

# 第 1 章 绪 论

20 世纪 60 年代末，数据库技术初露头角，随即得到迅速发展，成为数据处理的公用支撑技术，是计算机科学的重要分支，也是信息系统的核心和基础。它的出现极大地促进了计算机应用向教育、科研、金融、医疗等行业的渗透。同时，数据库的建设规模、数据库信息量的大小和使用频度已成为衡量一个国家信息化程度的重要标志。

## 1.1 数据库系统概述

### 1.1.1 数据库的几个基本概念

#### 1. 数据 (Data)

数据是数据库中存储的基本对象，是描述事物的符号记录。数据有多种表现形式，包括数字、文字、图像、声音、学生的档案记录、货物的运输情况等。为了方便计算机存储和处理日常生活中的事物，通常抽象出事物的特征并将其组成一个记录来描述。例如，在教师档案中，用姓名、性别、出生年份、所在系别、工作时间描述一个教师的基本情况，用记录形式表示为：(宋国，男，1998，信息管理系，2016)。这里的教师记录就是数据。结合其含义就能得到该教师的个人情况：宋国是一个教师，性别男，1998 年出生，2016 年进入信息管理系。如果不了解其语义，则无法理解该记录含义。可见，数据的形式还不能完全表达其内容，需要对数据进行解释。所以数据和关于数据的解释是不可分的，数据的解释是指对数据含义的说明，数据的含义称为数据的语义，数据与其语义是不可分的。

#### 2. 数据库 (DataBase, DB)

数据库是长期存储在计算机存储设备上的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余度、较高的数据独立性和易扩充性，并可共享。

#### 3. 数据库管理系统 (DataBase Management System, DBMS)

数据库管理系统是位于用户与操作系统之间的一层数据管理软件，研究如何科学地组织和存储数据、如何高效地获取和维护数据。它是数据库系统的一个重要组成部分，主要功能如下。

##### 1) 数据定义功能

DBMS 提供数据定义语言 (Data Definition Language, DDL)，用户通过它可以方便地对数据库中的数据对象进行定义。

##### 2) 数据操纵功能

DBMS 提供数据操纵定义语言 (Data Manipulation Language, DML)，用户通过它实现对数据的基本操作，如查询、插入、删除和修改等。

### 3) 数据库的运行管理

数据库在建立、运行和维护时由数据库管理系统统一管理、统一控制，以保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

### 4) 数据库的建立和维护功能

数据库的建立和维护功能包括数据库初始数据的输入、转换功能，数据库的转储、恢复功能，数据库的重组、重构造功能和性能监视、分析功能等。

## 4. 数据库系统(DataBase System, DBS)

数据库系统是指在计算机系统中引入数据库后的系统，一般由数据库、数据库管理系统(及其开发工具)、应用系统、数据库管理员(DataBase Administrator, DBA)和用户构成。数据库系统如图 1-1 所示。数据库在计算机系统的位置如图 1-2 所示。

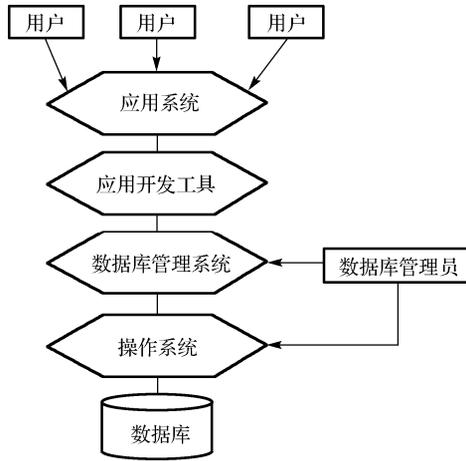


图 1-1 数据库系统

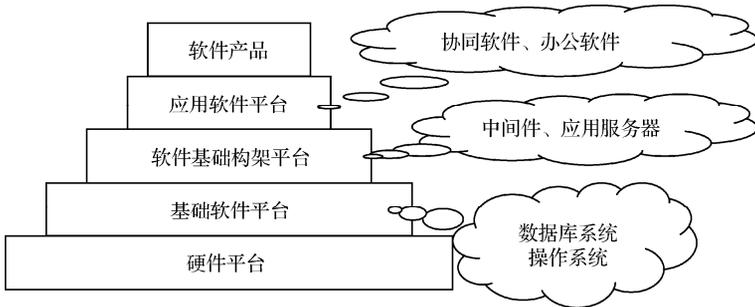


图 1-2 数据库在计算机系统的位置

### 1.1.2 数据管理技术的发展历史

数据库技术是应数据管理任务的需要而产生的。数据处理是指对各种数据进行收集、存储、加工和传播的一系列活动的总和。数据管理则是指对数据进行分类、组织、编码、存储、检索和维护，它是数据处理的中心问题。数据管理技术经历了人工管理(20 世纪 40 年代中—50 年代中)、文件系统(20 世纪 50 年代末—60 年代中)、数据库系统(20 世纪 60 年代末—现在)三个阶段。这三个阶段的特点及其比较如表 1-1 所示。

表 1-1 数据管理三个阶段比较表

		人工管理阶段	文件系统阶段	数据库系统阶段
背景	应用背景	科学计算	科学计算、管理	大规模管理
	硬件背景	无直接存取存储设备	磁盘的、磁鼓	大容量磁盘
	软件背景	没有操作系统	有文件系统	有数据库管理系统
	处理方式	批处理	联机实时处理、批处理	联机实时处理、分布处理、批处理
特点	数据的管理者	用户(程序员)	文件系统	数据库管理系统
	数据面向的对象	某一应用程序	某一应用	现实世界
	数据的共享程度	无共享, 冗余度极大	共享性差, 冗余度大	共享性高, 冗余度小
	数据的独立性	不独立, 完全来自于程序	独立性差	具有高度的物理独立性和一定的逻辑独立性
	数据的结构化	无结构	记录内有结构、整体无结构	整体结构化, 用数据模型描述
	数据控制能力	应用程序自己控制	应用程序自己控制	由数据库管理系统提供数据安全性、完整性、并发控制和恢复能力

