

第 1 章

数据库基础

本章学习提要 with 目标

本章介绍数据库的基本知识，包括数据库的基本概念、数据模型、基本结构等。通过对本章的学习，读者可以掌握数据库、数据库系统及数据库管理系统等基本概念，掌握数据库系统的特征、数据模型的分类、数据库系统的三级模式和两级映像，了解设计、管理和使用数据库的人员。

自从 20 世纪 60 年代中期数据库技术诞生以来，无论是在理论还是应用方面，此技术都已变得相当重要和成熟，成为计算机科学的重要分支。数据库技术是计算机领域发展最快的学科之一，也是应用很广、实用性很强的一门技术。目前，这一技术已从第一代的网状、层次数据库系统，第二代的关系数据库系统，发展到以面向对象模型为主要特征的第三代数据库系统。

随着计算机技术的飞速发展及其应用领域的扩大，特别是计算机网络和互联网的发展、基于计算机网络和数据库技术的管理信息系统和各类应用系统得到了突飞猛进的发展。如事务处理系统（TPS）、地理信息系统（GIS）、联机分析系统（OLAP）、决策支持系统（DSS）、企业资源计划（ERP）、客户关系管理（CRM）、数据仓库（DW）及数据挖掘（DM）等系统都是以数据库技术作为重要支撑。可以说，只要有计算机的地方，就在使用着数据库技术。因此，数据库技术的基本知识和基本技能已成为信息社会人们必备的部分。本章的 1.1 节将首先介绍数据库系统的基本概念和特征，1.2 节介绍数据模型及其分类，1.3 节介绍数据库管理系统的体系结构，1.4 节介绍设计、管理和使用数据库的人员。

1.1 数据库的基本概念

本节首先介绍一些数据库最常用的术语和基本概念，然后介绍与文件系统相比，数据库系统的特征与优势。



1.1.1 基本概念

1. 数据与数据处理

数据是数据库系统研究和处理的对象。所谓数据，通常是指可以用符号记录下来的、对事物的描述。描述事物的符号可以是数字，也可以是文字、图形、图像、声音等。例如，一篇文章、一幅地图、一首乐曲，这些都是数据。当然，这里研究的数据是指经过数字化存入计算机中的数据。

数据处理也被称为信息处理，是指从某些已知的数据出发，推导加工出一些新的数据的过程。数据处理不同于科学计算，科学计算通常比较简单，而数据处理则比较复杂。数据管理是数据处理的基础，其包括数据的收集、整理、存储、维护、检索等操作。

2. 数据库

数据库这个名词起源于 20 世纪 50 年代，当时美国为了战争，需要把各种情报集中在一起存储在计算机中，成为数据库。1963 年，美国 Honeywell 公司的 IDS (Integrated Data Store) 系统投入运行，揭开了数据库技术的序幕。1965 年，美国利用数据存储系统帮助 NASA 设计了阿波罗登月火箭，推动了数据库技术的发展。当时，在美国出现了形形色色的 Database 或 Databank 软件系统，但它们基本上都是文件系统的扩充。

1968 年，美国 IBM 公司推出层次模型的 IMS 数据库系统；1969 年，美国 CODASYL (Conference on Data Systems Languages, 数据系统语言协会) 组织的数据库任务组 (DBTG) 发表关于网状模型数据库的 DBTG 报告；1970 年，IBM 公司的科德 (E. F. Codd) 发表论文提出了关系模型数据库。这三件事情奠定了现代数据库技术发展的基础。

所谓数据库 (Database, DB)，是指长期存储在计算机中的、有组织的、可共享的数据集合。数据库中的数据往往按一定的数据模型组织、描述和存储，能为各种用户共享，具有较小的冗余度、较高的数据独立性、数据间联系密切等特点。

在计算机中，数据库是由很多数据文件及相关的辅助文件所组成，这些文件由一个被称为数据库管理系统 (Database Management System, DBMS) 的软件进行统一管理和维护。数据库中除了存储用户直接使用的数据外，还存储有另一类“元数据”，它们是有关数据库的定义信息，如数据类型、模式结构、使用权限等，这些数据的集合被称为数据字典 (Data Dictionary, DD)，它是数据库管理系统工作的依据，数据库管理系统通过 DD 对数据库中的数据进行管理和维护。

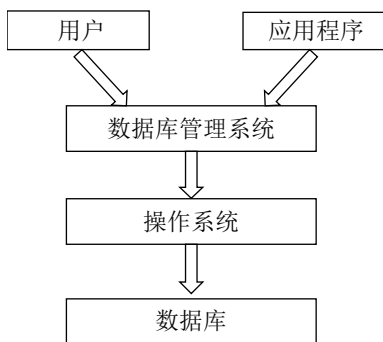


图 1.1 数据库管理系统

3. 数据库管理系统

如图 1.1 所示，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。它为用户或应用程序提供访问数据库的方法，包括数据库的建立、查询、更新及各种数据控制。

1) DBMS 的主要功能

(1) 数据定义功能。DBMS 提供了数据定义语言 (DDL)，数据库设计人员通过它可以方便地对数据库

中的相关内容进行定义。例如，对数据库、表、索引及数据的完整性进行定义。

(2) 数据操纵功能。DBMS 提供了数据的操纵语言 (DML)，用户通过它可以实现对数据库的基本操作。例如，对表中数据的查询、插入、删除和修改。

(3) 数据库运行控制功能 (保护功能)。这是 DBMS 的核心部分，它包括并发控制 (即处理多个用户同时使用某些数据时可能产生的问题)、安全性检查、完整性约束条件的检查和执行、数据库的内部维护 (例如，索引的自动维护) 等。所有对数据库的操作都要在这些控制程序的统一管理下进行，以保证数据的安全性、完整性以及多个用户对数据库的并发使用。

(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，数据库的重新组织功能和性能监视、分析功能等。这些功能通常是由一些实用程序完成的，它们是数据库管理系统的重要组成部分。

