

第 1 章



数据库基础知识



本章主要内容

本章主要介绍数据库的基本原理、概念、模型、结构和方法，如什么是数据库和数据库管理系统；数据库的 3 种数据模型，数据库系统的三级模式结构和数据独立性；数据库系统的体系结构；关系数据库和关系数据库的设计等。



本章难点提示

本章的难点如下：在掌握数据库的基本概念和原理的基础上，正确理解数据库系统的三级模式结构和数据库独立性的含义；在正确掌握关系数据库的基本概念、原理和模型后，能够根据实际需求，设计一个满足一定规范级别的关系数据库，设计关系的逻辑结构，满足各种业务和功能管理的信息需求，设计完整性约束以保证数据库数据的一致性和完整性。

数据库技术产生于 20 世纪 60 年代,是数据管理的最新技术,也是软件领域最重要的发展成果之一。数据库技术是信息系统的核心和基础,随着互联网、通信、移动技术的发展,数据库技术的应用深入社会生活的各个领域,包括金融银行、证券投资、商务销售、医疗卫生、政府部门、国防军工、科技发展等。

数据库技术经历了层次数据库、网状数据库和关系数据库等 3 代数据库的演变,造就了 4 个图灵奖,发展成为计算机的基础学科之一。图灵奖是计算机界的诺贝尔奖,第一位图灵奖得主是数据库技术的先驱者查尔斯·巴赫曼,他主持设计与开发了最早的网状数据库管理系统(Integrated Data Store, IDS);第二位图灵奖得主是关系数据库之父埃德加·科德,他在集合论的严格数学基础上,建立了关系数据库模型;第三位图灵奖得主是数据库技术及数据库事务处理专家詹姆斯·格雷(James Gray),他提出了数据库事务处理中的数据共享与封锁机理,突破了数据共享封锁线;第四位图灵奖得主是 MIT 的教授迈克尔·斯通布雷克,他是冲浪在数据潮头的实干家,创造了现代数据库系统的一系列奠基性概念和实现技术,创立了多家公司,成功地将数据库商业化。

数据库技术的发展,以数据模型和数据库管理系统为核心,形成了巨大的软件产业。数据库的建设规模、信息量大小,是衡量一个国家信息化程度的重要标志。

本章主要介绍数据库技术的基本概念、基本原理,数据库系统的构成,数据库管理系统的主要功能,数据库系统的三级模式结构和数据独立性,以及数据库的体系结构。

本章通过案例,以需求为驱动,介绍数据库设计的方法和过程,以及数据库应用系统开发的周期。如果读者对数据库原理比较熟悉,则本章内容可以略过。如果读者没有系统地接触过数据库原理,则学习本章对后续章节的学习非常必要。

1.1 数据库系统

数据库系统(Database System, DBS)指的是引入数据库后的计算机系统,包括数据库、数据库管理系统、数据库应用系统、数据库用户以及支撑软件系统运行的软硬件。在此仅对与数据库系统相关的部分概念进行简单介绍,更深入的知识请读者参考相关教材或书籍。

1.1.1 信息、数据与数据库

1. 信息

信息(Information)、物质和能量是客观世界的三大要素。信息就是对客观事物的反映,从本质上看,信息是对社会、自然界的事物特征、现象、本质及规律的描述,是由文字、符号、数字或声音、图形、图像,表现出来的消息、情报、指令、数据、信号等。例如,记录学生信息可以用文字和标点符号表示为“**学号: 20160431101; 姓名: 李勇; 性别: 男; 出生年月: 1988.2; 所在系别: 计算机**”。

信息源于物质和能量,可以感知,可存储、加工、传递、再生和共享,它具备 4 个基本特征:

可传载性,即信息可以依附于某种载体进行传递; 共享性,有别于物质和能量,信息的共享不仅不会产生损耗,还可以广泛地传播和扩散,使更多人共享; 可处理性,特别是经过人的分析、

综合和提炼，可以增加它的使用价值；时效性，只有既准确又及时的信息才有价值，一旦过时，就会变成无效信息。

2. 数据 (Data)

数据 (Data) 是表示信息的物理符号，是信息的具体表现。数据可以是数值数据，如某个具体数字，也可以是非数值数据，如文字、图形、图像和声音等。虽然数据有多种表现形式，但经过数字化处理后，都可以输入并存储到计算机中，并能成为其处理的符号序列。

3. 信息与数据的关系

数据是信息的符号表示或载体。信息是数据的内涵，是对数据的语义解释。

在计算机中，为了存储和处理某些事物，需要抽象出对这些事物感兴趣的特征组成一条记录来描述。例如，在学生档案中，如果人们感兴趣的是学生的姓名、性别、出生年月、籍贯、所在系别、入学日期等信息，就可以这样描述：(20160431101, 李勇, 男, 1988.2, 计算机)，因此这里的学生记录就是数据。它的含义即所含信息如下：李勇是一个大学生，学号为 1001，1988 年 2 月出生，性别为男，计算机专业。

数据的形式不能完全表达其内容，需要经过解释。数据的解释是指对数据含义的说明，数据的含义又称为数据的语义，也就是数据包含的信息。例如，记录数据(20160431101, 李勇, 男, 1988.2, 计算机)，如果没有语义解释，也可以理解为某一企业员工的记录，员工号为 20160431101，1988 年 2 月出生，计算机专业，等等。因此，信息是数据的内涵，数据是信息的符号表示，是载体；数据是符号化的信息，信息是语义化的数据。

4. 数据库

数据库 (Database, DB) 是长期存储在计算机内的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储，用于满足各种不同的信息需求，并且集中的数据彼此之间有相互的联系，具有较小的冗余度、较高的数据独立性和易扩展性。

数据库中的数据是以一定的数据模型进行组织存储的，数据库中的数据有一定的结构、组织存储方式和操作规则。例如，学生人事记录在数据库中的组织形式和结构如图 1-1 所示。



图 1-1 学生人事记录在数据库中的组织形式和结构

记录某个学生的人事信息如图 1-2 所示。

这样的数据组织方式节省存储空间，灵活性也相对得到了提高。但这种灵活性只是对一个应用而言的，因为一个学校和组织涉及许多应用，在数据库系统中不仅要考虑某个应用的数据结构，还要考虑整个组织的数据结构。例如，一个学校的管理信息系统包括学生的人事管理、学籍管理、选

