

第 6 章 数据库

随着计算机的发展与应用，数据库技术已经成为信息技术的重要组成部分，它是现代计算机信息系统和计算机应用系统的基础与核心。目前，计算机数据处理的应用已是计算机应用的主流，而数据处理的核心是数据管理。数据库技术主要是用于数据管理的技术。本章介绍数据库技术的一些基本知识。

本章主要学习内容

- I 数据库的基本概念
- I 数据库的组织模型
- I 关系数据库的基本知识
- I 关系数据库的设计
- I 结构化查询语言
- I Access 2007 数据库

6.1 数据库的基本概念

6.1.1 信息、数据与数据处理

简单地说，信息（Information）就是对客观事物属性的反映。这一反映指的是关于某一客观系统中某一事物的某一方面属性或某一时刻的表现形式。对事物的属性与表现形式的反映就是信息。数据（Data）实际上就是反映客观事物属性的物理符号的记录。数据的表现形式可以是文字、数字、符号、声音、图像等，它最终以消息、情报、知识等具体形式提供给人们作为决策的依据。数据是信息的具体表现形式，是信息的载体。

计算机中的数据一般分为两类，一类与程序仅有短时间的交互关系，随着程序进入内存，也随着程序的结束而消亡，这些数据也被称为临时性数据。另一类数据则是存储在计算机的外存上，计算机系统能够长期使用它们，这类数据被称为永久性数据。数据库中的数据就是这种永久性数据之一。

计算机中的数据有型（Type）与值（Value）之分，数据的型给出了数据的表示类型，如整型、实型、字符型等，而数据的值给出了符合给定类型的值，如整型值 150、实型值-128.38、字符型值“CHINA”等。

数据是需要处理的。所谓数据处理指的是对各种类型的数据进行收集、存储、分类、计算、加工、检索及传输的过程。在计算机应用普及的时代，计算机已成为了数据处理的主要工具。

6.1.2 数据库

数据库（DataBase，DB），顾名思义就是存放数据的仓库，只不过这个仓库是在计算机存

储设备上，而且数据是按一定的模型存放。人们收集并整理出一个应用所需的大量数据之后，应将其保存起来以便进一步加工处理，提取有用信息。在科学技术飞速发展的今天，人们的视野越来越广，数据量急剧增加。过去人们把数据存放在文件柜里，现在人们借助计算机技术和数据库技术科学地保存和管理大量而复杂的数据，以便能方便而充分地利用这些宝贵的数据资源。

利用数据库方法组织数据与利用文件系统方法组织数据相比，数据库组织方法具有更强的数据管理能力。数据库组织数据具有如下明显的特征。

1. 数据集中控制

在文件管理方法中，文件是分散的，每个用户或每种处理都有各自的文件，这些文件之间一般是没有联系的，因此，不能按照统一的方法来控制、维护与管理。而数据库管理很好地解决了这一问题，它可以集中控制、维护和管理相关数据。

2. 数据独立

数据库中的数据独立于应用程序。数据的独立性包括数据的物理独立性和逻辑独立性，它为数据库的使用、调整、优化和进一步扩充提供了方便，提高了数据库应用系统的稳定性。

3. 数据共享

数据库中的数据可以供多个用户使用，每个用户可以只与数据库中的一部分数据发生联系，且用户数据可以重叠，用户可以同时存取数据而互不影响，大大提高了数据库的使用效率。

4. 减少冗余

数据库中的数据不是面向应用，而是面向系统。数据的统一定义、组织和存储，集中管理避免了不必要的数据库冗余，也提高了数据的一致性。

5. 数据结构化

整个数据库按一定的结构形式组织，数据在记录内部和记录类型之间相互关联，用户可以通过不同的路径存取数据。

6. 统一的数据保护功能

在多用户共享数据资源的情况下，对用户使用数据有严格的检查，能够对数据库访问提供密码保护与存取权限控制，拒绝非法用户访问数据库，以确保数据的安全性、一致性和并发控制。

数据库是需要长期存放在计算机内，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，具有较小的冗余度，较高的数据独立性和易扩展性，并可为不同的用户共享。数据库中的数据是通过数据库管理系统来管理的。

6.1.3 数据库管理系统

数据库管理系统（DataBase Management System, DBMS）是一个管理数据库的软件系统。它为用户提供访问数据库的接口，应用程序只有通过它才能和数据库打交道。数据库管理系统是建立在操作系统基础之上，位于操作系统和用户之间的一个数据管理软件，任何数据操作都是在它的管理下进行的。

数据库管理系统主要功能有：

(1) 数据模式定义功能：即为数据库构建数据框架。

(2) 数据存取的物理构建功能：为数据模式的物理存取与构建提供有效的存取方法与手段。

(3) 数据操纵功能：为用户使用数据库的数据提供方便，如在数据管理系统中查询、插入、修改与删除数据操作都由数据库管理系统来实现。

(4) 数据的完整性、安全性定义与检查功能。数据的完整性和安全性是数据库保护的两个不同的方面，安全性是评价数据库系统的一个重要指标，数据安全性是指数据库中的数据不被破坏和泄漏，以及不被非法用户修改。它标志了程序和数据等信息的安全程度。一个完善的数据库系统应防止非法用户访问或使用数据库系统中的程序和数据。完整性是指数据库中的数据的正确性和相容性，它应反映客观现实世界中的实体的本来面貌。数据库系统提供相应的功能来保护数据的完整性。例如，防止合法用户在使用数据库时向数据库中加入不合语义的数据就是一种完整性保护。

(5) 数据库的并发控制与数据恢复功能。数据库可以由多个用户同时操作同时使用。在多用户环境下不止一个程序运行，也就是说，多个程序并发运行。如果对数据库的并发操作不加控制，就会产生不正确的数据，破坏数据的一致性。因此，数据库应提供并发功能。在数据库系统中，不但要有各种检验和控制，以减少出错的可能性，还得有相应的措施来实现出错后的恢复。因此，系统必须具有恢复功能。

(6) 数据的服务功能：该功能为数据库提供了如数据库的复制、转存、重组、性能监测、分析等功能。

为完成以上6种功能，数据库管理系统提供以下的数据语言：

(1) 数据定义语言：负责数据的模式定义与数据的物理存取构建。

(2) 数据操纵语言：负责数据的操纵，如数据查询与增加、删除与修改等。

(3) 数据控制语言：负责数据完整性、安全性的定义与检查及并发控制、故障恢复等。

目前，计算机厂商已开发出了多种具有优势的数据库管理系统，比较著名的系统有用于管理小型数据库的 Visual FoxPro、Access；管理中型数据库的 Sybase、SQL Server；管理大型数据库的 Oracle 等。

6.1.4 数据库系统

狭义地讲，数据库系统 (DataBase System, DBS) 是由数据库、数据库管理系统和用户组成的系统；广义地讲，它是由计算机硬件、操作系统、数据库管理系统，以及在它支持下建立起来的数据库、应用程序、用户和数据库管理员组成的一个整体。数据库系统可以用图 6-1 来描述。

数据库是为多用户共享的，因此需要有人进行规划、设计、协调、维护和管理，负责这些工作的人员称为数据库管理员 (DataBase Administrator, DBA)。

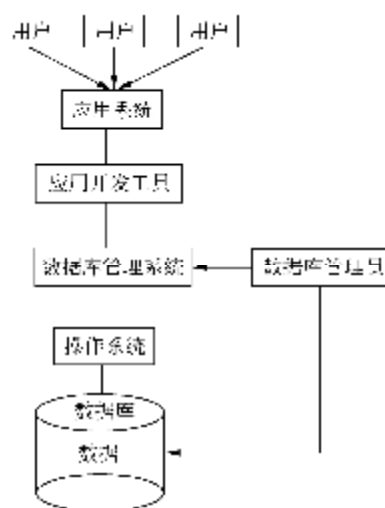


图 6-1 数据库系统

6.2 数据的组织模型

由于计算机不可能直接处理现实世界中的具体事物，因此，人们必须事先把具体事物转换成计算机能够处理的数据。数据组织模型定义了数据的逻辑模型，它也描述了数据库中不同数据之间的关系。在数据库设计发展过程中，曾使用过层次模型、网状模型和关系模型三种。

