

第1章 数据库系统概述



本章学习目标

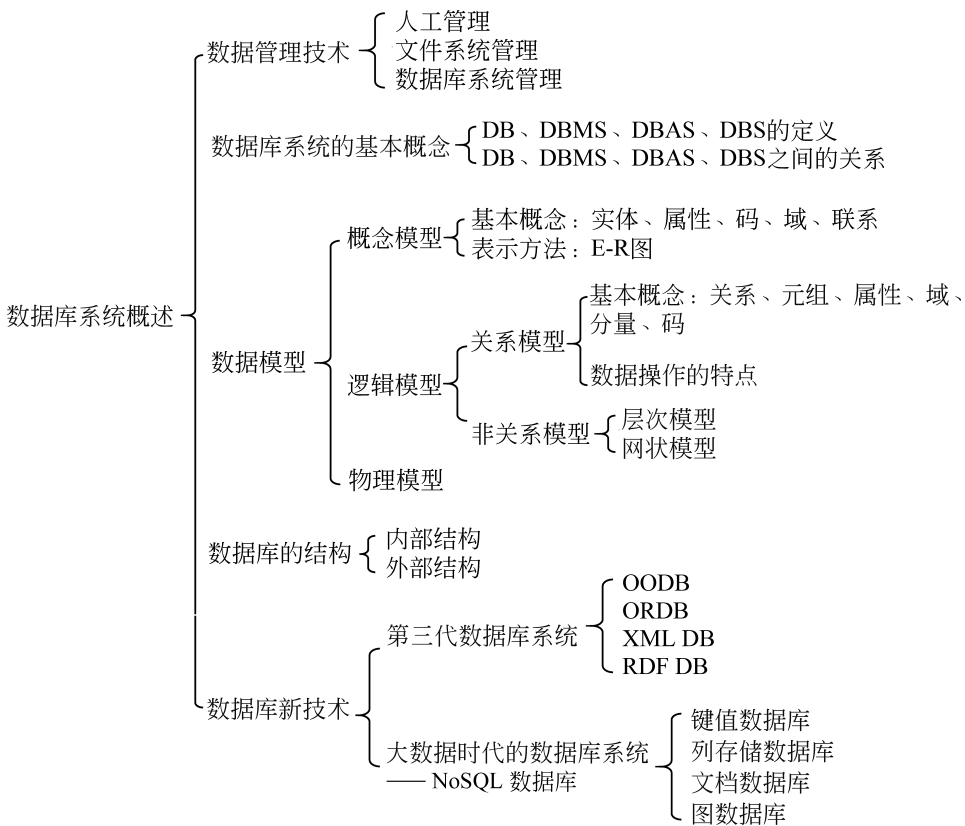
- (1) 了解数据管理技术发展的三个阶段。
- (2) 掌握数据库系统的基本概念。
- (3) 理解三个世界的划分及信息世界的基本概念。
- (4) 掌握关系模型的基本概念。
- (5) 掌握数据库系统的三级模式二级映像结构。
- (6) 了解数据库的外部结构和数据库新技术的发展。

重点：数据库系统的基本概念、信息世界的基本概念和数据库系统的三级模式二级映像结构。

难点：数据库系统的三级模式二级映像结构及其作用。



本章学习导航



众所周知,我们正处在一个高度信息化的社会,信息资源的获取、使用与管理已成为个人和企业的重要财富和资源。数据(Data)是信息(Information)的载体,数据库(Database, DB)是存储与管理大量数据的技术,其应用领域越来越广泛,如客户关系管理系统(Customer Relationship Management, CRM),教学管理、电子政务(E-Government)、人口普查等信息管理系统,订票、银行、电子商务(E-Business)等业务处理系统,地震救灾、电信营销等决策支持系统。数据库系统的建设规模、信息量的大小及其使用频度已经成为衡量一个企业乃至一个国家信息化程度高低的重要标志。

数据库课程是计算机相关专业的基础必修课,也是众多非计算机专业的重要选修课。数据库技术是现代计算机信息系统的基础和核心,是绝大多数应用软件必备的后台数据管理技术,因此我们要学好数据库课程,为后续相关课程的学习打下坚实的基础。

