目录

项目 3　初探 MySQL / 39

项目 4　操作 MySQL 数据库对象 / 55

项目 5　查询数据 / 81

项目 9　MySQL 用户安全性管理 / 175

项目 1 认识数据库

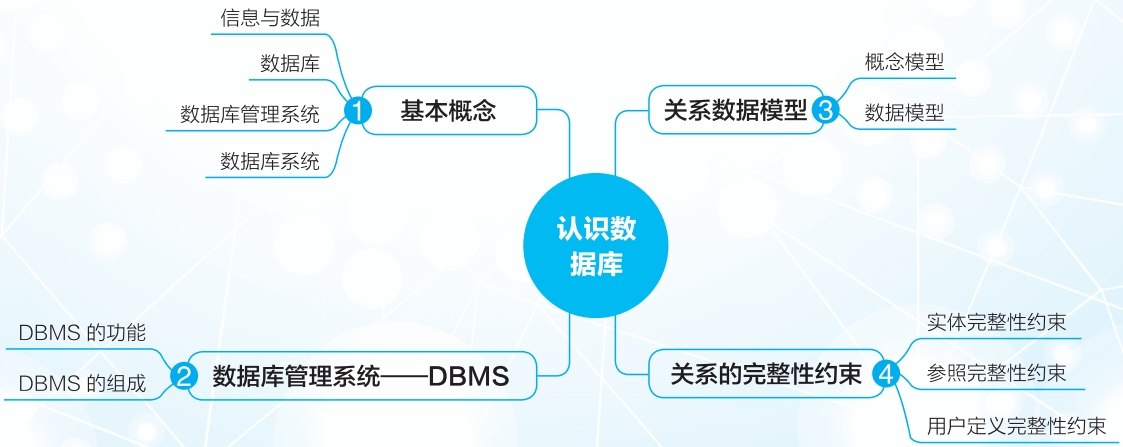
知识目标

1. 了解数据、数据库、数据库管理系统等基本概念。
2. 掌握数据库管理系统的功能和组成。
3. 掌握关系数据模型的概念。
4. 掌握关系的完整性的概念。

能力目标

1. 能区分数据、数据库等相关概念。
2. 能识别数据库管理系统。
3. 能区分概念模型和数据模型。
4. 能定义实体完整性和参照完整性。

知识导图



笔记

项目导入

小明是某高校一年级的学生，身为班级的学习委员，为了方便管理和统计本班同学的各科成绩，他打算自己开发一个学生成绩管理系统软件。

分析：

小明发现学生成绩管理系统软件的组成不仅需要程序的架构，还需要信息的管理。而信息的管理其实就是数据的管理，数据库技术就是一种数据管理技术，它产生于 20 世纪 60 年代，经过多年的发展，已经自成理论体系，成为计算机科学的一个重要分支。数据库技术体现了先进的数据管理思想，使计算机应用渗透到社会各个领域，在当今的信息社会中发挥着越来越大的作用。

小明通过对数据库的基本知识和基本概念的了解，进一步学习了信息与数据、数据库管理系统的基本概念及其组成、关系数据模型及关系的完整性。数据库对信息的有效管理在学生成绩管理系统的实现中起了关键性的作用。

任务 1.1 基本概念

子任务 1.1.1 信息与数据

信息是客观事物在人脑中的反映，是以各种方式传播的关于某一事物的消息、情报、知识。信息是一种资源。随着科学技术的发展，生产力水平大大提高，经济、文化、军事等各领域均迅猛发展，这就需要人们掌握大量的信息，并且能够研究和分析这些信息，从中得出有用的结论，再把它们应用到社会生产实践活动中去。计算机的问世和发展给人们提供了用计算机进行管理和处理信息的可能性。而人们在使用计算机管理和处理信息的同时，又进一步开发了信息资源，并利用信息资源进一步促进了生产力的发展和社会的进步。

数据（Data）是描述客观事物的符号记录。计算机中的数据是指经过数字化后能够由计算机处理的数字、字母、符号、声音、图形、图像等。数据是数据库中存储的基本对象。

为了了解世界以及相互交流，人们需要描述各种各样的事物。在日常生活中，我们通常直接用自然语言进行描述。而在计算机中，为了存储和处理这些抽象信息，就要抽取出对这些事物感兴趣的特征值，并用特定的符号来进行描述。

例如，在描述职工人事档案的时候，人们感兴趣的可能是员工编号、姓名、性别、年龄、生日、籍贯、家庭住址、政治面貌、职称、行政职务等基本信息。对于这些信息，可以用如下方式来描述：

（001、张三、女、36、1977 年 3 月 8 日、重庆市、重庆市渝中区、中共党员、高级工程师、处长）

这里的职工人事档案记录就是数据。对于以上记录中的每个数据项必须经过解释才能明确其含义。数据的含义称为数据的语义。上述记录可以解释为姓名为张三的女性员工，1977 年出生，现年 36 岁等。数据与其语义是不可分的。数据是信息的符号表示，信息则是数据的内涵，是对数据的语义解释。

