第 1 章 进入数据库技术

数据库技术诞生于 20 世纪 60 年代末期。经过几十年的发展，和数据库相关的理论研究和技术应用都有了非常大的发展。数据库技术是信息科学中一门重要的技术，在政府、企业等机构得到了广泛的应用。特别是 Internet 技术的发展，为数据库技术开辟了更广泛的应用舞台。

当前，商业的自动化和智能化，使得企业收集到了大量的数据，积累下来重要的资源。政府、企业等各类组织需要对大量的数据进行管理，从数据中获取信息和知识，从而进行决策。在 20 世纪 80 年代，美国信息资源管理学家霍顿（F. W. Horton）和马钱德（D. A. Marchand）等人指出：信息资源与人力、物力、财力和自然资源一样，都是企业的重要资源，因此，应该像管理其他资源那样管理信息资源。如今，数据库的建设规模、数据库信息量的规模及使用频度已成为衡量一个企业、组织乃至一个国家信息化程度的重要标志。本章重点介绍数据库系统的基本概念和数据库设计的步骤。

学习目标

- 了解数据库的基本概念
- 了解数据库的特点
- 熟悉数据库的发展历史
- 掌握数据库的三级数据模式结构
- 熟悉数据库的概念模型及 E-R 图
- 熟悉数据库的逻辑模型、数据模型
- 熟悉关系代数
- 了解数据库设计的基本步骤

1.1 导言

在人们的日常生活中，数据库技术已经广泛应用于各行业中。现通过一个超市连锁的示例，讨论为什么需要数据库。

假设某一超市连锁集团，在全国 50 多座大中型城市拥有 150～200 家超市，每个超市有 200～300 名工作人员，销售超万种商品。

那么，在管理这家公司时，需要掌握哪些信息？

首先，公司的管理层需要随时掌握各分店的进货情况、销货情况和库存情况；需要了解不同商品的供货市场情况，并对供货商进行维护；需要了解不同商品在不同地域的销售情况，以便及时调整销售策略；需要解决销售过程中的各种问题，对客户进行维护。在此过程中，公司还需要对员工的工作业绩进行考核。

随着市场规模的不断扩大，公司业务量的迅速增长，公司就必须有效地管理商品、供货商、客户和员工等数据，并且这类数据正在不断地积累和增长。

这样大量的数据已经不适应靠人工管理，比较好的方法是用数据库系统来管理。产生数据库的动因和使用数据库的目的正是为了及时地采集数据、合理地存储数据、有效地使用数据，保证数据的准确性、一致性和安全性，在需要的时间和地点获得有价值的信息。

那么，应该如何去管理数据、组织数据并有效地使用数据，从中得到有价值的信息呢？这正是本教材讨论的问题。

通常情况下，数据库技术所要解决的基本问题如下。

- 如何抽象现实世界中的对象，如何表达数据以及数据之间的联系。
- 如何方便、有效地维护和利用数据。

1.2 数据库系统

数据库就是数据的集合。一个数据库系统（Data Base System，简称 DBS）其主要组成部分是数据、数据库、数据库管理系统、应用程序及用户。数据存储在数据库中，用户和用户应用程序通过数据库管理系统对数据库中的数据进行管理和操作。

1.2.1 数据库系统的基本概念

1. 数据与信息

（1）数据（Data）是对客观事物的抽象描述，是用于承载信息的物理符号。也就是说，数据是信息的具体表现形式，信息包含在数据之中。数据的形式或者说数据的载体是多种多样的，它们可以是数值、文字、图形、图像、声音等。在计算机领域中，数据是指所有能输入到计算机并被计算机程序处理的数字、字母、符号和模拟量等的总称。比如，

80 是一个数据，可以是一名同学某门课的成绩，也可以是某个人的体重，还可以是一个班级的学生人数。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）信息（Information）是对客观世界中各种事物属性和运动状态的反映，是客观事物之间相互联系和相互作用的表征，表现的是客观事物运动状态和变化的实质内容。在通信和控制系统中，信息是一种普遍联系的形式。人通过获得、识别自然界和社会的不同信息来区别不同事物，得以认识和改造世界。

信息是有价值的，可以被感知的。信息可以通过载体传递，也可以通过信息处理工具进行存储、加工、传播、再生和增值。因此，信息是一种重要的资源。

数据和信息既有联系，又有区别。数据是信息的表现形式和载体；而信息是数据的内涵，是加载于数据之上、对数据做具有含义的解释。数据和信息是不可分离的，信息依赖数据来表达，数据则生动具体地表达出信息。数据本身没有意义，数据只有对实体行为产生影响时才成为信息。

2. 数据处理

数据处理（Data Processing）是从大量的原始数据中抽取出有价值的信息，也可以说是数据转换成信息的过程。数据处理的主要内容包括对数据的收集、存储、加工、分类、归并、计算、排序、转换、检索和传播等。数据处理的基本目的是从大量的、可能是杂乱无章的、难以理解的数据中抽取并推导出对于某些特定的人来说是有价值的、有意义的数据。

数据处理与数据管理是互相联系的，数据管理技术的优劣将对数据处理的效率产生直接影响。而数据库技术就是针对该需求目标进行研究并发展和完善起来的一个计算机应用技术的分支。

数据的三个表示范畴分为现实世界、信息世界和计算机世界。数据库设计的过程就是将数据的表示从现实世界抽象到信息世界（概念模型），再从信息世界转换到计算机世界（数据模型）。

3. 数据库

数据库（DataBase，DB）是长期存储在计算机内、有组织、可共享的数据集合，可以形象地理解为存储数据的仓库。数据库中的数据是以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展性，并可为多个用户共享。概括地讲，数据库中的数据具有永久存储、有组织和可共享三个特点。

数据库有很多种类，从存储各种数据表格的简单数据库到能够进行海量数据存储的大型数据库系统，在各方面都得到了广泛的应用。在信息化社会，充分有效地管理和利用各类信息资源，是进行科学研究和决策管理的前提条件。

4. 数据库管理系统

数据库管理系统（DataBase Management System，简称：DBMS）是一操纵和管理数据库的大型系统软件，能够科学地组织和存储数据、高效地获取和维护数据的环境。一般由

软件厂商提供，比如 Microsoft 公司的 SQL Server 和 Access 等。那么，数据库管理系统的主要功能如下。

（1）数据定义。数据库管理系统（DBMS）提供数据定义语言（Data Definition Language，DDL）。用户通过 DDL 可以对数据库中的数据对象进行定义。

（2）数据操作。数据库管理系统（DBMS）提供数据操作语言（Data Manipulation Language，DML）。通过使用 DML，用户可实现对数据的追加、删除、更新、查询等操作。

（3）数据库的建立和维护功能。数据库的建立和维护功能主要包括数据库初始数据的输入、转换，数据库的存储、恢复，数据库的重组、性能监视和分析等。

（4）数据库的运行管理。数据库系统的正常运行是由 DBMS 统一管理和控制的，以保证数据的安全性、完整性、并发性及发生故障后的系统恢复等。

（5）有效存取数据库信息的接口和工具。编程人员可通过程序开发工具与数据库接口编写数据库应用程序。数据库管理员可通过相应的软件工具对数据库进行管理。

5. 数据库系统

DBMS 是数据库系统的核心。一个完整的数据库系统，由保存数据的数据库、数据库管理系统、用户应用程序和用户组成。

1.2.2 数据库系统的特点

数据库系统的特点主要表现在以下几个方面：

1. 数据结构化

数据库中的数据是结构化的。这种结构化就是数据库管理系统所支持的数据模型。使用数据模型描述数据时，不仅描述了数据本身，同时描述了数据之间的联系。按照应用的需要，建立一种全局的数据结构，从而构成了一个内部紧密联系的数据整体。比如，关系数据库管理系统支持关系数据模型，关系模型的数据结构是“关系”满足一定条件的二维表。

2. 数据高度共享、低冗余度、易扩充

数据的共享度直接关系到数据的冗余度。数据库系统从整体看待和描述数据，数据不再面向某个应用，而是面向整个系统。因此，数据库中的数据可以高度共享。数据的高度共享本身就减少了数据的冗余，同时确保了数据的一致性，同一数据在系统中的多处引用是一致的。

3. 数据独立

数据的独立性是指数据库系统中的数据与应用程序之间是互不依赖的。数据库系统提供了两方面的映射功能，从而使数据既具有物理独立性，又具有逻辑独立性。物理独立性是指用户的应用程序与存储在磁盘上数据库中的数据是相互独立的；逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的。也就是说，数据的逻辑结构变了，用户程序也可以不变。数据独立性是数据库的一种特征和优点，它有利于在数据库结构改变时能保持应用程序尽可能不改变或少改变，以减少应用人员的开发工作量。