课管理等，可按图 1-3 所示的结构建立学校管理信息系统的学生数据模式。

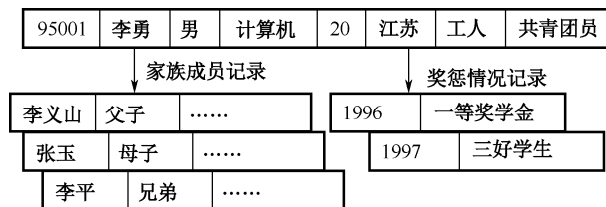


图 1-2 某个学生的人事信息

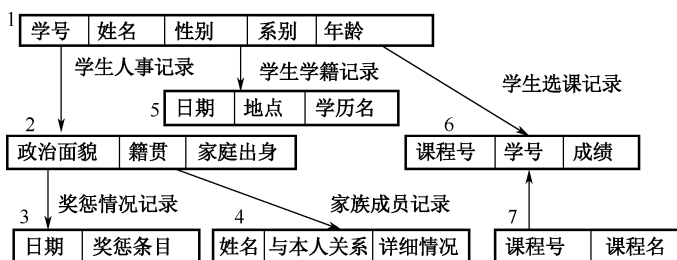


图 1-3 学校管理信息系统的学生数据模式

如图 1-3 所示结构中的学生数据为各个部门提供数据，并使数据结构化：人事管理用到图 1-3 中 1、2、3、4 记录的数据，学籍管理用到 1、5 记录的数据，选课管理用到 1、6、7 记录的数据，并且不仅描述了数据本身，还描述了数据之间的关联。

数据库中的数据是可共享的。这里的共享指的是不同地域、不同用户可以同时使用同一数据库中的同一数据。数据的共享性，能够最大限度地降低某一企事业单位信息数据的冗余度，并消除数据的不一致性。所谓数据的不一致性是指同一数据不同副本的值不一样。

数据库不仅存储数据以及数据之间的联系，还存储数据字典（对数据的描述，即数据的结构）。数据库技术对数据结构和数据同时进行管理，并将数据结构特征与应用程序尽可能隔离开来，即应用程序在一定程度上不受数据结构改变的影响，提高了数据的独立性。

数据库中的数据是海量的。一个单位、一个公司，原则上能构建一个数据库，各部门所需要的所有数据存储在同一个数据库中，并可在授权允许的情况下访问和使用其他部门提供的数据。

数据库中的数据是永久的。数据一旦成功地存储到数据库中，就会永久保存，除非用户删除数据。当然，要保证数据的永久性，还要有数据库恢复技术的支持。

1.1.2 数据库管理系统

数据库管理系统（Database Management System，DBMS）是位于用户和操作系统之间的一层数据管理软件，数据库在建立、运行和维护时由数据库管理系统统一管理、统一控制。DBMS 的主要任务是抽取一个应用所需要的大量数据，科学地组织这些数据并将其存储在数据库中，并能高效地获取、处理和维护数据。

DBMS 使用户能方便地定义数据和操纵数据，并能够保证数据的安全性、完整性、多用户对数

据的并发使用及发生故障后的系统恢复。其主要功能如下。

1. 数据定义功能

DBMS 提供了数据定义语言 (Data Definition Language , DDL), 用户通过它可以方便地定义数据对象。DDL 用于定义数据库的逻辑结构。

2. 数据操纵功能

DBMS 提供了数据操纵语言 (Data Manipulation Language , DML), 用户可以使用 DML 实现对数据库的各种操作, 如查询、插入、删除和修改等。

3. 数据组织、存储和管理功能

DBMS 要分类组织、存储和管理数据库中的各种数据, 包括数据字典、用户数据、数据的存取路径等, 从而提高存储空间的利用率和存取效率。

4. 数据库的运行管理功能

数据库的建立、运行和维护由 DBMS 统一管理和控制, 以保证数据的安全性、完整性、多用户对数据的并发使用, 以及发生故障后的系统恢复。

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输入、转换功能, 数据库转储、恢复功能, 数据库的重组织功能和性能监视、分析功能等。这些功能通常由一些实用程序完成。

6. 其他功能

DBMS 包括一些实用工具。例如, DBMS 与网络中其他软件系统的通信工具; 不同 DBMS 中的数据和文件系统数据的互相转换; 异构数据的互访和互操作等。

1.1.3 数据库系统

数据库系统是为适应数据处理的需要而发展起来的数据处理系统, 一般由软件、硬件、数据库、数据库管理员和用户构成, 如图 1-4 所示。硬件是指构成计算机系统的各种物理设备。软件主要包括操作系统、应用系统、应用开发工具和 DBMS。数据库由 DBMS 统一管理、统一控制, 数据的插入、修改和检索均要通过数据库管理系统。数据库管理员负责创

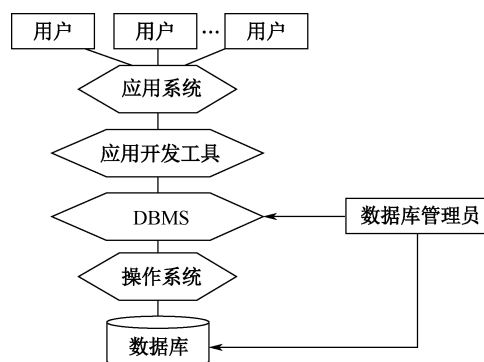


图 1-4 数据库系统的构成

建、监控和维护整个数据库，使数据能够被任何有权使用的人有效使用。

1.2 数据模型

1.2.1 数据模型的概念

数据模型是对现实世界中数据特征的抽象，用来描述数据、组织数据和对数据进行操作。通俗地讲，数据模型是对现实世界的模拟，描述了数据的结构、数据之间的联系、对数据的各种操作以及数据要满足的各种约束条件。

在开发实施数据库应用系统时，需要用到不同层次的数据模型(图 1-5)：概念模型(Conceptual Model)，逻辑模型(Logical Model)和物理模型(Physical Model)。