6.2.1 层次模型

在现实世界中，许多实体之间的联系就是一个自然的层次关系，如行政机构、家族关系等都是层次关系。图 6-2 是学校中系的层次模型。

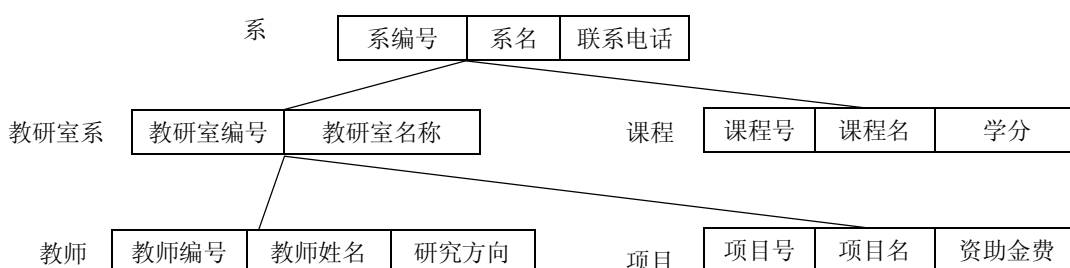


图 6-2 层次模型示例

层次模型用一棵倒置的树来描述数据之间的关系，树的结点表示实体集（多条记录），结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（多）”。通常把表示 1 的实体集放在上方，称为父结点，表示 M 的实体集放在下方，称为子结点。层次模型的结构特点是：

- （1）有且仅有一个根结点；
- （2）根结点以外的其他结点有且仅有一个父结点。

由此可见，层次模型只能表示“1—M”关系，而不能直接表示“M—M”关系。

在层次模型中，一个结点称为一个记录型，用来描述实体集。每个记录型可以有一条或多条记录，上层一条记录对应下层一条或多条记录，而下层每条记录只能对应上层一条记录。例如，系记录型有计算机系、电子系、外语系等记录。而计算机系的下层记录有软件、网络、应用等教研室和数据结构、操作系统、数据库等课程，软件研究室下层又有员工和项目记录值。

层次模型的优点是数据结构类似金字塔，不同层次之间的关联直接而且简单；缺点是由于数据纵向发展，横向关系难以建立，数据可能会重复出现，造成管理维护的不便。

6.2.2 网状模型

网状模型是一种比层次模型更具普遍性的结构，虽然该模型也使用倒置树型结构，但该模型能克服层次模型的一些缺点。网状模型与层次结构不同的是，网状模型的结点间可以任

意发生联系，能够表示各种复杂的联系。如图 6-3 所示是一个商品管理的网状型数据库。网状模型的优点是可以更直接地描述现实世界，可以避免数据的重复性，缺点是关联性比较复杂，尤其是当数据库变得越来越大时，关联性维护的复杂度更高。

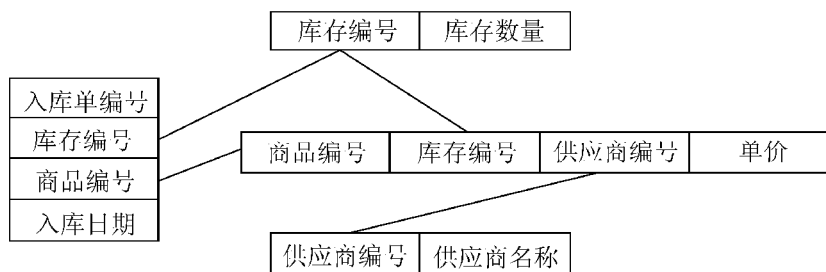


图 6-3 网状模型示例

6.2.3 关系模型

用二维表结构来表示实体及实体间联系的数据模型称为关系数据模型（简称关系模型），关系模型突破了层次模型和网状模型的许多局限。关系是指由行与列构成的二维表。在关系模型中，实体和实体间的联系都是用关系表示的。关系数据库是由若干关系组成的数据集合，如图 6-4 所示。关系模型是目前应用最广、理论最成熟的一种数据模型。

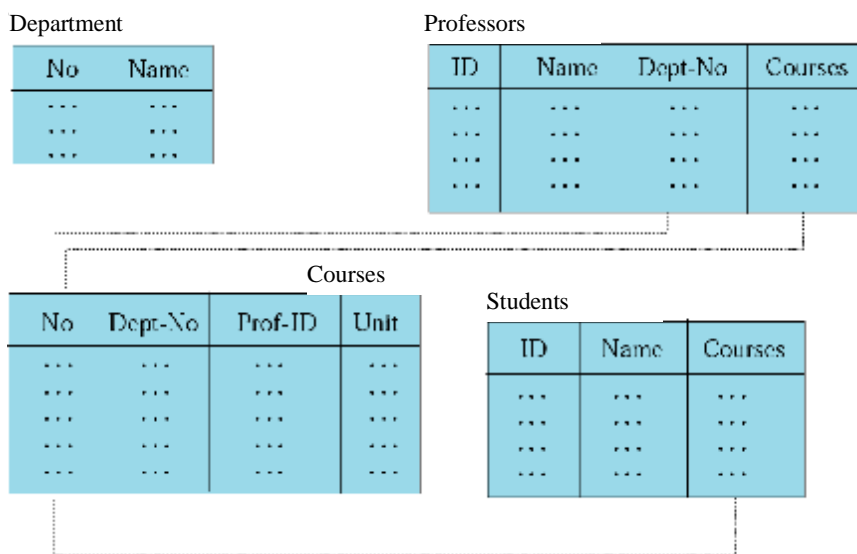


图 6-4 关系型数据库示例

6.3 关系数据库基本知识

关系数据库是采用关系模型作为数据组织方式的数据库。1970 年，IBM 公司的研究员 E.F.Codd（图灵奖获得者）发表了题为“大型共享数据库的关系模型”的论文，首次提出了数据库的关系模型，为关系数据库发展奠定了基础。关系数据库的特点在于它将每个具有相同

属性的数据独立地存储在一个表中。对任一表而言，用户可以新增、删除和修改表中的数据，而不会影响表中的其他数据。关系数据库产品一问世，就以其简单清晰的概念，易懂易学的数据库语言，深受广大用户喜爱。著名的 DB2、Oracle、Sybase、Informix 等都是关系数据库管理系统（Relational DataBase Management System RDBMS）。

6.3.1 常用的术语

1. 关系

一个关系就是一个二维表，每个关系有一个关系名。在关系型数据库管理系统中，每个关系（数据库中的表）用一个文件来存储，关系名一般用文件主名。表 6-1 是一个“学生情况”表，把这个文件存储时，可把关系名“学生情况”简写为 xsqq，系统会自动加上扩展名。扩展名随系统的不同而不同，在 SQL Server 中，扩展名为 .dbf，在 Access 中，扩展名 .mdb。

表 6-1 学生情况表

St_ID	Name	Class_No
970001	John	9501
.....
.....

