务端程序完成各种各样的操作。一般情况下，客户端和服务器端的程序通过标准 SQL 语言通信。

客户端应用程序根据使用者的不同，可以分成两类：一类是供数据库管理员使用，提供强大的图形界面和命令以便管理员最大限度的维护数据库的运转；另一类为程序开发人员使用，提供一整套完整的用户接口界面让开发人员通过程序实现操纵数据的目的，这些程序最终将提交给用户使用，即通常所讲的应用程序。

数据库应用程序主要完成用户的业务逻辑，被安装在用户的计算机上。应用程序和数据库管理系统一起完成用户的业务处理。在这个应用中，数据库管理系统负责数据的管理，提供数据共享功能，因此多个应用程序可以同时使用同一个数据库。应用程序使用数据库是通过 DBMS 实现的。

5. 数据库管理员(Data Base Administrator，DBA)

大型数据库通常由专业人员设计，还要配上专职数据库管理员(DBA)。DBA 的职责是维护和管理数据库，使之始终处于最佳状态。

1.3 数据库管理系统

数据库管理系统是数据库系统的核心，是为建立、使用和维护数据库而配置的软件，它建立在操作系统之上，位于操作系统和应用程序之间，负责对数据库中数据进行统一管理和控制。数据库管理系统通常由四部分组成。

1.3.1 数据定义语言

DBMS 提供的数据定义语言(Data Definition Language，DDL)用以定义数据库的概念模式、存储模式和外模式以及各模式间的映射和完整性约束。DBMS 具有模式翻译程序，负责将 DDL 定义的外模式、概念模式和存储模式翻译成相应的内部表示。翻译后的内部表示是数据库的框架，称为目标模式，存放在数据目录中，作为 DBMS 存取和管理数据的基本依据。根据这些定义，DBMS 可以从物理记录导出全局逻辑记录，又从全局逻辑记录导出用户所需的局部逻辑记录。

1.3.2 数据操纵语言

DBMS 提供的数据操纵语言(Data Manipulation Language，DML)用以数据库的检索、输入、修改、删除等的基本操作。DML 分为宿主型和自主型两类：宿主型 DML 本身不能独立使用，必须将其命令嵌入主语言中，例如，嵌入 C，COBOL，FORTRAN 等高级语言中。自主型 DML 又称为自含型 DML，它们一般是交互式命令语言，语法简单，可以独立使用。现有的微机数据库管理系统基本是自主型 DML，如 Informix，FoxPro 等。

1.3.3 数据控制语言

DBMS 一般也提供数据控制语言(Data Control Language，DCL)，以便让用户根据需要控制和管理数据库系统。

DBMS 提供了数据库运行过程中的控制管理程序，包括系统初始化程序、文件读写与维护程序、存取路径管理程序、缓冲区管理程序、安全性控制程序、并发控制程序、事务管理程序及运行日志管理程序等。它们在数据库运行过程中监视数据库的操作，管理数据库资源，处理多用户的并发操作。

1.3.4 实用程序

DBMS 提供了一些实用程序，包括数据初始装入程序、数据转储程序、数据库恢复程序、性能测试程序、数据库再组织程序、数据转换程序、通信程序等。

一个设计优良的 DBMS，应该具有友好的用户界面、比较完备的操作功能、较高的运行效率、清晰的系统结构和良好的开放性特点。所谓开放性，是指数据库设计人员能够方便地将自己开发的工具模块加入到 DBMS 中，这些新加进模块能够与 DBMS 紧密结合，一起运行。DBMS 开放性特点为建立规模较大的软件开发环境或应用系统提供了极大的方便，也使 DBMS 更具适应性、灵活性、可扩充性。

1.4　数据模型

模型是现实世界特征的模拟和抽象。数据模型就是用来抽象、表示和处理现实世界的数据和信息的工具。通俗地讲数据模型就是现实世界的模拟。

数据库系统均是基于某种数据模型的，数据模型是数据库系统的核心和基础。因此，了解数据模型的概念是学习数据库的基础。数据模型由三个基本的要素组成：数据的结构、数据的操作、数据的约束条件。其中数据的结构最具代表性。