## 1.2 数据模型

数据库技术是计算机领域中发展最快的技术之一。数据库技术的发展是沿着数据模型的主线推进的。模型,特别是具体模型对人们来说并不陌生。一张地图、一组建筑设计沙盘、一架精致的航模飞机都是具体的模型,都是一眼望去就会使人联想到现实生活的事物。模型是对现实世界中某个对象特征的模拟和抽象。

数据模型也是一种模型,它是对现实世界数据特征的抽象。通俗地讲数据模型就是现实世界的模拟。

由于计算机不可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处理的数据,也就是首先要数字化,把现实世界中的具体的人、物、活动、概念用数据模型这个工具来抽象、表示和处理。

数据模型是数据库系统的核心和基础。现有的数据库系统均是基于某种模型的。根据模型应用的不同目的,可分为两类:第一类是概念模型,也称信息模型,按用户的观点来对数据和信息进行建模,主要用于数据库设计;另一类模型是数据模型,包括网状模型、层次模型、关系模型等,它按计算机系统的观点对数据建模,主要用于数据库系统的实现。

### 1.2.1 信息的三个领域

在现实世界中,信息处于三个领域:现实世界、信息世界和机器世界。①现实世界是存在于人们头脑之外的客观世界,事物及其相互联系就处于这个世界中。②信息世界是现实世界在人们头脑中的反映。客观事物在信息世界中被称为实体,反映事物联系的是实体模型。③机器世界是信息世界中信息的数据化。现实世界中的事物及联系在这里用数据模型描述。信息所处三个领域的联系如图 1-3 所示。

可见,客观事物是信息之源,是设计数据库的出发点,也是使用数据库的最终归宿。实体模型与数据模型是对客观事物及其联系的两级抽象描述,数据库的核心问题是数据模型,数据模型由实体模型导出。

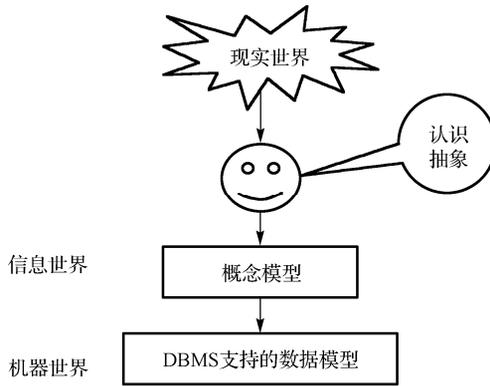


图 1-3 现实世界、信息世界和机器世界之间的联系

## 1.2.2 概念模型

由图 1-3 可以看出，概念模型实际上是现实世界到机器世界的一个中间层次。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，用于数据库设计。它是现实世界到机器世界的一个中间层次，是数据库设计的有力工具，是数据库设计人员和用户之间进行交流的语言。

在信息世界主要涉及以下一些概念。

### 1. 实体 (Entity)

实体是客观世界中存在的且可相互区分的事物，实体可以是人、事、物，也可以是抽象概念或联系。例如，一个职工、一个老师、一个学生、一门课、学生的一次选课、老师与院系的工作关系等都是实体，在 E-R 图中用矩形框表示。

### 2. 属性 (Attribute)

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、年龄、系别等属性组成。

### 3. 码 (Key)

唯一标识实体的属性集称为码。例如，学号是学生实体码。

### 4. 域 (Domain)

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，姓名为字符串集合。

### 5. 实体型 (Entity Type)

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、年龄、系别)就是一个实体型。

### 6. 实体集 (Entity Set)

同一类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 (Relationship)

现实世界中事物内部及事物之间的联系在信息世界中反映为实体内部的联系和实体之间的

联系。实体内部的联系通常是指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 (1:1)

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。例如乘客与车票、病人与床位之间的联系。

### 2) 一对多联系 (1:n)

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 ( $n \geq 0$ ) 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，班级与学生之间具有一对多联系。

### 3) 多对多联系 (m:n)

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 ( $n \geq 0$ ) 与之联系，反之，对于实体集  $B$  中的每一个实体，实体集  $A$  中也有  $m$  个实体 ( $m \geq 0$ ) 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，学生与课程之间具有多对多联系。

可以用图形来表示两个实体型之间的这三类关系，如图 1-4 所示。

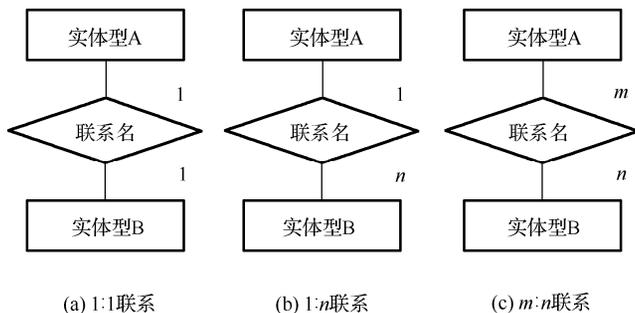


图 1-4 两个实体型之间的三类关系

## 1.2.3 概念模型的表示方法

概念模型是对信息世界建模，所以概念模型能够方便、准确地表示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、最为常用的是 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。

- ① 实体型：用矩形表示，矩形框内写明实体名。
- ② 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。
- ③ 联系：联系本身用菱形表示，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该联系连接起来。