4. 数据由数据库管理系统统一管理和控制

DBMS 提供以下几方面的数据管理与控制功能。

（1）数据安全性。数据安全性（Security）是指保护数据，防止不合法使用数据造成数据的泄密和破坏，要求每个用户只能按规定权限对某些数据以某种方式进行访问和处理。比如，部分用户对学生成绩只能查阅不能修改。

（2）数据完整性。数据完整性（Integrity）是指数据的正确性、有效性、相容性和一致性，即将数据控制在有效的范围内，或要求数据之间满足一定的关系。

（3）并发控制。当多用户的并发（Concurrency）进程同时存取、修改数据库时，可能会发生相互干扰而得到错误的结果，并使得数据库的完整性遭到破坏，因此必须对多用户的并发操作加以控制和协调。

（4）数据库恢复。计算机系统的硬件故障、软件故障、操作员的失误以及故意的破坏都会影响数据库中数据的正确性，甚至造成数据库部分或全部数据的丢失。DBMS 必须具有将数据库从错误状态恢复到某一已知的正确状态的能力，这就是数据库的恢复功能。

1.2.3 数据管理技术的产生和发展

数据库技术是应数据管理任务的需要而产生的。计算机数据管理随着计算机硬件、软件技术和计算机应用范围的发展而不断发展，经历了人工管理阶段、文件系统阶段和数据库系统阶段。

1. 人工管理阶段

20 世纪 50 年代中期以前，计算机主要用于数值计算。从硬件系统看，当时的外存储设备只有纸带、卡片、磁带，没有直接存取设备；从软件系统看，没有操作系统以及管理数据的软件；从数据看，数据量小，数据无结构，由用户直接管理，且数据间缺乏逻辑组织，数据依赖于特定的应用程序，缺乏独立性。人工管理数据阶段的特点如下。

（1）不能单独保存数据。由于当时的计算机主要用于科学计算，因而一般不需要将数据长期保存。数据与程序不独立（是一个整体），数据只为本程序所使用，并且数据只有与相应的程序一起保存才有价值。

（2）应用程序管理数据。数据需要由应用程序自己管理，没有相应的软件系统负责数据的管理工作。在程序中要规定数据的逻辑结构和物理结构，并且应用程序只包含自己所要用到的全部数据。用户编制程序时，必须全面考虑好相关的数据，包括数据的定义、存储结构以及存取方法等。

（3）数据不能共享。数据是面向应用的，不同的程序均有各自的数据，这些数据对不同的程序通常是不相同的，不可共享。即使不同的程序使用了相同的一组数据，这些数据也不能共享，程序中仍然需要各自加入这组数据，哪个部分都不能省略。基于这种数据的不可共享性，必然导致程序与程序之间存在大量的重复数据，浪费存储空间。

2. 文件系统阶段

20 世纪 50 年代后期到 60 年代中后期，计算机应用从科学计算发展到数据处理。1954

年出现了第一台商业数据处理的电子计算机 UNIVACI，标志着计算机开始应于加工数据为主的事务处理阶段。基于计算机的数据处理系统也从此迅速发展起来。这个阶段，硬件系统出现了磁鼓、磁盘等直接存取数据的存储设备；软件系统有了文件系统，处理方式也成批处理发展到了联机实时处理。那么，文件系统阶段的数据管理特点如下。

（1）数据可以长期保存。数据能够保存在存储设备上，可以对数据进行各种数据处理操作，包括查询、修改、增加、删除操作等。

（2）文件系统管理数据。数据以文件形式存储在存储设备上，有专门的文件系统软件对数据文件进行管理，应用程序按文件名访问数据文件，按记录进行存取，可以对数据文件进行数据操作。应用程序通过文件系统访问数据文件，使得程序与数据之间具有一定的独立性。

（3）数据共享差、数据冗余大。一个应用程序对应一个数据文件，即使在多个应用程序需要处理的部分有相同的数据时，也必须访问各自的数据文件，由此造成数据冗余，并可能导致数据不一致，数据不能正常共享。

（4）数据独立性差。数据文件与应用程序一一对应，数据文件改变时，应用程序就需要改变；同样，应用程序改变时，数据文件也需要改变。

3. 数据库系统阶段

20 世纪 60 年代后期以来，随计算机在数据管理领域的普遍应用，人们对数据管理技术提出了更高的要求：希望面向企业或部门，以数据为中心管理数据；减少数据的冗余，增强数据共享能力；同时要求程序和数据具有较高的独立性，当数据的逻辑结构改变时，不涉及数据的物理结构，也不能影响应用程序，以降低应用程序研制与维护的费用。数据库技术也正是在这样一个应用需求的基础上发展起来的。数据库管理系统管理数据具有的特点如下。

（1）数据结构化。采用数据模型表示复杂的数据结构。数据模型不仅描述数据本身的特征，还要描述数据之间的联系。数据联系是数据库与传统文件的根本区别。这样，数据不再面向特定的某个或多个应用，而是面向整个应用系统。如面向企业或部门，以数据为中心组织数据，形成综合性的数据库，为各应用所共享。

（2）数据的共享性高，冗余度低，易扩充。不同的应用程序根据处理要求，从数据库中获取需要的数据，这样就减少了数据的重复存储，也便于增加新的数据结构，维护数据的一致性。

（3）程序和数据有较高的独立性。数据的逻辑结构与物理结构之间的差别可以很大，用户以简单的逻辑结构操作数据而无须考虑数据的物理结构。

（4）由数据库管理系统统一管理数据。数据库管理系统（DBMS）对数据进行统一管理和控制，以保证数据的安全性、完整性，同时提供并发控制；具有良好的用户接口，用户可方便地开发和使用数据库。

1.3 数据库系统的三级模式结构

人们为数据库设计了一个严谨的体系结构，数据库领域公认的标准结构是三级模式结构，它包括外模式、模式（概念模式）和内模式，它有效地组织、管理数据，提高了数据库的逻辑独立性和物理独立性。用户级对应外模式，概念级对应模式（概念模式），物理级对应内模式，使不同级别的用户对数据库形成不同的视图（关于视图的概念在第 7 章详细阐述）。关于数据库系统的三级模式结构，如图 1-1 所示。

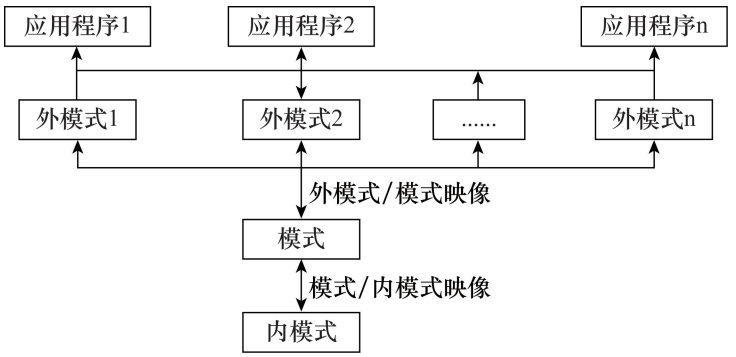

图 1-1 数据库系统的三级模式结构

1.3.1 数据模式的概念

1. 模式

模式也称概念模式或逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是对数据结构和属性的描述。概念模式是系统为了减小数据冗余、实现数据共享的目标，并对所有用户的数据进行综合抽象而得到的统一的公共数据的视图。一个数据库只能有一个概念模式，以概念模式为框架的数据库为概念数据库。比如，关系数据库用关系数据模型来描述数据的概念结构和数据之间的联系，以及数据的完整性规则。在关系数据模型中，对员工数据的一组描述就是一个模式，这个模式可以有多组不同的值与其对应，每一组对应的值称为模式的实例，比如，（2023092215，小王，男，销售部）就是上述模式的一个实例。

2. 外模式

外模式也称子模式或用户视图，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概念结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的概念表示。因此，外模式是模式的子集，一个数据库可以有多个外模式。外模式能很好地起到保护数据的作用，是数据库数据安全的一个有力措施。外模式使得每个用户只能访问到与其相关的数据，不能看到模式中的其他数据。

3. 内模式

内模式也称存储模式或物流模式，一个数据库只有一个内模式。它是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。其设计目标是将系统的模式组织成最优的物理模式，以提高数据的存取效率，改善系统的性能指标。以物理模式为框架的数据库称为物理数据库。

在数据库系统中，只有物理数据库才是真正存在的，它是存放在外存储器的实际数据文件。概念数据库、用户数据库和物理数据库三者的关系是：概念数据库是物理数据库的逻辑抽象形式；物理数据库是概念数据库的具体实现；用户数据库是概念数据库的子集，也是物理数据库子集的概念描述。

1.3.2 数据库系统的三级模式之间的映射

为了能够在内部实现数据库的三个抽象层次的联系和转换，数据库管理系统的三级模式之间提供了两层映射，分别为外模式/模式映射和模式/内模式映射。二级映射技术不仅在三级模式之间建立了联系，也保证了数据库系统中数据的逻辑独立性和物理独立性。关于两层映射的具体表述如下。