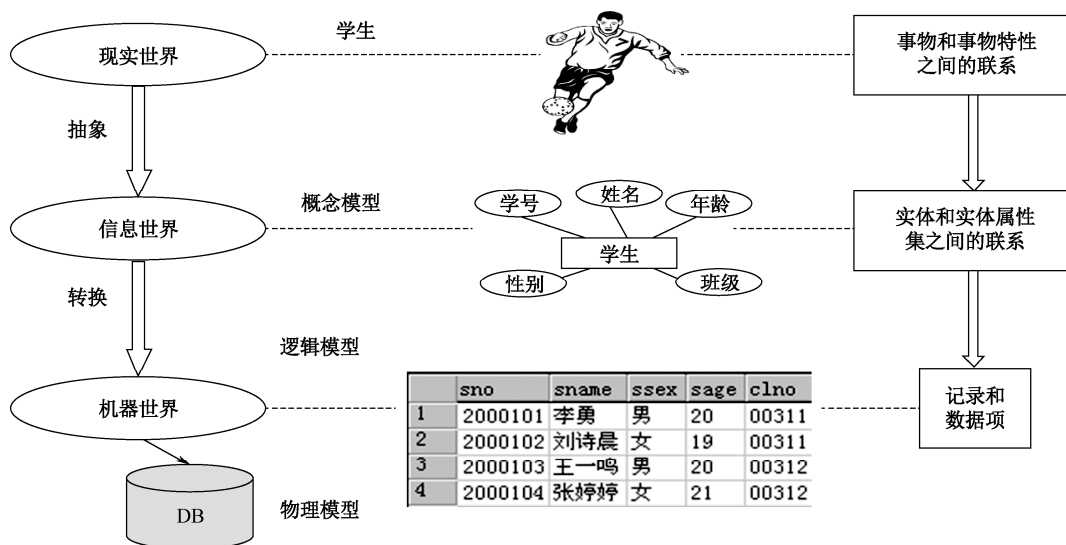


图 1-5 数据模型的 3 个层次

概念模型，也称信息模型，会按用户的观点对信息进行建模。

逻辑模型，是从计算机的观点对数据进行建模。最常用的逻辑数据模型有层次模型、网状模型和关系模型。层次模型中，数据以树状结构进行组织；网状模型中，数据的逻辑结构是图；关系模型是最重要也是最流行的数据模型，它以二维表的形式，存储数据和数据之间的联系。

物理模型，是对数据存储结构和特征的描述，即描述数据在数据库中的存储方式、存取方法。

1.2.2 数据模型的要素

数据模型是数据库系统的核心和基础。数据模型包括 3 个要素：结构要素、操作要素和完整性约束条件。

结构要素描述了数据库中数据的结构、数据之间的联系，是数据库的静态特性。

操作要素描述了数据库中数据对象所允许的操作的集合，是数据库的动态特性。对数据库的操作包括查询、增加、修改和删除4种。

完整性约束条件是一组约束规则，用于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、有效和相容。通俗地讲，完整性约束条件就是规定什么样的数据可以存储到数据库中，以及数据库中的数据可以做怎样的变化。

1.2.3 概念模型

概念模型是对现实世界的第一次抽象，用实体及属性描述客观事物及其性质，用实体之间的联系描述客观事物之间的联系，属于信息世界范畴。

概念模型具有以下两个特点。

(1) 真实性：真实客观地描述了现实世界中的信息实体和实体之间的联系。

(2) 独立性：不依赖于具体的计算机系统或具体的DBMS。因为客观真实，所以其是独立的（与具体的人也无关）。

1. 信息世界的几个概念

(1) 实体 (Entity)：客观存在并可相互区分的事物，可以是人、事、物，也可以是抽象的概念或联系。例如，学生、教师、课程、学生选修课程的事件、教师执教某一门课的协议等，都可以是实体。

(2) 属性 (Attribute)：实体具有的某一方面的特性。学生的学号、姓名、手机号码、邮件地址，教师的编号、姓名、职称，课程的类别、名称和学分等。

(3) 码 (Key)：也称关键字，是唯一标识实体的属性（组）。码可能不止一个。例如，在学生这个实体中，学生的学号、手机号码和邮件地址都可以是学生实体的码，都具有唯一性；而对于“选修事件”这个实体，码为属性组，即“学号+课程”。

(4) 域 (Domain)：属性的取值范围或集合。例如，学生学号是12位的数字串，性别的取值范围是‘男’或‘女’，课程成绩是0~100中的整数等。

(5) 实体型 (Entity Type)：实体型是对实体的描述，描述相同属性的实体具有的共同特征和性质。最简单的描述方法是用实体名及其属性名集合来抽象和刻画同类实体，其格式如下：实体名 (属性1, 属性2, ……)，构成码的属性用下划线标识。与实体型相对应的是实体值，实体值是单个个体，是实体型中的一个具体的值。例如，学生实体型可以描述为学生 (学号, 姓名, 性别, 出生年月, 专业, 手机号码, 邮件地址)，(‘20160431101’, ‘李勇’, ‘男’, ‘1988.2’, ‘计算机’, ‘136056783211’, ‘liyong@163.com’) 是具体的一个学生实体值，而全体学生就构成了学生实体集。型是描述，值是内容。

(6) 联系 (relationship)：在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中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，学生 (学号, 姓名, 性别, 班级, 专业) 中学号与姓名之间的联系。

两个实体之间的联系可分为以下 3 种。

- (1) 一对一 (1 : 1), 如班级——班长 (领导)。
- (2) 一对多 (1 : N), 如班级——学生 (包含)。
- (3) 多对多 (M : N), 如学生——课程 (选修)。

2. 概念模型的描述工具

概念模型是对现实世界的第一次抽象。陈品山于 1976 年提出实体-联系 (Entity-Relationship, E-R) 法来描述概念模型。该方法用图形化的工具 E-R 图来描述现实世界的概念模型。下面介绍 E-R 图的基本图形符号。

(1) 实体 (型): 用命名的矩形框表示实体, 矩形框的名称就是实体的名称。例如, 学生实体如图 1-6 所示。

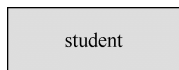


图 1-6 学生实体

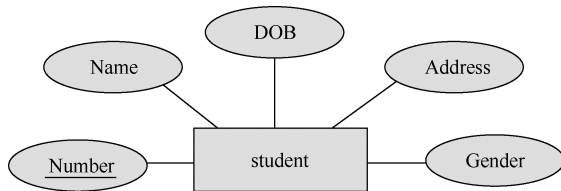


图 1-7 学生实体及其属性

(3) 联系: 用命名的菱形表示实体之间的联系, 并用线段将发生该联系的实体连接起来。具体示例如图 1-8 ~ 图 1-10 所示。

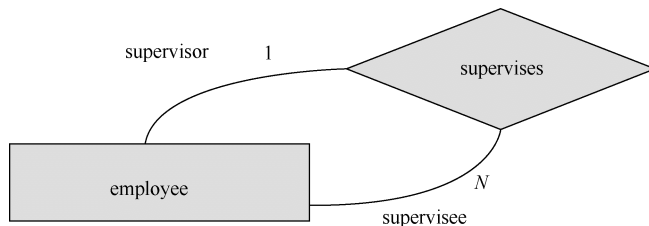


图 1-8 单元联系 (递归联系)