1.1 数据与信息

1.1.1 数据

数据是描述事物的符号记录,是信息的符号表示或载体,也是数据库中存储的基本对象。数据不仅包括常见的数字(如 89、-100、3.14、¥688),而且包括文本、图形、图像、音频、视频等。

数据本身无意义,具有客观性,经过解释后才能表示一定的意义。数据的含义是对数据的解释,称为数据的语义,数据与其语义是密不可分的。例如,89 可以表示一个人的年龄,也可以表示一个人的成绩或体重等。再如,一条人员档案记录(赵刚,男,199603,山东,计算机学院,2014),如果在学生管理系统中可以这样解释:赵刚同学,男,1996 年 3 月出生,籍贯山东省,就读于计算机学院,2014 年入学。但是,如果该记录存在于教工管理系统中,则可能得到这样的解释:赵刚老师,男,工号 199603,籍贯山东省,现工作于计算机学院,办公室房间号是 2014。由此可以看出,数据在不同的环境中有不同的语义解释,数据和语义是密不可分的。

1.1.2 信息

信息是数据的内涵,是数据的语义解释。数据经过处理后就转变成信息。例如,数据 40℃,若解释成体温 40℃,就表达出发烧的信息;若解释成气温 40℃,就表达出天气炎热的信息。

1.1.3 数据处理

数据处理是将数据转换成信息的过程,包括对数据的收集、存储、分类、加工、检索、维护等一系列活动,其目的是从大量的原始数据中抽取和推导出有价值的信息。数据、信息及数据处理之间的关系如图 1-1 所示。



图 1-1 数据、信息及数据处理之间的关系

1.2 数据管理技术的发展

计算机诞生之初的主要用途是进行科学计算。随着现代信息社会的飞速发展,计算机的应用范围扩展到存储与处理各种形式的海量数据,数据管理技术应运而生。数据管理就是对数据进行分类、组织、编码、存储、检索、传播和利用的一系列活动的总和。随着计算机软硬件的发展及应用需求的推动,数据管理技术经历了人工管理、文件系统和数据库系统三个阶段。

1.2.1 人工管理阶段

20 世纪 50 年代中期,计算机主要用于科学计算。当时,在硬件方面,计算机没有磁盘等能随机存取数据的存储设备,只有磁带、卡片和纸带等顺序存储设备;在软件方面,计算机没有操作系统,也没有管理数据的软件。数据处理主要采用批处理的方式,效率极低。这种情况下的数据管理方式称为人工管理,其特点如下。

(1) 数据不单独保存。该阶段计算机主要用于科学计算,一般不需要长期保存数据,且数据与程序是一个整体,数据只为本程序使用,因此数据不单独保存。

(2) 应用程序管理数据。当时没有相应的软件系统负责数据的管理,数据由应用程序自己设计、说明(定义)和管理。因此,每个应用程序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,程序员负担很重。

(3) 数据不共享。数据是面向应用程序的,一组数据只能对应一个程序。即使多个应用程序使用相同的数据,也必须各自定义,因此程序之间有大量的数据冗余。

(4) 数据不具有独立性。数据与程序是一体的,程序依赖数据,如果数据的类型、格式或输入/输出格式等逻辑结构发生变化,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。数据脱离了程序就没有任何存在的价值,无独立性。

人工管理阶段应用程序和数据之间的对应关系及示例如图 1-2 所示。



图 1-2 人工管理阶段应用程序和数据之间的对应关系及示例

1.2.2 文件系统阶段

20 世纪 50 年代后期到 60 年代中期是文件系统阶段,这时的计算机不仅用于科学计

算,还大量应用于数据处理。在硬件方面,计算机有了磁盘、磁鼓等直接存取设备,数据可以长期保存;在软件方面,计算机出现了操作系统,而且操作系统中的文件系统是专门进行数据管理的软件,用于管理以记录的形式存放在不同文件中的数据。数据的处理方式不仅有批处理,而且有联机实时处理。文件系统阶段的特点如下。