1. 外模式/模式映射

对于同一个模式可以有任意多个外模式，如图 1-1 所示。对于每一个外模式，数据库系统都有一个外模式/模式映射。当模式改变时，由数据库管理员对各个外模式/模式映射作相应的改变，可以使外模式保持不变。这样，依据数据外模式编写的应用程序就不用修改，保证了数据与程序之间的逻辑独立性，简称数据的逻辑独立性。

2. 模式/内模式映射

数据库中只有一个模式和一个内模式，所以模式/内模式映射是唯一的，它定义了数据库的全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时，由数据库管理员对模式/内模式映射做相应改变，可以使模式保持不变，应用程序相应的也不做变动。这样，保证了数据与程序的物理独立性。

1.4 概念模型

数据库概念模型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。

1.4.1 数据的表示范畴和描述

数据有三个表示范畴包括：现实世界、信息世界和计算机世界。

1. 现实世界

现实世界是存在于人脑之外的客观世界，事物及其相互联系就处于现实世界之中，它是可感知的世界。通过对现实世界的了解和认识，使得人们对要管理的对象、过程和方法有个概念模型。认识信息的现实世界并用概念模型加以描述的过程称为系统分析。

2. 信息世界

信息世界是现实世界在人们头脑中的反映，又称为观念世界。客观事物在信息世界中称为实体，反映事物间联系的是实体模型或概念模型。现实世界是物质的，相对而言信息世界是抽象的。

3. 计算机世界

信息世界中的信息，经过数字化处理形成计算机能够处理的数据，就进入了计算机世界。由于信息的表示方法和信息处理能力受到计算机硬件和软件的限制，因此所建立的数据模型应符合具体的计算机硬件系统和数据库管理系统（DBMS）的要求。

建立一个数据库系统，首先要深入到现实世界中进行系统需求分析，用概念模型真实地、全面地描述现实世界中的管理对象及联系，然后通过一定的方法将概念模型转换为数据模型，变成计算机能够处理的数据。它们之间的转换关系，如图 1-2 表示。

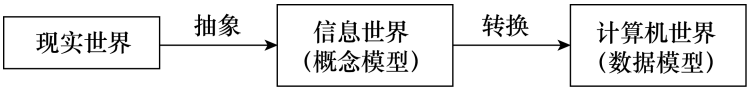

图 1-2　现实世界、信息世界和计算机世界的联系和转换

1.4.2 实体—联系模型

概念模型的表示方法有很多，其中使用最广泛的是实体-联系模型（Entity-Relationship Model），简称 E-R 模型。E-R 模型由实体集、属性和联系构成。

1. 实体—联系模型中的基本概念

关于 E-R 模型中的几个基本概念介绍如下。

（1）实体：现实世界中客观存在并可相互区别的事物称为实体（Entity）。实体是现实世界中的对象，可以是具体的人、事或物。例如，一名教师、一位学生、一台电脑都可称为实体。

（2）属性：实体所具有的某一特性称为属性（Attribute）。在 E-R 模型中，一个实体可以由若干个属性来描述。例如，可以用“产品名称”“型号”“材质”和“尺寸”等属性描述某种商品。这些属性的集合（产品名称，型号，材质，尺寸）表示了一个产品的部分特性。一个实体通常具有多种属性，应该使用哪些属性描述实体，取决于实际问题的需要或者最终期望得到哪些信息。那么，确定属性的两条原则如下：

① 属性必须是不可分的最小数据项，属性中不能包含其他属性，不能再具有需要描

述的性质。

② 同一属性不能与其他多个实体具有联系，在 E-R 图中所表示的联系是实体集之间的联系。

（3）实体集：具有相同属性的实体的集合称为实体集（Entity Set/Entity Class）。例如，商品是一个实体集。实体属性的每一组取值代表一个具体的实体。每个实体集只能表现一个主题。例如，学生实体集中不能包含教师，它们所要描述的容是有差异的，属性可能会有所不同。

（4）键：在描述实体集的所有属性中，可以唯一地标识每个实体的属性称为键（Key）或标识（Identifier）。首先，键是实体的属性；其次，这个属性可以唯一地标识实体集中的每个实体。因此，作为键的属性取值必须唯一且不能“空置”。例如，在学生实体集中，用学号属性唯一地标识每个学生实体。在学生实体集中，学号属性取值唯一，而且每一位学生一定有一个学号。因此，学号是学生实体集的键。每个实体集有一个键属性，其他属性只依赖键属性而存在。例如，学生实体中，学号属性值决定了姓名、性别、出生日期等属性的取值（记为：学号→姓名→性别→出生日期），反之则不行。

（5）实体型：具有相同特征和性质的实体一定具有相同的属性。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（Entity Type）。实体型的格式表示为：

```
实体名(属性 1,属性 2,...,属性 n)
```

例如，产品（产品编号，产品名称，产品型号，材质，尺寸，颜色，重量）就是一个实体型，其中带有下划线的属性是键。

（6）联系：现实世界上任何事物都不是孤立存在的，事物的内部和事物之间都是有联系（Relationship）的。实体集内部的联系体现在描述实体的属性之间的联系；实体集外部的联系是指实体集之间的联系，并且这种联系可以拥有属性。

2. 实体—联系模型的表示方法

在 E-R 图中提供了表示实体集、属性和联系的方法。

（1）实体集：用矩形表示，矩形框内写明实体集名称。

（2）属性：用椭圆形表示，并用无向边将其与相应的实体集连接起来。其中，作为键的属性用加下划线的方式进行表示。

例如，会员（会员 ID，用户名，密码，联系电话，会员积分，……）的 E-R 图，如图 1-3 所示。

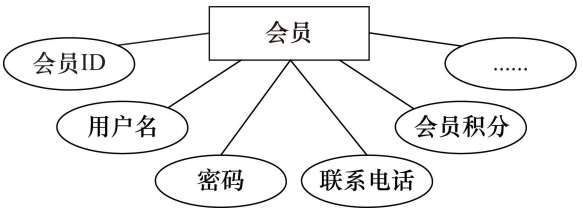

图 1-3　会员实体集的图形表示

（3）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集连接起来，同时在无向边旁标注上联系的类型。

3. 实体集之间的联系类型

实体集之间的联系通常有 3 种类型：一对一联系（1∶1）、一对多联系（1∶n）和多对多联系（m∶n）。

（1）一对一联系

对于实体集 A 中的每一个实体，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 具有一对一联系，记为 1∶1，如图 1-4 所示。

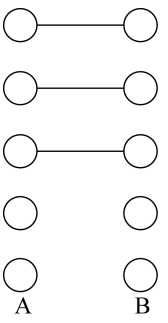

图 1-4 一对一联系示意图

【例 1-1】某某职业大学人力资源部门需要对各学院的工作人员进行管理。如果给定的需求分析如下，则建立此问题的 E-R 模型。

需求分析：

① 每个学院有一名院长，每位院长只在一个学院任职。

② 需要存储和管理的学院信息有学院名称、办公地址、电话。

③ 需要存储和管理的院长信息有姓名、性别、出生日期、职称。

这个问题中有两个实体对象，即学院实体集和院长实体集。描述学院实体集的属性是学院名称、办公地址和电话；描述院长实体集的属性是姓名、性别、出生日期和职称。但两个实体集中没有适合作为键的属性，因此为每一个学院编号，使编号能唯一地标识每一个学院；添加工号作为唯一地标识每一位院长的属性。

E-R 模型：

① 实体型。学院（学院编号，学院名称，办公地址，电话）院长（工号，姓名，性别，出生日期，职称）

② E-R 图，如图 1-5 所示。

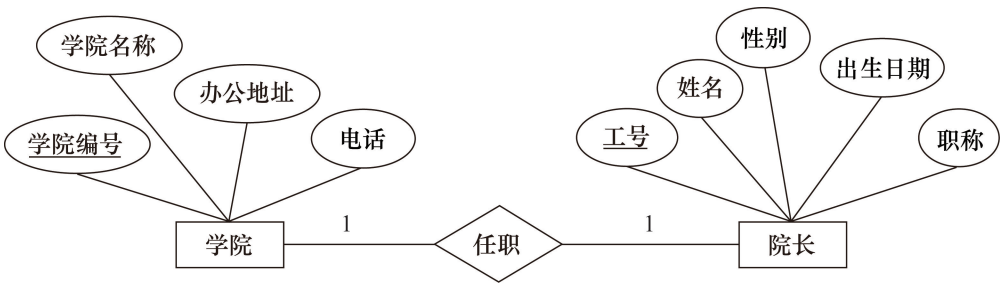

图 1-5 学院实体集与院长实体集的 E-R 图

③ 一对多联系。对于实体集 A 中的每一个实体，实体集 B 中至多有 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 具有一对多联系，记为 1∶n，如图 1-6 所示。