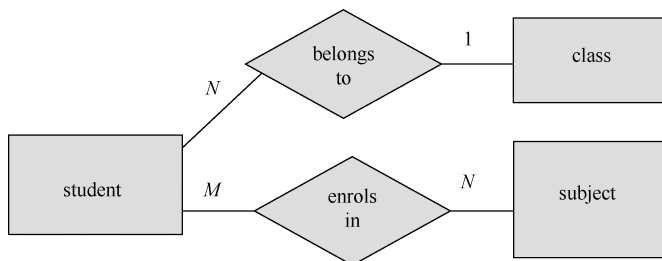


图 1-9 二元联系

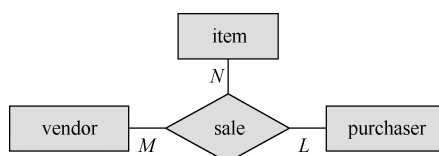


图 1-10 三元联系

(4) 联系的属性：不仅实体有属性，联系本身也可以有属性。具体示例如图 1-11 所示。

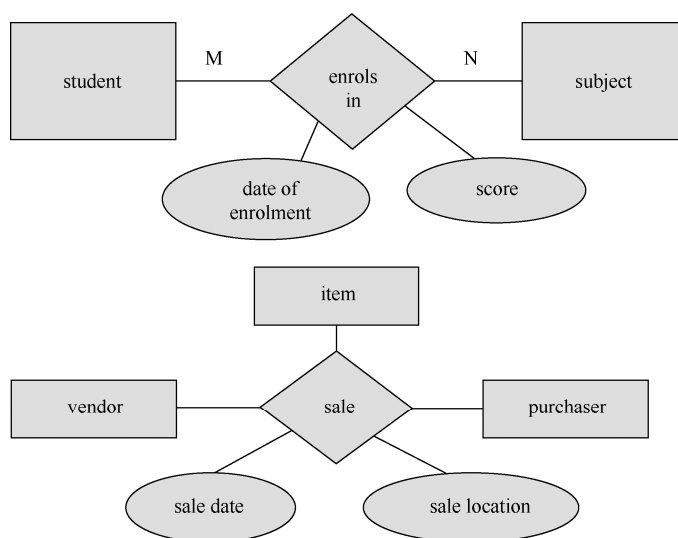


图 1-11 联系的属性

1.3 数据库系统结构

数据库系统的结构可以从不同角度进行划分。

从数据库管理系统的角度，数据库系统通常采用三级模式结构，这是数据库管理系统内部对数据库进行组织管理的结构。

从数据库最终用户的角度，数据库系统可分为单用户结构、主从式结构、文件服务器结构、客户机/服务器体系结构、分布式结构、浏览器/应用服务器/数据库服务器多层结构等，这是数据库系统的外部体系结构。

1.3.1 数据库模式

模式 (Schema) 是数据库中全体数据的逻辑结构和特征的描述，模式的一个具体值称为模式的一个实例 (Instance)。同一个模式可以有多个实例。模式是相对稳定的，而实例是相对变动的，因为数据库中的数据是在不断更新的。模式反映的是数据的结构及其联系，而实例反映的是数据库

某一时刻的状态及内容。

1.3.2 数据库系统的三级模式结构

数据库系统结构分为 3 层，即内模式、概念模式（模式）和外模式，三级模式和模式间的相互映射就组成了数据库系统的体系结构，如图 1-12 所示。这个三级结构有时称为“三级模式结构”，最早是在 1971 年的 DBTG 报告中提出的，后来收录在 1975 年美国的 ANSI/SPARC 报告中。虽然现在 DBMS 的产品多种多样，并在不同操作系统的支持下工作，但是大多数系统在总的体系结构上具有三级模式的结构特征。

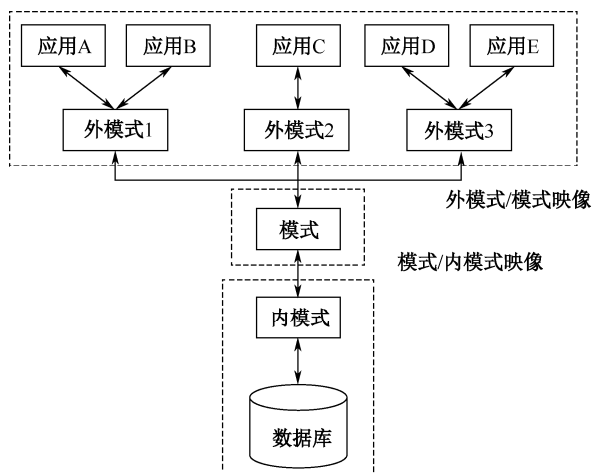


图 1-12 数据库系统的三级模式结构

从某个角度看到的数据特性称为“数据视图（Data View）”。

（1）外模式也称子模式或用户模式，是某数据库用户（包括应用程序员和最终用户）使用的局部数据的逻辑结构和特征的描述。

（2）模式涉及所有用户的数据定义，是全局的数据视图。全局数据视图的描述也称为“概念模式”。

（3）内模式最接近于物理存储设备，涉及实际数据存储的物理结构和存储方式。物理存储数据视图的描述称为“内模式”。

1. 模式

一个数据库只有一个概念模式，它以某一种数据模型为基础，统一综合地考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体。概念模式描述了所有实体、实体的属性和实体间的联系，数据的约束和语义信息。

概念模式由许多记录类型的值组成。例如，它可能包括部门记录值的集合、职工记录值的集合、供应商记录值的集合、零件记录值的集合等。

模式是数据库系统模式结构的中间层，与数据的物理存储细节和硬件环境无关，与具体的应用

程序、开发工具及高级程序设计语言无关。定义模式时,不仅要定义数据的逻辑结构,还要定义数据之间的联系,定义与数据有关的安全性、完整性要求。

在数据库管理系统中,描述概念模式的数据定义语言称为“模式 DDL”。

2. 外模式

外模式(用户可见的视图)是数据库用户能够(有权限)看见和使用的局部数据的逻辑结构和特征的描述,是数据库用户的数据视图,是与某个应用有关的数据的逻辑表示,是数据库模式的一个子集。