2) DBMS 的组成

数据库管理系统主要由数据库描述语言及其编译程序、数据库操作语言及其翻译程序、数据库管理和控制例行程序三部分组成。其中，数据库描述语言及其编译程序主要完成对数据库数据的物理结构和逻辑结构的定义；数据库操作语言及其翻译程序完成数据库数据的检索和存储；而管理和控制例行程序则用于完成数据的安全性控制、完整性控制、并发性控制、通信控制、数据存取、数据修改以及工作日志、数据库转储、数据库初始装入、数据库恢复、数据库重新组织等公用管理。

3) DBMS 与数据模型的关系

前已述及，数据库中的数据是根据特定的数据模型来组织和管理的，与之对应的，数据库管理系统总是基于某种数据模型，故可以把 DBMS 看成是某种数据模型在计算机系统上的具体实现。根据数据模型的不同，DBMS 可以分为层次型、网状型、关系型和面向对象型等，如利用关系模型建立的数据库管理系统就是关系型数据库管理系统等。目前商品化的数据库管理系统主要为关系型的，如大型系统中使用的 Oracle、DB2、Sybase 及微机上使用的 Access、Visual FoxPro 及 SQL Server 系列产品等。需要说明的是，不同的计算机系统由于缺乏统一的标准，故即使同一种数据模型的 DBMS，在用户接口、系统功能等方面也常常是不相同的，本书以 SQL Server 2008 系统为介绍对象。

4. 数据库系统

数据库系统 (Database System, DBS) 是指引入数据库技术后的计算机系统。数据库系统是一个复杂的系统，不仅是一组对数据进行管理的软件，还是一个数据库。DBS 是存储介质、处理对象和管理系统的集合体，由数据库、硬件、软件 (包括数据库管理软件、操作系统软件、应用软件等)、数据库管理员和用户组成，是一个按照数据库方法存储、维护和向应用系统提供数据支持的系统。

(1) 计算机支持系统。主要有硬件支持环境和软件支持系统 (如 DBMS、操作系统及开发工具)。

(2) 数据库。按一定的数据模型组织，长期存放在外存上的一组可共享的相关数据集合。

(3) 数据库管理系统。一个管理数据库的软件，简称 DBMS，它是数据库系统的核

心部件。

(4) 数据库应用程序。指满足某类用户要求的用于操纵和访问数据库的程序。

(5) 人员。数据库系统分析设计员、系统程序员、用户等。数据库用户通常又可分为两类：一类是批处理用户，也被称为应用程序用户，这类用户使用程序设计语言编写应用程序，对数据进行检索、插入、修改和删除等操作，并产生数据输出；另一类是联机用户，或被称为终端用户，他们使用终端命令或查询语言直接对数据库进行操作，这类用户通常是数据库管理员或系统维护人员。

1.1.2 数据库系统与文件系统

假设银行需要保存所有客户及其账户的信息。一种方法是将它们存储在操作系统文件中，由应用程序通过文件系统对它们进行存取，但这种方法会随着数据管理规模的扩大和数据量的急剧增加显露出一些缺陷。

第一，数据的冗余和不一致。在传统的文件处理模式下，应用程序所需要的所有文件的定义是该应用程序的一部分，不同的应用程序会定义不同的文件，特别的是，如果这些应用程序是在很长的一段时间内由不同的程序员创建，那么就会使相同的信息重复存储在不同的文件中。例如，储蓄账户管理程序所定义的文件中包含客户地址和电话等信息，若银行又要开设支票账户，就要开发相应的程序，那么定义的文件中可能也包含客户地址和电话等信息，这就造成了数据的冗余存储。这种冗余除了会导致存储和访问开销增大外，还可能导致数据不一致。又如，某个客户地址的变更可能在储蓄账户文件中得到反映，而在支票账户文件中却没有更新。

第二，数据间的联系弱。在传统的文件处理模式下，数据文件之间相互独立、缺乏联系，并且可能具有不同的格式，无法以一种方便而有效的方式获取数据。例如，要找出某一邮编地区各个客户的所有储蓄账户和支票账户的信息，此需求涉及储蓄账户文件和支票账户文件两个孤立的文件，并且它们也可能具有不同的客户地址格式，编写这样一个应用程序是比较困难的。

数据库系统克服了文件系统的上述缺陷，提供了对数据更有效的管理方式。数据库系统具有以下几个特征。

1. 数据结构自描述

数据库系统不仅包含数据库本身，其还通过系统目录（System Catalog）定义了数据库的结构、每个数据项的类型以及加在数据上的各种约束条件。系统中的任何应用都可以通过数据库管理系统软件从系统目录中提取数据库的定义，根据需求方便地获得对数据库全部或某些数据项的存取。例如，客户信息数据库中包括客户编号、姓名、住址、电话等信息，所以处理储蓄账户的应用程序、处理支票账户的应用程序以及打印客户列表的应用程序都可以使用该数据库获得所需要的某些客户信息。因此数据是面向整个系统的，可以被多个应用、多个用户所共享。数据共享可以大大减少数据冗余，避免数据的不一致性。

在传统的文件处理模式下，数据文件的定义一般被作为应用程序自身的一部分，其包

含的数据是面向特定的某个或几个应用程序的，并且程序对这些数据的操作只能以记录为单位，不能以数据项为单位。

2. 较高的数据独立性

在传统的文件处理模式下，数据文件的结构是由存取它的应用程序定义的。因此，文件结构的任何改变都将导致存取该文件的所有程序的改变。而在数据库系统中，大多数情况下可以避免这种改变，即便数据的逻辑结构改变了，应用程序也可以不变，也就是说数据库的逻辑结构和用户的应用程序是相互独立的，人们通常把这种特性称为逻辑独立性。例如，要为客户信息数据库增加一个“生日”数据项，用户只需通过 DBMS 改变系统目录中客户信息数据库结构的描述来实现，无须改变存取客户信息数据库的任何应用程序。

另外，数据库中的数据在磁盘上的存储也是由 DBMS 管理的。DBMS 向用户隐藏文件存储组织的细节，应用程序要处理的只是数据的逻辑结构。这样，就算数据的物理存储位置改变了，应用程序也不需改变，也就是说，数据库的物理结构和用户的应用程序是相互独立的，人们通常把这种特性称为物理独立性。