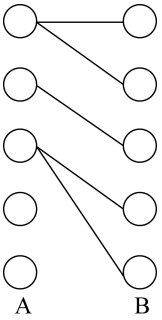

图 1-6　一对多联系示意图

【例 1-2】一家大型企业需要用计算机来管理

它分布在全国各地的分公司和员工信息。如果给定的需求信息如下，则建立此问题的概念模型。

需求分析：

① 某家企业有数个分公司分布在全国各地，每个分公司中有若干名员工，每名员工只在一个公司中工作。

② 需要管理的公司信息包括公司名称、地点、电话。

③ 需要管理的员工信息包括姓名、性别、出生日期和工资。

④ 此问题包含两个实体集：公司和员工。公司实体集与员工实体集之间的联系是一对多的联系。

⑤ 需要为每个公司编号，用以唯一地标识这个公司，因此公司实体的键是属性公司编号。

⑥ 需要为每位员工编号，用以唯一地标识每名员工，因此员工实体的键是属性员工编号。

E-R 模型：

① 实体型。公司（公司编号，公司名称，地点，电话）员工（员工编号，姓名，性别，出生日期，工资）

② E-R 图，如图 1-7 所示。

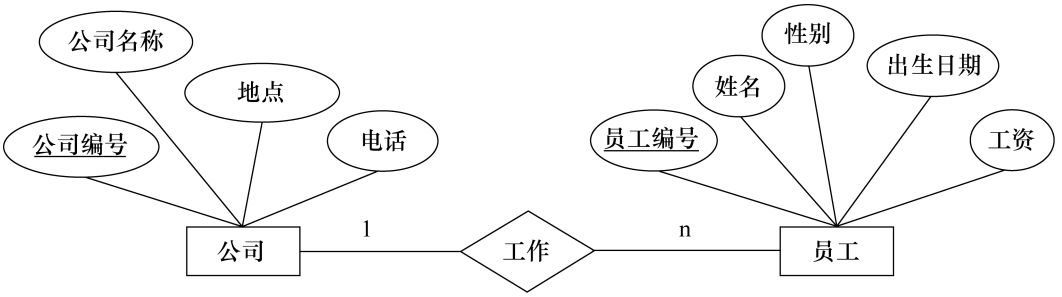

图 1-7　公司实体集与员工实体集的 E-R 图

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 中也有 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，如图 1-8 所示。

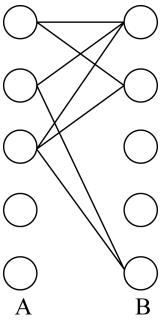

图 1-8 多对多联系示意图

【例 1-3】考虑某某职业大学中的学生选修课程的情况。如果给定的需求分析如下，则为其建立 E-R 模型。

需求分析：

① 每名学生可以选修多门课程，每门课程可被多名学生选修。

② 需要管理的课程信息包括：课程编号、课程名称、任课教师和学分。

③ 需要管理的学生信息包括：学号、姓名、性别、出生日期和所属院系。

④ 每门课程结束之后，教师会给出每名学生的学习成绩。

E-R 模型：

① 实体型。课程（课程编号，课程名称，教师编号，学分）学生（学号，姓名，性别，出生日期，所属院系）

② E-R 图，如图 1-9 所示。

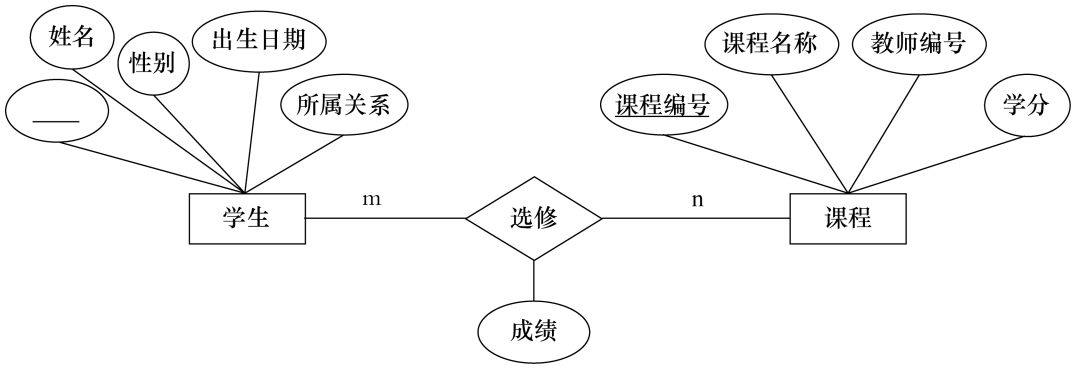

图 1-9 学生实体集与课程实体集的 E-R 图

1.5 逻辑模型

逻辑模型（Logical Model），是指数据的逻辑结构。是一种面向数据库系统的模型，是具体的 DBMS 所支持的数据模型。

1.5.1 数据模型

数据模型是一种用来表达数据的工具。在计算机中表示数据的数据模型应该能够精确地描述数据的静态特性、动态特性和完整性约束条件。因此，数据模型

通常是由数据结构、数据完整性约束和数据操作 3 部分内容构成的，其各内容表述如下。

1. 数据结构

数据结构是所研究对象类型的集合。这些对象是数据库的组成成分，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的联系有关的对象。

数据结构用于描述数据的静态特性，是数据模型中最重要的方面。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，采用层次结构、网状结构和关系结构的数据模型分别被称作层次模型、网状模型和关系模型。

2. 数据操作

数据操作是对数据库中各类对象的实例允许执行的操作的集合，包括操作对象和相关的操作规则。数据库主要有查询和更新（含插入、删除、修改）两大类操作。

数据操作用于描述数据的动态特性。数据模型必须对数据库中全部数据操作进行定义，指明每项数据操作的确切含义、操作对象、操作符号、操作规则以及对操作的语言约束等。

3. 数据完整性约束

数据完整性约束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、有效和一致。

1.5.2 常见的数据模型

数据库领域中常见数据模型包括层次模型、网状模型、关系模型和面向对象模型等，其各模型内容如下。

1. 层次模型

层次模型数据库系统中最早出现的数据模型，用树形结构来表示各类实体以及实体间的联系。层次模型的特点是：

（1）有且仅有一个结点没有父结点，它称作根结点；

（2）其他结点有且仅有一个父节点，如上图 1-10 所示。

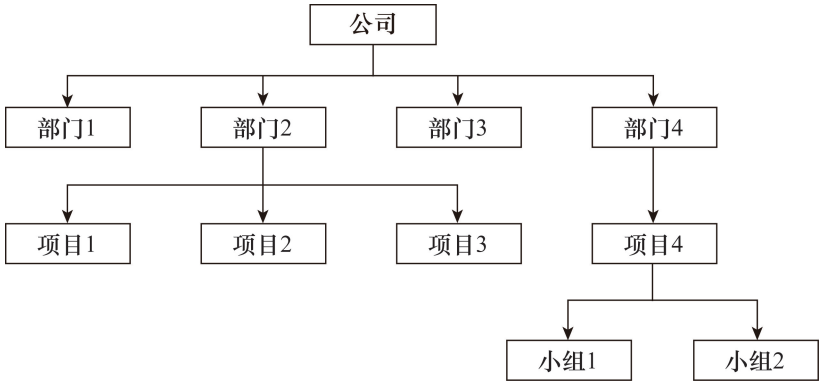

图 1-10　层次模型

2. 网状模型

现实世界实体之间的联系有很多种，层次模型也难以表达实体之间比较复杂的联系。网状模型的结构比层次模型的结构更具有普遍性，它允许多个节点没有父节点，也允许结点有多于一个的父节点，如图 1-11 所示。此外，网状模型还允许两个结点之间有多种联系。

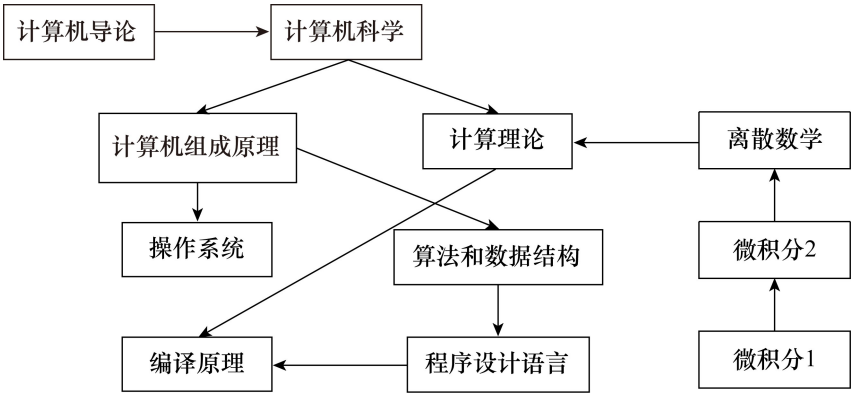

图 1-11 网状模型

网状模型其优点是能够更直接地描述现实世界，一个结点可以有多个父结点，允许复合链，具有良好的性能，存取效率比较高。其缺点是结构复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于用户掌握。