2. 属性

二维表中的列称为属性（也称字段），属性在表中是列头。每一个属性表示了存储在它下面的数据含义。表中的每一列在关系范围内的名称必须唯一。在数据库管理系统中，每个字段须定义名称、数据类型与数据宽度等属性。表 6-1 “学生情况”表有 3 个属性（字段），它们分别是 St_ID（学号）、Name（姓名）与 Class_No（班级）。

3. 记录

在一个二维表中，表中的行称为元组，元组也称为记录。在二维表中，记录必须有唯一性标识。如，在“学生情况”表中，St_ID（学号）是记录的唯一性标识，因为表中不可能有 St_ID 相同的记录。表 6-1 “学生情况”表有 3 条记录。

4. 关系模式

关系模式是对关系结构的描述。一个关系模式对应一个关系的结构，关系模式简化表示的方法为：关系名（属性名 1，属性名 2，...，属性名 n）。如表 6-1 的关系模式也可以简化描述成 xsqq(St_ID, Name, Class_No)。

5. 域

在关系数据库中，域指属性域，是指属性（字段）的取值范围。例如，student 表中的“性别”字段的取值是“男”或“女”，所以该字段的域是“男”和“女”两个值。假如某一个表中的字段存放学生的某门课程的成绩，按传统成绩计分方式，成绩是在 [0, 100] 之间，那成绩字段的域就为 [0, 100]。

6. 键码

键码（Key）是关系模型中的一个重要概念，在关系中用来标识行的一列或多列。在图 6-5 中，student 表的“学号”、“姓名”、“年龄”与“性别”都是键码。

7. 候选关键字

候选关键字 (Candidate Key) 是唯一标识表中的一行的一个属性或属性集。如图 6-5 中的“学号”、“课程号”就是候选关键字，如果表中没有姓名相同的记录，那姓名也可以作为候选关键字。候选关键字也称候选键。

8. 主关键字

主关键字 (Primary Key) 是被挑选出来作为表行的唯一标识的候选关键字。一个表只有一个主关键字，主关键字又称为主键。在图 6-5 中，student 表与 course 表分别有“学号”与“课程号”主关键字。

9. 公共关键字

在关系型数据库中，关系之间的联系是通过相容或相同的属性或属性组来表示的。如果两个关系中具有相容或相同的属性或属性组，那么这个属性或属性组被称为这两个关系的公共关键字 (Common Key)。在图 6-5 中，student 表与 sc 表有公共关键字“学号”，sc 表与 course 表有公共关键字“课程号”。

10. 外关键字

如果公共关键字在一个关系中是主关键字，那么这个公共关键字被称为另一个关系的外关键字 (Foreign Key)，如图 6-5 所示。由此可见，外关键字表示了两个关系之间的联系。外关键字又称作外键。

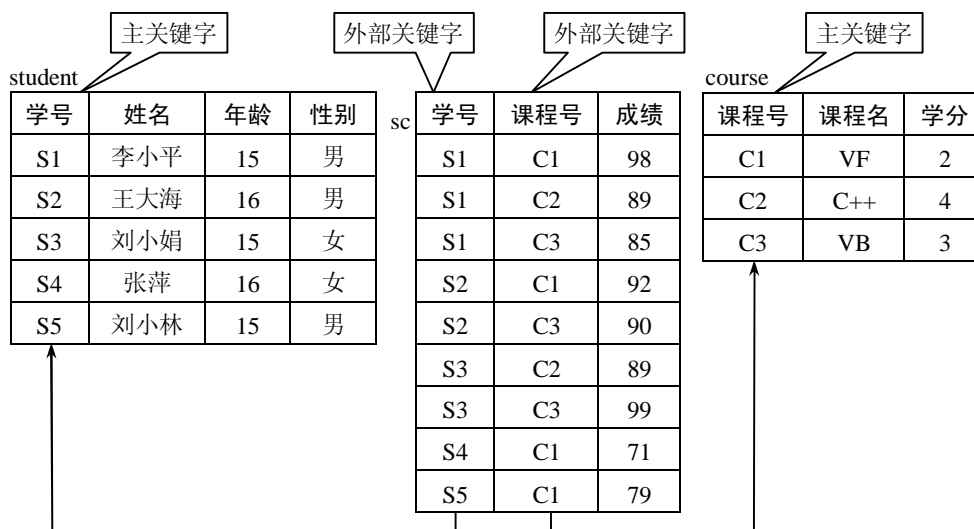


图 6-5 关系型数据库

6.3.2 关系的基本特点

关系必须规范化 (关系的规范化内容将在本章的后续内容中介绍)。规范化是指关系模型中的每一个关系模式都必须符合关系的基本要求。关系有如下基本特征:

- I 关系中的每个属性必须是不可分的数据单元，即表中不能有表。
- I 二维表中元组个数是有限的，即元组个数的有限性。
- I 二维表中元组不能重复，即元组的唯一性。

- l 二维表中元组的次序可以任意交换，即元组的次序无关性。
- l 二维表中属性名不能相同，即属性名的唯一性。
- l 二维表中属性可任意交换次序，即属性的次序无关性。

6.3.3 关系的操作

在关系型数据库中，定义一些操作来通过已知的关系创建出新的关系。这些操作很多，这里主要介绍插入、删除、更新、选择、投影、连接操作。

1. 插入

插入操作是应用于一个关系的操作，该操作是向二维表中插入一条记录。图 6-6 给出了在 course 表中插入一条记录的示例。

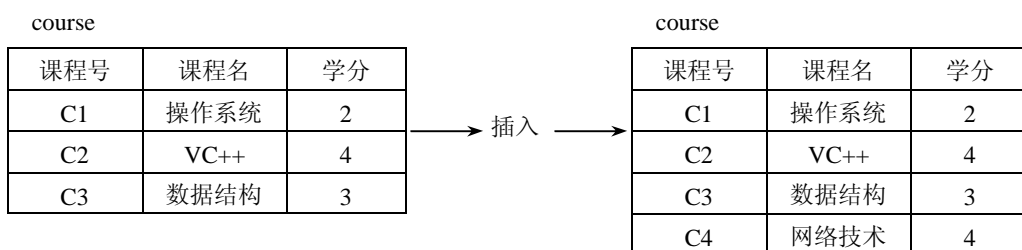


图 6-6 记录的插入操作

2. 删除

删除操作也是应用于一个关系的操作。该操作是根据要求删去表中相应的记录。图 6-7 给出了在 course 表中删除课程名为“数据结构”记录的示例。

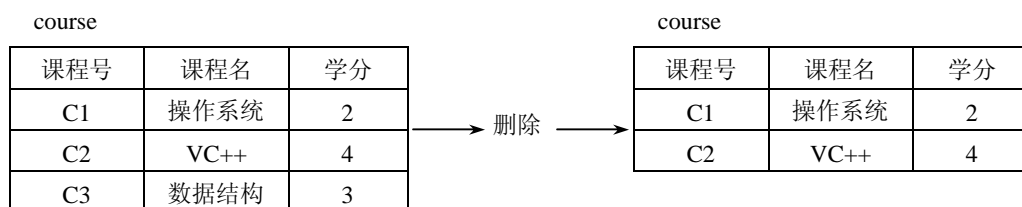


图 6-7 记录的删除操作

3. 更新

更新也是应用于一个关系的操作，用于更新记录中部分属性的值。图 6-8 给出了在 course 表中更新第三条记录“学分”字段的值的示例。

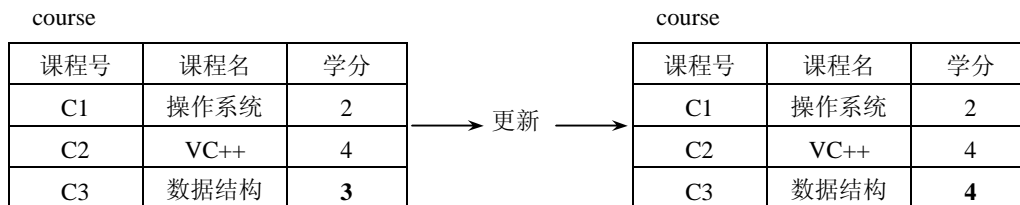


图 6-8 字段值的更新

4. 选择

选择操作是应用于一个关系并产生另一个新关系的操作。新关系中的元组（记录）是原

关系中元组（记录）的子集。选择操作根据要求从原关系中选择部分记录，属性（字段）的个数与原关系一致。图 6-9 给出了在 `course` 表中选择学分数大于或等于 3 的课程情况并生成 `course1` 表。生成的表与原表字段数相同，只是记录数不同而已。