3. 统一的数据管理与控制

数据库系统统一提供以下四个方面的数据管理与控制功能。

(1) 数据完整性。其可以保证数据库始终包含正确的数据。用户可以设计完整性规则以确保数据满足某些特定的约束。例如，银行账户的余额永远不会低于某个预定的值（如 1 元）。

(2) 数据库的可恢复性。在发生故障时，系统需要有把数据库恢复到最近某时刻的正确状态。对很多应用程序来说，这样的保证是至关重要的。例如，要把 A 账户的 500 元转入 B 账户。假设系统首先从 A 账户上减去了 500 元，但在转入 B 账户之前系统发生了故障，这 500 元没来得及存入 B 账户，这就造成了数据库状态的不一致，此时，就需要把数据库恢复到未作转账之前的正确状态。为此，数据库系统需要采用事务的概念保证转账事务涉及的借和贷两个操作要么全部发生，要么根本不发生。

(3) 数据的并发控制。其可以避免并发操作之间相互干扰，防止数据库的完整性被破坏。为了提高总体性能和加快响应速度，许多系统都允许多个用户同时更新数据。例如，某账户中原有余额 500 元，甲、乙两个客户同时从该账户中取款，分别取出 50 元和 100 元。此时，每个取款操作首先要读取账户余额，在其上减去取款额，然后将结果写回。如果甲、乙客户的取款操作并行执行，可能他们读到的余额都是 500 元，取款后系统将分别写回 450 元和 400 元。账户中到底剩下 450 元还是 400 元要视甲、乙两个操作最后写回的结果而定，但实际上这两种结果都是错误的，正确的结果应该是 350 元。为了消除这种情况发生的可能性，数据库系统需要提供并发控制功能。

(4) 数据安全性。其作用是保证数据的安全，防止数据丢失或被窃取、破坏。数据库系统的所有用户并非都可以访问所有数据，例如，用户通过 ATM 自动取款机只能访问自己的账户，而无权访问其他账户。

从文件系统发展到数据库系统是信息处理领域的一个重大变化。在文件系统下，信息处理的传统方式如图 1.2 所示，在此模式下，程序设计处于中心地位，数据只起着服从程

序设计需要的作用；而在数据库模式下，信息处理的方式发生了改变，如图 1.3 所示，数据占据了中心地位，数据库的设计成为信息系统首先要解决的问题，而利用这些数据的应用程序的设计则要围绕既定的数据库结构来进行。

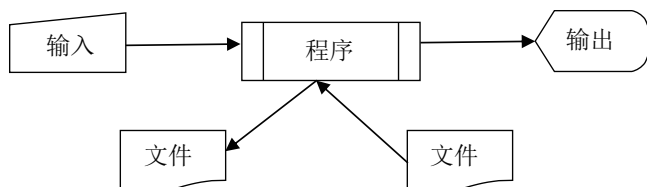


图 1.2 文件系统下的信息处理方式

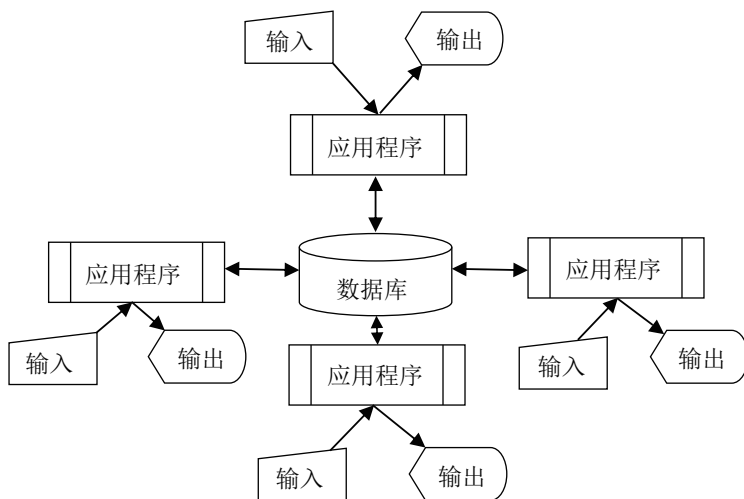


图 1.3 数据库系统下的信息处理方式

1.2 数据模型

1.2.1 数据模型的分类

数据库结构的基础是数据模型，数据模型描述了数据的类型、数据间的联系和施加在数据上的约束条件。大多数数据模型还包括一个对数据库进行检索和更新的基本操作的集合。因此，数据模型所描述的内容有三个方面的内容，分别是数据结构、数据操作和数据约束。

根据不同的应用层次，数据模型可分为概念数据模型、物理数据模型和逻辑数据模型三类。

概念数据模型又称概念模型，其主要用于客观世界的建模，着重于刻画客观世界中复杂事物的结构和相互间的内在联系。概念模型与具体的数据库管理系统和计算机物理实现无关，它在数据库设计中被广泛使用，是数据库设计人员和用户之间进行交流的语言。采

用概念数据模型，数据库设计人员可以在设计的开始阶段把主要精力用于了解和描述现实世界，而把涉及数据库管理系统和物理实现的一些具体问题推迟到后续阶段考虑。常见的概念模型有实体 - 联系模型、扩充的实体 - 联系模型、面向对象模型和谓词模型等。

物理数据模型又称物理模型，它描述的是数据在计算机中实际存储的方式，涉及物理块、索引等概念，它是面向计算机专家而不是面向用户的数据模型。

逻辑数据模型又称逻辑模型，本书后面如果提到数据模型，在没有具体指出是哪类数据模型时一般指的就是逻辑数据模型。逻辑模型介于概念模型与物理模型之间，隐藏了一些数据存储的细节，但可以在计算机中被直接实现。逻辑模型着重数据库系统级的实现。概念模型表示的数据只有被转换为逻辑模型表示的数据后，才能在数据库中得以实现。目前使用最广泛的逻辑数据模型是关系模型。