(1) 数据以文件的形式长期保存在外存储器中。由于计算机大量用于数据处理,因此数据需要长期保存在外存储器上,以反复进行查询、修改、插入和删除等操作。

(2) 由文件系统对数据进行管理。操作系统中的文件系统把数据组织成相互独立的数据文件,利用“按文件名访问,按记录进行存取”的管理技术对文件进行修改、增加和删除等操作。

(3) 数据共享性差,冗余度大。在文件系统中,一个(或一组)文件基本上对应一个应用程序,即文件是面向应用的。当不同的应用程序使用部分相同的数据时,文件系统也必须建立各自的文件,而不能共享相同的数据。因此,数据的冗余度大,浪费存储空间,而且大量冗余的数据在进行更新操作时容易造成数据不一致。

(4) 数据独立性差。文件系统中的文件是为某一特定应用服务的,文件的逻辑结构对该应用程序来说是优化的,如果要对现有的数据增加一些新的应用会很困难,系统不容易扩充。数据和应用程序相互依赖,一旦改变数据的逻辑结构,则必须改变相应的应用程序,而应用程序的变化(如采用另一种语言编写)也需要修改数据结构。因此,数据和应用程序之间缺乏独立性。

文件系统阶段应用程序和数据之间的对应关系及示例如图 1-3 所示。

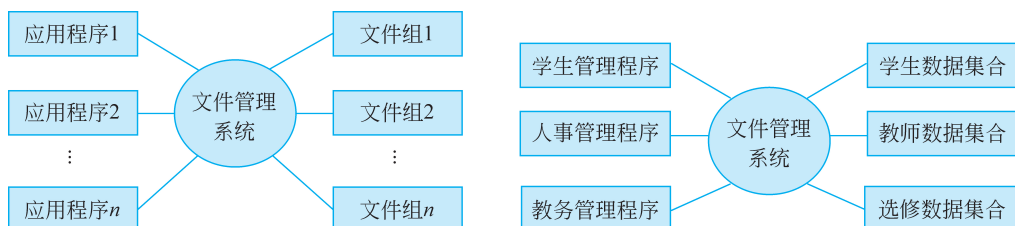


图 1-3 文件系统阶段应用程序和数据之间的对应关系及示例

1.2.3 数据库系统阶段

20 世纪 60 年代后期,计算机软硬件有了进一步的发展。随着计算机管理对象的规模越来越大,应用范围越来越广,数据量急剧增长,用户对数据共享的要求也越来越强烈。在硬件方面,计算机出现了大容量磁盘,使联机存取大量数据成为可能。硬件价格下降,软件价格上升,使得开发和维护系统软件的成本增加。文件系统管理数据的方法已不能满足应用系统的需求,为解决多用户、多应用程序共享数据的要求,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(Database Management System,DBMS)。

数据库系统阶段应用程序与数据之间的对应关系及示例如图 1-4 所示。

从文件系统到数据库系统,标志着数据管理技术的飞跃。和文件系统阶段相比,数据库系统阶段数据管理的特点如下。

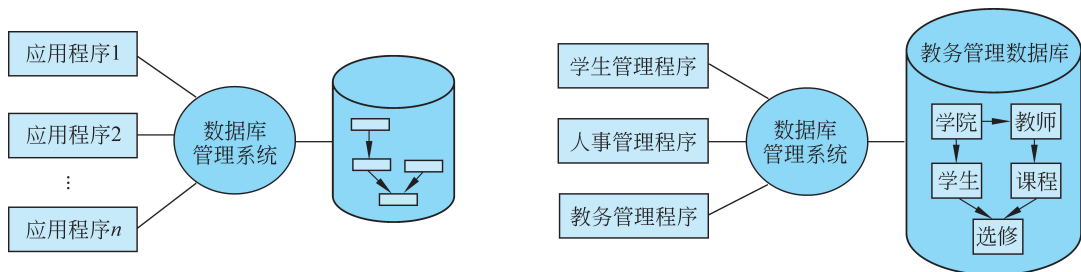


图 1-4 数据库系统阶段应用程序和数据之间的对应关系及示例

1. 数据结构化

在文件系统中,尽管记录内部是有结构的,但记录之间没有联系;而数据库系统则实现了整体数据的结构化,这是数据库的主要特征之一,也是数据库系统与文件系统的本质区别。

数据库中数据整体的结构化是指在数据库中的数据都不属于任何一个应用,而是面向全组织的、公共的。不仅数据内部是结构化的(如都以记录的形式存储数据),而且整体是结构化的,是一个有机整体。