3. 关系模型

关系模型是目前最重要也是应用最广泛的数据模型。简言之，关系就是一张二维表，由行和列组成，如表 1-1 所示。

表 1-1 “学生”关系表

教学部门	姓名	性别	年龄
信息技术学院	王平	男	18
经济管理学院	叶珊	女	17
艺术设计学院	黄钟	男	20
教育技术学院	璐萍	女	22

关系模型的概念单一，数据操作方法统一，使用户易懂易用。在关系数据库系统中，用户根据数据的逻辑模式和子模式进行数据操作，而不必关心数据的物理模式情况，使用方便，数据的独立性、安全性和保密性都很好。目前，市面流行的商用数据库管理系统几乎都支持关系模型，非关系模型的产品也大都加上了关系接口。

4. 面向对象模型

尽管关系模型简单灵活，但是现实世界中仍存在一些复杂的数据结构，难以用关系模

型进行描述，人们迫切需要语义表达更强的数据模型。面向对象方法与数据库相结合所构成的数据模型称为面向对象模型。面向对象模型用面向对象的观点来描述现实世界实体的逻辑组织、对象间的联系，其表达能力丰富，具有对象可复用、维护方便等优点，是正在发展的数据模型，也是数据库的发展方向之一。

1.6 关系数据库

关系数据库是数据库应用的主流，许多数据库管理系统的数据模型都是基于关系数据模型开发的，数据库的操作借助于集合代数等概念和方法来处理数据库中的数据，需具有坚实的数学基础。

1.6.1 关系模型的基本概念

一个关系模型的逻辑结构是一张二维表格，即关系。在关系数据模型中，实体集以及实体集间的各种联系均用关系表示。其基本概念如下。

1. 关系

关系（Relation）即一个二维表格。

2. 属性

表（关系）的每一列必须有一个名字，称为属性（Attribute）。

3. 元组

表（关系）的每一行称为一个元组（Tuple）。

4. 域

表（关系）的每个属性有一个取值范围，称为域（Domain），域是一组具有相同数据类型的值的集合。

5. 关键字

关键字又称主属性，可以唯一地标识一个元组（一行）的一个属性或多个属性的组合。起到这样作用的关键字（Key）有两类：

（1）主关键字（Primary Key）。一个关系中只能有一个主关键字，用以唯一地标识元组。同时，主关键字也用于关系之间的联系。

（2）候选关键字（Candidate Key）。一个关系中可以唯一地标识一个元组（一行）的一个属性或多个属性的组合，一个关系中可以有多个候选关键字。

有时，关系中只有一个候选关键字，把这个候选关键字定义为主关键字后，关系中将没有候选关键字。当关系中有多个候选关键字时，可指定其中一个为主关键字。例如，学生表中的序号和学生 ID 都可以唯一地标识元组，将学生 ID 指定为主关键字，那么序号就是候选关键字。

另外，如果某个关系中的一个属性或属性组合不是所在关系的主关键字或候选关键

字，但却是其他关系的主关键字，对这个关系而言，则称其为外部关键字（Foreign Key）。

1.6.2 关系数据库的基本性质

关系数据库的基本性质主要表现在以下几个方面。

（1）一个关系就是一个二维表格，表的每一列是一个属性，每一列有唯一的列名。

（2）表的每一列都必须是不可再分的数据项，表的每一列的数据类型相同，数据来自同一个域。

（3）表的每一行是一个元组，表中不能有重复的元组，用主关键字来保证元组的唯一性。

（4）不同的列可以出自同一个域，但列名不能相同。表中列的顺序可以任意交换，行的顺序也可以任意交换。

1.6.3 关系数据模式的规范化

关系数据库的设计主要是关系模式的设计，关系模式设计的好坏将直接影响到数据库设计的成败。为了使关系模式设计的方法趋于完备，数据库开发者们研究了关系规范化理论。

1. 函数依赖及其对关系的影响

函数依赖是属性之间的一种联系，普遍存在于现实生活中。例如，学籍科通过学生的学生编号，可以查询到该编号的学生身份证号码。又比如，表 1-2 是职工工资关系的二维表格，用一种称为关系模式的形式表示为：

职工工资(工号,姓名,性别,出生日期,工资级别,工资额)

由于每名职工有唯一的工号，一个工号只对应一名职工，一名职工只对应一个工资级别，因此工号的值确定后，姓名、性别、出生日期、工资级别等的值也就被唯一地确定了。属性间的这种依赖关系类似数学中的函数。因此称工号函数决定姓名，或者说姓名函数依赖于工号，记为：工号→姓名；同样有工号→性别，工号→出生日期等。如下表 1-2 所示。

表 1-2 职工工资关系表

工号	姓名	性别	出生年月	工资级别	工资额
001010	张三	男	9901	3	6000
001011	李四	男	9702	5	5000
001012	王五	女	9805	4	4000
001013	赵六	女	2010	6	2000
001014	李蓉	女	9902	6	2000

上述关系模式存在如下问题，主要表现在 4 个方面。

（1）数据冗余（Data Redundancy）。如职工很多，工资级别有限，每一级别的工资数额反复存储多次。

（2）插入异常（Insertation Anomaly）。如果没有职工具有 2 级工资，则 2 级工资的工资数额就难以插入。

（3）删除异常（Deletion Anomaly）。如果仅有职工赵六具有 6 级工资，如果将李蓉删除，则有关 6 级工资的工资数额信息也随之删除了。

（4）更新异常（Update Anomaly）。如果将 6 级工资的工资数额调为 2500，则需要找到每个具有 6 级工资的职工，逐一修改。

一个关系之所以会产生上述问题，是由于关系中存在某些函数依赖引起的。通常，如果把太多的信息放在一个关系中，出现的诸如冗余之类的问题称为“异常”。

规范化是为了设计出好的关系模型。规范化理论正是用来改造关系模式，通过分解关系模式来消除其中不合适的数据依赖，以解决更新异常、插入异常、删除异常和数据冗余问顾。

上述修改后的关系，如表 1-3 和表 1-4 所示。

表 1-3　职工关系

工号	姓名	性别	出生年月	工资级别
001010	张三	男	9901	3
001011	李四	男	9702	5
001012	王五	女	9805	4
001013	赵六	女	2010	6
001014	李蓉	女	9902	6

表 1-4　工资关系

工资级别	工资额
3	6000
5	5000
4	4000
6	2000

2. 关系模式的规范化

每个规范化的关系只有一个主题。如果某个关系有两个或多个主题，就应该分解为多个关系。规范化的过程就是不断分解关系的过程。

在数据库技术的发展过程中，人们每发现一种异常，就研究一种规则防止异常出现，使设计关系的准则得以不断改进。范式（Normal Form）是指规范化的关系模式，由于规范化的程度不同，就产生了不同的范式。

从 1971 年起，E. F. Codd 相继提出了第一范式（Fist Normal Forms，1NF），第二范式（Second Normal Form，2NF）和第三范式（Third Normal Form，3NF），Codd 与 Boyce 合作提出了 Boyce-Codd 范式。在 1976-1978 年，Fagin Delobe 以及 Zaniolo 又定义了第四范式。到目前为止，已经提出了第六范式。在进行关系数据库设计时，至少要符合第一范式的要求，在第一范式基础上再满足另外一些约束条件就产生了第二范式，其余范式依次类推。一般来说，在设计数据库时，只需满足第三范式就行了。

（1）第一范式（1NF）。第一范式是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多个值，即实体中的某个属性不能有多个值或者不能有重复的属性。如果出现重复的属性，就需要定义一个新的实体，新的实体由重复的属性构成，新实体与原实体之间为一对多关系。在第一范式中，表的每一行只包含一个实例的信息。简言之，第一范式就是无重复的列。例如，表 1-2 的职工工资关系中所有的属性都是不可再分的简单属

性，满足第一范式的关系模式，列的取值只能是原子数据；每一列的数据类型相同，每一列有唯一的列名（属性）；列的先后顺序无关紧要，行的先后顺序也无关紧要。

（2）第二范式（2NF）。第二范式是在第一范式的基础上建立起来的，即满足第二范式必须先满足第一范式。第二范式要求数据库表中的每个实例或行必须能被唯一地区分。在第二范式中，要求实体的属性完全依赖于主关键字。所谓完全依赖，是指不能存在仅依赖主关键字一部分的属性，如果存在依赖关键字一部分的属性，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分，通常需要为表加上一个列，以存储各个实例的唯一标识。简单地说，第二范式就是属性完全依赖于主关键字。例如，在学生选修课程的关系模式（学号，课程号，所在院系，成绩）中，（学号，课程号）组合起来作为关键字。模式中，非主属性“成绩”完全依赖于该关键字，而非主属性“所在系”则部分依赖于该关键字。因此选修不属于2NF。根据2NF的定义，可将其分解为学生（学号，所在院系）和选课（学号，课程号，成绩）两个关系。