从文件系统中，可以看到一个简单数据模型的基本框架。其数据的结构包括了文件、记录、字段等概念；其数据的操作包括打开、关闭、读、写等文件操作；其约束条件表现在字段有类型和长度的定义。当然这个简单的数据模型没有描述数据间的联系。

数据模型从概念上描述了系统的静态特征、动态特征和约束条件。静态特征是指数据的基本结构、数据间的联系；动态特征是指定义在数据上的操作；约束条件主要是完整性约束。

1.4.1　数据模型的概念

数据库是某个企业、组织或部门所涉及的数据的集合，它不仅要反映数据本身的内容，而且要反映数据之间的联系。由于计算机不能直接处理现实世界中的具体事物，所以人们必须事先把具体事物转换成计算机能够处理的数据。在数据库中用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息。

数据模型应满足三方面要求：一是能比较真实地模拟现实世界；二是容易为人所理解；三是便于在计算机上实现。一种数据模型要很好地满足这些方面的要求是很困难的。在数据库系统中针对不同的使用对象和应用目的，应采用不同的数据模型。

不同的数据模型实际上是提供给我们模型化数据和信息的不同工具。根据模型应用的不同目的，可以将这些模型划分为两类，它们分属于两个不同的层次。

第一类模型是概念模型，也称信息模型，它是按用户的观点来对数据和信息建模，主要用于数据库设计。另一类模型是数据模型，主要包括网状模型、层次模型和关系模型，它是按计算机系统的观点对数据建模，主要用于 DBMS 的实现。

数据模型是数据库系统的核心和基础。各种机器上实现的 DBMS 软件都是基于某种数据模型的。

为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统，不是某一个 DBMS 支持的数据模型，而是概念上的模型；然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型，这一过程如图 1-2 所示。

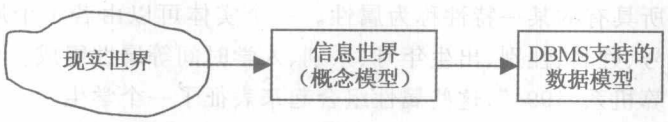


图 1-2　现实世界中客观对象的抽象过程

1.4.2 数据模型的组成要素

一般地讲，数据模型是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特性、动态特性和完整性约束条件。因此数据模型通常由数据结构、数据操作和完整性约束三部分组成。

1. 数据结构

数据结构是所研究的对象类型的集合。这些对象是数据库的组成成分，它们包括两类，一类是与数据类型、内容、性质有关的对象；另一类是与数据之间联系有关的对象。

数据结构是刻画一个数据模型性质最重要的方面，它是对系统静态特性的描述。在数据库系统中，人们通常按照其数据结构的类型来命名数据模型。例如层次结构、网状结构和关系结构的数据模型分别命名为层次模型、网状模型和关系模型。

2. 数据操作

数据操作是指对数据库中各种对象的实例允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有检索和更新(包括插入、删除、修改)两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则以及实现操作的语言。数据操作是对系统动态特性的描述。

3. 数据的约束条件

数据的约束条件是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和依存规则，用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容。

数据模型应该反映和规定本数据模型必须遵守的基本的通用的、完整性约束条件。此外，数据模型还应该提供定义完整性约束条件的机制，以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。例如，年龄不能取负值，学生累计成绩不得有三门以上不及格等。

1.4.3 概念模型

概念模型是现实世界到机器世界的一个中间层次，它用于信息世界的建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的有力工具，也是数据库设计人员和用户之间进行交流的语言，因此概念模型一方面应该具有较强的语义表达能力，能够方便、直接地表达应用中的各种语义知识，另一方面它还应该简单、清晰、易于用户理解。

1. 信息世界中的基本概念

(1)实体：客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物，也可以是抽象的概念或联系，例如一个教师、一个学生、一门课、学生的一次选课、教师与系的工作关系等都是实体。

(2)属性：实体所具有的某一特性称为属性。一个实体可以由若干个属性来刻画。例如，学生实体可以由学号、姓名、性别、出生年月、系别、入学时间等属性组成。例如："94002268，张三，男，1976/07，计算机系，1994"，这些属性组合起来表征了一个学生。