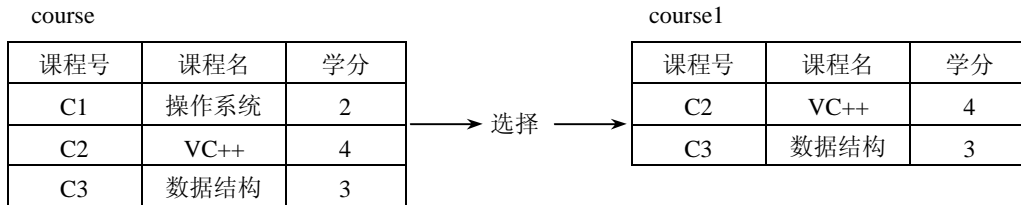


图 6-9 选择操作

5. 投影

投影操作也是一种一元操作，它应用于一个关系并产生另外一个新关系，新关系中的属性（字段）是原关系中属性（字段）的子集。投影操作根据要求从原关系中选择部分字段，生成组（记录）的个数与原关系一致。图 6-10 给出了利用 `course` 表生成课程号与课程名的关系表——`course2` 表。生成的表与原表记录数相同，只是字段数不同而已。

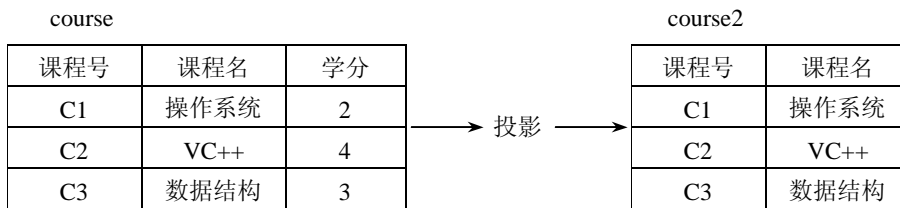


图 6-10 投影操作

6. 连接

连接操作是一个二元操作，它基于两个关系的共有属性把两个关系组织起来。连接操作十分复杂，结果随连接操作的情况变化而变化。图 6-11 给出连接 `course` 表与教师担任课程情况表 `taught-by`，生成一个信息更加全面的关系课程综合情况表。该表既包含了课程号、课程名与学分，也包含了担任该课程教学教师的信息的 `course-all` 表。

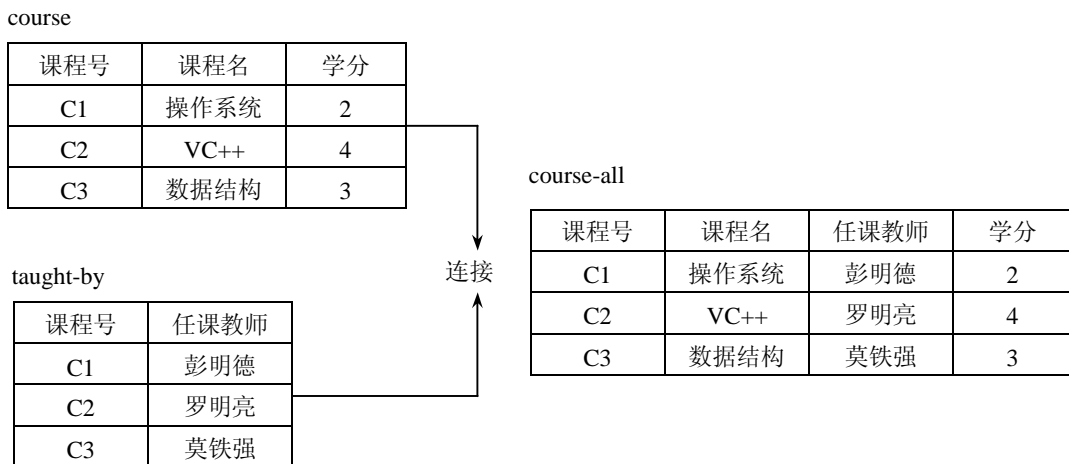


图 6-11 连接操作

6.3.4 关系型数据库的主要对象

关系型数据库对象是一种数据库组件，是数据库的主要组成部分。常见的数据库对象有以下几种。

1. 表 (Table)

数据库中的表由行 (Row) 和列 (Column) 组成。列由同类的信息组成，每列又称为一个“字段”，每列的标题称为列名。行包括了若干列信息项，一行数据称为一条“记录”，它表达有一定意义的信息组合，是一个含义非常明确的实体。一个数据库表由一条或多条记录组成，没有记录的表称为空表。每个表中通常都有一个主关键字，用于唯一地标识一条记录。

2. 索引 (Index)

索引是根据指定的数据库表列建立起来的顺序，它提供了快速访问数据的途径。什么是索引文件？一本书的目录可看作是一个索引文件，该文件中只有章节标题及其所在的页码，根据此目录文件可很快地找到所需内容，而不必一页页地翻阅。索引所基于的字段称为索引项。索引文件中建立了索引项值与记录号的对应关系。这样，系统查找记录时只要在打开表的同时打开索引文件，先在索引文件中快速找到要查的索引项值，而对应的记录号便确定了所查记录在表中的位置。

3. 视图 (View)

视图看上去同表一样，具有一组命名的列和数据项，但它其实是一个虚拟的表，在数据库中并不实际存在。视图是由查询数据库表产生的，它限制了用户能看到和修改的数据。显而易见，视图可以用来控制用户对数据的访问，并能简化数据的显示，即通过视图只显示用户需要的数据信息。

4. 图表 (Diagram)

图表就是数据库表之间的关系示意图，利用它可以编辑表与表之间的关系。图 6-12 显示了一个数据库有 3 个表以及每个表之间的关联关系。

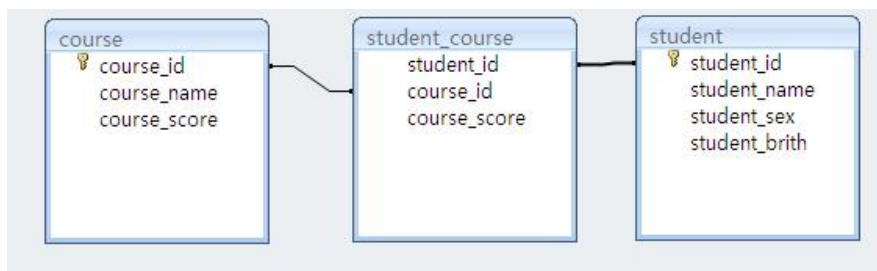


图 6-12 数据库的图表

5. 默认值 (Default)

在表中创建列或插入数据时，默认值用于指定其具体值的列或列数据项赋予事先设定好的值。

数据库中还有规则 (Rule)、触发器 (Trigger)、存储过程 (Stored Procedure) 等对象。这些概念超出了本书范围，在此不作解释，读者可参考其他书籍的相关内容。

6.3.5 表之间的关系

在关系数据库中，每一个表都是一个实体对象，表本身具有完整的结构。但数据库中的表不是孤立的，数据库的表与表之间以关键字相互联系着，数据库依靠表之间的关联把数据以有意义的方式联系在一起。数据库中表之间的关系有如下三种类型。

1. 一对一 (one-to-one)

如果表 A 中的每一条记录，在 B 表中至多有一条记录（也可以没有）与之对应，反之亦然，那么称 A 表和 B 表具有一对一关系。如图 6-13 所示，xscj 表与 xsqq 表是一对一的关系。

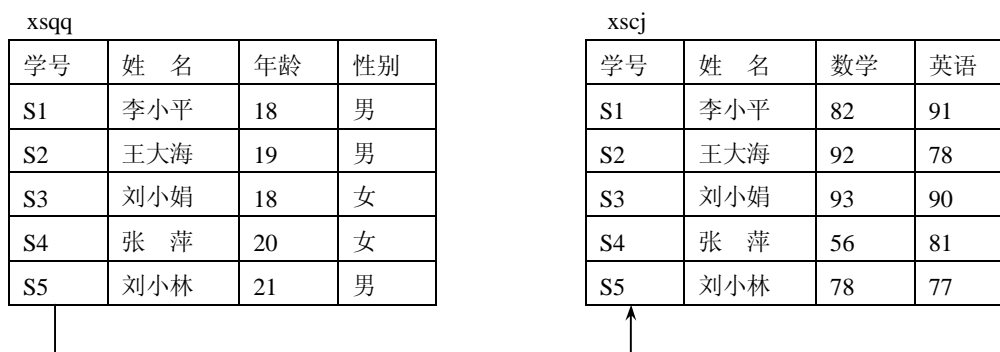


图 6-13 xscj 表与 xsqq 表之间的关系

2. 一对多 (one-to-many)