例如,某高校教务管理系统中有学生(学号,姓名,性别,出生日期,照片,兴趣爱好,所在学院)、课程(课程号,课程名,直接先修课,学分)和选修(学号,课程号,成绩)等数据,分别对应三个文件。若采用文件管理系统进行处理,则三个文件单独存储和使用,每个文件内部是有结构的,即文件由记录组成,每条记录由若干属性组成;但文件与文件之间没有联系,如选修文件中的某条记录的学号不是学生文件中的某个学号,这表明该记录中的成绩不是已有的某个学生的成绩或该成绩表示一个并不存在的学生的成绩。同理,选修文件的课程号也可能不是课程文件中的某一个课程号。

若采用数据库系统存储这三个文件,则学生、课程、选修三个文件之间是有联系的,如图 1-4 所示,箭头的方向表示二者之间的一对多联系。选修文件中的学号必须是学生文件中的某个学号,即成绩必须是某个已存在的相关学生的成绩。

2. 数据的共享性高,冗余度低,易扩充

在数据库系统中,数据是面向整个系统的,可被多个应用程序共享,因此冗余度低。如图 1-4 中的数据库系统建成后,既可用于学生管理系统,也可用于人事管理系统。

3. 数据独立性高

数据独立性包括数据的物理独立性和逻辑独立性,分别表示当数据的物理存储和逻辑结构发生改变时,应用程序都不必发生改变,即数据与应用程序完全独立(详见 1.5 节)。

4. 数据由 DBMS 统一管理和控制

数据库的共享是并发的共享,即多个用户可以同时存取数据库中的数据,甚至可以同时存取数据库中的同一个数据。因此,DBMS 必须提供相应的数据控制功能,主要包括数据的安全性(Security)保护、数据的完整性(Integrity)检查、并发(Concurrency)控制和故障恢复(Recovery)。

1.3 数据库系统的基本概念

1.3.1 数据库

数据库是长期存储在计算机内、有组织、可共享的大量数据的集合。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展性,并可为各种用户共享。数据库中的数据通常是面向部门或企业等应用环境的整体数据,如高校教务管理系统中的学院信息、学生信息、教师信息、课程信息和选修信息等数据。

1.3.2 数据库管理系统

数据库管理系统是管理数据库的系统软件,它运行于用户与操作系统之间,如图 1-5 所示。

DBMS 是整个数据库系统的核心部分,用户对数据库的一切操作都由它统一管理和控制,包括数据的定义、查询、更新、完整性约束、安全性保护、多用户的并发控制、数据库故障的恢复等操作。DBMS 的主要功能如下。

1. 数据定义功能

DBMS 提供数据定义语言(Data Definition Language, DDL),用户通过 DDL 可以方便地对数据库中的数据对象进行定义。例如,创建基本表时,为保证数据库安全而定义的用户口令和存取权限、为保证正确语义而定义的完整性规则等。

2. 数据操纵功能

DBMS 提供数据操纵语言(Data Manipulation Language, DML),实现对数据库的基本操作,包括查询、插入、修改、删除等。

3. 数据组织、存储和管理

DBMS 要分类组织、存储和管理各种数据,包括数据字典、用户数据和数据的存取路径等,要确定以何种文件结构和存取方式在存储级上组织这些数据,以及如何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率和方便存取,提供多种存取方法(如索引查找、Hash 查找、顺序查找等)来提高存取效率。

4. 数据库运行管理

数据库在建立、运行和维护时由 DBMS 统一管理、统一控制。DBMS 通过对数据的安全性控制、完整性控制、多用户环境下的并发控制及数据库的恢复来确保数据正确和有效,以及数据库系统的正常运行。