信息和数据都是现象所反映的知识，这是它们的共同点。因此当不需要严格区分的时候，可以把这两个概念不加区分地使用，如“数据处理”与“信息处理”是同义的，“数

据资源”和“信息资源”是同义的。

对数据进行收集、整理、组织、存储、传播、检索、分类、加工、计算、打印报表、输出等一系列活动称为数据处理或信息处理。

在数据处理的一系列活动中，数据的收集、整理、组织、存储、传播、检索、分类等是基本环节，被称为数据管理或信息管理。

数据管理是数据处理的基本环节，数据管理技术的优劣直接影响数据处理的效果。而数据库技术就是一种先进的数据管理技术。

笔记

子任务 1.1.2　数据库

数据库（DataBase，DB），顾名思义，就是存放数据的仓库，是长期存储在计算机内的、有组织的、可共享的相关数据集合。

数据库中保存的是以一定的组织方式存储在一起的相互有关的数据整体，即数据库不仅保存数据，还保存数据与数据之间的联系。数据库中的数据可以被多个应用程序的用户所使用，从而达到数据共享的目的。

数据库中的数据与应用程序之间彼此独立。在数据库中，数据的组织和存储方法与应用程序互不依赖、相互独立。应用程序不再与一个孤立的数据文件相对应，它所涉及的数据取自整体数据集合的某个子集，作为逻辑文件与应用程序相对应，并通过系统软件数据库管理系统实现逻辑文件与物理数据之间的映射。

数据库中的数据不是孤立的，而是相互关联的。在数据库中不仅存放了数据本身，还存放了数据与数据之间的联系。例如，在教学管理系统中，数据库不仅存放了关于学生和课程的数据，而且还存放了学生及其所选修的课程，这就反映了学生数据与课程数据之间的联系。

综上所述，数据库是以一定的组织方式存放在一起的、能够被多个用户所共享的、与应用程序彼此独立的、相互关联的数据集合。

注意

数据库和数据仓库（Data Warehouse）不是同一个概念。数据仓库是在数据库技术基础上发展起来的一个新的应用领域。

子任务 1.1.3　数据库管理系统

数据库管理系统（Database Management System，DBMS）是一个系统软件，负责对数据库资源进行统一管理和控制，其职能是建立数据库、维护数据库，接受并完成用户提出的访问数据等各种请求，并且为数据库的安全性和完整性提供保证。数据库管理系统位于用户与操作系统之间。通过数据库管理系统，用户可以不必过问数据存放的细节而方便地建立、使用和维护数据库。用户可通过数据库管理系统访问数据库中的数据，数据库管理员也可通过数据库管理系统对数据库进行维护。它可使多个应用程序和用户用不同的方法同时或在不同时刻去建立、修改和询问数据库。

子任务 1.1.4　数据库系统

1. 数据库系统的定义

数据库系统（Database Systems）是由数据库及其管理软件组成的系统。它是为适应数据处理的需要而发展起来的一种较为理想的数据处理的核心机构。数据库系统是一个为实

笔记

际可运行的存储、维护和应用系统提供数据的软件系统，是存储介质、处理对象和管理系统的集合体。

2. 数据库系统的构成部分

数据库系统包括数据、硬件、软件和用户 4 个部分。

（1）数据是构成数据库的主体，是数据库系统的管理对象。

（2）硬件是构成计算机系统的各种物理设备，包括存储所需的外部设备。硬件的配置应满足整个数据库系统的需要。

（3）软件包括操作系统、数据库管理系统及应用程序。数据库管理系统是数据库系统的核心软件，是在操作系统的支持下，解决如何科学地组织和存储数据、如何高效获取和维护数据的系统软件。

（4）用户包括专业用户、非专业用户和数据库管理员。

① 专业用户指应用程序员。他们负责设计和编制应用程序，通过应用程序存取和维护数据库，为最终用户准备应用程序。

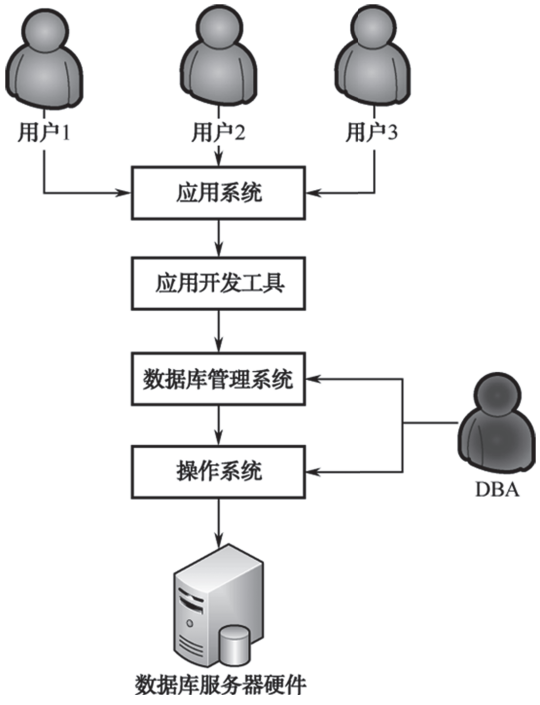


图 1-1　数据库系统结构

② 非专业用户，即最终用户，是非计算机专业人员。他们通过应用系统提供的用户接口界面以交互方式操作使用数据库。交互式操作通常为菜单驱动、图形显示、表格操作等。

③ 数据库管理员（Database Administrator，DBA），即负责管理和维护数据库服务器的人员。对于大型数据库系统，要求配备专门的数据库管理员，其主要职责是：