现以某校教学管理为例建立实体模型，E-R 图如图 1-5 所示。

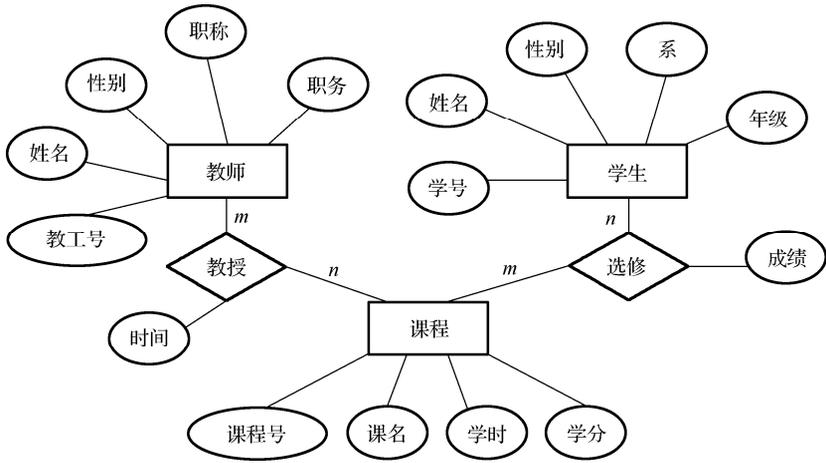


图 1-5 某学校教学管理 E-R 图

## 1.3 数据模型

### 1.3.1 数据模型的组成要素

数据模型 (Data Model) 是现实世界数据特征的抽象，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。数据模型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，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

数据结构是对系统静态特性的描述，是所研究的对象类型的集合。这些对象是数据库的组成部分，包括两类：一类是与数据类型、内容、性质有关的对象，如层次模型中的数据项、关系模型中的关系；另一类是与数据之间联系有关的对象，如网状模型中的系型 (Set Type)。

数据结构是刻画一个数据模型性质最重要的方面。因此在数据库系统中，通常按照数据结构的类型来命名数据模型。例如，层次结构的数据模型命名为层次模型，关系结构的数据模型命名为关系模型。

#### 2. 数据操作

数据操作是对系统动态特性的描述，是对数据库中各种对象 (型) 的实例 (值) 允许执行的操作的集合，包括操作及有关的操作规则。数据操作主要有检索和更新 (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、操作规则及实现操作的语言。

#### 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储存规则，用以限定符合数据模型的数据库状态及状态的变化，以保证数据正确、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。例如，在关系模型中，任何关系必须满足实体完整性和参照完整性两个条件。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。

### 1.3.2 最常用的数据模型

目前, 数据库领域中最常用的数据模型有五种, 它们是:

- 层次模型(Hierarchical Model);
- 网状模型(Network Model);
- 关系模型(Relational Model);
- 面向对象模型(Object Oriented Model);
- 对象关系模型(Object Relational Model)。

其中层次模型和网状模型统称为非关系模型。

非关系模型的数据库系统盛行于 20 世纪 70 年代至 80 年代初。在非关系模型中, 实体用记录表示, 实体的属性对应记录的数据项(或字段)。实体之间的联系在非关系模型中转换成记录之间的两两联系。非关系模型中数据结构的单位是基本层次联系(见图 1-6)。基本层次联系: 两个记录及它们之间的一对多(包括一对一)的联系。

20 世纪 80 年代以来, 面向对象的方法和技术对计算机各个领域都产生了深远的影响, 促进了数据库中面向对象数据模型的研究和发展。

下面简要介绍层次模型、网状模型和关系模型。

#### 1. 层次模型

层次模型是数据库系统中最早出现的数据模型, 层次模型用树形结构来表示各类实体及实体间的联系。层次数据库系统的典型代表是 IBM 公司的 IMS (Information Management System) 数据库管理系统。

##### 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。

- ① 有且只有一个结点没有双亲结点, 这个结点称为根结点;
- ② 根以外的其他结点有且只有一个双亲结点。

在层次模型中, 每个节点表示一个记录类型, 记录之间的联系用节点之间的连线(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。

每个记录型可以包含若干个字段, 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, 则它能唯一地表示一个记录值。在层次模型中, 同一双亲的子女结点称为兄弟结点(Twin 或 Sibling), 没有子女结点的结点称为叶结点。图 1-7 是一个教员学生层次数据库模型。该层次数据库有四个记录型。记录型“系”是根结点, 由“系编号”、“系名”、“办公地点”三个字段组成。它有两个子女“教研室”和“学生”。记录型“教研室”是“系”的子女结点, 同时又是“教员”的双亲结点, 它由“教研室编号”、“教研室名”两个字段组成。记录类型“学生”由“学号”、“姓名”、“成绩”三个字段组成。记录“教员”由“职工号”、“姓名”、“研究方向”三个字段组成。“学生”与“教员”是叶结点, 它们没有子女结点。由“系”到“教研室”、由“教研室”到“教员”、由“系”到“学生”均是一对多关系。

##### 2) 多对多联系在层次模型中的表示

因为层次数据模型只能表示一对多(包括一对一)的联系, 现实世界中大部分是多对多联系, 在层次模型中采用分解法表示多对多联系, 分解法有两种: 冗余结点法和虚拟结点法。

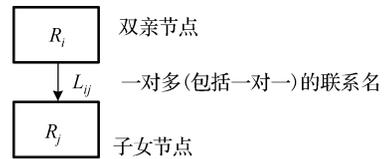


图 1-6 基本层次联系

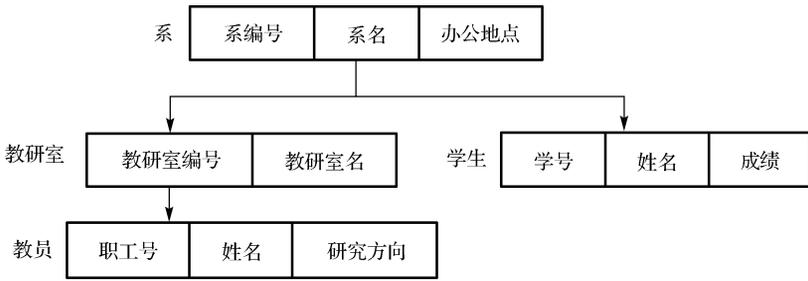


图 1-7 教员学生层次数据库模型

图 1-8(a) 是一个简单的多对多联系：一个学生可以选修多门课程，一门课程可由多个学生选修。“学生”字段由“学号”、“姓名”、“成绩”三个字段组成，“课程”由“课程号”和“课程名”两个字段组成。图 1-8(b) 采用冗余节点法，即通过增设两个冗余节点将图 1-8(a) 的多对多联系转换成两个一对多联系，图 1-8(c) 采用虚拟节点分解方法，即将图 1-8(b) 中的冗余节点转换为虚拟节点。虚拟节点就是一个指针，指向所替代的节点。