5. 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装入和转换,数据库的转储、恢复、重

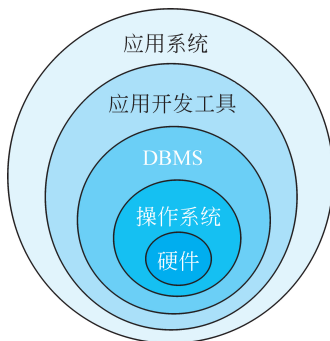


图 1-5 DBMS 在计算机系统中的地位

组织,系统性能的监视和分析等功能,这些功能通常由一些应用程序完成。

6. 其他功能

其他功能包括 DBMS 与网络中其他软件系统的通信功能、两个 DBMS 系统的数据转换功能、异构数据库之间的互访和互操作功能等。

1.3.3 数据库应用系统

数据库应用系统(Database Application System,DBAS)是以数据库为基础、在 DBMS 的支持下使用应用开发工具建立的面向用户的应用系统,系统一般拥有友好的、人性化的图形用户界面,通过数据库语言或相应的数据访问接口存取数据库中的数据。例如,教务管理系统、人事管理系统等都是数据库应用系统。

1.3.4 数据库用户

数据库用户主要是指开发、管理和使用数据库的各类人员,包括数据库管理员(Database Administrator, DBA)、系统分析人员(System Analyst)、数据库设计人员(Database Designer)、应用程序开发人员(Application Developer)和最终用户(End User)。

DBA 是支持数据库系统的专业技术人员,负责全面管理和控制数据库系统,拥有对数据库的最高操作权限。DBA 的具体职责:决定数据库中的信息内容和结构;决定数据库的存储结构和存储策略;定义数据的安全性要求和完整性约束条件;监督数据库的使用和运行,以及数据库的改进和重组重构。

系统分析人员主要负责应用系统的需求分析和规范说明,参与数据库的概要设计。

数据库设计人员参与需求调查和系统分析,负责数据库中数据的确定和数据库各级模式的设计。

应用程序开发人员负责设计和编写访问数据库的应用系统的程序模块,并对程序进行调试和安装。

最终用户是数据库应用程序的使用者,包括不经常访问数据库的偶然用户,如企业中的中高级管理人员;通过应用程序界面对数据库进行查询和数据更新工作的简单用户,如银行职员和售货员等;能直接使用数据库语言访问数据库的具有较高科学技术背景的人员,如工程师和科学技术工作者。

1.3.5 数据库系统

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、DBMS 及其开发工具、应用系统、DBA 和用户组成,如图 1-6 所示。

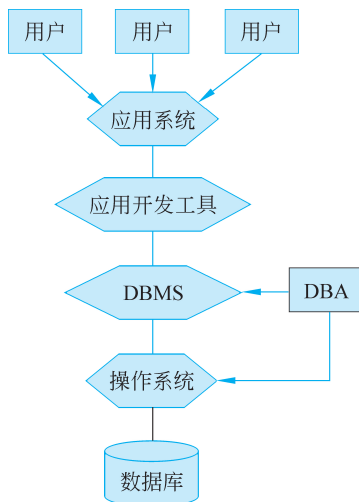


图 1-6 数据库系统的组成

1.4 数据模型

模型(Model)是对现实世界中某个对象特征的抽象,如楼盘模型、分子模型。数据模型(Data Model)是对现实世界数据特征的抽象,是数据库管理的数学形式框架。

由于计算机不能直接处理现实世界中的人、事物、活动及其联系,因此计算机要事先对具体事物及其联系进行特征抽象,转换成自身能够处理的数据。

数据模型是数据库的框架,该框架描述了数据及其联系的组织方式、表达方式和存取路径,是数据库系统的核心和基础。各种 DBMS 软件都是基于某种数据模型的,数据模型的数据结构直接影响到数据库系统的其他部分性能,也是数据定义和数据操纵语言的基础。