- 参与数据库设计的全过程；
- 定义数据库的安全性和完整性约束条件；
- 决定数据库的存储和读取策略；
- 监督控制数据库的使用和运行并及时处理运行程序中出现的问题；
- 改进数据库系统和重组数据库。

数据库系统结构如图 1-1 所示。

任务 1.2　数据库管理系统——DBMS

子任务 1.2.1 DBMS 的功能

DBMS 是数据库系统的一个重要组成部分，是位于用户与操作系统之间的一层数据管理软件。它的主要功能有以下几个方面。

1. 数据定义功能

DBMS 提供数据定义语言（Data Definition Language，DDL），其可以定义数据库的结构、数据库的完整性约束条件和保证完整性的触发机制等。

笔记

2. 数据操纵功能

DBMS 提供数据操纵语言（Data Manipulation Language，DML）。用户可以使用数据操纵语言操纵数据，从而实现对数据库中数据的查询、插入、修改、删除等基本操作。国际标准数据库操作语言——SQL 语言，就是数据操纵语言的一种。

3. 数据库控制系统

DBMS 提供一系列系统运行控制程序，负责在数据库运行过程中对数据库进行管理和控制，其主要表现在以下几个方面：①当许多用户同时访问数据库时，协调各个用户的访问；②对数据库进行安全检查，核对用户标识、口令，对照授权表检验访问的合法性等；③对数据库进行完整性约束条件的检查和执行，在对数据库进行操作之前或之后，核对数据库完整性约束条件，从而决定是否运行操作执行或清除操作执行后的影响；④对数据库的内部维护如索引、数据字典的自动维护等。所有访问数据库的操作都要在这些控制程序的统一管理下进行，以保证数据的正确有效。

4. 数据库的建立与维护功能

除了上述几个方面的功能外，DBMS 还提供一些理性维护公用程序，主要体现为：①负责数据库初始数据的输入；②记录工作日志；③监视数据库性能；④在性能变差时重新组织数据库；⑤在用户要求或系统设备发生变化时修改和更新数据库；⑥在系统软硬件发生故障时恢复数据库。

子任务 1.2.2 DBMS 的组成

DBMS 主要由以下组件组成。

1. 数据定义语言及其翻译处理程序

DBMS 一般都提供数据定义语言供用户定义数据库的各种模式，翻译程序负责将它们翻译成相应的内部表示，即生成目标模式。

2. 数据操纵语言及其编译（或解释）程序

DBMS 提供了数据操纵语言以实现对数据库的检索、插入、修改、删除等基本操作。DML 分为宿主型 DML 和自主型 DML 两类。

3. 数据库运行控制程序

DBMS 提供了一些系统运行控制程序负责数据库运行过程中的控制与管理，它们负责监视数据库运行过程中有关数据的所有操作，控制、管理数据库资源，处理多用户的并发操作等。

4. 实用程序

DBMS 提供一些实用程序，数据库用户可以利用这些实用程序完成数据库的建立与维护，以及数据格式的转换与通信。

任务 1.3　关系数据模型

提到模型我们自然会联想到建筑模型、飞机模型等事物。广义地说，模型是对现实世界特征的模拟和抽象。在数据库中，用数据模型（Data Model）对现实世界进行抽象。数据模型应满足以下 3 个方面的要求：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。

笔记

在数据库系统中，应针对不同的使用对象和应用目的，采用不同的数据模型。不同的数据可以提供给用户模型化的数据和信息。根据模型应用的目的，可以将数据模型分为两种类型：一类是概念模型，也称为信息模型。它是独立于计算机之外的模型，如实体—联系模型，这种模型不涉及信息在计算机中如何表示，而是用来描述某一特定范围内人们所关心的信息结构，它是按照用户的观点对数据和信息建模，主要用于数据库的设计。另一类是数据模型，它是直接面向计算机的，是按照计算机系统的观点对数据进行建模，主要用于 DBMS 的实现，通常称为基本数据模型或数据模型。数据库中基本数据模型有网状模型、层次模型和关系模型等。

子任务 1.3.1 概念模型

为了把现实世界的具体事物或事物之间的联系表示成 DBMS 所支持的数据模型，人们必须将现实世界的事物及其之间的联系进行抽象后转换为信息世界的概念模型，再将信息世界的概念模型转换为机器世界的数据模型，也就是说，首先，把现实世界的客观对象抽象成一种信息结构，这种信息结构并不依赖于具体的计算机系统和 DBMS。其次，把概念模型转换为某一计算机系统上的某一 DBMS 所支持的数据模型。因此，概念模型是从现实世界到机器世界的一个中间层次。它是整个数据模型的基础。

下面介绍有关概念模型的几个术语。

1. 实体（Entity）

客观存在并可相互区别的事物称为实体。实体可以是人，也可以是物；可以是实际对象，也可以是概念；可以是事物本身，也可以是指事物之间的联系，如一个学生、一辆轿车、一张椅子、一个部门等。

2. 属性（Attribute）

实体所具有的每个特性称为属性。例如，学生实体可以由学号、姓名、专业名、性别、出生日期、身高等属性组成。比如，101101、林琳、计算机软件、男、1991—8—10、175.5，这些属性组合起来表征了一个学生。