在关系模型出现以前，层次模型和网状模型这两种数据模型曾被广泛应用过。由于这两种模型和底层的实现联系很紧密，会使数据建模复杂化，因此除了少数情况下，现在已经很少使用它们了。

除此以外，如今还出现了一些新型的逻辑模型，例如，面向对象的数据模型和对象关系数据模型。前者可以被看成是实体 - 联系模型增加了封装、方法和对象标识等特征后形成的扩展；后者结合了面向对象的数据模型和关系数据模型的特征。为阐明数据模型的概念，下面先简单介绍一下最为广泛应用的两类数据模型：实体 - 联系模型和关系模型，后面的章节中再详细介绍这两种模型。

1.2.2 实体 - 联系模型

概念数据模型是对现实世界的第一层次的数据抽象，也是数据库设计员与用户之间交流的语言。表示概念数据模型的方法有多种，但目前最为常用的是实体 - 联系模型（简称 E-R 模型）。

1. E-R 模型中的基本概念

1) 实体 (entity)

客观存在并可相互区别的事物都被称为实体。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，张山、王涛、计算机系、离散数学、教材、教学楼等都是实体。

2) 属性 (attribute)

实体通常具有若干特征，每个特征被称为实体的一个属性。例如，每个学生实体都具有学号、姓名、年龄等属性。属性可以分为基本属性和复合属性两类。基本属性是不可再分的属性，例如，性别、年龄。复合属性是可以再被分解为更细小属性的属性，例如，出生日期就可以进一步被分解为出生年、月、日。

3) 类 (class) 或实体类型 (entity type)

具有相同属性的某一类实体必然具有共同的特征和性质，因此，用实体名及其属性名集合来抽象刻画同类实体，被称为类或实体类型。例如，学生（学号，姓名，性别，年龄）就是一个类或实体类型。

4) 实体集 (entity set)

若干同类实体的集合被称为实体集。

5) 关键字 (key)

能唯一标识实体集中每一个不同实体的属性的集合被称为关键字 (也称码或键)。

6) 域 (domain)

属性的取值范围被称作域。例如, 性别的域为集合 {男, 女}。

7) 联系 (relationship)

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 E-R 模型中反映为实体之间的联系。实体之间的联系又分为实体集之间的联系、实体集内部的联系、子类联系、3 个实体集之间的联系等。

(1) 实体集之间的联系。指一个实体集中的实体与另一个实体集的实体之间的联系, 其包括 3 种, 即一对一联系 (1:1)、一对多联系 (1:n)、多对多联系 (m:n), 其分别叙述如下。

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, 则可称实体集 A 与实体集 B 具有一对一联系, 记为 (1:1), 如图 1.4 所示。

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 中至多有一个实体与之联系, 则可称实体集 A 与实体集 B 具有一对多联系, 记为 (1:n), 如图 1.5 所示。

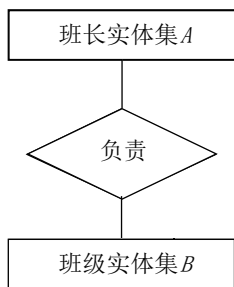


图 1.4 一对一联系 (1:1)

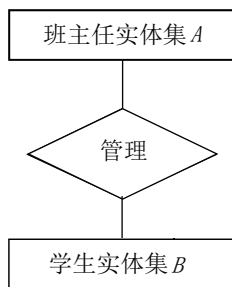


图 1.5 一对多联系 (1:n)

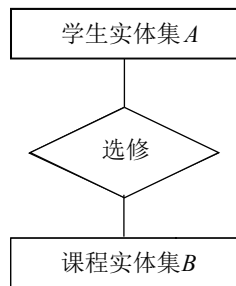


图 1.6 多对多联系 (m:m)

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, 则可称实体集 A 与实体集 B 具有多对多联系, 记为 (m:n), 如图 1.6 所示。

(2) 实体集内部的联系指在同一个实体集内的实体之间的联系。这种联系也同样存在 3 种类型, 即 1:1、1:n 和 m:n 三种类型, 如图 1.7 所示。

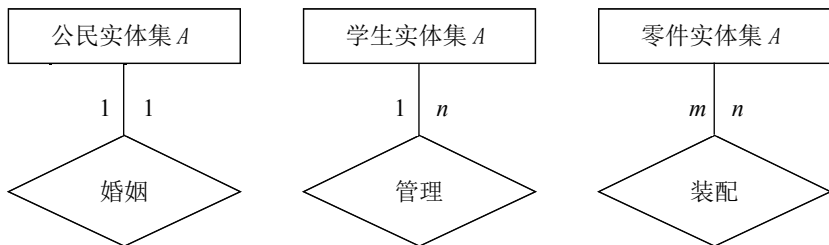


图 1.7 实体集内部 3 种联系

(3) 子类联系 (is-a)。如果实体集 B 继承 (包含) 了实体集 A 中的所有属性, 除此之外, 实体集 B 还具有自己一些特殊的属性, 此时可称 B 的类是 A 的类的子类, 如图 1.8 所示。

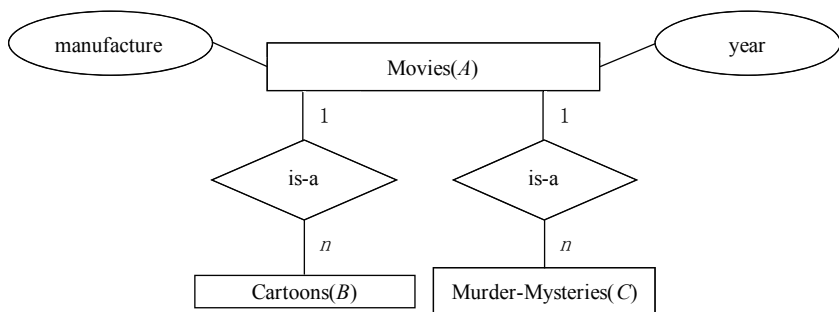


图 1.8 子类联系

图 1.8 中 $Cartoons(B)$ 类和 $Murder-Mysteries(C)$ 类是 $Movies(A)$ 类的子类, 显然实体集 A 与实体集 B 或 C 之间是一对多联系, 这包括了一对一甚至“一对 0”联系。