（3）第三范式（3NF）。第三范式是在第二范式的基础上建立起来的，即满足第三范式必须满足第二范式。第三范式要求关系表中不存在非关键字对任一候选关键字的传递函数依赖。传递函数依赖是指如果存在“A→B→C”的决定关系，则C传递函数依赖A。也就是说，第三范式要求关系表不包含其他表中已包含的非主关键字段信息。例如，关系表中职工工资（职工号，级别，工资）中因为存在：职工号→级别，级别→工资这样的传递函数依赖，因此它不属于3NF。将其分解，修改为两个关系：职工级别（职工号，级别）和级别工资（级别，工资），则每张表都属于3NF。

由第一范式（1NF）、第二范式（2NF）和第三范式（3NF）的定义，总结出规范化的规则如下所示。

① 每个关系只有一个实体集，每个实体集只有一个主题，一个实体集对应一个关系。

② 属性中只包含原子数据。

③ 每个关系有一个主关键字，用来唯一地标识关系中的元组。

④ 关系中不能有重复属性；所有属性完全依赖关键字；所有非关键字属性相互独立。

⑤ 元组的顺序无关，属性的顺序无关。

1.6.4 关系的完整性约束

在关系数据库中往往包含多张表，有些表之间存在一定的联系，不同的表之间还可能出现相同的属性。这些都给数据库的数据维护带来了挑战，因为数据库必须保证所有表中的数据值与其所描述的应用对象的状态一致。关系模型通过关系完整性约束条件来保证数据的正确性和一致性。

关系完整性约束包括域完整性、实体完整性、参照完整性和用户定义完整性。现在的数据库系统都在不同程度上支持完整性规则的检查。

1. 域完整性

域完整性是对数据表中字段属性的约束，如规定该字段的数据类型、格式、值域范

围、是否允许空值等约束规则，通常是由确定关系结构时所定义的字段属性决定的。例如，规定某个关系中的“性别”字段只能取值“男”或“女”。

2. 实体完整性

实体完整性是指关系的所有主关键字对应的主属性都不能取空值，而且主关键字的值不能重复。例如，学生选课的关系选课（学号，课程号，成绩）中，学号和课程号共同组成主关键字，则学号和课程号两个属性都不能为空。因为，没有学号的成绩或没有课程号的成绩是不存在的。

3. 参照完整性

参照完整性又称引用完整性，与关系之间的联系有关，要求关系中不允许引用不存在的实体。在关系数据库中，关系之间可能存在着引用关系，对数据库进行修改时，可能会破坏关系之间的参照完整性。所以，为了保证数据库中数据的完整性，应该对数据库的修改加以限制，这些限制包括插入约束、删除约束和更新约束。

（1）插入约束。在向表中输入一条新记录时，系统要检查新记录的外键值是否在主表中已经存在。如果存在，则允许执行插入操作，否则拒绝插入。

（2）删除约束。如果删除主表中的一条记录，则相关表中凡是外键的值与主表的主键值相同的记录也会被同时删除，将此称为级联删除。

（3）更新约束。如果修改主表中主关键字的值，则相关表中相应记录的外键值也随之被修改，将此称为级联更新。

1.6.5 关系代数

数据模型的建立是在对现实世界抽象的基础上优化数据存储，其目的是为了使用数据。在关系数据模型中，使用关系代数来建立数据操纵的模型。那么，关系代数是一种抽象的查询语言，是关系数据操纵语言的传统表达方式，它用关系运算来表达数据查询。

1. 关系运算符

关系代数的运算对象是关系，运算结果也是关系，其操作运算符包括传统集合运算符、专门的关系运算符、比较运算符和逻辑运算符，如表 1-5 所示。

表 1-5　关系代数运算符

类别	运算符	说明	类别	运算符	说明
传统集合运算符	∩	交	比较运算符	>	大于
	∪	并		≥	大于等于
	−	差		<	小于
	×	笛卡尔积		≤	小于等于
				=	等于
				≠	不等于

续表

类别	运算符	说明	类别	运算符	说明
专门关系运算符	σ Π $\square$ $\div$	选择 投影 连接 除	逻辑运算符	$\wedge$ $\vee$ $\neg$	与 或 非

其中，集合运算将关系看成元组的集合，其运算是针对行进行的，而专门关系运算符不仅操作行也可以操作列。

2. 传统的集合运算

传统的集合运算是双目运算，包括并、交、和笛卡尔积运算。

（1）关系的并。关系 R 和 S 的并是由关系 R 和关系 S 的所有元组合并，再删去重复的元组，组成的新关系，记为 $R \cup S$。

$$R \cup S = \{t \mid t \in R \ \vee \ t \in S\}$$

（2）关系的差。关系 R 和 S 的差是由属于 R 但不属于 S 的所有的元组组成的集合，即关系 R 中删除与关系 S 中相同的元组，组成的新关系，记为 $R-S$。

$$R-S = \{t \mid t \in R \wedge t \notin S\}$$

（3）关系的交。关系 R 和 S 的交是由既属于 R 又属于 S 的元组组成的集合，即在两个关系 R 与 S 中取相同的元组组成新的关系，记为 $R \cap S$。

$$R \cap S = \{t \mid t \in R \wedge t \in S\}$$

（4）笛卡尔积。设关系 R 和 S 分别有 n 和 m 列，若关系 R 中有 i 行，关系 S 中有 j 行，则关系 R 和 S 的笛卡尔积是由 $R \times S = n+m$ 列且 $R \times S$ 有 $i\,j$ 行集合组成的新关系。记为$R \times S$。

【例 1-4】设有 3 个关系 R、S 和 T，分别求出 $R \cup S$、$R-S$、$R \cap S$ 和 $R \times T$ 的运算结果。如图 1-12 所示。

R

A	B
a	b
b	c
c	a

S

A	B
a	c
b	a
c	b

T

B	C
a	a
b	c

图 1-12 关系 R、S 和 T

运算结果，如图 1-13 所示。

$R \cup S$

A	B
a	b
a	c
c	a
b	a
c	b

$R \cap S$

A	B
a	c

$R-S$

A	B
a	b
c	a

$R \times T$

A	B	B	C
a	b	a	a
a	b	b	c
a	c	a	a
a	c	b	c
c	a	a	a
c	a	b	c

图 1-13　传统集合运算

3. 专门的关系运算

专门的关系运算包括选择、投影、连接和除等运算。

（1）选择运算。从关系中找出满足给定条件的那些元组称为选择。其中的条件是以逻辑表达式给出的，值为真的元组将被选取，这种运算是从行的角度抽取元组。经过选择运算得到的结果元组组成新的关系，其关系模式不变，但元组的数目小于等于原来关系中元组的个数，是原关系的子集。选择运算记为：$\sigma_F(R)$。

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = TRUE\}$$

其中，R 为一个关系，F 为逻辑函数，函数 F 中可以包含比较运算符和逻辑运算符。

在企业会员系统中，有会员关系 Users 如表 1-6 所示。

表 1-6　会员（Users）关系

会员 ID	用户名	性别	QQ	会员积分
1	Lili2001	男	21787636	231
2	Anhn	女	221515645	23
3	小王	男	1235752	87
4	Lipeng	女	234987452	134
5	张三	男	674654287	568

【例 1-5】查询会员（Users）关系中，性别为“男”的会员信息。

其关系运算表达式可以描述为 σ 性别=’男’（Users）或 $\sigma_{3='男'}$（Users），其中 3 表示关系中的第 3 列。运算结果，如表 1-7 所示。

表 1-7　选择运算

会员 ID	用户名	性别	QQ	会员积分
1	Lili2001	男	21787636	231
2	小王	男	1235752	87
3	张三	男	674654287	568

（2）投影运算。从关系模式中挑选若干属性组成新的关系称为投影。这是从列的角度进行运算，相当于对关系进行垂直分解。投影后的新关系所包含的属性少于或等于原关系，若新关系中包含重复元组，则要删除重复元组。投影运算记为：$\Pi_X(R)$。

$$\Pi_X(R) = \{ t[x] \mid t \in R \}$$

其中，R 是一个关系，x 是 R 中的属性列。

【例 1-6】查询会员（Users）的用户名、性别和会员积分。

其关系运算表达式可以描述为 $\Pi_{用户名,性别,会员积分}$（Users）或 $\Pi_{2,3,5}$（Users）。运算结果，如表 1-8 所示。

表 1-8 投影运算

用户名	性别	会员积分
Lili2001	男	231
Anhn	女	23
小王	男	87
Lipeng	女	134
张三	男	568

（3）连接运算。连接运算是从两个关系的笛卡尔积中选择属性值满足一定条件的元组，去处过程通过连接条件来控制，连接是对关系的结合。连接运算通常分为 θ 连接和自然连接。