如果表 A 中的每一项在 B 表中有 n 项与之联系；反之，表 B 中的每一项，在 A 表中至多有一项与之联系，则称表 A 与表 B 具有一对多关系。如图 6-14 所示 xsqq 表与 kccj 表之间的关系是一对多的关系。

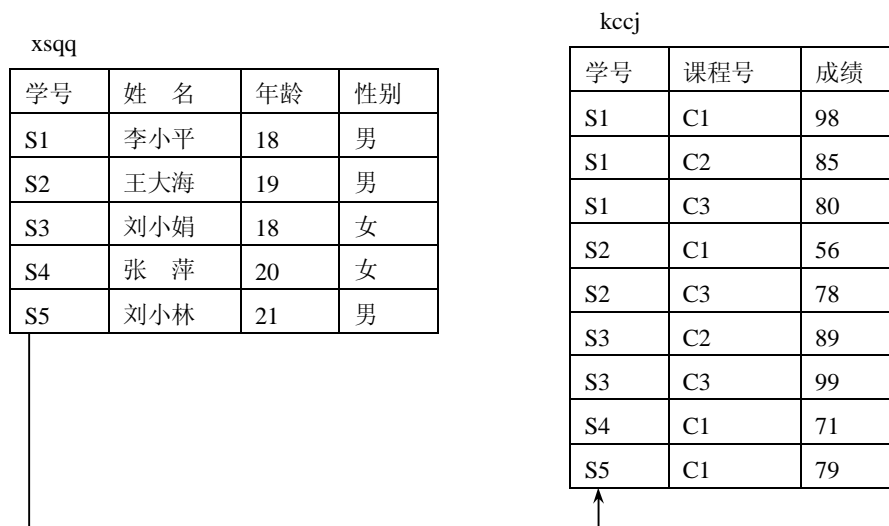


图 6-14 kccj 表与 xsqq 表之间的关系

3. 多对多 (many-to-many)

如果 A 表中的每一条记录，在 B 表中有 n 条记录与之联系，反之，表 B 中的每一条记录

在 A 表中有 m 条记录与之相联系，则称表 A 与 B 表具有多对多关系。多对多的关系需要引入中间表，也叫做联系表，来实现中间表与表 A、表 B 的一对多的关系，因为关系型系统不能直接实现多对多的关系。图 6-5 中的 sc 表就是一个联系表，实现 student 表与 course 表中实体之间的多对多的关系。

6.4 关系数据库设计

数据库设计是指对于一个给定的应用，构造最优的数据库模式，建立数据库，使之能够有效地存储数据，满足用户的各种应用需求。数据库设计的目标是正确反映应用的实际情况。在数据库系统应用中，数据由 DBMS 进行独立地管理，大大减少了数据对程序的依赖性，因而数据库的设计也逐渐成为一项独立的开发活动。

6.4.1 数据库设计过程

一般来说，数据库的设计都要经历需求分析、概念设计、实现设计和物理设计 4 个阶段，图 6-15 显示了数据库的设计过程及每一过程产生的文档。

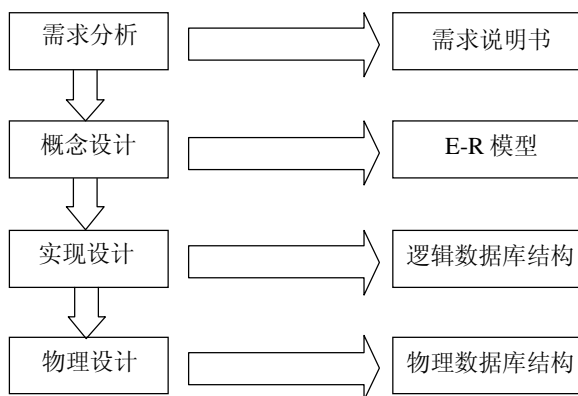


图 6-15 数据库设计过程和产生的文档

1. 需求分析

需求分析是整个设计过程的基础，是最困难、最耗时的一个阶段。需求分析的目的是分析系统的需求。该过程的主要任务是从数据库的所有用户中收集对数据的需求和对数据处理的要求，并把这些需求写成用户和设计人员都能接受的说明书。

例如，要为某学校的学生成绩管理系统设计数据库。数据库工程师通过对学校学生成绩管理进行调查，确定学生成绩管理系统中的实体有学生、班级、课程、专业与系。在该系统中对这些实体要求的属性分别是：

- (1) 学生：学号、姓名、性别、出生日期、联系方式。
- (2) 班级：班级编号、班级名称。
- (3) 课程：课程编号、课程名称、周学时、课程性质（考试/考查）、类型（公共基础/专业基础/专业课/公共选修/专业选修）。
- (4) 专业：专业编号、专业名称。

(5) 系：系编号、系名称、系办地址。

这些数据是设计学生成绩管理数据库的重要依据。

2. 概念设计

概念设计是整个数据库设计的关键。它的目的是将需求说明书中关于数据的需求，综合为一个统一的 DBMS 概念模型。首先根据单个应用的需求，画出能反映每一应用需求的局部 E-R 模型。然后将这些 E-R 模型图合并起来，消除冗余和可能存在的矛盾，得出系统总体的 E-R 模型。

实体-关系图 (Entity-Relationship, E-R) 是由 P.P.Chen 于 1976 年首次提出，提供不受任何 DBMS 约束的面向用户的表达方法，在数据库设计中被广泛用作数据建模的工具。E-R 数据模型问世后，经历了许多次修改和扩充。

E-R 模型的构成成分是实体集、属性集和关系集，其表示方法如下：

(1) 实体用矩形框表示，矩形框内写上实体名。

(2) 实体的属性用椭圆形表示，框内写上属性名，并用无向边与其实体相连。

(3) 实体间的联系用菱形框表示，且根据适当的含义为联系命名，名字写在菱形框中，用无向连线将参加联系的实体矩形框分别与菱形框相连，并在连线上标明联系的类型，即 1:1、1:m 或 n:m。

在此，仍以前面介绍的某学校的学生成绩管理系统数据库设计中的学生与课程为例建立 E-R 图。由需求分析可知，学生实体包含的属性有学号、姓名、性别、出生日期、联系方式，课程实体包含的属性有课程编号、课程名称、周学时、课程性质、类型。可用图 6-16 描述两个实体以及它们的属性。

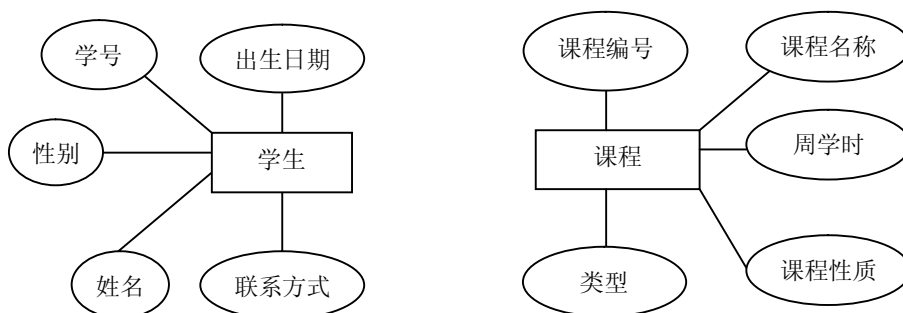


图 6-16 学生与课程实体以及它们的属性

学生实体与课程实体之间的联系命名为“学习”，学生与课程之间的关系显而易见是多对多的关系，因为一个学生学习多门课程，一门课程被多个学生学习。另外学生学习课程会得到成绩。由此，可以画出学生与课程的 E-R 图，如图 6-17 所示。

