

第 1 章 数据库基础概述

当今社会是一个信息化的社会,信息已经成为各行各业的重要资源。数据是信息的载体,数据库是相互关联的数据集合。数据库能利用计算机保存和管理大量复杂的数据,快速而有效地为多个不同的用户和应用程序提供数据,帮助人们有效利用数据资源。目前,数据库应用已遍及生活中的各个角落,例如,学校的教学管理系统、图书馆的图书借阅系统、车站及航空公司的售票系统、电信局的计费系统、超市的售货系统、银行的业务系统、工厂的管理信息系统等。

数据库技术已经成为先进信息技术的重要组成部分,是现代计算机信息系统及计算机应用系统的基础和核心。因此,掌握数据库技术是全面认识计算机系统的重要环节,也是适应信息化时代的重要基础。

本章主要介绍数据库系统的基本概念、数据模型、数据库系统结构、关系数据理论和数据库系统体系结构。

1.1 数据库系统概述

在系统地介绍数据库的概念之前,这里首先介绍数据库最常用的一些基本概念。

1.1.1 数据管理技术的产生和发展

数据是现实世界中实体(或客体)在计算机中的符号表示。数据不仅可以是数字,还可以是文字、图表、图像、声音等。每个组织都保存了大量复杂的数据。例如,银行有关储蓄存款、贷款业务、信用卡管理、投资理财等方面的数据;医院有关病历、药品、医生、病房、财务等方面的数据;超市有关商品、销售情况、进货情况、员工等方面的信息。数据是一个组织的重要资源,有时甚至比其他资源更珍贵,因此必须对组织的各种数据实现有效管理。数据管理是指对数据的分类、组织、编码、存储、检索和维护等操作。数据库的核心任务就是数据管理。

数据库技术并不是最早的数据管理技术。在计算机诞生的初期,计算机主要用于科学计算,虽然当时同样存在数据管理的问题,但当时的数据管理是以人工的方式进行的,后来发展到文件系统,再后来才是数据库。也就是说,数据管理主要经历了人工管理阶段、文件系统阶段和数据库系统阶段。

1.1.1.1 人工管理阶段

人工管理阶段是指计算机诞生的初期(20 世纪 50 年代中期以前)。这个时期的计算机技术,从硬件看还没有磁盘这种可直接存取的存储设备,从软件看还没有操作系统,更

没有管理数据的软件。在人工管理阶段,程序与数据之间的对应关系如图 1-1 所示。

这个阶段数据管理的特点是:

(1) 数据不保存。因为计算机主要用于科学计算,通常不需要长期保存数据,只是在要完成某一个计算或课题时才输入数据,不仅原始数据不保存,计算结果也不保存。

(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。应用程序中不仅要规定数据的逻辑结构,而且要设计物理结构,包括存储结构存取方法、输入方式等,因此程序员负担很重。

(3) 数据不共享。数据是面向应用的,一组数据只能对应一个程序。当多个应用程序涉及某些相同的数据时,由于必须各自定义,无法相互利用、相互参照,因此程序与程序之间有大量的冗余数据。

(4) 数据不具有独立性。数据的逻辑结构或物理结构发生变化后,必须对应用程序做相应的修改,这进一步加重了程序员的负担。

1.1.1.2 文件系统阶段

文件系统阶段是指 20 世纪 50 年代后期到 60 年代中期这一阶段。在这个阶段,计算机不仅大量用于科学计算,也开始大量用于信息管理。像磁盘这样直接存取的存储设备已经出现,在软件方面也有了操作系统和高级语言,有专门用于数据管理的软件——文件系统(或操作系统的文件管理部分)。在文件系统阶段,程序与数据之间的对应关系如图 1-2 所示。