1.4.1 三个世界的划分

为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型,人们首先对现实世界的事物及其联系进行特征抽取,形成信息世界的概念模型,再将概念模型转换为机器世界中某一 DBMS 支持的逻辑模型和物理模型,如图 1-7 所示。

1. 现实世界

现实世界即客观存在的世界。

客观世界中存在着各种事物,它们都有自己的特征,人们总是选用感兴趣的最能表征一个事物的若干特征来描述该事物。例如,要描述一位员工,人们常选用员工号、员工名、性别、年龄来描述,有了这些特征,人们就能区分不同的员工。

客观世界中,事物之间又是相互联系的,而这种联系可能是多方面的,但人们只选择那些感兴趣的联系。例如,在超市管理系统中,人们可以选择“销售”这一联系来表示员工和商品之间的联系。

2. 信息世界及其基本概念

信息世界是现实世界在人们头脑中的反映,现实世界经过人脑的分析、归纳和抽象形成信息,这些信息经过记录、整理、归类 and 格式化后就构成了信息世界。信息世界中常用的主要概念如下。

(1) 实体(Entity): 客观存在并且可以相互区分的事物称为实体。实体可以是具体的人或物,如张三、桌子等;也可以是抽象的事件或概念,如一场比赛。

(2) 属性(Attribute): 实体具有的某一特性称为属性。一个实体可由若干属性来描述。例如,员工实体可由员工号、员工名、性别、年龄等属性描述,那么(2018001,张三,男,29)这组属性值就构成了一个具体的员工实体。属性有属性名和属性值之分,如“员工名”是

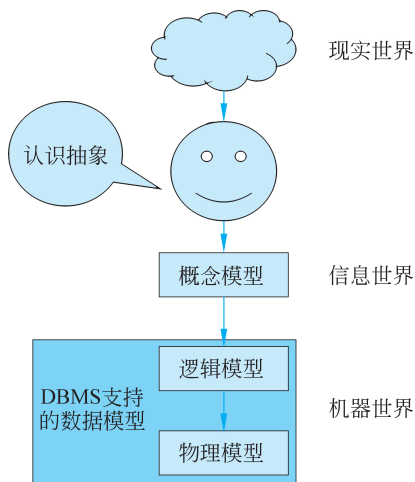


图 1-7 现实世界中客观对象的抽象过程

属性名,“张三”是属性值。

(3) 码(Key): 能唯一标识实体的属性或属性的组合称为码,也称为键。例如,“员工”实体中的“员工号”属性是它的码。

(4) 域(Domain): 属性的取值范围称为该属性的域。例如,“员工”实体的“性别”属性的域为(“男”,“女”)。

(5) 实体型(Entity Type): 具有相同属性的实体具有相同的特征和性质,用实体名及其所有属性的集合描述同类实体,称为实体型。例如,员工(员工号,员工名,性别,年龄)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,全体员工就是一个实体集。同一实体集中没有完全相同的两个实体。

 **注意:** 在不引起混淆的情况下,通常将实体型和实体集称为实体。

(7) 联系(Relationship): 在现实世界中,事物内部及事物之间都是有联系的。实体内部的联系是指组成实体的各属性之间的联系,实体之间的联系是指不同实体集之间的联系。这里只讨论实体集之间的联系,分为以下三种情况。

① 两个实体集之间的联系。两个实体集之间的联系类型有三种。

a. 一对一联系(1:1): 如果对于实体集 E_1 中的每一个实体,实体集 E_2 中至多有一个实体与之联系,反之亦然,则称实体集 E_1 与实体集 E_2 具有一对一联系,记为 1:1。例如,一个班级只有一个班长,每个班长只能在一个班级任职,则班级与班长之间具有一对一联系。

b. 一对多联系(1:n): 如果对于实体集 E_1 中的每一个实体,实体集 E_2 中至多有 n 个实体($n \geq 0$)与之联系;对于实体集 E_2 中的每一个实体,实体集 E_1 中至多有一个实体与之联系,则称实体集 E_1 与实体集 E_2 具有一对多联系,记为 1:n。例如,一个班级有多名学生,一名学生只能属于一个班级,则班级和学生之间具有一对多联系。

c. 多对多联系($m:n$): 如果对于实体集 E_1 中的每一个实体,实体集 E_2 中有 n 个实体($n \geq 0$)与之联系;对于实体集 E_2 中的每一个实体,实体集 E_1 中也有 m 个实体($m \geq 0$)与之联系,则称实体集 E_1 与实体集 E_2 具有多对多联系,记为 $m:n$ 。例如,一名学生可以选修多门课程,每门课程可由多名学生选修,则学生和课程之间具有多对多联系。