通过进一步的分析，可得到学生成绩管理系统数据库的 E-R 图，如图 6-18 所示。

3. 实现设计

实现设计的目的是将 E-R 模型转换为某一特定的 DBMS 能够接受的逻辑模式，也就是说把 E-R 图中的实体与实体之间的联系用关系来描述。对关系数据库，主要是完成结构的设计与表的关联设计。

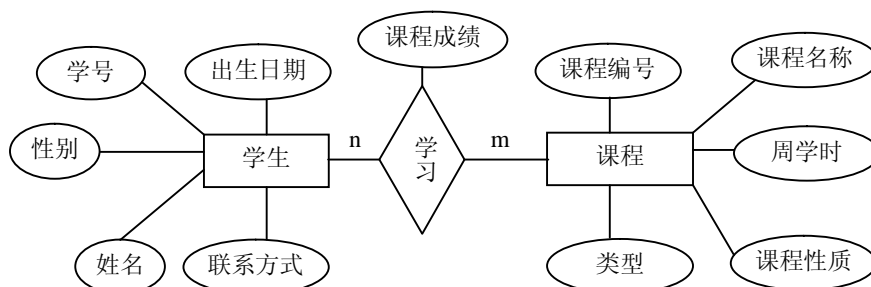


图 6-17 学生与课程实体的 E-R 图

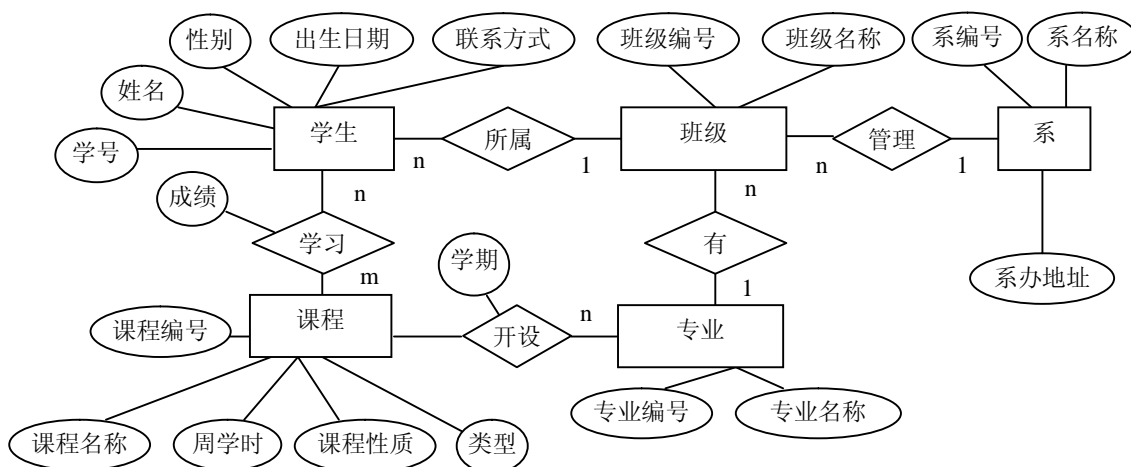


图 6-18 学生成绩管理数据库的 E-R 图

在此，仍以学生成绩管理系统中学生与课程的 E-R 图讨论数据库的实现设计。实现设计实质上就是把实体及实体之间的联系转化为关系。由图 6-18 可知，每个实体对应的关系分别如下（其中带下划线的属性为主键）：

实体名：学生

对应的关系：学生（学号，姓名，性别，出生日期，联系方式）

实体名：课程

对应的关系：课程（课程编号，课程名称，周学时，课程性质，类型）

把实体转换成关系后，还须把联系也转换成关系。转换而成的关系的属性包括联系的两个关系的主键及联系自身的属性。“学习”联系的两个表的主键为“学号”与“课程编号”，“学习”的属性为“成绩”，由此可得联系的关系如下（其中带下划线的属性为外键）：

实体：学生课程成绩

对应的关系：学生课程成绩（学号，课程编号，成绩）

这样就完成了数据的简单逻辑设计。

注意：如果两个实体之间是多对多的关系，一定要为两个实体设计关联表（也称中间表），因为，在关系数据库系统中不能直接实现多对多的关联。如果是一对一或一对多关联，可为两个实体设计关联表，也可以不设计关联表。

4. 物理设计

物理设计的目的在于确定数据库的存储结构。其主要任务包括：确定数据库文件和索引，文件的记录格式和物理结构，选择存取方法，决定访问路径和外存储器的分配策略等。不过这些工作大部分可由 DBMS 来完成，仅有一小部分工作由设计人员完成。例如，物理设计应确定列类型和数据库文件的长度。实际上，由于借助 DBMS，这部分工作难度比实现设计要容易得多。这些内容将在 6.6 节的 Access 2007 中详细介绍。

对于一个程序员，需要了解最多的应该是逻辑设计阶段。因为数据库不管设计好坏，都可以存储数据，但在存储的效率上可能有很大的差别。可以说，逻辑设计阶段是关系数据库存取效率的重要阶段。

6.4.2 关系数据库规范化

在数据库的逻辑设计阶段，常常使用关系规范化理论来指导关系数据库设计。规范化基本思想为：每个关系都应该满足一定的规范，从而使关系模式设计合理，达到减少冗余，提高查询效率的目的。

为了建立冗余较小、结构合理的数据库，将关系数据库中关系应满足的规范划分为若干等级，每一等级称为一个“范式”（Normal Forms, NF）。

范式的概念最早是由 E.F.Codd 提出的，他从 1971 年开始相继提出了三级规范化形式，即满足最低要求的第一范式（1NF），在 1NF 基础上又满足某些特性的第二范式（2NF），在 2NF 基础上再满足一些要求的第三范式（3NF）。1974 年，E.F.Codd 和 Boyce 共同提出了一个新的范式概念，即 Boyce-Codd 范式，简称 BC 范式。1976 年 Fagin 提出了第四范式（4NF），后来又有人定义了第五范式（5NF）。至此，在关系数据库规范中建立了一个范式系列：1NF、2NF、3NF、BCNF、4NF 和 5NF。这 6 种范式一级比一级要求更严格。它们之间的关系如图 6-19 所示。一般数据库的设计至少要符合第三范式。下面详细介绍在关系数据库中常用的第一范式、第二范式与第三范式。

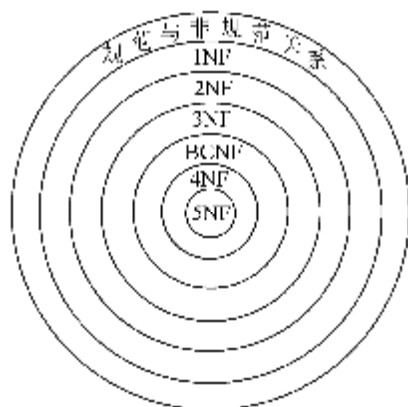


图 6-19 范式之间的关系

1. 第一范式（1NF）

在关系数据库范式设计中，第一范式是对关系模型的基本要求，不满足第一范式的数据库就不是关系数据库。

所谓第一范式是指数据库表的每一列都是不可再分割的基本数据项，同一列不能有多个值，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，就可能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。例如表 6-2 是不符合第一范式要求的表。

表 6-2 “员工联系”表

员工姓名	员工地址	员工联系电话	
		座机	手机
史真真	北京市西城区	010-8800880	13901022549
王颖	北京市朝阳区	010-8800220	13901022345
宋昆	北京市朝阳区	010-8678550	13701022446
李辰	北京市大兴区	010-6754320	13501034567
张莫	北京市丰台区	010-7766889	13367568386
王朋	北京市海淀区	010-5678900	13901033445
赵讯	北京市通州区	010-3344556	13701023456

为了让它符合第一范式要求，把员工联系电话列拆分，得到如表 6-3 所示的符合第一范式要求的表。

表 6-3 “员工联系”表

员工姓名	员工地址	座机	手机
史真真	北京市西城区	010-8800880	13901022549
王颖	北京市朝阳区	010-8800220	13901022345
宋昆	北京市朝阳区	010-8678550	13701022446
李辰	北京市大兴区	010-6754320	13501034567
张莫	北京市丰台区	010-7766889	13367568386
王朋	北京市海淀区	010-5678900	13901033445
赵讯	北京市通州区	010-3344556	13701023456

又如，表 6-4 “产品销售员情况”表也不符合第一范式要求。因为客户信息数据项是可分割的。为了达到第一范式要求，可设计成两个关系，即销售员情况表与客户信息表。如表 6-5 与表 6-6 所示。

表 6-4 “销售员信息”表

员工号	姓名	客户信息	办公电话
.....