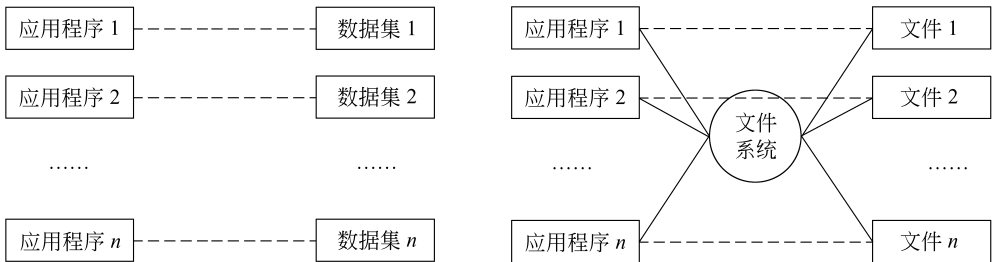


图 1-1 人工管理阶段程序与数据的对应关系 图 1-2 文件系统阶段程序与数据之间的对应关系

这个阶段数据管理的特点是:

(1) 由于存储设备的出现,数据可以长期保存在磁盘上,也可以反复使用,即可以对文件进行查询、更新和删除等操作。

(2) 操作系统提供了文件管理功能和访问文件的存取方法,程序和数据之间有了数据存取的接口,程序开始通过文件名和数据打交道,可以不再关心数据的物理存放位置。因此,这时也有了数据的物理结构和逻辑结构的区别。程序和数据之间有了一定的独立性。

(3) 文件的形式多样化。由于有了磁盘这样的直接存取存储设备,文件也就不再局限于顺序文件,也有了索引文件、链表文件等。因此,对文件的访问可以是顺序访问,也可以是直接访问;但文件之间是独立的,它们之间的联系要通过程序去构造,文件的共享性也比较差。

(4) 数据的存取基本上以记录为单位。

尽管文件系统阶段较手工阶段已经有了长足进步,但仍然存在如下缺陷:

(1) 数据冗余度大。由于文件都是为特定的用途设计的,因此会造成同样的数据在多个文件中重复存储,导致数据冗余度大,容易造成数据的不一致。

(2) 数据独立性差。应用程序是根据文件结构编写的,文件结构一旦改变,应用程序也必须随之进行修改,程序和数据之间的独立性较差。

(3) 数据联系弱。文件与文件之间是独立的,文件之间的联系必须通过程序来构造。因此,文件是一个不具有弹性的、无结构的数据集合,不能反映现实世界事物之间的联系。

1.1.1.3 数据库系统阶段

数据库系统阶段是指 20 世纪 60 年代后期至今。在这个阶段,计算机主要用于大规模的管理,涉及的数据量急剧增长。这时硬件已有大容量磁盘,价格下降;软件则价格上升,为编制和维护系统软件及应用程序所需的成本相对增加;在处理方式上,联机实时处理要求更多,并开始提出和考虑分布处理。在这种背景下,以文件系统作为数据管理手段已经不能满足应用的需求,于是为解决多用户、多应用共享数据的需求,使数据为尽可能多的应用服务,数据库技术应运而生,出现了统一管理数据的专门软件系统——数据库管理系统。在数据库系统阶段,程序与数据之间的对应关系如图 1-3 所示。

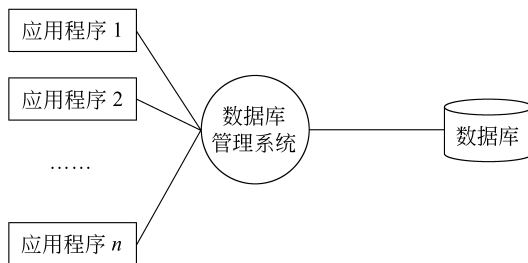


图 1-3 数据库系统阶段程序与数据的对应关系

数据库是指长期存储在计算机存储设备上的相互关联的、可以被用户共享的数据集合。用数据库系统来管理数据比文件系统具有明显的优点,从文件系统到数据库系统,标志着数据管理技术的飞跃。