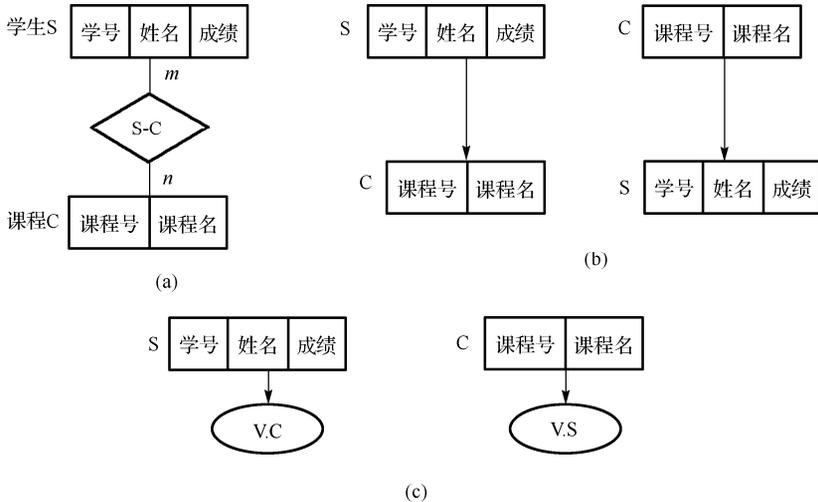


图 1-8 用层次模型表示多对多联系

### 3) 层次模型的数据操纵与完整性约束

层次模型的数据操纵主要有查询、插入、删除和修改。层次模型的完整性约束条件如下：

- (1) 无相应的双亲结点值就不能插入子女结点值；
- (2) 如果删除双亲结点值，则相应的子女结点值也被同时删除；
- (3) 更新操作时，应更新所有的相应记录，以保证数据的一致性。

### 4) 层次数据模型的存储结构

层次数据库中不仅要存储数据本身，还要存储数据之间的层次联系。常用的实现方法有以下两种。

#### (1) 邻接法

按照层次树前序穿越的顺序把所有记录值依次邻接存放，即通过物理空间的位置相邻来实现（或隐含）层次顺序。例如，对于图 1-9(a) 的数据库，按邻接法存放图 1-9(b) 中以根记录 A1 为父的层次记录实例集，应按图 1-10 所示的方法存放。

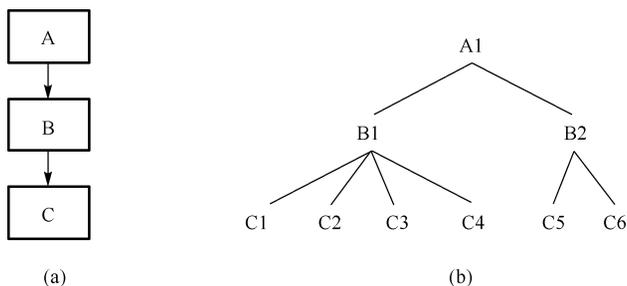


图 1-9 层次数据库及其实例

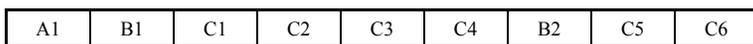


图 1-10 邻接法

## (2) 链接法

链接法用指引元来反映数据之间的层次联系。如图 1-11 所示，图 1-11 (a) 中的每个记录设两类指引元，分别指向最左边的子女和最近的兄弟，这种连接方法称为子女-兄弟链接法；图 1-11 (b) 按树的前序穿越顺序链接各记录值，这种链接方法称为层次序列链接法。

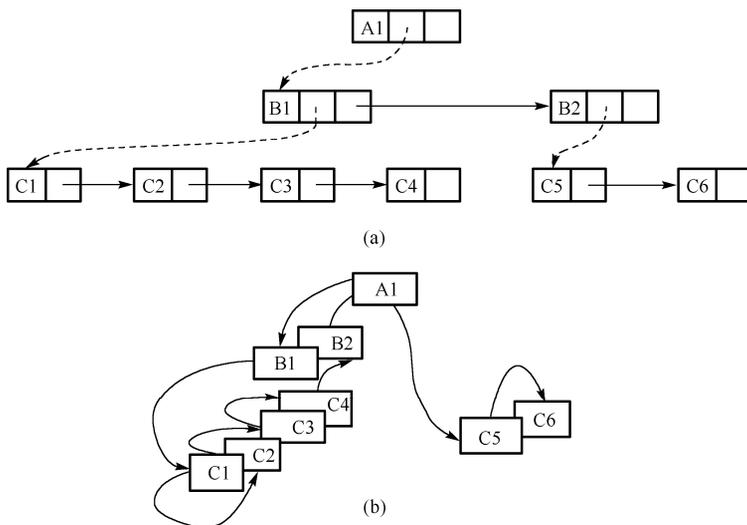


图 1-11 链接法

## 5) 层次模型的优缺点

层次模型的优点主要有：

- (1) 层次模型的数据结构简单清晰；
- (2) 查询效率高，性能优于关系模型，不低于网状模型；
- (3) 层次数据模型提供了良好的完整性支持。

层次模型的缺点主要有：

- (1) 多对多联系表示不自然；
- (2) 对插入和删除操作的限制多，应用程序的编写比较复杂；
- (3) 查询子女结点时必须通过双亲结点；
- (4) 由于结构严密，层次命令趋于程序化。