表 6-5 “销售员信息”表

员工号	姓名	办公电话
.....

表 6-6 “客户信息”表

客户号	客户名称	客户地址	员工号
.....	

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体或行必须可以被唯一地区分。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。如表 6-3,该关系不满足第二范式,在表 6-3 中加上列“员工编号”,因为每个员工编号是唯一的,因此,每个员工可以被唯一区分。表 6-7 符合第二范式要求,这个唯一属性列被称为主关键字或主键。

表 6-7 “员工联系”表

员工编号	员工姓名	员工地址	固定电话	移动电话机
A0001	史真真	北京市西城区	010-8800880	13901022549
A0002	王颖	北京市朝阳区	010-8800220	13901022345
A0003	宋昆	北京市朝阳区	010-8678550	13701022446
A0004	李辰	北京市大兴区	010-6754320	13501034567
A0005	张莫	北京市丰台区	010-7766889	13367568386
A0006	王朋	北京市海淀区	010-5678900	13901033445
A0007	赵讯	北京市通州区	010-3344556	13701023456

第二范式也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,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。简而言之,第二范式就是非主属性非部分依赖于主关键字,如表 6-8 也不满足第二范式,为了让它满足第二范式,必须把不依赖于主键“学号”的“系编号”与“所在系”列从原表中抽取出来形成一个新的关系,如表 6-9 与表 6-10 所示。

表 6-8 学生情况表

学号	姓名	性别	系编号	所在系
2008001	刘林	女	07	外语系
2008005	王海波	男	07	外语系
2008035	彭珊	女	01	计算机系
2008126	易杨	女	03	机械系

表 6-9 学生情况表

学号	姓名	性别	系编号
2008001	刘林	女	07
2008005	王海波	男	07
2008035	彭珊	女	01
2008126	易杨	女	03

表 6-10 系编号表

系编号	系名
01	计算机系
03	机械系
07	外语系 1

3. 第三范式（3NF）

满足第三范式必须先满足第二范式。也就是说，第三范式要求一个数据库表中不包含已在其他表中包含的非关键字信息。例如，存在一个业务员信息表，有业务员编号、业务员姓名、家庭住址、电话等信息。那么另一表中的客户信息中列出业务员编号后就不能再将业务员姓名、家庭住址、电话等与业务员有关的信息加入客户信息中。如果不存在业务员信息，则根据第三范式也应该构建它，否则就会有大量的数据冗余。简而言之，第三范式就是属性不依赖于其他非主属性。

范式设计的目的是规范化，规范化的目的是为了**保证数据结构更合理，能消除存储异常，使数据冗余尽量小，便于数据的插入、删除和更新**。范式设计的原则是遵从概念单一化“一事一地”原则，即一个关系模式描述一个实体或实体间的一种联系，规范的实质就概念的单一化。范式设计方法是将关系模式投影分解成两个或两个以上的关系模式。分解要求分解后的关系模式集合应当与原关系模式“等价”，即经过自然连接可以恢复原关系而不丢失信息，并保持属性间合理的联系。

总之，设计范式是符合某一种级别的关系模式的集合。数据库的设计范式是数据库设计所需要满足的规范，满足这些规范的数据库是简洁的、结构明晰的，同时，也能保证数据不会发生插入（insert）、删除（delete）和更新（update）操作异常。

6.5 结构化查询语言

结构化查询语言（Structured Query Language，SQL）最早是由 IBM 的圣约瑟研究实验室为其关系数据库管理系统 SYSTEMR 开发的一种查询语言。由于 SQL 语言结构简洁，功能强大，简单易学，因此，自从 IBM 公司 1981 年推出以来，SQL 语言得到了广泛的应用。如今无论是像 Oracle、Sybase、Informix、SQL Server 这些大型的数据库管理系统，还是像 Visual Foxpro、Access 等小型数据库管理系统都支持 SQL 语言。

SQL 语言由数据定义语言、数据操纵语言、数据查询语言（Select 语句）与数据控制语言 4 部分组成。本节介绍前三种语言。

6.5.1 数据定义语言

数据定义语言（DDL）用来定义数据的结构，如创建、修改或者删除数据库对象，常用的数据定义语言有 create、drop 与 alter 等。

1. create

创建一张新关系（表）可以使用 create table 命令，create table 命令格式也很复杂，简单的命令格式如下：

```
create table table_name  
(column_name data_type(DEFAULT or constant_expression))
```

说明: table_name 是新表的名称, column_name 是表中的列名, data_type 指定列的数据类型, 括号中 DEFAULT 与 constant_expression 用于指定字段的宽度为默认值或 constant_expression 的宽度。

例如, 下面的 SQL 语句将建立一个学生表 student。

```
create table student  
(  
    ID_Card varchar(18),  
    Sname varchar(10),  
    Ssex char(2),  
    Sage int,  
    Sschool_number char(6),  
)
```

2. drop

drop 用于删除表。语法格式如下:

```
drop table table_name
```

其中 table_name 就是要删除的表名, 如要删除 loan 表, 语句格式如下:

```
drop table loan
```

alter 命令很复杂, 在此不作介绍。

6.5.2 数据操纵语言

数据操纵语言实现对数据库中表记录的插入、更新与删除操作。这里介绍 insert、update 与 delete 命令的作用与使用方法。

1. insert

该命令实现对数据表记录的插入操作。命令的格式为:

```
insert into relation-name value(...)
```

其中, relation-name 为关系名, value 为定义插入记录的各个字段的值。

如图 6-6 的插入操作可以通过如下的 SQL 语句实现:

```
insert into course value("C4","网络技术",4)。
```

注意: value 中的属性值如果是字符型数据须用引号。

2. delete

该命令实现对数据表记录的删除操作。命令的格式为:

```
delete from relation-name where criteria
```

其中, relation-name 为执行删除操作关系名, criteria 指出删除的记录要满足的条件。

如图 6-7 的删除操作可以通过如下的 SQL 语句实现:

```
delete from course where 课程名="数据结构"
```

3. update

该命令实现对数据表记录的更新操作。命令的格式为:

```
update relation-name  
set attribute1=value1, attribute2=value2,...  
where criteria
```

其中，relation-name 为关系名，attribute1=value1, attribute2=value2 定义更新，criteria 指定更新的条件。

如图 6-18 的更新操作可以通过如下的 SQL 语句实现：

```
update course set 学分=4 where 课程名="数据结构"。
```

6.5.3 数据查询语言

在 SQL 中，最基础与最重要的操作就是查询操作。数据库中数据的查询通过 select 语句来实现。事实上，二维表的选择、投影与连接等操作都可通过查询语句来实现。select 语句的格式很复杂，功能很强大。在此，我们仅介绍使用 select 语句来实现二维表的选择、投影与连接运算。

1. 选择

选择操作使用如下的 select 语句格式：

```
select * from relation-name where criteria
```

其中，*表示选择表的所有属性。

如图 6-9 的选择操作可以通过如下的 select 语句实现：

```
select * from course where 学分>=3
```

2. 投影

投影操作使用如下的 select 语句格式：

```
select attribute-list from relation-name
```

如图 6-10 的投影操作可以通过如下的 select 语句实现：

```
select 课程号,课程名 from course
```

3. 连接

连接操作是一个二元操作，连接操作使用的 SQL 命令格式为：

```
select attribute-list from relation-name1,relation-name2 where criteria
```