① θ 连接

$$R \bowtie S = \{ t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta \quad t_s[B] \}$$

$$A \ \theta \ B$$

θ 连接是从关系 R 和 S 的笛卡尔积中选取属性值满足一个条件运算符 θ 的元组，其关系运算定义如下：

其中，A 和 B 是关系 R 和 S 中第 A 列和第 B 列的值或列序号。当 θ 为符号“=”时，该连接操作称为等值连接。

②自然连接

自然连接是去除重复属性的等值连接，它是连接运算的特例，是最常用的连接运算。其关系运算定义如下：

$$R \bowtie S = \{ t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[A] \}$$

其中关系 R 和 S 具有同名属性 A。

在网络商城系统中，有商品类别（Goods Type）和商品（Goods）两个关系，如表 1-9 和 1-10 所示。

表 1-9　商品（Goods）关系

商品 ID	商品编号	商品名称	价格	类别 ID
1	1001	迷彩帽	63	1
2	2001	牛肉干	94	2
3	2002	零食礼包	145	2
4	3005	运动鞋	400	1

表 1-10　商品类别（Goods Type）关系

类别 ID	类别名称
1	服饰
2	零食

【例 1-7】查询类别为服饰的商品信息。

设 Goods 关系为 R，Goods Type 关系为 S，由于两个关系中有共同的属性类别 ID，则进行的连接运算为自然连接，其关系运算表达式可以描述如下：

$$\sigma_{\text{类别名称}='\text{服饰}'}(R \bowtie S)$$

其运算结果，如表 1-11 所示。

表 1-11　自然连接运算

商品 ID	类别 ID	商品编号	商品名称	价格	类别名称
1	1	1001	迷彩帽	63	服饰
4	1	3005	运动鞋	400	服饰

【例 1-8】查询类别为服饰的商品信息，列出商品编号，商品名称，价格和类别名称。设 Goods 关系为 R，Goods Type 关系为 S，其关系运算表达式可以描述如下：

$$\Pi_{\text{商品编号,商品名称,价格,类别名称}}(\sigma_{\text{类别名称}='\text{服饰}'}(R \bowtie S))$$

其运算结果，如表 1-12 所示。

表 1-12　选择、投影和自然连接运算

商品编号	商品名称	价格	类别名称
1001	迷彩帽	63	服饰
3005	运动鞋	400	服饰

（4）除法运算。在关系代数中，除法运算可理解为笛卡尔积的逆运算。设被除关系 R 有 m 元关系，除关系 S 有 n 元关系，那么它们的商为 $m-n$ 元关系。记为 $R \div S$。其中在 R 中每个元组 i 与 S 中的每个元组 j 组成的新元组必在关系 R 中。

【例 1-9】设有如下关系 R 和 S，如图 1-14所示。求 $R \div S$ 的运算结果。

$R \div S$ 的运算结果，如表 1-13 所示。

R

A	B	C	D
2	1	a	c
2	2	a	d
3	2	b	d
3	2	b	c
2	1	b	d

S

C	D	E
a	c	5
a	c	2
b	d	6

图 1-14　除法运算示例

表 1-13 除法运算结果

A	B
2	1

商的构成原则是将被关系 R 中的 $m-n$ 列，按其值分成若干组，检查每一组的 n 列值的集合是否包含除关系 S，若包含则取 $m-n$ 列的值作为商的一个元组，否则不取。

1.7 数据库设计的基本步骤

现实世界的信息结构复杂且应用环境多种多样，在很长一段时间内，数据库设计是采用手工试凑法进行的。用手工试凑法设计数据库与设计人员的经验和水平有直接关系，它更像是一种技艺而不是工程技术。这种方法缺乏科学的理论和工程方法支持，数据库的质量很难得到保证。经过人们的不断努力，提出了各种各样的数据库设计方法，并提出了多种数据库系统设计的准则和规程，这些方法被称为规范设计法。

新奥尔良方法是规范设计法中的一种，它将数据库设计分为 4 个阶段：需求分析、概念设计、逻辑设计和物理设计。其后，许多科学家进行了改进，认为数据库设计应分为 6 个阶段进行：需求分析、概念设计、逻辑设计、物理设计、数据库实施、数据库运行和维护，如图 1-15 所示。

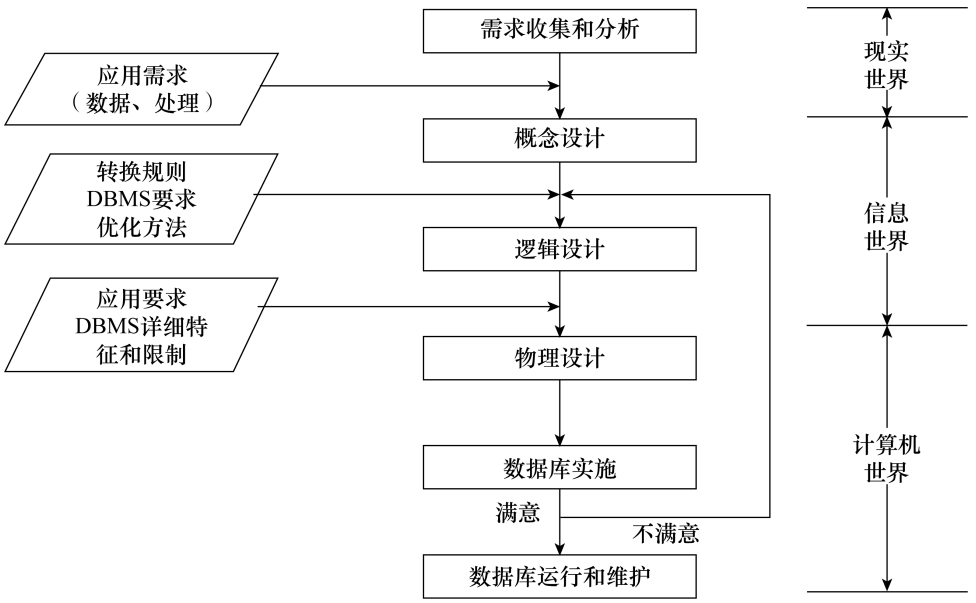

图 1-15 数据库设计步骤

在数据库设计的不同阶段，实现的具体方法也不同，有基于 $E-R$ 模型的数据库设计方法、基于 3NF 的设计方法以及基于抽象语法规范的设计方法等。

1.7.1 需求分析

需求分析是数据库设计过程中的首要步骤，是最耗费时间、最困难的一步，也是整个数据库设计过程的基础。本阶段的主要任务是对现实世界中要处理的对象进行详细调查，然后通过分析，逐步明确客户和用户对系统的需求，包括数据需求和业务处理需求。

需求分析是否做的充分和准确，直接决定了对用户需求进行分析和表达后，必须把分析结果提交给用户，取得用户认可。否则，会导致整个数据库设计失败而返工重做。

1.7.2 概念设计

概念设计是数据库设计的关键，通过综合、归纳和抽象用户需求，形成一个具体DBMS的概念模型，也就是绘制数据库设计的*E*-*R*图，*E*-*R*图又称*E*-*R*模型。*E*-*R*图主要用于在项目团队内部，设计人员和客户之间进行沟通，确认需求信息的正确性和完整性。

概念模型是数据模型的前身，它比数据模型更独立于计算机、更抽象、也更加稳定。自数据库技术广泛应用以来，出现了不少数据库概念设计方法，可简单归纳为4种。

（1）自顶向下。首先定义全局概念结构的框架，然后逐步细化为完整的全局概念结构。

（2）自底向上。首先定义各局部应用的概念结构，然后将它们集成起来，得到全局概念结构的设计方法。

（3）逐步扩张。首先定义最重要的核心概念结构，然后向外扩充，生成其他概念结构，直至完成总体念结构。

（4）混合策略。采用自顶向下与自底向上相结合的方法，首先用自顶向下策略设计一个全局概念结构框架，然后以它为骨架，通过自底向上策略设计各局部概念的结构。

概念结构的设计可以分为两步：一是抽象数据并设计局部视图；二是集成局部视图得到全局的概念结构。设计步骤，如图1-16所示。

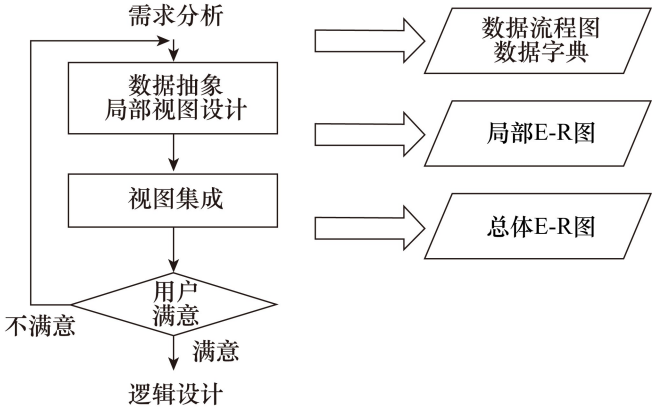

图1-16 概念结构设计步骤

1.7.3 逻辑设计

数据库的逻辑设计与选用的 DBMS 有关。目前，一般的 DBMS 都是关系型的，因此数据库逻辑设计阶段主要的任务是在概念设计的基础上，首先利用一些映射规则得到一组初始关系模式集，然后用关系规范化理论对关系模式进行优化，以获得质量良好的数据库设计。