(4) 3 个实体集之间的联系。如果实体集 A 与实体集 B 存在以上 3 种联系的任意 1 种, 且实体集 C 与实体集 B 也存在以上 3 种联系的任意 1 种, 那么 3 个实体集之间必然存在某种联系, 如图 1.9 所示。

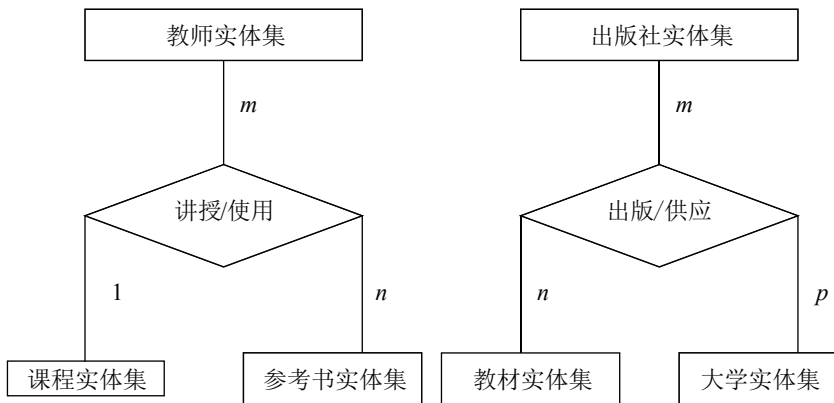


图 1.9 3 个实体集之间的联系

2. E-R 模型图示方法

1) 实体集 (实体类型、类)

用矩形表示, 矩形内写名称。

2) 联系

用菱形表示，菱形内写联系名，并标明联系类型（1:1、1:n 或 m:n 等）。

3) 属性

用椭圆表示，椭圆内写属性名。

例如，教师、学生、课程的 E-R 概念模型，如图 1.10 ~图 1.11 所示。

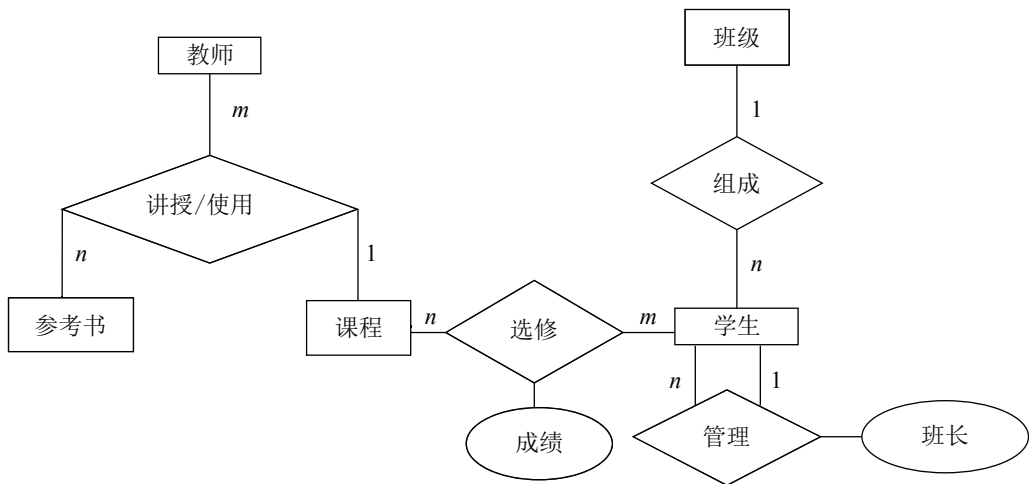


图 1.10 实体集之间联系的 E-R 图

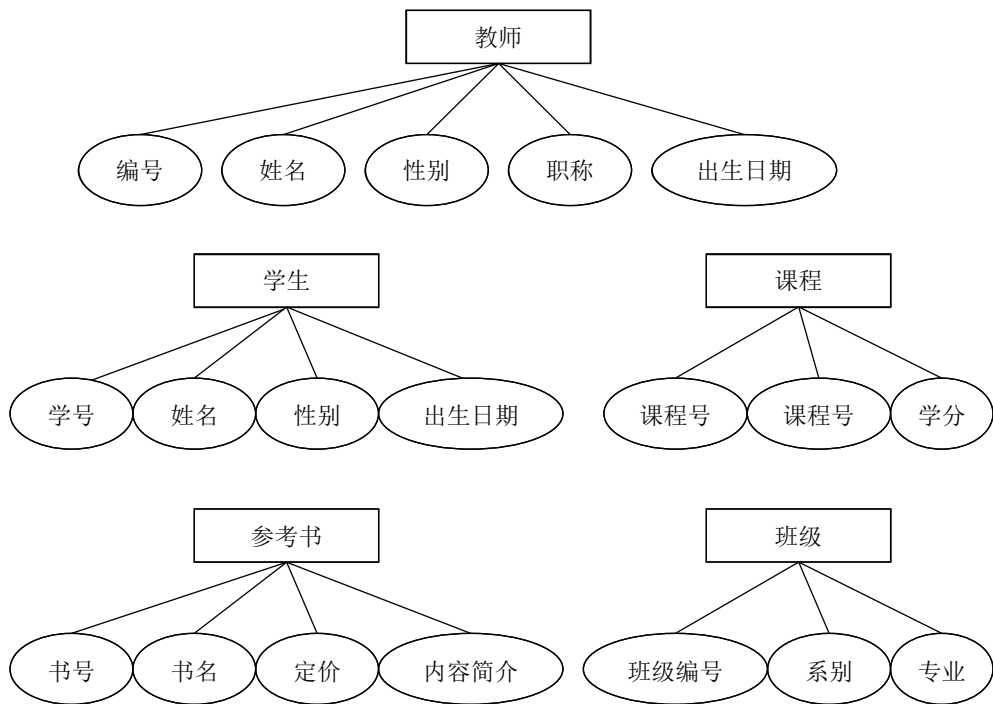


图 1.11 实体集属性 E-R 图

1.2.3 典型的逻辑数据模型

比较典型的逻辑数据模型是层次模型、网状模型和关系模型。现在所使用的数据库99%以上都是建立在关系模型基础上的，近年来面向对象模型也有一定的发展，下面将对前3种模型分别进行介绍。