数据库系统主要有以下优点。

1. 数据库是相互关联的数据的集合

现实世界的信息是相互关联的,用数据库管理数据可以将数据之间的关联关系也体现出来。数据库中的数据不是孤立的,数据与数据之间是相互关联的。在数据库中不仅能反映数据本身,还能反映数据与数据之间的联系。例如,在银行贷款管理中,数据库中不仅要存放银行和法人两类数据,还要存放哪些法人在哪些银行进行贷款的信息,这就反映了银行数据与法人数据之间的联系。

2. 具有较小的数据冗余

在文件系统中,每个应用都拥有各自的文件,即同一数据可能存放在不同的文件中,

这带来了大量的数据冗余。在数据库系统中,数据成为统一的逻辑结构,每一个数据项的值可以只存储一次,从而最大限度地控制了数据冗余。所谓控制数据冗余,是指数据库系统可以把数据冗余限制在最少,系统也可保留必要的数据冗余。事实上,由于应用业务或技术上的原因,如数据合法性检验、数据存取效率等方面的需要,同一数据可能在数据库中保持多个副本。但是在数据库系统中,冗余是受控的。系统知道冗余,保留必需的冗余也是系统预定的。

3. 具有较高的数据独立性

数据独立性是指数据的组织和存储方法与应用程序互不依赖、彼此独立的特性。在数据库技术之前,数据文件的组织方式和应用程序是密切相关的,当改变数据结构时,相应的应用程序也随之修改,从而极大地增加了应用程序的开发代价和维护代价。数据库技术可以使数据的组织和存储方法与应用程序互不依赖,从而大幅降低应用程序的开发代价和维护代价。

4. 具有安全控制机制,能够保证数据的安全、可靠

数据库技术要能够保证数据库中的数据是安全的、可靠的。数据库要有一套安全机制,以便有效地防止数据库中的数据被非法使用或非法修改;数据库还要有一套完整的备份和恢复机制,以便保证当数据遭到破坏时(软件或硬件故障引起的),能够立刻完全恢复数据,从而保证系统能够连续、可靠地运行。

5. 最大限度地保证数据的正确性

保证数据正确的特性在数据库中称为数据完整性。可以在数据库中通过建立一些约束条件来保证数据库中的数据是正确的。例如,北京市电话的区号是 010,当输入 027-56785436 时,数据库能够主动拒绝这类错误。

6. 数据可以共享并能保证数据的一致性

数据库中的数据是共享的,允许多个用户同时使用相同的数据,并能保证各个用户之间对数据的操作不发生矛盾和冲突,即在多个用户同时使用数据库时,能够保证数据的一致性和正确性。

1.1.2 数据库系统的组成

数据库系统(DataBase System,DBS)是指在计算机系统中引入数据库后的系统,一般由数据库(DataBase,DB)、数据库管理系统(DataBase Management System,DBMS)及其开发工具、应用系统、数据库用户和管理员(DataBase Administrator,DBA)构成。其中,数据库是系统的核心,DBMS 及其开发工具和数据库管理员是系统的基础,应用系统和用户是系统服务的对象。在不引起混淆的情况下,数据库系统常常简称为数据库。

数据库系统的组成如图 1-4 所示。数据库系统在整个计算机系统的位置如图 1-5 所示。

数据库系统对硬件的要求除了一般计算机系统对硬件的要求外,还要求有足够大的内存,以便存放操作系统、DBMS 的核心模块、数据缓冲区和应用程序;有容量足够大的磁盘等直接存取设备,以便存放数据和作数据备份;有较高的数据传输速率。