(3)实体标识符：能够唯一标识一个实体的属性集称为实体标识符，也称为关键码，简称码或键。例如学号是学生实体的码。

(4)域：属性的取值范围称为该属性的域。例如，学号的域为8位整数，姓名的域为字符串集合，性别的域为{男，女}。

(5)实体型：具有相同属性的实体必然具有共同的特征和性质。用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。例如，学生(学号，姓名，性别，出生年份，系别，入学时间)就是一个实体型。

(6)实体集：同型实体的集合称为实体集。例如，全体学生就是一个实体集。

(7)联系：在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中反映为实体(型)内部的联系和实体(型)之间的联系。实体内部的联系通常指的是组成实体的各属性之间的联系。实体之间的联系通常是指不同实体集之间的联系。

两个实体型之间的联系可以分为三类：

(1)一对一联系：如果对于实体集 A 中的每一个实体，实体集 B 中至多有一个(也可以没有)实体与之联系；反之亦然，则称实体集 A 与实体集 B 具有一对一联系，记为 1∶1。

例如，学校里面，一个班级只有一个班长，而一个班长只在一个班中任职，则班级与班长之间具有一对一联系。

(2)一对多联系：如果对于实体集 A 中的每一个实体，实体集 B 中有 N 个实体(N≥0)与之联系；反之，对于实体集 B 中的每一个实体，实体集 A 中至多只有一个实体与之联系，则称实体集 A 与实体集 B 有一对多联系，记为 1∶N。

例如，一个班级中有若干名学生，而每个学生只在一个班级中学习，则班级与学生之间具有一对多联系。

(3)多对多联系：如果对于实体集 A 中的每一个实体，实体集 B 中有 N 个实体(N≥0)与之联系；反之，对于实体集 B 中的每一个实体，实体集 A 中有 M 个实体(M≥0)与之联系，则称实体集 A 与实体集 B 具有多对多联系，记为 M∶N。

例如，一门课程同时有若干个学生选修，而一个学生可以同时选修多门课程，则课程与学生之间具有多对多联系。

实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。

可以用图形来表示两个实体型之间的这三类联系，如图 1-3 所示。

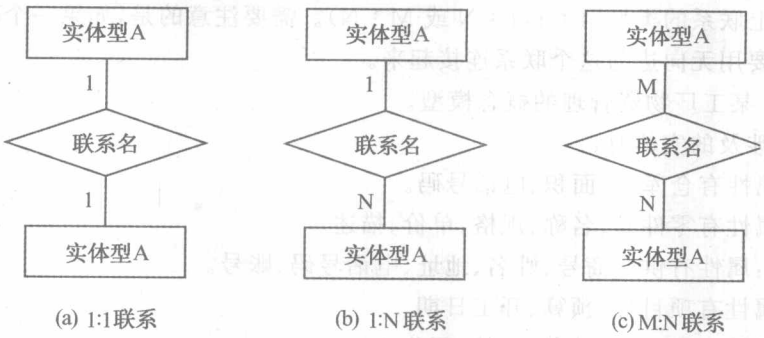


图 1-3　两个实体型之间的三类联系

一般地，两个以上的实体型之间也存在着一对一、一对多、多对多联系。

若实体集 $E_1, E_2, \cdots, E_n$ 存在联系，对于实体集 E_j $(1 \leqslant j \leqslant N$，且 $j \neq i)$ 中给定的实体，最多只和 E_i 中的一个实体相联系，则说 E_i 与 E_j 之间的联系是一对多的。

例如，对于课程、教师与参考书三个实体型，如果一门课程可以有若干个教师讲授，使用若干本参考书，而每一个教师只讲授一门课程，每一本参考书只供一门课程使用，则课程与教师、参考书之间的联系是一对多的，如图 1-4(a)所示。

又如，有三个实体型：供应商、项目、零件，一个供应商可以供给多个项目多种零件，而每个项目可以使用多个供应商供应的零件，每种零件可由不同供应商供给，由此看出供应商、项目、零件三者之间是多对多的联系，如图 1-4(b)所示。