每个属性都有一个取值范围，称为该属性的值域。值域的类型可以是整型、实型或字符型等。例如，学号的值域为由若干位数字构成的字符串集合，姓名的值域为字符串集合，年龄的值域为整数，性别的值域为（男，女）。

3. 关键字（Key）

能唯一地标识实体属性的集合称为关键字（或码）。例如，学生的学号就是学生实体的码。

4. 域（Domain）

属性的取值范围称作域。例如，学号和姓名的域为字符串集合，性别的域为（男，女）。

5. 实体型（Entity Type）

一类实体所具有的共同特征或属性的集合称为实体型。

一般用实体名及其属性来抽象地刻画一类实体的实体型。例如，学号、姓名、专业名、性别、出生日期、身高等就是一个实体型。

6. 实体集（Entity Set）

同型实体的集合叫作实体集。例如，全体学生、所有汽车、所有学校、所有课程、所

有零件都称为实体集。由此可以看到，事物的若干属性值的集合可表征一个实体，而若干属性所组成的集合可表征一个实体的类型，简称为“实体型”，同类型的实体集合组成实体集。

笔记

7. 联系（Relationship）

现实世界的事物之间是有联系的。一般存在两类联系：一类是实体内部组成实体的属性之间的联系；另一类是实体之间的联系。在考虑实体内部的联系时，可将属性看作实体。一般来说，两个实体之间的联系可分为三种。

（1）一对一（1∶1）联系。若对于实体集 A 中的每一个实体，实体集 B 中至多有唯一的一个实体与之联系，反之亦然，则称实体集 A 与实体集 B 具有一对一联系，记作 1∶1，如图 1-2 所示。

例如，在学校里，一个班只有一个正班长，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。观众与座位、乘客与车票、病人与病床、学校与校长之间都是一对一的联系。

（2）一对多（1∶n）联系。若对于实体集 A 中的每个实体，实体集 B 中有 n 个实体（$n \geqslant 0$）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，记作 1∶n，如图 1-3 所示。

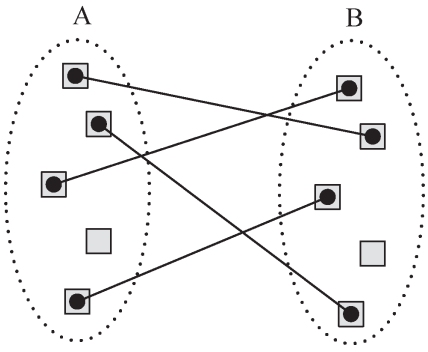


图 1-2　一对一联系

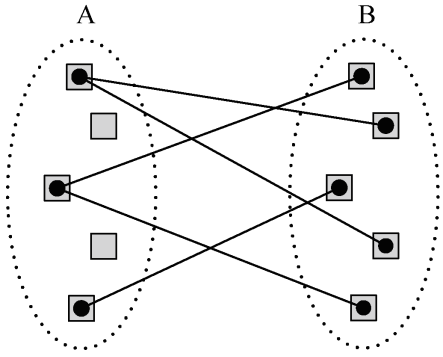


图 1-3　一对多联系

例如，一个班级中有若干名学生，而一个学生只能在一个班级中学习，则班级与学生之间具有一对多的联系，宿舍与学生之间也是一对多的联系。

（3）多对多（m∶n）联系。若对于实体集 A 中的每一个实体，实体集 B 中有 n 个实体（$n \geqslant 0$）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（$m \geqslant 0$）与之对应，则称实体集 A 与实体集 B 具有多对多联系，记作 m∶n，如图 1-4 所示。

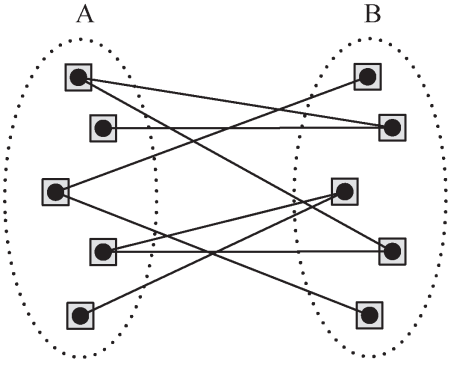


图 1-4　多对多联系

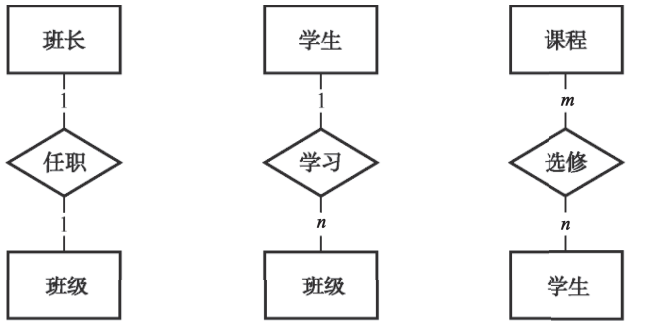
例如，一个学生可以选修若干门课程，而一门课程也可以由若干名学生选修，所以学生与课程之间就是多对多的联系。