一个数据库可以有多个外模式,反映了不同的用户应用需求、看待数据的方式以及对数据保密的要求;对于模式中同一数据,在外模式中的结构、类型、长度、保密级别等都可以因为不同用户(或应用)的个性化需求而不同。

外模式是保证数据库安全性的一个有力措施。每个用户只能看见和访问所对应的外模式中的数据,数据库中的其余数据是不可见的。可以通过 DDL 定义视图来定义用户的外模式,用户使用 DML 语句对视图进行操作,完成需要的数据处理。

3. 内模式

内模式也称存储模式/物理模式,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、记录的存储方式(顺序存储,按照 B 树结构存储,按 Hash 方法存储)、索引的组织方式、数据是否压缩存储、数据是否加密、数据存储记录结构的规定,一个数据库只有一个内模式。

注意:内模式和物理层仍然不同。内部记录并不涉及物理记录,也不涉及设备的约束。比内模式更接近物理存储和访问的那些软件机制是操作系统的一部分,即文件系统。

1.3.3 数据库的二级映像功能与数据独立性

数据库系统的三级模式结构是数据的 3 个抽象级别。它把数据的具体组织留给 DBMS 去做,用户只要抽象地处理数据即可,而不必关心数据在计算机中的表示和存储。不同抽象级别的数据之间的转换,通过定义三级结构之间的映像来实现:外模式/概念模式映像、概念模式/内模式映像,如图 1-12 所示。

这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

1. 外模式/模式映像

其定义了外模式与模式之间的映像关系。由于外模式和模式的数据结构可能不一致,即记录类型、字段类型的命名和组成可能不一样,因此,需要这个映像说明外部记录和概念记录之间的对应性。

当模式发生改变时,可以通过修改相应的外模式/模式的映像,使外模式保持不变,应用程序是依据数据的外模式编写的,从而应用程序不必修改,保证了数据与程序的逻辑独立性,简称数据的逻辑独立性。

2. 模式/内模式映像

其定义了模式和内模式间的映像关系,即定义了数据库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后,可以通过修改模式/内模式映像来使模式保持不变,从而应用程序也不必改变,保证了数据与程序的物理独立性,简称数据的物理独立性。

1.3.4 数据库系统用户结构

从最终用户角度来看,数据库系统分为单用户结构、主从式结构、客户机/服务器结构和分布式结构。

1. 单用户结构数据库系统

单用户结构数据库系统是一种早期最简单的数据库系统,如图 1-13 所示。在这种系统中,整个数据库系统(包括应用程序、DBMS、数据)都装在一台计算机中,由一个用户独占,不同计算机之间不能共享数据。

2. 主从式结构数据库系统

主从式结构是指一个主机带多个终端的多用户结构,如图 1-14 所示。在这种结构中,数据库系统(包括应用程序、DBMS、数据)都集中存放在主机中,所有处理任务都由主机来完成,各个用户通过主机的终端并发地存取数据库,共享数据资源。

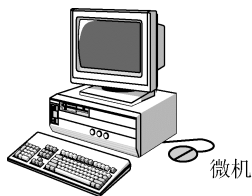


图 1-13 单用户结构数据库系统

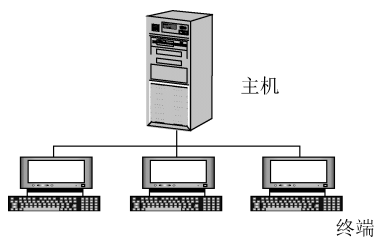


图 1-14 主从式结构数据库系统

3. 客户机/服务器结构数据库系统

主从式结构数据库系统中的主机是一台通用计算机,既执行 DBMS 功能,又执行应用程序。随着工作站功能的增强和广泛使用,人们开始把 DBMS 的功能和应用分开,网络中某个(些)结点上的计算机专门用于执行 DBMS 功能,称为数据库服务器,简称服务器;其他结点上的计算机安装 DBMS 的外围应用开发工具,支持用户的应用,称为客户机,这就是客户机/服务器结构的数据库系统,如图 1-15 所示。

在客户机/服务器结构中,客户端的用户请求被传送到服务器,服务器进行处理后,只将结果(而不是整个数据)返回给用户,从而显著减少了网络中的数据传输量,提高了系统的性能、吞吐量和负载能力。此外,客户机/服务器结构的数据库往往更加开放。客户机与服务器一般能在多种

不同的硬件和软件平台上运行，可以使用不同厂商的数据库应用开发工具，应用程序具有更强的可移植性，也可以减少软件维护开销。

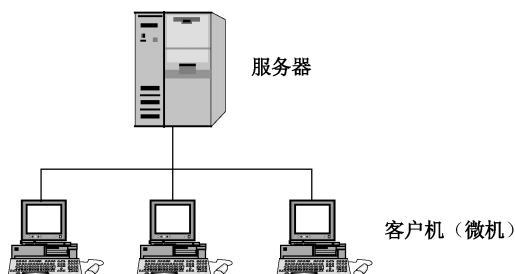


图 1-15 客户机/服务器结构数据库系统

4. 分布式结构数据库系统

分布式结构是指数据库中的数据在逻辑上是一个整体，但分布在计算机网络的不同结点上。网络中的每个结点都可以独立处理本地数据库中的数据，执行局部应用，也可以同时存取和处理多个异地数据库中的数据，执行全局应用，如图 1-16 所示。它的优点是适应了地理上分散的公司、团体和组织对于数据库应用的需求；缺点是数据的分布存放给数据的处理、管理与维护带来了困难，当用户需要经常访问远程数据时，系统效率会明显地受到网络交通的制约。