## 2. 网状模型

网状数据库系统采用网状模型作为数据的组织方式,典型代表是 DBTG 系统,也称 CODASYL 系统。CODASYL 系统是 20 世纪 70 年代由 DBTG 提出的一个系统方案,奠定了数据库系统的基本概念、方法和技术。后来不少的系统都采用 DBTG 模型或简化的 DBTG 模型。例如, Cullinet Software Inc.公司的 IDMS、Univac 公司的 DMS1100、Honeywell 公司的 IDS/2、HP 公司的 IMAGE 等。

### 1) 网状模型的数据结构

网状模型是满足下面两个条件的基本层次联系的集合:

- (1) 允许一个以上的结点无双亲;
- (2) 一个结点可以有多个的双亲。

可以看出网状模型与层次模型的区别:网状模型允许多个结点没有双亲结点;网状模型允许结点有多个双亲结点;网状模型允许两个结点之间有多种联系(复合联系);网状模型可以更直接地去描述现实世界;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)之间一对多的父子关系。

图 1-12 是用网状模型表示的学生选课数据库。

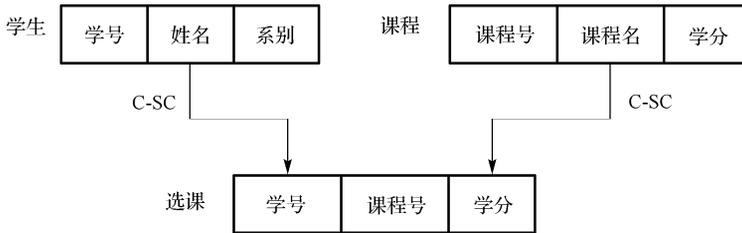


图 1-12 学生选课网状数据库模式

### 2) 网状模型的操纵与完整性约束

网状数据库系统(如 DBTG)对数据操纵加了一些限制,提供了一定的完整性约束:

- (1) 码: 唯一标识记录的数据项的集合。
- (2) 一个联系中双亲记录与子女记录之间是一对多联系。
- (3) 支持双亲记录和子女记录之间某些约束条件。

### 3) 网状模型的存储结构

网状模型的存储结构中,关键是如何实现记录之间的联系。常用的方法是链接法,包括单向连接、双向链接、环状链接和向首链接等。

图 1-13 为学生选课网状数据库的一个存储示意图。图中实线链表示 S-SC 联系,虚线链表示 C-SC 联系。

### 4) 网状数据模型的优缺点

网状数据模型的优点主要有:

- (1) 能够更直接地描述现实世界,如一个结点可以有多个双亲;
- (2) 具有良好的性能,存取效率较高。

网状模型的缺点主要有:

(1) 结构比较复杂,而且随着应用环境的扩大,数据库的结构变得越来越复杂,不利于最终用户掌握;

(2) DDL、DML 语言复杂，用户不容易使用。

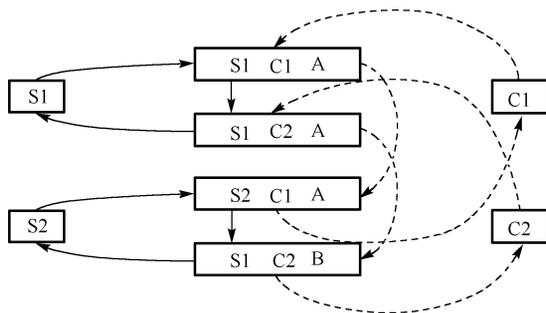


图 1-13 学生选课网状数据库实例

### 3. 关系模型

关系数据库系统采用关系模型作为数据的组织方式。1970 年美国 IBM 公司 San Jose 研究室的研究员 E. F. Codd 首次提出了数据库系统的关系模型。目前，计算机厂商新推出的数据库管理系统几乎都支持关系模型。

#### 1) 关系模型的数据结构

关系模型是建立在严格的数学概念的基础上的。在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。现在以教师登记表为例（见表 1-2）介绍关系模型中的一些术语。

表 1-2 教师登记表

教师号	姓名	年龄	系别
09001	赵谦	26	历史系
09002	孙立	27	中文系
...	...	...	...

#### (1) 关系 (Relation)

一个关系对应通常所说的一张表。

#### (2) 元组 (Tuple)

表中的一行即为一个元组。

#### (3) 属性 (Attribute)

表中的一列即为一个属性，给每一个属性所起的一个名称即属性名。

#### (4) 主码 (Key)

主码是表中的某个属性组，它可以唯一确定一个元组。

#### (5) 域 (Domain)

属性的取值范围即为域。

#### (6) 分量

分量是元组中的一个属性值。

#### (7) 关系模式

关系模式是对关系的描述。一般表示为：关系名(属性 1, 属性 2, ..., 属性  $n$ )