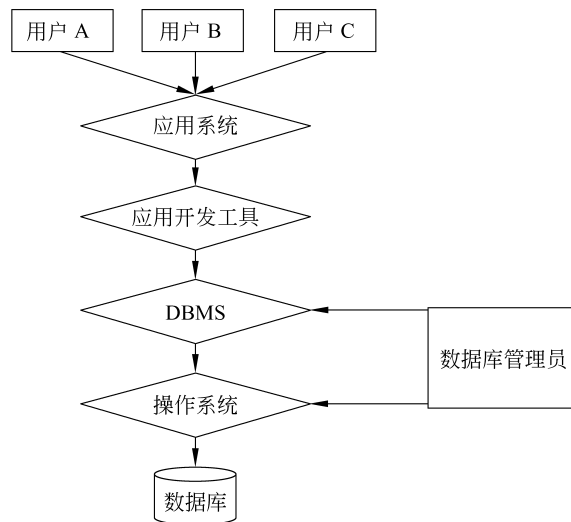


图 1-4 数据库系统的组成

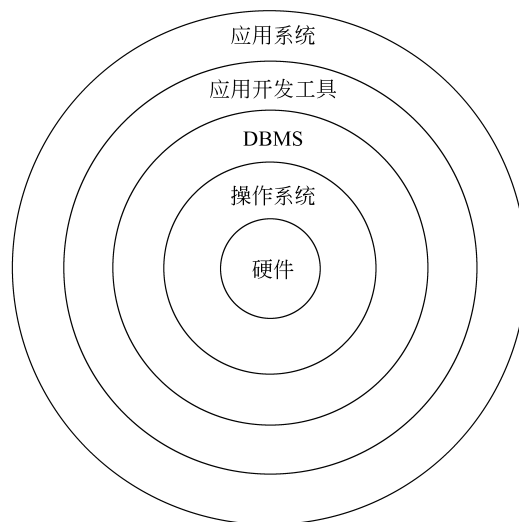


图 1-5 数据库系统在整个计算机系统的位置

数据库系统的软件主要有 DBMS、支持 DBMS 运行的操作系统、具有与数据库接口的高级程序设计语言及其编译程序、以 DBMS 为核心的应用开发工具和为特定应用开发的数据库应用系统。

数据库用户和数据库管理员是数据库系统的重要组成部分，它们的作用是开发、管理和使用数据库系统。不同人员涉及不同的数据抽象级别，对应不同的数据视图。

1.1.3 数据库管理系统

1.1.1 节提到的数据库的各种功能和特性，并不是数据库中的数据固有的，而是靠管

理或支持数据库的系统软件——数据库管理系统提供的。一个完备的数据库管理系统应该具备 1.1.1 节提到的各种功能,它的任务就是对数据资源进行管理,并且使之能为多个用户共享,同时保证数据的安全性、可靠性、完整性、一致性和高度独立性。

具体来说,一个数据库管理系统应该具备如下功能。

(1) 数据定义功能:定义数据库的结构和数据库的存储结构、定义数据库中数据之间的联系、定义数据的完整性约束条件等。

(2) 数据操纵功能:对数据库中的数据进行查询、插入、删除和修改操作。

(3) 数据维护功能:为了提高数据库的性能,可以重新组织数据库的存储结构,为了保证数据库的安全性和可靠性,可以完成数据库的备份和恢复等。

(4) 数据控制功能:实现对数据库的安全性控制、完整性控制、多用户环境下的并发控制等各方面的控制。

(5) 数据通信功能:在分布式数据库或提供网络操作功能的数据库中还必须提供数据的通信功能。

(6) 数据服务功能:数据库不是一个孤立的系统,通常可以被其他软件访问,可以与其他系统交换数据;数据库中的数据也不是简单的操纵和查询,还可以提供很多服务,如数据分析服务等。

1.2 数据模型

数据库中不仅存储数据本身,还要存储数据与数据之间的联系,这种数据及其联系是需要描述和定义的,数据模型正是用于完成这一任务的。

1.2.1 数据模型的概念、分类及构成

1.2.1.1 概念

模型是对现实世界特征的模拟和抽象,它可以帮助人们描述和了解现实世界。人们可以将现实世界的物质抽象为模型。同时,看到模型,人们就能想象现实世界的物质。数

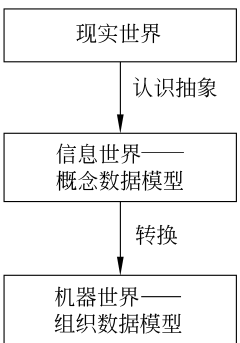


图 1-6 对现实世界的抽象过程

据模型(Data Model)也是一种模型,它是现实世界数据特征的抽象。设计数据库系统时,一般要求用图或表的形式抽象地反映数据彼此之间的关系,这被称为建立数据模型。现有的数据库系统都是基于某种数据模型的。