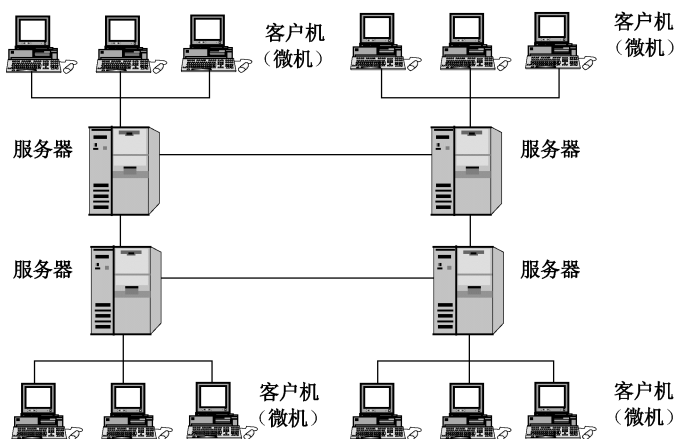


图 1-16 分布式结构数据库系统

1.4 关系数据库

1.4.1 关系数据库概述

在一个给定的应用领域中，所有关系及关系之间联系的集合构成一个关系数据库。

1. 关系的概念

那么,什么是关系?在关系数据库中,一个关系就是一张唯一命名的二维表,由相同结构的行和不同名称的列组成。如图 1-17 所示,表中的一行就是一个元组(也称记录),表中的列为一个属性,给每个属性取一个名即为其属性名(也称字段名)。属性的取值范围是属性的域。能唯一区分一个元组的属性或者属性组,是关系的候选码;可以选定其中一个候选码作为关系的主码,一个关系有且只有一个主码。分量是某个元组在某列上的取值。

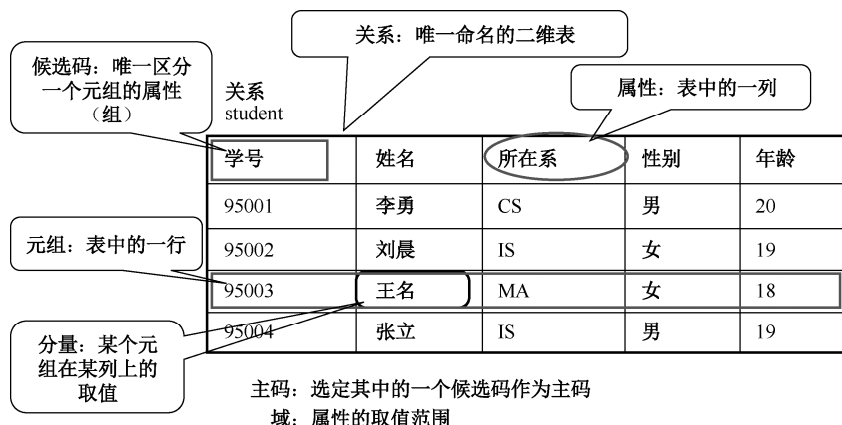


图 1-17 学生关系(表)

2. 关系的基本性质

关系具有以下 6 个基本性质。

关系中每个列的取值是不可分的数据项,是数据库中的最基本单位。

关系中列是同质的,即同一列的数据具有相同的数据类型及约束。

关系中列的名称是唯一的,不同的列可以来自相同的域。

关系中行的顺序可以是任意的。

关系中列的顺序是无所谓的。

关系中不允许有完全相同的两个行。每个关系都有主码关键字(Key)的属性集合,用于唯一地标识关系中的各记录行。

3. 关系间的联系

实际问题往往需要多个关系,关系和关系是有联系的,这种联系也用关系表示。例如,有两个关系,分别是学生关系和课程关系:

学生(学号,姓名,所在系,性别,年龄),主码是学号,记录所有在校学生的信息,如图 1-17 所示;

课程(课程号,课程名,学分,先修课程号),主码是课程号,记录学校能够开出以供学生选修的所有课程信息,如图 1-18 所示。

课程号	课程名	学分	先修课程号
1	数据库	4	5
2	数学	2	
3	信息系统	4	1
4	操作系统	3	6
5	数据结构	4	7
6	数据处理	2	
7	PASCAL	4	6

图 1-18 课程关系（表）

在学校中，学生要选修多门课程，每一门课程都有可能被多个学生选修，学生和课程在选修这件事情发生的时候就要建立联系，这样的联系要用第三个关系来表示，即：

选修（学号，课程号，成绩），主码是学号+课程号，记录学生选修课程的信息，如图 1-19 所示。

学号	课程号	成绩
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80
...	...	...

图 1-19 选修关系（表）

关系之间的联系，最终转换为不同关系中数据之间的联系，表现为取值之间的约束和关联。这样的约束和关联用外码机制来实现。

设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码，如果 F 与基本关系 S 的主码相对应，则称 F 是基本关系 R 的外码 (Foreign Key)，并称基本关系 R 为参照关系 (Referencing Relation)，基本关系 S 为被参照关系 (Referenced Relation) 或目标关系 (Target Relation)。

例如，有以下 3 个关系：

学生（学号，姓名，系别，性别，先修课程号，年龄），主码是学号，记录所有在校学生的信息；

课程（课程号，课程名，学分，先修课程号），主码是课程号，记录学校能够开出以供学生选修的所有课程信息；

选修（学号，课程号，成绩），主码是学号+课程号，记录学生选修课程的信息。

其中，选修关系中的“学号”要和学生关系中的主码“学号”相对应，而课程号和课程关系中的主码“课程号”相对应，也就是说选修关系中有两个外码：“学号”和“课程号”。

外码机制描述的是关系数据库中表与表之间的关系，也就是数据之间的联系。例如，选修关系的两个外码描述了选修关系中选课记录与学生信息和课程信息之间的关联，选课信息一定是学生关