例如，上面的关系可以描述为：教师(教师号, 姓名, 年龄, 系别)。

关系必须是规范化的，满足一定的规范条件。最基本的规范条件：关系的每一个分量必须是一个不可分的数据项，不允许表中还有表。

## 2) 关系数据模型的操纵与完整性约束

数据操作是集合操作，主要包括查询、插入、删除和更新数据。操作对象和操作结果都是关系。关系的完整性约束条件包括三大类：实体完整性、参照完整性和用户定义的完整性。

## 3) 关系数据模型的存储结构

在关系数据模型中，实体及实体间的联系都用表来表示，表以文件形式存储。有的 DBMS 一个表对应一个操作系统文件，有的 DBMS 自己设计文件结构。

## 4) 关系数据模型的优缺点

关系数据模型的优点主要有：

(1) 建立在严格的数学概念的基础上；

(2) 概念单一，实体和各类联系都用关系来表示，数据的检索结果也是关系；

(3) 关系模型的存取路径对用户透明，具有更高的数据独立性，更好的安全保密性，简化了程序员的工作和数据库开发建立的工作。

关系数据模型的缺点主要有：

(1) 存取路径对用户透明导致查询效率往往不如非关系数据模型；

(2) 为提高性能，必须对用户的查询请求进行优化，增加了开发 DBMS 的难度。

# 1.4 数据库的体系结构

## 1.4.1 数据库的分级结构

数据库分为三级：外模式、模式和内模式，如图 1-14 所示，掌握数据库的三级结构及其联系与转换是深入学习数据库的必由之路。

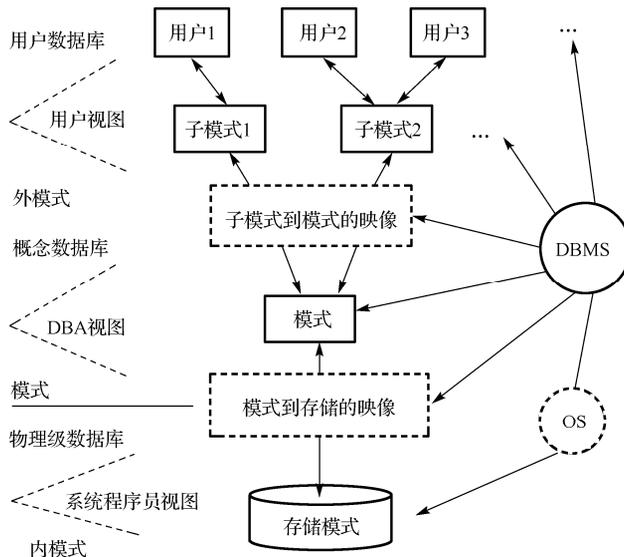


图 1-14 数据库的分级结构

用户级数据库也称为用户视图 (View) 或子模式，对应于外模式，它是单个用户看到并获准使用的那部分数据的逻辑结构 (称为局部逻辑结构)，用户根据系统给出的子模型，用询问语言或应用程序去操作数据库中的数据。

概念级数据库对应于概念模式，简称模式，是对数据库所有用户数据的整体逻辑描述(故称为数据库的整体逻辑结构)，又称为 DBA 视图，即数据库管理员看到的数据库，它是所有用户视图的一个最小并集。设立概念级的目的是把用户视图有机地结合成一个逻辑整体，统一地考虑所有用户要求，它涉及的仍是数据库中所有对象的逻辑关系，而不是它们的物理情况。

物理级数据库对应于内模式，又称为存储模式。它包含数据库的全部存储数据，这些被存储在内外存介质上的数据也称为原始数据，是用户操作(加工)的对象。从机器的角度看，它们是指令操作处理的位串、字符和字；从系统程序员的角度看，这些数据是用一定的文件组织方法组织起来的一个个物理文件(或存储文件)，系统程序员编制专门的访问程序，实现对文件中数据的访问，所以物理级数据库也称为系统程序员视图。

对一个数据库系统来说，实际上存在的只是物理级数据库，它是数据访问的基础。概念级数据库只不过是物理级数据库的一种抽象(逻辑)描述，用户级数据库是用户与数据库的接口。用户根据子模式进行操作，数据库管理系统通过子模式到模式的映像将操作与概念级联系起来，又通过模式到存储模式的映像与物理级联系起来。这样一来，用户可以在较高的抽象级别上处理数据，而把数据组织的物理细节留给系统。事实上，DBMS 的中心工作之一就是完成三级数据库之间的转换，把用户对数据库的操作转化到物理级去执行。

## 1.4.2 模式及映像

### 1. 模式 (Schema)

模式也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个模式，模式主体是数据模型，此外，一般还包括允许的操作、数据完整性和安全保密等方面的控制。DBMS 提供模式描述语言(模式 DDL)来严格定义模式。

用语言书写的模式称为源模式，机器不能直接使用，必须将此模式用机器代码表示，变为机器书写的模式，称为目标模式。目标模式通常设计成表格形式、树结构或网络结构。

### 2. 外模式 (External Schema)

外模式也称子模式 (Subschema) 或用户模式，它是数据库用户(包括应用程序员和最终用户)能够看见和使用的局部数据的逻辑结构和特征描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。从逻辑关系上看，外模式是模式的一部分，从模式用某种规则(如关系方法中的关系运算)可以导出外模式，但外模式也可以做出不同外子模式的改变，如在外模式中略去模式的某些记录类型、数据项，改变模式中某些数据项的数据库特征，改变模式中的安全、完整约束条件等。子模式与模式之间的对应关系称为子模式到模式的映像，描述子模式的语言称为子模式 DDL，外模式也有相应的源形式与目标形式。

### 3. 内模式 (Internal Schema)

内模式也称存储模式 (Storage Schema)，是数据在物理存储结构方面的描述，是数据库内部的表示方式，一个数据库只有一个内模式，它除了定义所有的内部记录类型外，还定义一些索引、存储分配及恢复等方面的细节。例如，记录的存储方式是堆存储，还是按照某个属性值的升(降)序存储，还是按照属性值聚簇存储；索引按照什么方式组织，是 B+ 树索引还是 Hash 索引等，但更具体的物理存储细节，如从磁盘读写某些数据块等，存储模式一般不予考虑，而是交给操作系统去完成。

所以，三级模式是对数据的三个抽象级别，它把数据的具体组织留给 DBMS 管理，使用户能逻辑地、抽象地处理数据，而不必关心数据在计算机中的具体表示方式与存储方式。为了能够在

系统内部实现这3个抽象层次的联系和转换,数据库管理系统在这三级模式之间提供了两层映像,即外模式/模式映像和模式/内模式映像,正是这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

#### 4. 外模式/模式映像

外模式到模式的映像主要给出外部级与概念级的对应关系。这两级的数据结构及数据量纲可能不一致,外模式中某些数据项甚至是由若干数据导出的,在数据库中并不真实存在。因此在映像中需要说明外模式中的记录类型和数据项如何对应模式中的记录和数据项及导出的规则步骤(若为导出项)。在外部级与概念级之间存在这样一个映像,就能保证当模式发生变化时,只要修改记录级数据项的对应关系或导出规则,可能不必修改外模式,用户根据外模式设计的应用程序就可继续沿用,达到所谓的逻辑数据独立性。通常情况下,外模式到模式的映像的模式中描述。