概念模型向关系模型转换要解决的问题是如何将实体以及实体之间的联系转换为关系模式，以及如何确定这些关系模式的属性和主关键字。概念模型向关系模型的转换步骤，如图 1-17 所示。

图 1-17　逻辑结构设计的步骤

1. 实体的转换

概念模型的表现形式是 *E-R* 图，由实体、实体的属性和实体之间的联系 3 个要素组成。从 *E-R* 图转换为关系模式的方法如下。

（1）为每个实体定义一个关系，实体的名字就是关系的名字，实体的属性就是关系的属性，实体的键是关系的主关键字。

（2）用规范化准则检查每个关系，上述设计可能需要改变，也可能不用改变。依据关系规范化准则，在定义实体时就应遵循每个实体只有一个主题的原则。

2. 实体间关系的转换

关系之间的联系是通过外部关键字来体现的。前面讨论过实体之间的联系通常有 3 种类型：一对一联系、一对多联系和多对多联系。

1.7.4 物理设计

数据库的物理结构主要是指数据库在物理设备上的存储结构和存取方法。数据库物理设计的任务就是利用所选 DBMS 提供的手段为设计好的逻辑数据模型选择一个符合应用要求的物理结构。由于不同的数据库系统所提供的物理环境、存取方法和存储结构有很大差别，能提供给设计人员使用的设计变量、参数范围也大不相同，因此没有通用的物理设计方法可遵循，只能给出一般的设计内容和原则。

关系数据库物理设计的主要内容包括选择存取方法和存储结构，即确定关系、索引、聚簇、日志、备份等的存储安排和存储结构，确定系统配置等。数据库物理设计过程中需要对经项目组开会讨论确定 *E-R* 图后，根据项目的技术实现，团队开发能力及项目的成本预算，选择具体的数据库项目进行物理实现。

1.7.5 数据库实施

设计人员采用 DBMS 提供的数据定义语言、工具及宿主语言，根据逻辑设计和物理设计的结果严格建立数据库，编制与调试应用程序，组织数据入库，并进行试运行阶段。试运行阶段由于系统还不稳定，软硬件故障随时都有可能发生，因此，应分期分批地组织数据入库，并做好数据库的备份和恢复工作。

1.7.6 数据库运行和维护

数据库应用系统经过试运行后即可投入正式运行，这标志着数据库开发工作基本完成。在数据库运行阶段，数据库经常性的维护工作主要是由数据库管理员完成的，主要包括备份系统数据、恢复数据库系统、产生用户信息表、为信息表授权、监视系统运行状况、及时处理统错误、保证系统数据安全和定期更改用户口令等。

1. 数据库的备份和恢复

数据库的备份和恢复是系统正式运行后最重要的维护工作之一。数据库管理员要针对不同的应用要求制定不同的备份计划，以保证一旦发生故障尽快将数据库恢复到某种一致的状态，并尽可能减少对数据库的破坏。

2. 数据库的安全性、完整性控制

数据库运行过程中，由于应用环境的变化，对安全性的要求也会发生变化。为保证系统数据的安全，系统管理员必须依据系统的实际情况，执行一系列的安全保障措施。其中，周期性地更改用户口令是比较常用且十分有效的措施。

3. 数据库性能的监督、分析和改造

数据库运行过程中，监督系统运行，对检测数据进行分析，并找出改进系统性能的方法，是数据库管理员的又一重要任务。目前有些 DBMS 产品提供了检测系统性能的参数工具，数据管理员可以利用这些工具方便地得到系统运行过程中一系列性能参数的值。通过分析这些数据，系统管理员可判断当前系统的运行状况。

4. 数据库的重组织与重构造

数据库运行一段时间后，由于记录不断增、删、改，会使数据库的物理存储情况变坏，降低了数据的存取效率，使得数据库的性能下降。这时，数据库管理员就要对数据库进行重组织或部分重组织。在重组织过程中，按原设计要求重新安排存储位置、回收垃圾、减少指针链等，以提高系统性能。

如今随着数据库应用环境的变化，会导致实体及实体间的联系也发生变化，原有的数据库设计方法可能不能很好地满足新的需求，此时需要对数据库技术进行重构。数据库重构的主要工作是根据新环境调整数据库的设计模式和内模式、增加新的数据项、改变数据项的类型、改变数据库的容量、增加或删除索引以及修改完整性约束条件等。重构数据库的程度是有限的，如果应用变化太大或重构代价太大，则表明现有数据库的生命周期已经

结束，需要开发新的数据库管理系统。

知识点小结

本章详细介绍了数据库相关技术的基础知识，以及数据库系统的基本概念、特点、产生和发展过程。通过本章节的学习，希望读者了解什么是数据库、数据模式的概念、模式之间的映射、概念模型、逻辑模型以及数据库设计的基本步骤等，为以后的学习打下良好基础。

习　题

一、选择题

1. 下面列出的（　　）是数据库管理系统的简称。

   A. DB　　B. DBA　　C. DBMS　　D. DBS

2. 数据库发展的 3 个阶段中，没有专门的软件对数据进行管理的阶段是（　　）。

   A. 人工管理阶段和文件系统阶段　　B. 只有文件系统阶段

   C. 文件系统阶段和数据库阶段　　D. 只有人工管理阶段

3. 下面（　　）项不是数据模型的常用组成部分。

   A. 数据结构　　B. 数据表

   C. 数据操作　　D. 完整性约束

4. 关于结构化查询语言描述错误的是（　　）。

   A. 结构化查询语言简称 MySQL

   B. 结构化查询语言是一种应用于关系数据库查询的结构化语言

   C. 最早由 Boyce 和 Chambedin 在 1974 年提出，称为 SEQUEL 语言

   D. SQL 是一种介于关系代数和关系演算之间的语言，具有丰富的查询功能

5. 数据库、数据库系统和数据库管理系统之间的关系是（　　）。

   A. 数据库系统包括数据库和数据库管理系统

   B. 数据库管理系统包括数据库和数据库系统

   C. 数据库包括数据库系统和数据库管理系统

   D. 数据库系统就是数据库，也就是数据库管理系统

6. 关于数据库三级模式结构描述正确的是（　　）。

   A. 内模式也称逻辑模式或概念模式

   B. 模式是保证数据安全性的一个有力措施

   C. 数据库系统的三级模式结构是指模式、外模式和内模式

   D. 模式也称用户模式

7. 开发人员在设计 *E*-*R* 图时，通常使用（　　）表示联系。

A. 矩形框　　B. 椭圆　　C. 菱形　　D. 无向边

8. 数据库的数据独立性是指（　　）。

A. 不会因为数据的存储策略变化而影响系统存储结构

B. 不会因为系统存储结构变化而影响数据的逻辑结构

C. 不会因为数据存储结构与逻辑结构的变化而影响应用程序

D. 不会因为某些数据的变化而影响其他数据

9. 两个实体 A 和 B，如果 A 中的每一个值在 B 中有多个实体值与其对应，反之在 B 中每一个实体值在 A 中至多有一个实体值与之对应，那么则称 A 和 B 为（　　）。

A. 一对一关系　　B. 一对多关系

C. 多对多关系　　D. 以上都不是

10. 通常用以下的顺序来完成数据库的设计工作（　　）。

A. 概念设计、物理设计、逻辑设计　　B. 逻辑设计、概念设计、物理设计

C. 概念设计、逻辑设计、物理设计　　D. 物理设计、概念设计、逻辑设计

一、填空题

1. _____简称_____是一种应用于关系数据库查询的结构化语言。

2. SQL 语言为完成其核心功能只用了 6 个动词，分别是_____、_____、_____、_____、_____和 GRANT（REVOKE）。

3. _____图也称“实体-关系图”，用于描述现实世界的事物，以及事物与事物之间的关系。其中_____表示实体，_____表示关系。

4. 开发人员在设计 *E*-*R* 图时，使用_____表示实体，在_____内写明实体名，使用_____表示属性，并且使用_____将其与实体连接起来。

5. 通常情况下，实体间存在_____、_____、_____ 3 种联系。

6. 在 *E*-*R* 图中，每个实体通常对应一张_____，实体的属性对应于_____。

7. 在数据库管理系统中，常用的数据模型包括_____、_____、_____和_____。

8. 在关系模型中，若属性 A 是关系 *R* 的主码，则在 *R* 的任何元组中，属性 A 的取值都不允许为空，这种约束称为_____。