概念模型的表示方法最常用的是实体—联系方法（Entity-Relationship Approach），简称 E-R 方法。该方法是由 P.P.S.Chen 在 1976 年提出的。E-R 方法用 E-R 图来描述某一组织的概念模型，在这里仅介绍 E-R 图的要点。在 E-R 图中：

笔记

- 长方形框表示实体集，框内写上实体型的名称。
- 椭圆框表示实体的属性，并用无向边把实体框及其属性框连接起来。
- 菱形框表示实体间的联系，框内写上联系名，用无向边把菱形框及其有关的实体框连接起来，在旁边标明联系的种类（1∶1，1∶n 或 m∶n）。如果联系也具有属性，则把属性框和菱形框也用无向边连接上。

例如，学生、班长、班级、课程联系的 E-R 模型表示如图 1-5 所示。



（a）一对一模型　（b）一对多模型　（c）多对多模型

图 1-5　实体间的联系

【例 1.1】用 E-R 图来表示工厂库房管理系统的概念模型。

该库房信息如下所示。

- 库房：库房号、库房名、库房容量。
- 零件：零件号、零件名、规格型号。
- 职工：职工号、职工名、工种。

其中，每个库房有若干职工，每个职工只能在一个库房工作；每个库房可存放若干种零件，每种零件可存放在不同的库房中。

因此，给系统建立概念模型其实就是为其设计对应的 E-R 模型图，其关键步骤如下。

第一步，确定实体和实体的属性，如图 1-6 所示。

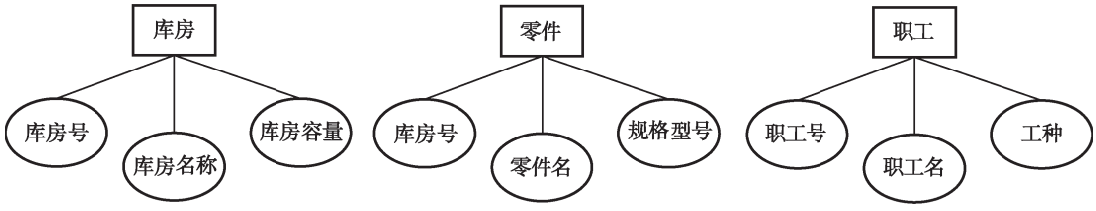


图 1-6　实体及其属性 E-R 图

第二步，确定实体之间的联系及联系的类型，如图 1-7 所示。

第三步，给实体和联系加上属性，如图 1-8 所示。

实体和属性之间并没有可以截然划分的界限，如何划分实体与属性有两个原则可作参考：一是作为实体属性的事物本身没有再需要刻画的特征，即属性不能再有属性来描述；二是属性不能与其他实体具有联系，联系只发生在实体之间。凡是满足上面两条原则的事物，一般可作为属性对待。

笔记

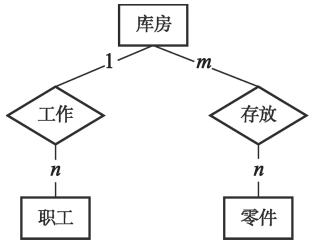


图 1-7　实体及其联系 E-R 图

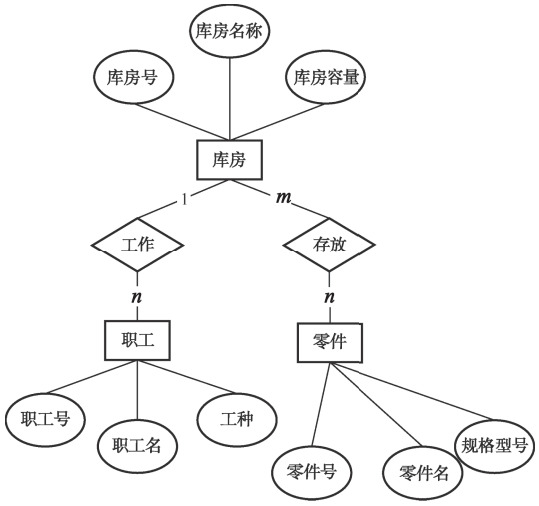


图 1-8　实体及其联系属性 E-R 图

例如，在图 1-9 中，职称没有属性来描述，并且与实体之间没有发生联系，所以将职称作为职工的属性。而在图 1-10 中，职称与工资、福利等挂钩，即它还需要工资、福利等属性来描述，所以将它作为实体。

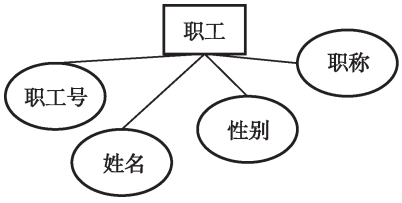


图 1-9　职称作为属性

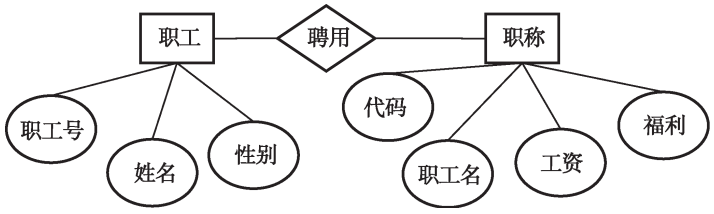


图 1-10　职称作为实体

如何划分实体和联系也有一个可供参考的原则：当描述发生在实体集之间的行为时，最好采用联系集。例如，读者和图书之间的借、还书行为，顾客和商品之间的购买行为，都应该作为联系。