#### 5. 模式/内模式映像

模式到内模式的映像主要给出概念级数据与物理级数据之间的对应关系。表现在两个方面:一方面是数据结构的变换,另一方面是逻辑数据如何在物理设备上定位。在概念级与内部级之间存在这样一个映像,就能保证当存储模式发生变化时(如存储设备、文件组织方法、存储位置等发生变化),只要修改此映像即可,而模式尽量不受影响,从而使得对内模式和应用程序的影响更小,达到所谓的物理数据独立性。通常情况下,模式到内模式的映像在内模式中描述。

数据与程序之间的独立性,使得数据的定义和描述可以从应用程序中分离出去。另外,由于数据的存取由 DBMS 管理,用户不必考虑存取路径等细节,从而简化了应用程序的编制,大大减少了应用程序的维护和修改。

## 1.5 数据库管理系统

数据库管理系统(DBMS)是一个非常复杂的软件系统,是数据库系统具有数据共享、并发访问、数据独立等特性的根本保证,对数据库系统的所有操作和各种运行控制最终都是通过 DBMS 实现的。

### 1.5.1 DBMS 的功能

DBMS 的主要职责就是有效地实现数据库三级之间的转换,即把用户(或应用程序)对数据库的一次访问,从用户级带到概念级,再导向物理级,转换为存储数据的操作。因此其功能应包括下面几类。

#### 1. 数据库的定义

DBMS 总是提供数据定义语言 DDL 用于描述模式、外模式和内模式及各模式之间的映像,描述的内容包括数据的结构、数据的完整性约束条件和访问控制条件等,并负责将这些模式的源形式转换成目标形式,存在系统的数据字典中,供以后操作或控制数据时查用。

#### 2. 数据库的操作及查询优化

DBMS 总是提供数据操作语言 DML 实现对数据库的操作,基本操作包括检索、插入、删除和修改。用户只需根据外模式给出操作要求,其处理过程的确定和优化则由 DBMS 完成。查询处理和优化机制的好坏直接反映了 DBMS 的性能。

#### 3. 数据库的控制运行

数据库方法的最大优势在于允许多个用户并发地访问数据库,充分实现共享,DBMS 必须提

供并发控制机制、访问控制机制和数据完整性约束机制，从而避免多个读写操作并发执行可能引起的冲突、数据失密或安全性、完整性被破坏等一系列问题。

#### 4. 数据库的恢复和维护

这些维护信息可将数据库恢复到一致状态。此外，当数据库性能下降或系统软/硬件设备变化时，也能重新组织或更新数据库。

#### 5. 数据库的数据管理

数据库中物理存在的数据包括两部分：一部分是元数据，即描述数据的数据，主要是前述的三类模式，它们构成了数据字典(DD)的主体，DD 由 DBMS 管理、使用；另一部分是原始数据，它们构成物理存在的数据。DBMS 一般提供多种文件组织方法，供数据库设计人员选用。数据一旦按某种组织方法装入数据库，其后对它的检索和更新都由 DBMS 的专门程序完成。

#### 6. 数据库的多种接口

数据库一旦设计完成，可能供多类用户使用，包括常规用户、应用程序的开发者、DBA 等。为适应不同用户的需求，DBMS 常提供各种接口，近年来还普遍增加了图形接口，用户使用起来更直观、方便。

### 1.5.2 DBMS 的程序组成

从程序的角度看，DBMS 是完成上述各项功能的许多程序模块组成的一个集合，其中一个或几个程序一起完成 DBMS 的一项工作，或一个程序完成几项工作，以设计方便和系统性能良好为原则，所以各个 DBMS 的功能不完全一样，包含的程序也不同。其主要程序如下。

#### 1. 语言处理方面

- (1) 模式 DDL 翻译程序：把模式 DDL 源形式翻译成机器可读的目标形式。
- (2) 外模式 DDL 翻译程序：把外模式 DDL 源形式翻译成目标形式。
- (3) DML 处理程序：把应用程序的 DML 语句转化成主语言的一个过程调用语句。
- (4) 终端选文解释程序：解释终端询问的意义，决定操作的执行过程。
- (5) 数据库控制命令解释程序：解释每个控制命令的含义，决定怎样执行。

#### 2. 系统运行控制方面

- (1) 系统总控程序：DBMS 的神经中枢，它控制、协调 DBMS 各个程序的活动，使其有条不紊地运行。
- (2) 访问控制程序：内容包括核对用户标识、口令，核对授权表，检验访问的合法性等，它决定一个访问是否能够进入系统。
- (3) 并发控制程序：在许多用户同时访问数据库时协调各个用户的访问。
- (4) 保密控制程序：在执行操作之前，核对保密规定。
- (5) 数据完整性控制程序：在执行操作前或后，核对数据库完整约束条件，从而决定是否允许操作执行，或清除已执行操作的影响。
- (6) 数据访问程序：根据用户访问请求，实施对数据的访问。
- (7) 通信控制程序：实现用户程序与 DBMS 之间的通信。

#### 3. 系统建立、维护方面

- (1) 数据装入程序：把大批原始数据按某种文件组织方法(顺序、索引、Hash 等)存储到内外存介质上，完成数据库的装入。

(2) 工作日志程序：负责记载进入数据库的所有访问。

(3) 性能监督程序：监督操作执行时间与存储空间占用情况，做出系统性能估算，以决定数据库是否需要重新组织。

(4) 重新组织程序：当数据库系统性能变坏时需要数据重新进行物理组织，或者按原组织方法重新装入，或者改变原组织方法、采用新的结构。

(5) 系统恢复程序：当软/硬件设备遭到破坏时，该程序把数据库恢复到可用状态。

#### 4. 用户接口方面

目前不少 DBMS 的接口软件的规模往往超过 DBMS 的核心软件。

### 1.5.3 数据语言

数据语言包括数据描述语言 (DDL) 和数据操作语言 (DML) 两大部分，前者负责描述和定义数据的各种特性，后者说明对数据进行的操作。

#### 1. 数据描述语言

数据描述语言用于描述数据库中各种对象的特征，其功能包括以下四个方面。

(1) 描述数据的逻辑结构。因为数据库的逻辑描述限于概念级和用户级，所以在数据库中又分模式数据描述语言和子模式数据描述语言。前者描述模式，它必须具备定义数据模型的功能和容易阅读的优点，是一种与现行普通程序语言不同的格式化的陈述性语言，在采用 SQL 标准关系数据库系统中，模式 DDL 都是统一的。后者描述子模式，它可以是用户使用的程序语言 (如 C、Fortran 等) 的扩充。不论怎样设计模式与子模式 DDL，描述数据的逻辑组织一般要包括如下内容。