1. 层次模型 (Hierarchical Model)

层次模型是4种模型中最早出现的，其采用分多个层次的结构来组织数据，以简单、直观地表现信息世界中的各种实体，如图1.12所示。

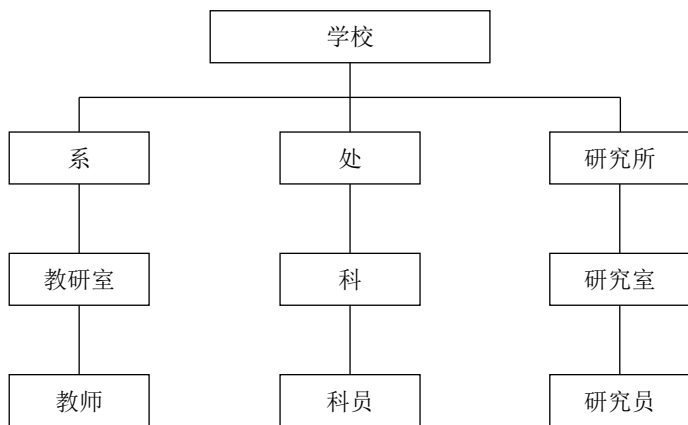


图 1.12 层次模型

层次模型 E-R 图满足以下 2 个性质。

- (1) 该模型中有且只有一个结点，没有父 (parent) 结点，这个结点被称为根结点。
- (2) 除根结点外其他结点有且仅有一个父结点与之相连。

2. 网状模型 (Network Model)

网状模型的特点便是具有网状结构，该模型中的结点可以有多个父结点，结点之间也往往为多对多的关系，如图1.13所示。

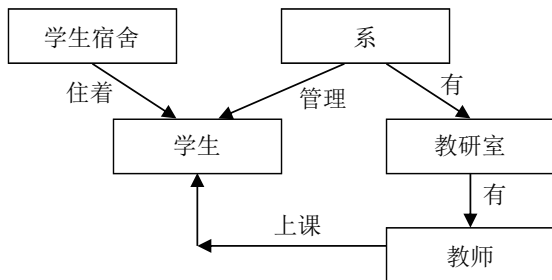


图 1.13 网状模型

网状模型 E-R 图满足以下 2 个性质。

- (1) 该模型允许一个结点没有父结点。
- (2) 该模型中一个结点可以有两个或两个以上的父结点。

实际上，以上定义并不十分严格。在网状模型中，结点并无父结点或子结点的概念。

3. 关系模型 (Relational Model)

关系模型是建立在集合之间“关系”基础上的，从用户角度看该模型是由多个二维表构成的。正是与“关系”这种概念的天生血缘关系，关系模型具有严格的数学理论基础，这也是其得到广泛应用的原因。

在关系模型中，经常用到下列术语。

1) 属性 (attribute) 或字段 (field)

二维表中的一列即为一个属性，其名称为属性名。

2) 关系模式 (relational schema)

对应一个二维表的表头，表示该表的结构。关系模式是对一类实体特征的结构化描述，一般表示为：关系名 (属性 1, 属性 2, ..., 属性 n)。

3) 关系 (relation)

对应一个二维表。

4) 元组 (tuple) 或记录 (record)

对应二维表中的一行。

5) 键、候选键 (candidate key)、码或关键字 (key)

二维表可以唯一确定记录 (元组) 的属性集合。例如，学生数据表中的学号。

一个关系可以有多个键，其中一个会成为主键 (primary key)。

6) 域 (domain)

属性的取值范围。

7) 分量 (attribute value)

元组中的一个属性值。例如，元组 (20080001, 张三, 女, 1990.2.23) 中的“张三”“女”等都是该元组的分量。

关系数据模型是以集合论中的关系概念为基础发展起来的数据模型，它把记录集合定义为一张二维表，即关系。表的每一行是一条记录，表示一个实体；每一列是记录中的一个字段，表示实体的一个属性。关系模型既能反映实体集之间的一对一联系，也能反映实体集之间的一对多和多对多联系。如表 1.1、表 1.2 及表 1.3 就构成了一个典型的关系模型实例。

表 1.1 学生基本情况表

学生学号	学生姓名	学生性别	出生日期	是否团员	学生籍贯	所在班级
3031023101	张 山	男	08/28/84	是	江苏	计应 0231
3031023102	武云峰	男	05/02/83	是	山东	计应 0231
3031023103	孙玉凤	女	12/10/84	否	江苏	计应 0231
1011024101	王加玲	女	10/08/84	是	天津	机电 0241
1011024102	周云天	男	01/02/82	是	山西	机电 0241
1011024103	东方明亮	女	05/01/83	否	天津	机电 0241
1011024104	张洁艳	女	06/30/82	是	山西	机电 0241

表 1.2 课程信息表

课程号	课程名	课程类型	课时数
10001	电子技术	考试	80
10002	机械制图	考查	60
10003	数控机床	选修	50
20001	商务基础	考查	60
20002	会计电算化	考试	68
30001	计算机应用	考查	80
30002	数据库原理	考试	76

表 1.3 学生成绩表

学号	课程号	学期	成绩	学分
3031023101	30001	1	69.5	3
3031023101	30002	2	78.0	5
3031023103	30001	1	90.5	3
3031023103	30002	2	81.0	5
3031023104	30002	2	92.0	5
1011024101	10001	3	74.5	5
1011024101	10002	3	80.0	5