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。

计算机不能直接处理现实世界中的具体事物,所以人们必须把具体事物抽象并转换成计算机能够处理的数据。一般要经过两个阶段:首先将现实世界中的客观对象抽象为信息世界的概念数据模型;然后将信息世界的概念数据模型转换成机器世界的组织数据模型,如图 1-6 所示。

1.2.1.2 三个领域

为了能够很好地理解数据模型,下面先介绍现实世界、信息世界和机器世界这三个领域。

1. 现实世界

现实世界是存在于人们头脑外的客观世界。在现实世界中,事物之间不是相互孤立的,而是相互联系的。事物及其之间的联系正是建立数据库的原始数据。现实世界中的原始数据是错综复杂的,数据量是很大的。例如,银行贷款管理、企业人事管理、超市销售管理等。

2. 信息世界

信息世界是现实世界在人脑中的反映,它搜集、整理现实世界的原始数据,找出数据之间的联系和规律,并用形式化方法表示出来,实现人与人之间的信息交流。信息世界最主要的特征是可以反映数据之间的联系。

3. 机器世界

机器世界是数据库的处理对象。信息世界的信息经过加工、编码转换成机器世界的的数据,这些数据必须具有自己特定的数据结构,能反映信息世界中数据之间的联系。计算机能对这些数据进行处理,并向用户展示经过处理的数据。

1.2.1.3 数据模型的分类

在数据库系统中,针对不同的使用对象和应用目的,往往采用不同的数据模型。根据数据模型的不同应用目的,可以将这些模型划分为两类,它们分属于两个不同的层次。

第一类模型是概念数据模型。它面向现实世界,从数据的语义视角来抽取模型,按用户的观点对数据和信息建模,强调语义表达能力,建模容易、方便,概念简单、清晰,便于用户理解,是现实世界到信息世界的第一层抽象,是用户和数据库设计人员之间进行交流的语言。概念数据模型主要用在数据库的设计阶段,与 DBMS 无关。常用的概念数据模型是实体联系模型。

第二类模型是组织数据模型,也称为数据模型。它是一种基于记录的模型,主要包括网状模型、层次模型、关系模型等。组织数据模型是面向机器世界的,它按照计算机系统的观点对数据建模,从数据的组织层次来描述数据,一般与实际数据库对应。例如,网状模型、层次模型、关系模型分别与网状数据库、层次数据库和关系数据库对应,可以在机器上实现。这类模型有更严格的形式化定义,常需要加上一些限制或规定。组织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 软件都是基于某种组织数据模型的。

设计数据库系统时,通常利用第一类模型作初步设计,之后按一定方法转换为第二类模型,再进一步设计全系统的数据库结构,最终在计算机上实现。

1.2.1.4 数据模型的构成元素

数据模型包括三部分:数据结构、数据操作和数据的约束条件。

1. 数据结构

数据结构用于描述数据库系统的静态特性,包括数据库中数据的组成、特性及其相互间联系。在数据库系统中通常按数据结构的类型来命名数据模型,如网状结构、层次结构和关系结构的模型分别被命名为网状模型、层次模型和关系模型。

2. 数据操作

数据操作用于描述数据库系统的动态特性,是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。数据库的数据操作主要有查询、插入、删除和更新。数据模型要给出这些操作的确切定义、操作符号、操作规则及实现操作的语言。

3. 数据的约束条件

数据的约束条件也用于描述数据库系统的静态特性,是一组数据完整性规则的集合。它给定数据模型中数据及其联系所具有的制约依存规则,用于限定符合数据模型的数据库状态及其变化,以保证数据的完整性。