同一个实体集内的各实体之间也可以存在一对一、一对多、多对多的联系。例如，职工实体集内部具有领导与被领导的联系，即：某一职工“领导”若干名职工，而一个职工仅被另外一个职工直接领导，因此这是一对多的联系。一个实体型之间一对多联系示例如图 1-5 所示。

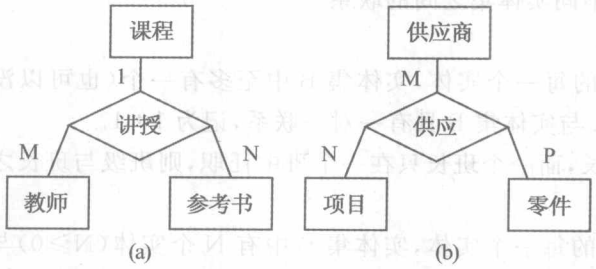


图 1-4　三个实体型之间的联系示例图

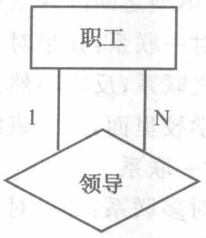


图 1-5　一对多联系示例

2. 概念模型的表示方法

概念模型是对信息世界建模，所以概念模型应该能够方便、准确地表示出上述信息世界中的常用概念。概念模型的表示方法很多，其中最为著名和常用的是 P. P. S. Chen 于 1976 年提出的实体－联系方法(Entity-Relationship Approach)。该方法用 E-R 图来描述现实世界的概念模型，E-R 方法也称为 E-R 模型。

E-R 图提供了表示实体型、属性和联系的方法：

- 实体型：用矩形表示，矩形框内写明实体名。
- 属性：用椭圆形表示，并用无向边将其与相应的实体连接起来。
- 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型(1∶1，1∶N 或 M∶N)。需要注意的是，如果一个联系具有属性，则这些属性也要用无向边与这个联系连接起来。

【例 1-1】 某工厂物资管理的概念模型。

物资管理涉及的实体有：

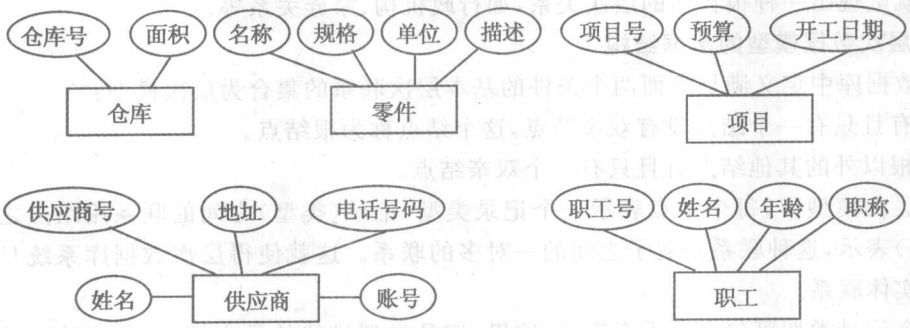
- 仓库：属性有仓库号、面积、电话号码。
- 零件：属性有零件号、名称、规格、单价、描述。
- 供应商：属性有供应商号、姓名、地址、电话号码、账号。
- 项目：属性有项目号、预算、开工日期。
- 职工：属性有职工号、姓名、年龄、职称。

这些实体之间的联系描述如下：

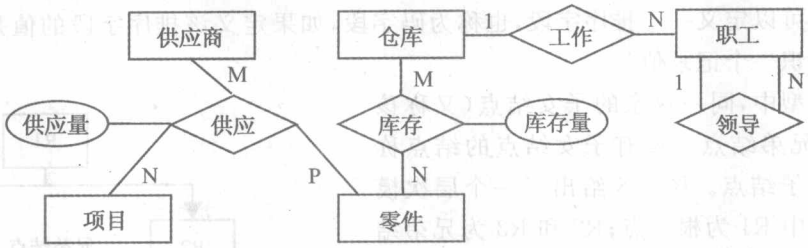
- 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中，因此仓库和零件具有多对多的联系。用库存量来表示某种零件在某个仓库中的数量。
- 一个仓库有多个职工当仓库保管员，一个职工只能在一个仓库工作，因此仓库和职工之间是一对多的联系。
- 职工之间具有领导和被领导的关系。即仓库主任领导若干保管员，因此职工实体集中具有一对多的联系。