划分联系的属性应遵循如下原则：与联系中的所有实体都有关的属性应作为联系的属性。例如，学生和课程的选课联系中的成绩属性，顾客和商品之间的销售联系中的数量等属性都应该作为联系的属性。

【例 1.2】假设某公司在多个地区设有分公司来经销本公司的各种产品，每个分公司聘用多名职工，且每名职工只属于一个分公司。而且，分公司有公司名称、地区和联系电话等属性，产品有产品编码、名称和价格等属性，职工有职工编号、姓名和性别等属性，每个分公司销售产品有数量属性。根据上述的语义画出 E-R 图。

根据题意可确定实体为：分公司、职工和产品。分公司与职工之间为 1∶*n* 的联系，分公司与产品之间是 *m*∶*n* 的联系。数量应该作为分公司与产品之间联系的属性。其 E-R 图如图 1-11 所示。

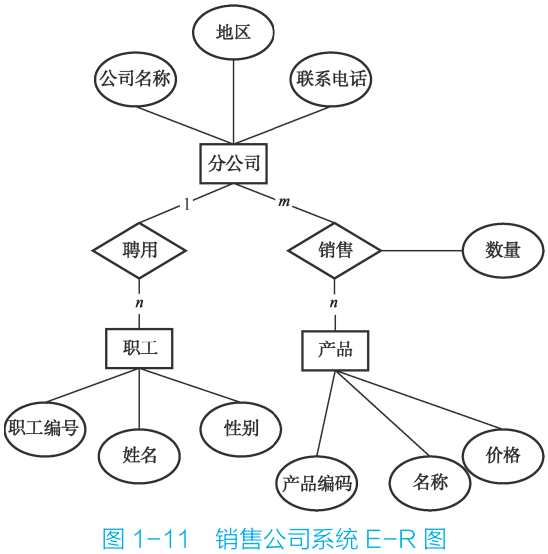


图 1-11　销售公司系统 E-R 图

笔记

子任务 1.3.2 数据模型

数据模型是数据库系统中关于数据和联系的逻辑组织的形式表示。每一个具体的数据库都由一个相应的数据模型定义。每一种数据模型都以不同的数据抽象与表示能力来反映客观事物，有其不同的处理数据联系的方式。数据模型的主要任务就是研究记录类型之间的联系。数据模型是数据库系统的核心和基础，每个 DBMS 软件都是基于某种数据模型开发的。

数据模型有 3 个要素分别为数据结构、数据操作和完整性规则。数据结构用于描述系统的静态特性，人们通常按照其数据结构的类型来命名数据模型。数据操作用于描述数据的动态特征。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以及状态的变化。

目前，数据库领域采用的数据模型有层次模型、网状模型、关系模型和面向对象的数据模型，其中应用最广泛的是关系模型。

1. 层次模型

用树形结构表示实体之间联系的模型叫作层次模型。层次模型是最早用于商品数据库管理系统的数据模型。其典型代表是于 1969 年问世的、由 IBM 公司开发的信息管理系统（Information Management System，IMS）。在现实世界中、许多实体集之间的联系就是一个自然的层次关系。例如，行政机构、家族关系等都是层次关系。图 1-12 展示的就是学校中的部门层次模型。

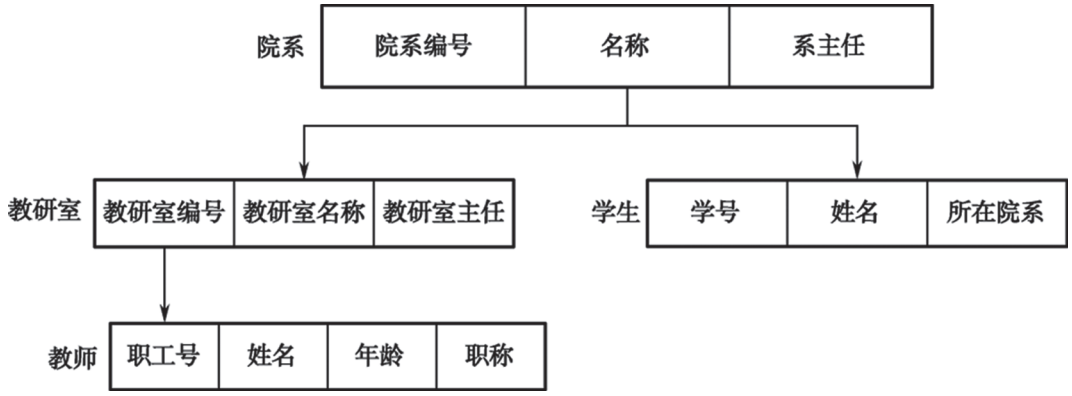


图 1-12　层次模型示例

层次模型的表示方法是：树的节点表示实体集（记录的型），节点之间的连线表示相连两实体集之间的关系，这种关系只能是“1 ： M”的。通常把表示“1”的实体集放在上方，称为父节点；把表示“M”的实体集放在下方，称为子节点。层次模型的结构特点有以下两点：