系中的某一学生选修了课程关系中的某门课程，如图 1-20 所示。

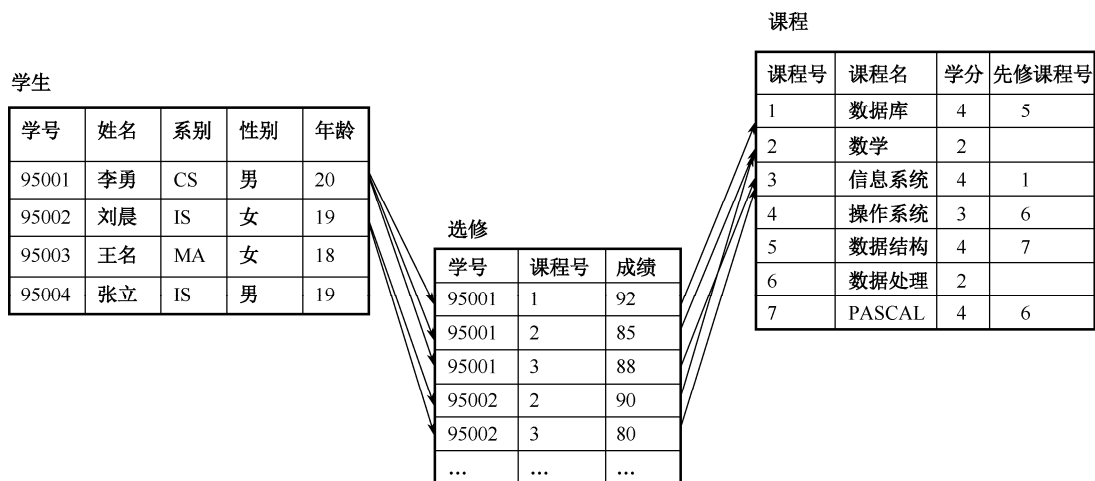


图 1-20 表间的联系（数据之间的联系）

4. 关系的完整性

1) 实体完整性

关系模型中以主码为唯一性标志，主码中的属性（即主属性）不能为空值。

规则 1.1 实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。

简单来讲，就是关系中的主码的取值必须是唯一的和非空的，这里非空指的是每一个构成主码的属性都要非空。例如，学生的学号必须是非空的和唯一的；而选修关系的主码包含两个属性——学号和课程号，因此学号和课程号的组合的取值必须是唯一的，而且均不能取空值。这和实际应用也是非常符合的。试想，在一个选课记录中，假设学号为空值，课程号为“C001”，则表示有一个不知道学号的学生选修了“C001”课程，但只要是在校的学生，就一定要有学号的。

2) 参照完整性

规则 1.2 参照完整性规则：若属性（或属性组）F 是基本关系 R 的外码，它与基本关系 S 的主码 K_s 相对应（基本关系 R 和 S 不一定是不同的关系），则 R 中的每个元组在 F 上的值或者取空值（F 的每个属性值均为空值），或者等于 S 中某个元组的主码值。

如在图 1-19 中，选修关系的学号和课程号都是外码，学号和课程号的取值就要符合参照完整性规则，即要确保选修关系中学号的取值与学生关系中某一学生的学号相等，且课程号的取值与课程关系中某一门课程的课程号相等，参照完整性限定的选课信息记录是该校某在校生选修了某一门课程。

3) 用户定义的完整性

用户定义的完整性约束就是针对某一具体关系数据库的约束条件，它反映了某一具体应用所涉及的数据必须满足的语义要求。

例如，学生的学号的编码格式、成绩的取值范围或取值集合等。

1.4.2 关系数据库设计

什么是数据库设计呢？广义地讲，是数据库及其应用系统的设计，即设计整个数据库应用系统，如图书管理系统、办公自动化系统、电子政务系统、财务管理系统、电子商务平台等。狭义地讲，是数据库设计本身，即设计数据库的各级模式并建立数据库，这是数据库应用系统设计的一部分。本书以某学校的图书管理系统设计和实现为目标，设计数据库和应用系统，并使用 Visual FoxPro 开发工具实现图书管理系统的基本核心功能；本节主要介绍狭义的数据库设计，以图书管理系统的数据库设计为例，介绍数据库设计的方法、技术和步骤。

数据库设计要基于一个给定的应用环境，根据用户的信息需求、处理需求和数据库的支撑环境，利用数据模型和应用程序模拟现实世界中该单位的数据结构和处理活动的过程，是数据设计和数据处理设计的结合。规范化的数据库设计要求数据库中的数据达到最大程度的共享、最小的冗余度，以及较高的独立性，并保证数据的一致性、正确性和相容性。

1. 数据库设计方法

早期数据库设计主要采用手工与经验相结合的方法。设计的质量往往与设计人员的经验、水平有直接关系，缺乏科学理论和工程方法的支持，设计质量难以保证，常常是数据库运行一段时间后又不同程度地发现各种问题，需要进行修改甚至重新设计，增加了系统维护的代价。

为此，人们努力探索，提出了各种数据库设计方法。

(1) 新奥尔良 (New Orleans) 方法。该方法运用软件工程的思想，按一定的设计规程，应用工程化方法，把数据库设计分为若干阶段和步骤，并采用一些辅助手段实现每一过程。

(2) 基于 E-R 模型的数据库设计方法。该方法用 E-R 模型来设计数据库的概念模型，是数据库设计阶段广泛采用的方法。

(3) 3NF (第三范式) 的设计方法。该方法以关系数据库理论为指导来设计数据库的逻辑模型，是设计关系数据库时在逻辑阶段可以采用的一种有效方法。

(4) 对象定义语言 (Object Definition Language, ODL) 方法。这是面向对象的数据库设计方法。该方法用面向对象的概念和术语来说明数据库结构。

以上方法都属于规范设计法，本质上看仍然是手工设计方法，其基本思想是过程迭代和逐步求精。在实际数据库应用系统设计过程中，一般会遵循一定的步骤，运用相应的设计方法和工具，基于需求分析所得到的数据管理需求、系统的业务功能处理需求以及业务处理规则和完整性需求，设计数据库的概念结构、逻辑结构和物理结构，创建数据库和开发应用系统。

2. 数据库设计步骤

按照规范化设计的方法，考虑数据库及其应用开发全过程，将数据库应用系统设计分为以下 6 个阶段：需求分析、概念结构设计、逻辑结构设计、物理结构设计、数据库系统实施、数据库系统运行和维护。

在数据库设计过程中，需求分析和概念结构设计可以独立于任何数据库管理系统进行；逻辑结构设计、物理结构与选用的数据模型以及 DBMS 密切相关。

数据库设计之前，首先必须选定参加数据库设计的人员，包括系统分析员、数据库设计师、应

用开发者、数据库管理员和用户代表。系统分析和数据库设计人员是数据库设计的核心人员，他们将自始至终参与数据库设计，他们的水平决定了数据库系统的质量。用户和数据库管理员在数据库设计中也是举足轻重的，他们主要参加需求分析和数据库的运行和维护。应用开发者包括程序员和操作员，分别负责编制程序和准备软硬件环境，他们在系统实施阶段参与进来。

设计一个完善的数据库应用系统是不可能一蹴而就的，它往往是上述 6 个阶段的不断反复。在设计过程中要把数据库的设计和对数据库中数据处理的设计紧密结合起来。

3. 需求分析

需求分析是数据库应用系统设计的基础，也是最困难和最耗时间的一步。本阶段的主要任务就是通过调查、收集和分析，获得用户对数据库的信息需求、处理需求、安全性和完整性需求。信息需求指的是需要在数据库中存储的数据信息以及信息的内容、格式、获取方式等；处理需求指的是用户希望数据库应用系统完成什么处理功能，处理的响应时间有什么要求，处理的方式是什么。每一项处理功能用到的、产生的以及需要存储、传播的数据信息的总和就是数据库的信息需求。这一阶段会用数据流图和数据字典等工具描述需求分析的结果，完成需求分析报告。