① 描述数据模型各个部分 (称为数据逻辑单位) 的特征，通常是：给出数据各个逻辑单位的无二义性命名，如数据模型名、记录类型名、数据项名等；说明各个最小逻辑单位的数据特征，如数据类型是字母、数字还是字符，数据长度及取值范围等；说明数据单位的自然含义，如数据项是表示姓名还是城市、表示年龄还是分数等。

② 描述各数据逻辑单位之间的联系，以及数据表示的对象之间和对象内部的联系，通常说明：数据的一个逻辑单位按照什么规则包含哪些更小的逻辑单位，如一个记录类型一次包含哪些数据项；哪个或哪些数据项组合成关键字使用；数据项之间的完整性约束条件；各逻辑单位按什么规则形成一个整体的数据结构，如记录类型怎样构成树或网络。

(2) 描述数据的物理特征。数据库设计称之为存储结构描述语言，描述的内容包括：① 系统中建立了哪些物理文件，它用文件目录说明，文件目录是系统的一个特殊文件；② 说明每个物理文件数据的文件组织方法，如顺序、索引、Hash、树结构、倒排结构等；③ 说明每个物理文件的文件名称、文件的存储设备、开始地址、允许占用的空间等。

(3) 描述存储映像即逻辑数据到物理数据的映像。它说明：① 每个逻辑单位的数据存放在哪个文件中，如哪个记录类型对应于哪个文件，存放在哪个区域；② 逻辑数据到物理数据的转换，如十进制变二进制，数据的存储紧缩过程等。

(4) 描述访问规则。它包括：① 用户与子模式的对应关系；② 用户身份检验，如用户口令、声音、指纹的核对；③ 对用户的授权；④ 数据的加密锁与密码。

上述内容只是数据描述的一些基本内容，并不是详尽无遗的，各个数据库系统都根据自己的情况决定包括哪些内容。

## 2. 数据操作语言

数据操作语言是用户与数据库系统的接口之一，是用户操作数据库中数据的工具。在设计数据操作语言时，一般要做到：描述操作准确，无二义性；功能齐全，操作能力强，用户希望使用的操作应尽量满足：语言自然、直观，容易掌握，使用方便。

一般来说，数据操作是一些操作语句组成的集合。DML 有两类，一类嵌入在宿主语言中使用，称为宿主型 DML，如嵌入在 COBOL、Fortran、C、Ada 等高级语言中；另一类可以独立交互使用，称为自主型或自含型 DML，如 SQL (SQL 也可以嵌在主语言中使用)、QBE 等。当前各种数据库系统的数据语言虽然在形式上差别较大，但其功能基本相似，通常包括如下几方面的操作：

- (1) 从数据库中检索数据；
- (2) 向数据库中添加数据；
- (3) 删除数据库中某些已过时的、没有保留价值的原有数据；
- (4) 修改某些属性发生变化的数据项的值，使其能确切反映变化后的情况；
- (5) 用于并发访问控制的操作。

### 1.5.4 数据字典

数据字典 (Data Dictionary) 的主要任务是描述 (或定义) 数据库系统中各类对象的性质和属性 (包括其自然语言含义)、对象之间的交叉联系和它们的使用规则。数据字典描述时给每个对象一个唯一的标识 (称为内码)，以示区别。数据字典是复杂的，也要用数据模型描述，又把它称为关于数据的“数据”或“元数据”，它也有源形式与目标形式的问题。它的目标形式是包括模式、子模式表、用户表、物理文件或区域表、内码与自然语言对照表等在内的一组字典表格。

一个数据库系统所涉及的对象大致可分为如下几类：

- (1) 与数据组织结构有关的对象，包括数据库 (一个系统可能容纳多个数据库)、模式、子模式、记录类型 (或类)、数据项、物理文件及索引等。
- (2) 与系统运行、配置有关的对象，包括存储过程、事务、终端、客户机等。
- (3) 与询问优化有关的对象，包括访问例程、代价估算所用各类信息。
- (4) 与完整、安全控制有关的对象，包括用户、角色、用户标识、访问授权、密钥及各类完整约束条件等。
- (5) 与系统监控有关的对象，包括触发子、审计项目及数据字典本身的变化情况等。
- (6) 数据字典的用处在于：满足 DBMS 快速查找有关对象的要求。供数据库管理员掌握整个系统运行的下列情况：①系统现有的数据库、用户；②当前具有的模式数目及其名称，每个模式包含的子模式与记录类型；③每个子模式包含的记录类型与用户；④某个记录类型 (或数据项) 所属的子模式和对应的处理文件或区域等。

## 1.6 小 结

本章对数据库系统进行了概述，介绍了数据库的基本概念和数据管理的发展过程。数据库中的数据是按照一定的结构和模型进行组织的，重点要掌握 E-R 模型的基本概念。层次模型、网状模型和关系模型是三种主要的数据库模型，本章详细介绍了这几种模型的优缺点和相关概念。

从数据库管理系统的角度看，数据库系统通常采用三级模式结构，这是数据库系统内部的结构。数据库系统的三级模式和二级映像保证了数据库系统的逻辑独立性和物理独立性。