（1）有且仅有一个根节点。

（2）根节点以外的其他节点有且仅有一个父节点。

因此，层次模型只能表示“1 ： M”关系，而不能直接表示“M ： N”关系。

在层次模型中，一个节点称为一个记录型，用来描述实体集。每个记录型可以有一个或多个记录值，上层的一个记录值对应下层的一个或多个记录值，而下层的一个记录值只能对应上层的一个记录值。

笔记

2. 网状模型

网络模型的典型代表是数据模型（Database Task Group，DBTG）。DBTG 是美国 CODASYL（Conference on Data System Language）组织的下属机构，它于 1971 年提出 DBTG 报告。DBTG 报告是一个网络模型的数据描述语言和数据操纵语言的规范文本。

网状模型（Network Model）是一种更具普遍性的结构，从图论的角度讲，它是一个不加任何条件限制的无向图。

网状模型是以记录为节点的网状结构，它满足了以下 3 个条件：①可以有任意个节点无双亲；②允许节点有一个以上的双亲；③允许两个节点之间有一种或两种以上的联系。

在网状模型的 DBTG 标准中，基本结构是简单二级树，称作系。系的基本数据单位是记录，它相当于 E-R 模型中的实体集；记录由若干数据项组成，它相当于 E-R 模型中的属性。图 1-13 为教师授课数据库的网状模型。

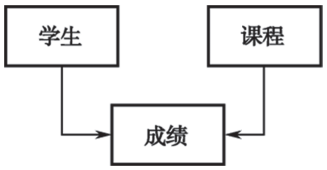


图 1-13　教师授课网状模型示例

网状模型明显优于层次模型，其在一定程度上支持数据的重构，具有一定的数据独立性和共享特性，并且运行效率较高，但它在应用时存在以下问题。

（1）网状结构较为复杂，增加了用户查询和定位的困难。它要求用户熟悉数据的逻辑结构，知道自身所处的位置。

（2）网状数据操作命令具有过程式性质。

（3）不直接支持对于层次结构的表达。

3. 关系模型

网状数据库和层次数据库已经很好地解决了数据的集中和共享问题，但是在数据独立性和抽象级别上仍有很大欠缺。用户在对这两种数据库进行存取时，仍然需要明确数据的存储结构，指出数据的存取路径。而后来出现的关系数据库较好地解决了这些问题。关系数据库理论出现于 20 世纪 60 年代末 70 年代初。1970 年，IBM 的研究员 E.F.Codd 博士在发表的“大型共享数据库数据的关系模型”一文中提出了关系模型的概念。

目前，关系模型是数据库领域最为重要的一种数据模型。关系模型的本质是一张二维表。在关系模型中，一张二维表就称为一个关系，如表 1-1 所示。自 20 世纪 80 年代以来，计算机厂商推出的 DBMS 几乎都是关系数据库，如 Oracle、Sybase、Informix、MS SQL Server、Visual FoxPro 等。

表 1-1　2013 年某地大学学生注册数量

学　院	新　生	毕 业 生	变　动
重庆大学	210	204	+6
邮电学院	123	113	+10
交通大学	184	121	+63
五一研究所	108	110	-2

注：虚构数据，仅用作图表示例。

（1）关系模型的优点。关系模型是应用最广泛的一种数据模型，它具有以下优点：

① 能够以简单、灵活的方式表达现实世界中各种实体及其相互间的关系，使用与维护也很方便。关系模型通过规范化的关系为用户提供了一种简单的用户逻辑结构。

笔记

② 关系模型具有严密的数学基础和操作代数基础，如关系代数、关系演算等，可将关系分开，或将两个关系合并，使数据的操纵具有高度的灵活性。

③ 在关系模型中，数据间的关系具有对称性，因此，关系之间的寻找在正、反两个方向上的难易程度是一样的，而在其他模型如层次模型中，从根节点出发寻找叶子的过程容易解决，而相反的过程则很困难。

（2）关系模型的缺点。目前，绝大多数数据库系统采用关系模型，但它也存在如下问题：

① 实现效率不高。

② 描述对象语义的能力较弱。

③ 不能直接支持层次结构，因此不能直接支持对于概括、分类和聚合的模拟，即不满足管理复杂对象的要求，不允许嵌套元组和嵌套关系存在。

④ 模型的可扩充性较差。

⑤ 模拟和操纵复杂对象的能力较弱。

4. 面向对象的数据模型

数据库的发展集中表现为数据模型的发展。从最初的层次模型、网状模型发展到关系模型，数据库技术产生了巨大的飞跃。关系模型的提出，是数据库发展史上具有划时代意义的重大事件。然而，20 世纪 80 年代，随着数据库应用领域对数据库需求的增多，传统的关系数据模型开始暴露出许多弱点。如今，许多应用程序都要求有操纵声音、视频和图像数据的能力。传统的数据管理系统并不能满足这样的要求，因为这些数据类型不便于用行和列的二维表进行存储，如 CAD 数据、图形数据等。这些数据需要更高级的数据库技术表达，所以出现了面向对象的数据模型。