1.3 数据库管理系统的体系结构

1.3.1 数据库管理系统的功能

数据库管理系统是数据库系统的核心，它的主要功能包括以下几个方面。

1. 数据模式定义

数据库管理系统提供数据模式定义语言 DDL (Data Definition Language) 来定义数据库的结构，即为数据库构造数据框架。这既包括数据的逻辑结构，也包括数据的物理结构。

2. 数据操纵

数据库管理系统提供数据操纵语言 DML (Data Manipulation Language) 来实现数据的查询、插入、删除和修改。

3. 数据控制

数据库管理系统提供数据控制语言 DCL (Data Control Language) 来实现数据的完整性、安全性定义与检查，以及数据的并发控制和故障恢复。

4. 数据维护

数据库管理系统还提供一些实用程序 (utilities) 来实现数据的拷贝、转储、重组和性能监测分析等。

1.3.2 数据库管理系统体系结构

数据库管理系统总是基于某种数据模型，其可以分为层次型、网状型、关系型和面向对象型等。虽然数据库管理系统多种多样，需要在不同的操作系统支持下工作，但是绝大多数数据库管理系统在总的体系结构上都采用三级模式结构，即物理模式、逻辑模式和用户模式，并且提供两级映像功能，即用户模式 / 逻辑模式映像和逻辑模式 / 物理模式映像，如图 1.14 所示。

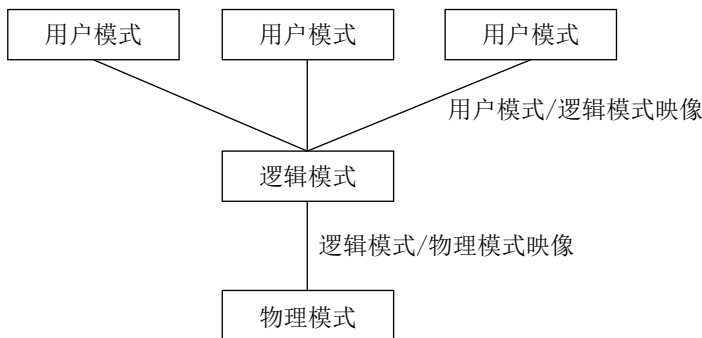


图 1.14 数据库管理系统体系结构

1.3.3 三级模式

随着时间的推移，信息会被插入或删除，数据库也就发生了改变。特定时刻存储在数据库中的信息的集合被称作数据库的一个实例。数据库的总体结构被称为数据库模式。数据库模式即使发生变化，也是不频繁的。

数据库管理系统的三级模式体系结构表达了数据的三个抽象层次，物理模式在物理层描述了数据库的结构；逻辑模式在逻辑层描述了数据库的结构；用户模式在用户层描述了数据库的结构。数据库管理系统通过这几个层次上的抽象来对用户隐藏复杂性，简化系统的用户界面。

1. 物理模式

物理模式也称内模式，它描述数据的物理存储方式，详细地描述复杂的底层数据结构。例如，数据存储的文件组织方式、索引方式，是否压缩存储等。

2. 逻辑模式

逻辑模式也称概念模式，它是对数据库中全部数据的整体逻辑结构的描述，是所有用户的公共数据视图。它描述数据库中存储了什么数据，以及这些数据间存在什么联系。例如，数据记录由哪些数据项组成，数据项的名称、类型、取值范围等，数据之间的联系，以及数据的安全性、完整性要求等。

逻辑模式用相对简单的结构来描述整个数据库，用户只需要使用逻辑模式即可，不需要关心隐藏在逻辑模式下的复杂的物理模式。

3. 用户模式

用户模式也称外模式，它是用户的数据视图，只描述数据库的某个部分。用户模式由逻辑模式导出，一个逻辑模式可以导出若干个用户模式。逻辑模式给出数据库系统全局的数据描述，而用户模式则给出面向每个用户的局部描述。

尽管逻辑模式使用了比较简单的结构，但由于数据库规模巨大，因此其仍存在一定程度的复杂性。数据库系统的很多用户并不需要关心数据库的整个逻辑模式，而只需要关心与其有关的模式。一方面，用户模式可以屏蔽大量无关的信息，使用户与数据库系统的交互更简单。另一方面，用户模式也有利于保护数据。每个用户只能看见和访问其自身所对

应的用户模式中的数据，数据库中的其余数据对其而言是不可见的。例如，银行的出纳员只能看见数据库中关于客户账户的信息，而不能访问客户的密码等信息。

1.3.4 两级映像与数据独立性

数据库管理系统的三级模式体系结构是数据的三个抽象层次，它把数据具体的物理组织方式留给物理模式，使用户不必关心数据在计算机中的具体存储实现。三级模式间的联系与转换是通过两级映像实现的，即用户模式 / 逻辑模式映像和逻辑模式 / 物理模式映像。同时，两级映像还保证了数据库系统中数据的独立性。

1. 用户模式 / 逻辑模式映像与数据的逻辑独立性

用户模式 / 逻辑模式映像给出了用户模式与逻辑模式的对应关系，该映像由数据库管理系统提供定义机制。

当逻辑模式改变时，如增加了新的数据项，那么系统可以通过 DBMS 对用户模式 / 逻辑模式的映像作相应改变，使用户模式保持不变，从而使基于用户模式的应用程序保持不变，这保证了数据与应用程序的逻辑独立性，也即数据的逻辑独立性。

2. 逻辑模式 / 物理模式映像与数据的物理独立性

物理模式描述的是数据的物理存储结构，逻辑模式描述的是数据的全局逻辑结构。一个数据库只有一个物理模式和一个逻辑模式。逻辑模式 / 物理模式映像给出了逻辑模式与物理模式的对应关系，该映像由数据库管理系统提供定义机制。

当物理模式改变时，例如，服务器系统改变了文件的组织方式，可以通过 DBMS 对逻辑模式 / 物理模式的映像作相应改变，使逻辑模式保持不变，从而使应用程序保持不变，这保证了数据与应用程序的物理独立性，也即数据的物理独立性。

数据与程序间的独立性保证了用户模式的稳定性，进而保证了应用程序的稳定性，使应用程序的编写不必考虑数据复杂的整体结构，更不必考虑数据复杂的存储细节，这大大降低了应用程序的维护与开发费用。