实际上,一对一联系是一对多联系的特例,而一对多联系又是多对多联系的特例。

② 三个或三个以上实体集之间的联系。一般地,三个或三个以上的实体集之间也存在一对一、一对多和多对多联系。下面仅给出多个实体集之间多对多联系的定义。

若实体集 E_j ($j=1,2,\dots,i-1,i+1,\dots,n$) 中的给定实体和 E_i 中的多个实体相联系,则 E_i 与 $E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_n$ 之间的联系是多对多的。例如,供应商、项目和零件三个实体集之间具有多对多联系,其语义是:一个供应商可以供给多个项目多种零件,每个项目可以使用多个供应商供应的多种零件,每种零件可由不同供应商供给且可应用于不同的项目。

③ 单个实体集之间的联系。单个实体集与它自身之间也存在一对一、一对多和多对多联系。

a. 一对一联系(1:1): 例如,职工实体集中每位职工的配偶(也是职工)最多有一个,则职工实体集中的“配偶”联系是一对一的。

b. 一对多联系(1:n): 例如,职工实体集中每位领导(也是职工)管理若干职工,每位职

工只被一位领导管理,则职工实体集中的“领导”联系是一对多的。

c. 多对多联系($m:n$): 如果零件实体集中的每种零件都由若干种子零件构成,每种种子零件又参与了多种母零件的组装,那么零件实体集中的“构成”联系是多对多的。

(8) 概念模型(Conceptual Model)的表示方法。概念模型是对信息世界的建模,它能方便、准确地描述信息世界中的概念。概念模型有很多表示方法,其中最著名、最常用的是华裔科学家陈品山(Peter Pin-Shan Chen)于1976年提出的实体—联系方法(Entity-Relationship Approach),也称为实体—联系模型(Entity-Relationship Model),该方法用E-R图表示实体、属性及实体间的联系。



概念模型的表示方法 E-R图

① 实体用矩形表示,矩形框内写明实体名。例如,“部门”实体和“员工”实体的表示方法如图1-8所示。



图1-8 E-R图中实体的表示方法

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。例如,“部门”和“员工”实体的属性表示方法如图1-9所示。



图1-9 E-R图中属性的表示方法

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 $m:n$)。例如,“部门”和“员工”实体间的“属于”联系表示方法如图1-10所示。

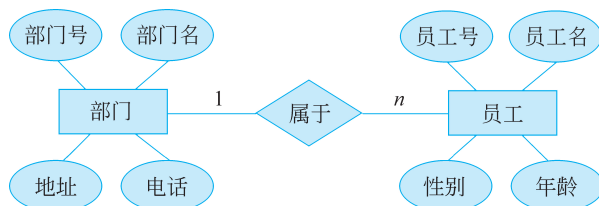


图1-10 E-R图中联系的表示方法

联系也可能有属性,联系的属性用椭圆形表示,并用无向边与该联系连接起来。例如,“属于”联系可以有“工作时间”属性,其表示方法如图1-11所示。

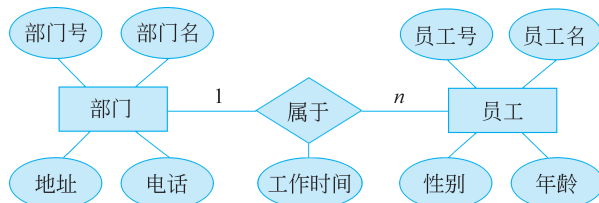


图1-11 E-R图中联系的属性的表示方法