面向对象的数据模型采用面向对象程序设计方法的核心概念和基本思想，其包括以下几点：

（1）对象。一切可识别的实体。

（2）对象标识。现实世界中的任何实体都可统一地用对象表示，每一个对象都有唯一的标识，称为对象标识（Object Identifier，OID）。就像商品都有唯一的条形码一样，这在关系数据库中也有。OID 与对象的物理存储位置无关，也与数据的描述方式和值无关。

（3）封装。每一个对象是其状态和行为的封装。面向对象技术是把数据和行为封装在一起，使得数据应用更灵活。从对象外部看，对象的状态和行为是不可见的，只能通过显示定义的消息传递来存取。

（4）类。把类似的对象归并在一起，称为类，类中每个对象称为实例，同一类的对象具有相同的实例变量和方法，可在类中统一说明，而不必在类的每个实例中重复说明，这样就减少了信息冗余。

面向对象数据模型中类的概念相当于 E-R 模型中实体集的概念。

（5）继承性。继承性允许不同类的对象共享它们公共部分的结构和特性。继承性可以用超类和子类的层次联系实现。一个子类可以继承某一个超类的结构和特性，称为单继承性；一个子类也可以继承多个超类的结构和特性，称为多继承性。

（6）消息。由于对象是封装的，对象与外部的通信一般只能通过显示的消息传递，即消息从外部传送给对象，存取和调用对象中的属性和方法，在内部执行所要求的操作，操

作的结果仍以消息的形式返回。

面向对象的模型不但继承了关系数据库的许多优良性能，还能处理多媒体数据，并支持面向对象的程序设计。因此，它已成为目前数据库中最有前途和生命力的发展方向。

笔记

任务 1.4　关系的完整性约束

关系模型有三类完整性约束条件：实体完整性约束、参照完整性约束和用户定义完整性约束。这三类约束条件中前两类是关系模型必须满足的完整性约束条件，由关系系统自动支持，而后一类约束条件是用户针对特定的数据库设置的约束条件。

子任务 1.4.1 实体完整性约束

在关系模式中，能唯一标识一个元组的属性或属性组称为候选码，选中其中一个为主码，而包含在任何一个候选码中的属性称为主属性，不包含在任何候选码中的属性称为非主属性。

实体完整性规则：若属性（指一个或一组属性）A 是基本关系 R 的主属性，则属性 A 不能取空值。

这个规则很容易理解，因为主码能唯一标识关系中的元组，若构成主码的主属性取空值（“不知道”或“无意义”的值），便失去其唯一标识功能。例如，关系模式为：学生（学号、姓名、性别、年龄、籍贯、专业名称），其中学号是主码，而主码对应的属性只有学号，所以学号也是主属性。根据实体完整性约束规则，学号不能取空值。若学号取空值，那么这个元组就没有意义了。在学生选课关系模式——选修（学号、课程编码、成绩）中，属性组“学号，课程编码”为主码，所以“学号”和“课程编码”的属性均不能取空值。

实体完整性约束规则是针对基本关系而言的，即针对现实世界的一个实体集，而现实世界中的实体是可区分的。该规则的目的是利用关系模式中的主码或主属性来区分现实世界中实体集中的实体，所以主属性不能取空值。

子任务 1.4.2 参照完整性约束

在关系模型中，实体与实体之间的联系同样采用关系模式来描述。通过引用对应实体关系模式的主码来表示对应实体之间的联系。

例如，关系模式：

部门（部门编码、部门名称、电话、办公地址）

职工（职工编码、姓名、性别、年龄、籍贯、所属部门编码）

其中，职工关系模式中的“所属部门编码”与部门关系模式中的主码“部门编码”相对应，所以“所属部门编码”是职工关系模式中的外码。职工关系模式通过外码来描述与部门关系模式的关联。职工关系中的每个元组（每个元组描述一个职工实体）通过外码表示该职工所属的部门。当然，被参照关系的主码和参照关系的外码可以同名，也可以不同名；被参照关系与参照关系可以是不同关系，也可以是同一关系。

例如：

职工（职工编码、姓名、性别、年龄、籍贯、所属部门编码、班组长编码）

笔记

其中的“班组长编码”与本身的主码“职工编码”相对应，属性“班组长编码”是外码，职工关系模式既是参照关系也是被参照关系。

参照完整性约束规则：若属性 F 是基本关系 R 的外码，且 F 与基本关系 S 的主码 K 相对应，则对于 R 中每个元组在 F 上的值必须为：或者取空值，或者等于 S 中某个元组的主码值。

在职工关系中，某一个职工“所属部门编码”或者取空值，表示该职工尚未分配到指定部门；或者等于部门关系中某个元组的“部门编码”，表示该职工隶属于指定部门。若既不为空值，又不等于被参照关系——部门中某个元组的“部门编码”分量，则表示该职工被分配到一个不存在的部门，就违背了参照完整性规则。所以，参照完整性规则就是定义外码与主码之间的引用规则，也是关系模式之间的关联规则。

子任务 1.4.3 用户定义完整性约束

用户定义完整性是针对某一具体数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求，关系模型应提供定义和检验这一类完整性机制，以便用统一的系统方法处理它们，而不是由应用程序来承担这一功能。

例如，在职工关系中，职工年龄的取值范围应该限定为 18~60，学生选课的成绩取值范围应该限定为 0~100。而关系模型应该为用户提供定义和检验这一类完整性的约束机制，以保证数据的正确性。