数据模型的这三方面内容完整描述了一个数据模型,而其中的数据结构是首要内容。

1.2.2 实体—联系模型

概念模型主要描述现实世界中实体以及实体和实体之间的联系。P. P. S. Chen 于 1976 年提出的实体—联系(Entity-Relationship, E-R)模型是支持概念模型的最常用方法。E-R 模型使用的工具称为 E-R 图,它描述的是现实世界的信息结构。

E-R 模型主要包含三个要素:实体、属性和联系。

1.2.2.1 实体

现实世界中所管理的对象称为实体(Entity)。实体的定义为:客观存在并可以相互区分的客观事物或抽象事件。例如,职工、学生、银行、桌子都是客观事物,球赛、上课都是抽象事件,它们都是现实世界管理的对象,都是实体。

在关系数据库中,一般一个实体被映射成一个关系表,表中的一行对应一个可区分的现实世界对象,称为实体实例(Entity Instance)。比如,“银行”实体中的每家银行都是“银行”实体的一个实例。这里所提到的实体在其他书中有用实体集或实体型表示的,这里所提到的实体实例在其他书中有用实体表示的。

在 E-R 图中用矩形框表示实体,在框内写上实体,图 1-7 分别显示了银行、雇员和球赛三个实体的 E-R 图表示。

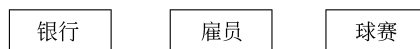


图 1-7 实体示例

1.2.2.2 属性

实体所具有的某一特性称为属性(Attribute)。一个实体可以由若干个属性来刻画。例如,雇员可以由雇员号、雇员名、工资和经理号来刻画。人们可以根据实体的特性来区分实体。但并不是每个特性都可以用来区分实体,因此又把用于区分实体的实体特性称为标识属性。例如,雇员的雇员号是标识属性,用雇员号可以区分一个个雇员,而工资则不是标识属性。

在 E-R 图中用椭圆框或圆角矩形框表示实体的属性,框内写上属性名,并用连线连到对应实体。可以在标识属性下加下画线。图 1-8 显示了用 E-R 图表示的雇员实体及其属性。

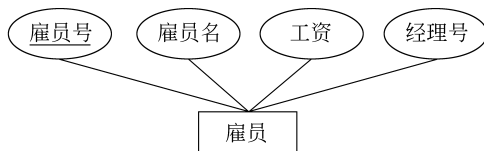


图 1-8 雇员实体及其属性

1.2.2.3 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。例如,图 1-8 中雇员实体的属性“雇员号”与“经理号”之间就有关联关系,即经理号的取值受雇员号取值的约束(因为经理也是雇员,也有雇员号),这就是实体内部的联系。实体之间的联系通常是指不同实体之间的联系。例如,在银行贷款管理信息系统中,银行实体和法人实体之间就存在“贷款”联系。我们主要研究的是实体之间的联系。实体之间的联系用菱形框表示,框内写上联系名,然后用连线与相关的实体相连。实体之间的联系按联系方式可分为三类。

(1) 一对一联系(1:1)

如果实体 A 与实体 B 之间存在联系,并且对于实体 A 中的任一个实例,实体 B 中至多有一个(也可以没有)实例与之关联,反之亦然,则称实体 A 与实体 B 具有一对一联系,记作 1:1。例如,飞机的乘客和座位之间、学校与校长之间都是 1:1 联系,要注意的是 1:1 联系不一定是一一对应,如图 1-9 所示。联系本身也有属性,图 1-9 中乘客与座位的联系——“乘坐”的属性为“乘坐时间”。

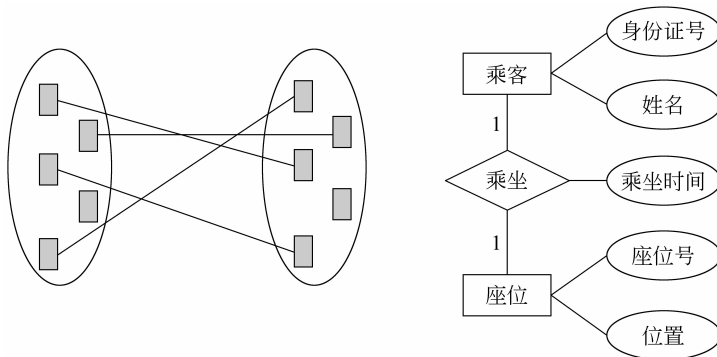


图 1-9 一对一联系示例

(2) 一对多联系(1:n)