1.4 设计、管理和使用数据库的人员

对一个小型的个人数据库来说，一般由用户自己来定义、操纵和管理数据库。但是，对于拥有上百名用户的大型数据库来说，其需要很多人共同参与设计、使用和维护。这些人员按其工作的不同可以分为数据库设计人员、数据库管理员和数据库用户。

1.4.1 数据库设计人员

数据库设计人员的职责是识别存储于数据库中的数据、选择适当的结构来表示和存储

这些数据并设计出数据库的各级模式。数据库设计人员必须参加用户需求调查和系统分析，然后提出合理的设计方案。通常情况下，数据库设计人员是数据库管理员的候选者。

1.4.2 数据库管理员

大型数据库系统，需要有专门的管理人员来监督和管理数据库、数据库管理系统及其相关软件，这样的人员被称为数据库管理员 DBA (database administrator)。DBA 的职责包括以下内容。

(1) 定义逻辑模式。DBA 通过 DBMS 提供的逻辑模式 DDL 来创建最初的数据库逻辑模式，决定数据库中要存放的信息内容和结构。

(2) 定义物理模式。DBA 通过 DBMS 提供的物理模式 DDL 来定义数据库的存储结构和存取方式，以获得较高的空间利用率和存取效率。

(3) 授予存取权限。DBA 的重要职责是保证数据库的安全性和完整性。DBA 通过数据控制语言 DCL 规定各个用户对数据库的存取权限以及数据的完整性约束条件。

(4) 日常维护。DBA 通过数据库管理系统提供的实用程序进行的日常维护活动有以下两点。

①定期备份数据库，从而在系统发生故障时将数据库恢复到正确状态。

②监视数据库的运行，包括系统的空间使用情况、处理效率等性能指标，以保证良好的系统性能。DBA 可能要根据实际运行情况进行诸如数据的重组织、重新设计，以及磁盘空间升级等工作。

1.4.3 数据库用户

数据库用户是满足查询、更新以及产生报表等需要而访问数据库的人员。数据库主要是为了满足他们的使用而存在的。数据库系统的用户可以分为以下几类。

(1) 初级用户。这类用户在数据库用户中占有相当的比率，他们的主要工作是经常性地查询和修改数据库，一般都是通过应用程序存取数据库。例如，银行出纳员使用数据库进行存款和取款操作。

(2) 应用程序员。应用程序员是编写应用程序的计算机专业人员，这类人员应该熟悉 DBMS 所能提供的各个方面的功能，并充分利用这些功能完成他们的任务。

(3) 资深用户。资深用户包括工程师、科学家、商业分析师等全面了解其领域知识的人员。他们一般都比较熟悉 DBMS 的功能，能够直接使用数据库语言访问数据库，可以在 DBMS 的帮助下满足其复杂需求。

本章小结

本章介绍了数据库的基本概念。数据是客观世界的现象与事物在计算机中的抽象，其通

常是指可以用符号记录下来的对事物的描述。数据库是数据的集合，它具有统一的结构形式，存放于统一的存储介质内，并由统一机构管理。数据库管理系统是统一管理数据库的一种系统软件。数据库系统是指引入数据库技术后的计算机系统，它是一个可实际运行的，向应用系统提供支撑的系统。数据库应用系统利用数据库系统作应用开发，以满足用户的应用需求。

相比文件系统，数据库系统提供了对数据更有效的管理。其不仅包含数据库本身，还定义了数据库的结构，即具有自描述特征，从而使数据能面向整个系统。数据库系统具有高共享性与低冗余性，具有较高的数据独立性，能够为数据提供统一管理与控制，支持数据完整性、数据库可恢复性、数据并发控制和数据安全性。

数据模型是数据库结构的基础，能够描述数据结构、定义其上操作及约束条件。数据模型分三个层次：概念模型、逻辑模型和物理模型。概念模型是一种面向客观世界和用户的模型，与具体的数据库管理系统及具体的计算机平台无关，常见的概念模型有实体-联系模型、扩充的实体-联系模型、面向对象模型及谓词模型等。逻辑模型是一种面向数据库系统的模型，着重于在数据库系统一级的实现，较为成熟的逻辑模型有：层次模型、网状模型、关系模型、面向对象模型以及对象关系模型等。物理模型是一种面向计算机物理表示的模型，它给出了数据模型在计算机上物理结构的表示方法。

数据库管理系统是数据库系统的核心，其功能包括：数据模式定义功能，数据操纵功能，数据控制功能和数据维护功能。虽然数据库管理系统多种多样，但是绝大多数系统在总的体系结构上采用三级模式结构，即逻辑模式、物理模式和用户模式，并且提供两级映像功能，即用户模式/逻辑模式映像和逻辑模式/物理模式映像。数据库管理系统通过三级模式使用户和全局设计者不必关心数据库的具体实现与物理背景，从而隐藏了复杂性，简化了系统的用户界面。两级映像保证了数据库系统中数据的独立性。

针对小型数据库，用户一般自己来定义、操纵和管理数据库。但是，针对大型数据库，就需要很多人共同参与设计、使用和维护的工作。这些人员按工作的不同可以分为数据库管理员、数据库设计人员和数据库用户。

练习与思考

- 1.1 试解释下列术语：数据库，数据库管理系统，数据库系统，数据库应用系统。
- 1.2 试述数据库管理系统与文件系统的联系与区别。
- 1.3 什么是数据模型？它分哪几种类型？
- 1.4 试解释下列概念：概念模型、逻辑模型、物理模型、用户模式、逻辑模式、物理模式。
- 1.5 数据库管理系统的功能有哪些？
- 1.6 为什么数据库要实现三级模式体系结构？
- 1.7 试解释下列概念：用户模式/逻辑模式映像、逻辑模式/物理模式映像。
- 1.8 试述逻辑模式在数据库中的重要地位。
- 1.9 什么是数据独立性？
- 1.10 什么是 DBA？DBA 的职责是什么？