4. 概念结构设计

概念结构设计是整个数据库设计的关键，它通过对用户需求进行综合、归纳与抽象，形成独立于具体 DBMS 的概念模型。

概念结构设计的步骤如下。

- (1) 对需求分析得到的数据需求进行综合、归纳、抽象，得到所有的信息实体。
- (2) 分析实体之间的联系。
- (3) 分析每个实体的属性，绘制 E-R 图。
- (4) 确定每个属性的域。
- (5) 分析每个实体的候选码和主码。
- (6) 检查模型的冗余度。
- (7) 验证模型能否支持用户需求。
- (8) 与用户进行确认。

通常，数据库的概念结构用 E-R 图来表示。

5. 逻辑结构设计

关系模型的逻辑结构是一组关系模式的集合。E-R 图则是由实体、实体的属性和实体之间的联系 3 个要素组成的。所以，将 E-R 图转换为关系模型，实际上就是要将实体、实体的属性和实体之间的联系转换为关系（表）。这种转换一般遵循如下原则。

1) 实体与实体属性的转换。

实体与实体属性的转换即一个实体型转换为一个关系（表）。实体的属性就是关系的属性。实体的码就是关系的码。例如，图 1-17 中学生实体 Student 及其属性可以转换为如下关系，其中学号 Number 为学生关系的码。

Student(Number, Name, DOB, Gender, Age)

例如，图 1-21 中教师 Teacher 实体与班级 Class 实体转换为两个关系（表）。

Teacher(Tno, Tname, Ttitle, Tsex)

Class(ClassNo, ClassName, StuNumber)

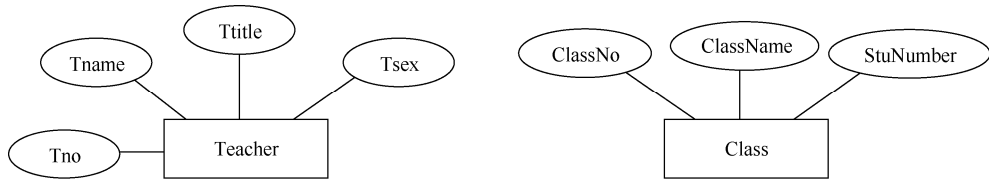


图 1-21 Teacher 实体&班级 Class 实体

2) 实体间联系的转换

(1) 一个 1 : 1 联系可以转换为一个独立的关系表，也可以将任意一端关系中的码属性添加到另一端的表（表）中。

如果转换为一个独立的关系（表），则与该联系相连的各实体的码及联系本身的属性均转换为关系的属性，每个实体的码均是该关系的候选码，任选其中一个作为主码即可。

如果不转换为独立的关系（表），则需要在一个关系（表）的属性中加入另一个关系（表）的码和联系本身的属性，而原来的码不变。

例如，假设一个班级只能由一个班主任（教师）管理，一个班主任也只能管理一个班，则教师与班级之间具有一对一联系，如图 1-22 所示。



图 1-22 1 : 1 联系

将其转换为关系有以下 3 种方法。

转换成一个独立的关系（表）：

Manages (Tno, ClassNo)

将“Teacher”关系（表）中的主码属性职工号“Tno”加入到“Class”关系（表）中，并作为“Class”关系（表）的外码，参照“Teacher”关系（表）的主码：

Class (ClassNo, ClassName, StuNumber, Tno)

将“Class”关系（表）中的主码属性班级号“ClassNo”加入到“Teacher”关系（表）中，并作为“Teacher”关系（表）的外码，参照“Class”关系（表）的主码：

Teacher(Tno, Tname, Ttitle, Tsex, ClassNo)

(2) 一个 1 : N 联系的转换可将“1”端关系中的码加入“N”端实体对应的关系模式中，并作为 N 端关系模式的外码，而原来的码不变。



图 1-23 1 : N 联系

例如，假如有一个学生“BelongsTo”的联系，即一个学生只能属于一个班级，一个班级可能有多个学生，该联系为1：N联系，将其转换为关系模式的方法如下。

将“1”端实体的主码属性班级号“ClassNo”添加到“N”端实体对应的关系Student关系（表）中，并作为该关系（表）的外码，原主码不变。

Student(Number , Name , DOB , Gender , ClassNo , Age)

（3）一个M：N联系转换为一个关系：必须转换为一个独立关系（表），与该联系相连的各实体的码及联系本身的属性均转换为新关系的属性，而关系（表）的码为各实体码的组合，如图1-24所示。

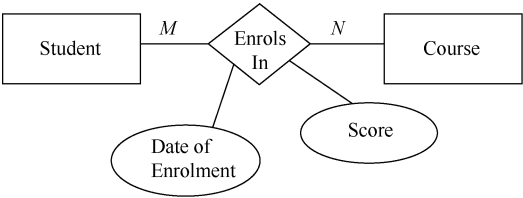


图 1-24 M：N 联系

例如，假如有一个学生选修课程“EnrolsIn”的联系，即一个学生可以选修多门课程（Course（Cno , Cname , Ccredit , Ctype)), 一门课程可以被多个学生选修，每个学生选修课程都有选课时间，考试后有一个成绩。该联系是一个M：N联系，可将其转换为如下关系（表）。

EnrolsIn (Number , Cno , DateofEnrolment , Score)

6. 图书管理系统数据库

下面给出简化版的图书管理系统数据库。实际中由于涉及不同图书馆的不同业务功能，系统比较复杂，本书实例中忽略了许多细节，保留图书馆最基本的功能业务，包括图书信息管理和查询、读者信息管理、图书借阅和归还等功能。为了记录这些业务处理所需要的数据信息，需要设计关系（表）存储图书基本信息、读者信息、图书借阅信息、用户口令信息等。具体关系（表）的结构设计如下。

1) 书籍情况表（Book）

这里设计Book关系（表）来存储图书馆所有图书的信息，包括图书编号、书名、作者姓名、出版社名称、出版日期、类别、馆藏册数、图书定价等信息，图书编号是唯一的。

Book（图书编号，书名，作者，出版社，出版日期，类别，册数，定价，备注），其中主码是“图书编号”其结构如表1-1所示。

表 1-1 Book 的结构

属性（字段）名	类 型	宽 度	小数位数	约 束	能否取空值
图书编号	字符型	10		主码，以ts开头的数字串	否
书名	字符型	40			否
作者	字符型	20			否
出版社	字符型	20			否