如果实体 A 与实体 B 之间存在联系,并且对于实体 A 中的任一实例,实体 B 中有 n 个实例($n \geq 0$)与之对应;而对于实体 B 中的任一个实例,实体 A 中至多有一个实例与之对应,则称实体 A 到实体 B 的联系是一对多的,记为 1:n。例如,部门和职工之间、学校和

学生之间都是 $1:n$ 联系,如图 1-10 所示。

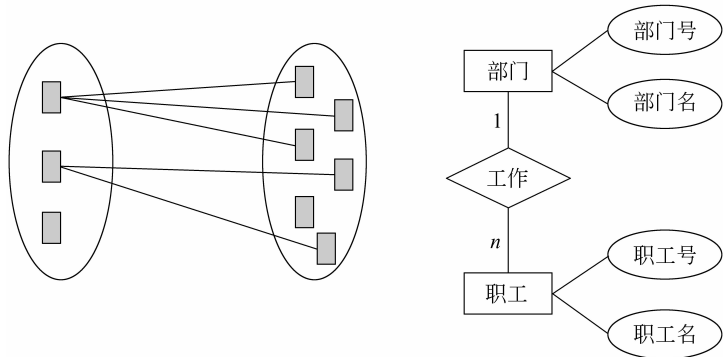


图 1-10 一对多联系示例

(3) 多对多联系($m:n$)

如果实体 A 与实体 B 之间存在联系,并且对于实体 A 中的任一实例,实体 B 中有 n 个实例($n \geq 0$)与之对应;而对实体 B 中的任一实例,实体 A 中有 m 个实例($m \geq 0$)与之对应,则称实体 A 到实体 B 的联系是多对多的,记为 $m:n$ 。例如,银行和法人之间、商场和顾客之间都是 $m:n$ 联系,如图 1-11 所示。

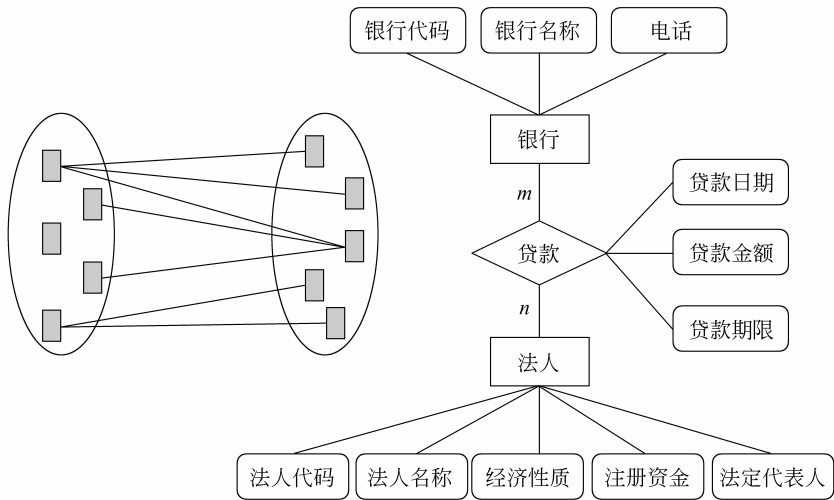


图 1-11 多对多联系示例

1.2.3 关系数据模型

关系数据模型就是用关系表示现实世界中实体以及实体之间联系的数据模型。关系数据模型包括关系数据结构、关系数据操作和关系完整性约束三个重要要素。

1.2.3.1 关系模型的数据结构

关系数据结构非常简单,在关系数据模型中,现实世界中的实体及实体之间的联系均