● 供应商、项目和零件三者之间具有多对多的联系。即一个供应商可以供给若干项目多种零件，每个项目可以使用不同供应商供应的零件，每种零件可由不同供应商供给。

该工厂的物资管理 E-R 图如图 1-6 所示。



(a) 实体及其属性图



(b) 实体及其联系图

图 1-6　工厂物资管理 E-R 图

【例 1-2】 某信用卡账户管理系统概念模型。

某信用卡系统主要包含用户和信用卡两个实体，可以建立下面简单的 E-R 模型，如图 1-7 所示。

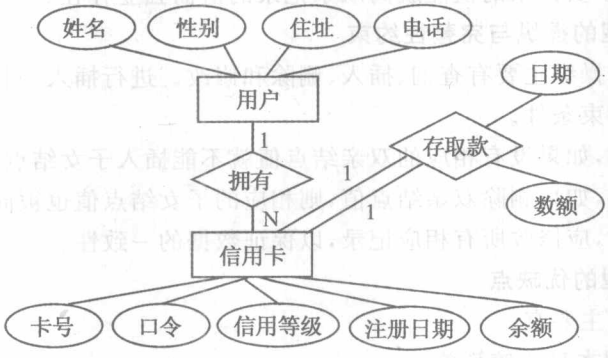


图 1-7　某信用卡账户管理系统 E-R 图

1.4.4　层次模型

层次模型是数据库系统中最早出现的数据模型，层次数据库系统采用层次模型作为数据的组织方式。层次数据库系统的典型代表是 IBM 公司的 IMS(Information Management Sys-

tem)数据库管理系统，这是 1968 年 IBM 公司推出的第一个大型的商用数据库管理系统，曾经得到广泛的使用。

层次模型用树形结构来表示各类实体以及实体间的联系。现实世界中许多实体之间的联系本来就呈现出一种很自然的层次关系，如行政机构、家族关系等。

1. 层次数据模型的数据结构

在数据库中定义满足下面两个条件的基本层次联系的集合为层次模型：

- 有且只有一个结点没有双亲结点，这个结点称为根结点。
- 根以外的其他结点有且只有一个双亲结点。

在层次模型中，每个结点表示一个记录类型，记录(类型)之间的联系用结点之间的连线(有向边)表示，这种联系是父子之间的一对多的联系。这就使得层次数据库系统只能处理一对多的实体联系。

每个记录类型可包含若干个字段，这里，记录类型描述的是实体，字段描述的是实体的属性。各个记录类型及其字段都必须命名。各个记录类型、同一记录类型中各个字段不能同名。每个记录类型可以定义一个排序字段，也称为码字段，如果定义该排序字段的值是唯一的，则它能唯一地标识一个记录值。

在层次模型中，同一双亲的子女结点(又称孩子结点)称为兄弟结点。没有子女结点的结点称为叶结点或叶子结点。图 1-8 给出了一个层次模型的例子。其中 R1 为根结点；R2 和 R3 为兄弟结点，是 R1 的子女结点；R4 和 R5 为兄弟结点，是 R2 的子女结点；R3，R4 和 R5 都为叶结点。

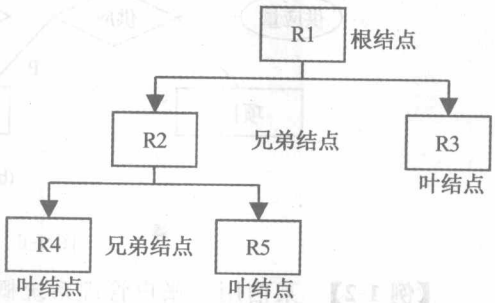


图 1-8 层次模型示例

从图中可以看出层次模型像一棵倒立的树，结点的双亲是唯一的。

层次模型的一个基本的特点是，任何一个给定的记录值只有按其路径查看时，才能显出它的全部意义，没有一个子女记录的值能脱离双亲记录的值而独立存在。

2. 层次数据模型的操纵与完整性约束

层次模型的数据操纵主要有查询、插入、删除和修改。进行插入、删除、修改操作时要满足层次模型的完整性约束条件。

进行插入操作时，如果没有相应的双亲结点值就不能插入子女结点值。

进行删除操作时，如果删除双亲结点值，则相应的子女结点值也被同时删除。

进行修改操作时，应修改所有相应记录，以保证数据的一致性。

3. 层次数据模型的优缺点

层次模型的优点主要有：

- 层次数据模型本身比较简单。
- 对于实体间联系是固定的，且对于预先定义好的应用系统，采用层次模型来实现，其性能优于关系模型，不低于网状模型。
- 层次数据模型提供了良好的完整性支持。

可见用层次模型对具有一对多层次关系的部门描述非常自然、直观，容易理解。这是层次数据库的突出优点。

层次模型的缺点主要有：

- 现实世界中很多联系是非层次性的，如多对多联系、一个结点具有多个双亲等，层次模型表示这类联系的方法很笨拙，只能通过引入冗余数据或创建非自然的数据组织（引入虚拟结点）来解决。
- 对插入和删除操作的限制比较多。
- 查询子女结点必须通过双亲结点。
- 由于结构严密，层次命令趋于程序化。

1.4.5　网状模型

在现实世界中事物之间的联系更多的是非层次关系的，用层次模型表示非树形结构是很不直接的，网状模型则可以克服这一弊病。

网状数据库系统采用网状模型作为数据的组织方式。网状数据模型的典型代表是DBTG系统，亦称CODASYL系统。这是20世纪70年代数据系统语言研究会CODASYL（Conference On Data System Language）下属的数据库任务组（Data Base Task Group 简称 DBTG）提出的一个系统方案。DBTG系统虽然不是实际的软件系统，但是它提出的基本概念、方法和技术具有普遍意义。它对于网状数据库系统的研制和发展起了重大的影响。后来不少的系统都采用DBTG模型或者简化的DBTG模型。

1. 网状数据模型的数据结构

在数据库中，把满足以下两个条件的基本层次联系集合称为网状模型：

- 允许一个以上的结点无双亲。
- 一个结点可以有多于一个的双亲。

网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，允许多个结点没有双亲结点，允许结点有多个双亲结点，此外它还允许两个结点之间有多种联系。因此网状模型可以更直接地去描述现实世界。而层次模型实际上是网状模型的一个特例。

与层次模型一样，网状模型中每个结点表示一个记录类型（实体），每个记录类型可包含若干个字段（实体的属性），结点间的连线表示记录类型（实体）之间一对多的父子联系。

从定义可以看出，层次模型中子女结点与双亲结点的联系是唯一的，而在网状模型中这种联系可以不唯一。因此，要为每个联系命名，并指出与该联系有关的双亲记录和子女记录。例如图1-9(a)中R3有两个双亲记录R1和R2，因此把R1与R3之间的联系命名为L1，R2与R3之间的联系命名为L2。其他例子的情况类似。

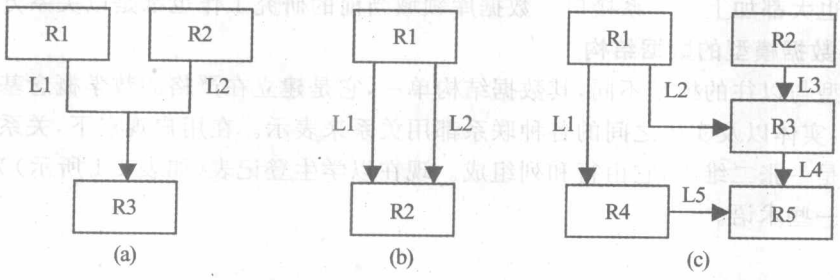


图1-9　网状模型示例

实际的商品化网状数据库系统对网状数据结构都有不同的限制，例如HP公司的IM-