其中，属性列表是原来两个输入关系的属性的组合，where 子句明确定义了用于连接的共有属性。

如图 6-11 的连接操作可以通过如下的 select 语句实现：

```
select course.课程号, course.课程名, course.学分, taught-by.任课教师 from course, taught-by  
where course.课程号= taught-by.课程号。
```

注意：“course.课程号”中的句点是面向对象程序设计的符号，句点把对象与属性分开。

“course.课程号”表达的涵义是 course 对象中的“课程号”属性，也就是说 course 表的“课程号”字段。

6.6 Access 2007 数据库

Access 2007 是微软公司最新发布的 Office 2007 办公软件的重要组成部分之一，是一款重要的关系数据库产品。Access 是目前最流行的桌面数据库管理系统之一，它以其强大的功能和直

观的操作界面，深受用户的喜爱。本节依次介绍 Access 2007 的界面组成，数据库创建与维护，表的创建与使用，数据表之间联系的创建方法等内容。

6.6.1 数据库创建与维护

1. 数据库的创建

Access 2007 提供创建一个新的空白数据库或通过模板创建数据库等方法。由于用户使用 Access 来创建数据库时，应已完成了数据库的设计，因此，此处仅介绍空数据库的创建方法。创建空数据库的具体操作方法是：


(1) 单击 Windows “开始”按钮或桌面快捷方式启动 Access 2007，将出现如图 6-20 所示的“开始使用 Microsoft Office Access”界面。



图 6-20 “开始使用 Microsoft Office Access”界面

(2) 在“开始使用 Microsoft Office Access”页面上的“新建空白数据库”栏下，单击“空白数据库”图标。

(3) 在“空白数据库”窗格的“文件名”文本框中，输入数据库文件名，单击“创建”按钮就可完成空数据库的创建任务。

当然用户也可以单击“Office 按钮”，使用菜单中的“新建”命令来创建空数据库。注意数据库的扩展名为 .accdb。



2. 数据库的打开

要使用已创建的数据库，首先应打开数据库。

在打开（或创建然后打开）数据库时，Access 会将该数据库的文件名和位置添加到最近使用文档的内部列表中。此列表显示在“开始使用 Microsoft Office Access”页面中，用户可以轻松打开最近使用的数据库。打开最近使用的数据库方法是：

在“开始使用 Microsoft Office Access”页面的“打开最近的数据库”栏下，单击要打开的数据库，Access 将打开数据库。

使用 Microsoft Office Access 按钮打开数据库的方法是：

单击“Office 按钮”，然后单击要打开的数据库（如果该数据库出现在菜单的右窗格中）。或单击“Office 按钮”，然后单击“打开”命令，在出现“打开”对话框时，输入数

数据库文件名，然后单击“打开”按钮，该数据库随即打开。

在 Access 2007 中，打开数据库有 4 种方式，可通过“打开”对话框右下角“打开”按钮的下拉列表进行选择，如图 6-21 所示。不同打开方式的特点是：

- l 打开：此方式允许用户查看数据库与编辑数据库。
- l 只读方式：此方式只允许用户查看数据库，而不能编辑数据库。
- l 独占方式：网络数据库的访问方式，此方式禁止网络上其他用户再打开该数据库。
- l 独占只读方式：网络数据库的访问方式，具有只读和独占两种方式的特征，即只能查看不能编辑数据库，且不允许其他用户再打开数据库。

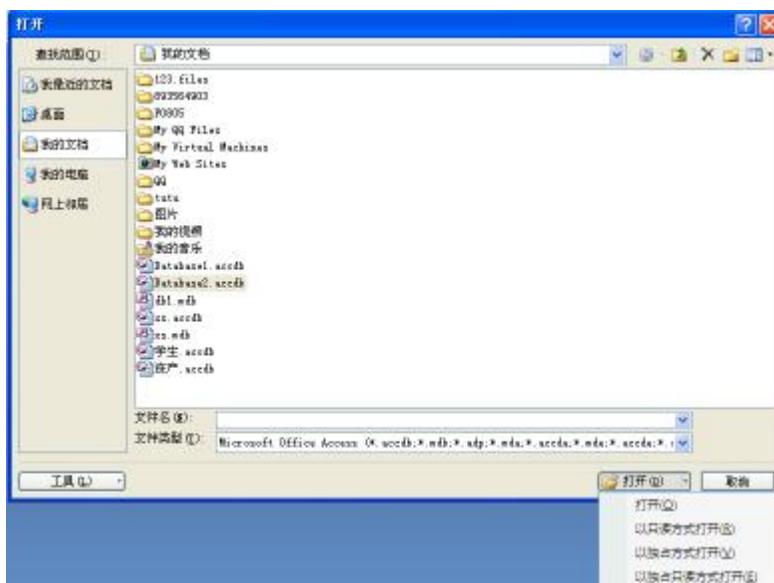



图 6-21 数据库的“打开”对话框

3. 数据库的关闭

如果退出 Access 数据库管理系统，数据库会自动关闭。关闭数据库还可单击主窗口的“关闭”按钮，或单击“Office 按钮”, 执行“关闭数据库”命令。

6.6.2 数据表的创建



1. 表的创建

表是数据库最基本的对象。建立了空的数据库后，就可向数据库中添加表对象。在 Access 2007 中，创建表的具体方法是：

(1) 选择功能区中“创建”选项卡，单击“表设计器”命令，系统弹出设计器界面，如图 6-22 所示。

(2) 在“字段名称”文本框中输入字段名，在“字段类型”下拉列表框中选择适当的数据类型。

(3) 定义完全部字段后，设置一个字段为主键。

(4) 单击“Office 按钮”旁的“保存”按钮，这时出现“另存为”对话框，如图 6-23 所示。

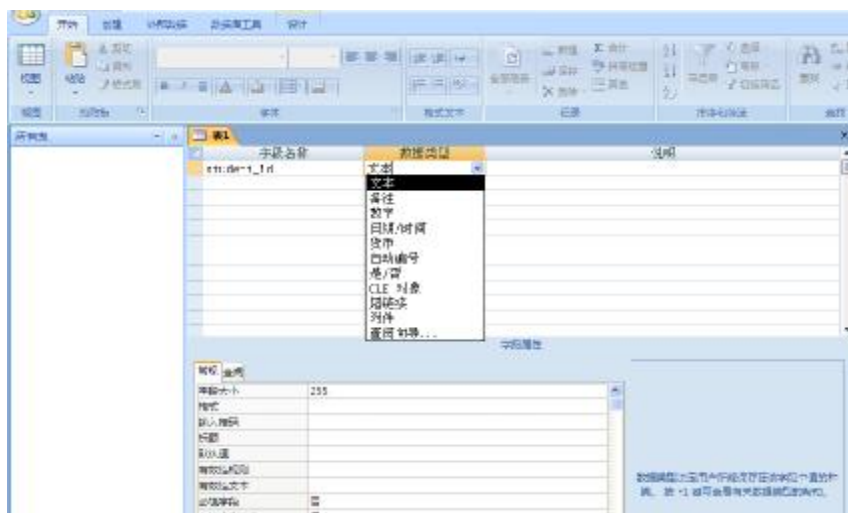


图 6-22 表设计器界面



图 6-23 “另存为”对话框

(5) 在“另存为”对话框中的“表名称”文本框中输入表的名称。

(6) 单击“确定”按钮，完成表的创建，此时在数据库窗口的表对象中会出现新建表的名称。

2. 字段的定义

定义字段是建立表结构的中心环节，字段的定义包括字段名、字段类型与字段属性的定义。在 Access 中，字段名必须以字母开头，后面可以包括字母与数字，长度最多为 64 个字符。
注意：在数据库管理系统中，为了方便查询等操作，一般不用汉字作字段名。

3. 字段数据类型

在表的设计视图中，每一个字段都有数据类型，Access 允许定义 11 种数据类型，分别是文本、备注、数字、日期/时间、货币、自动编号、是/否、OLE 对象、超链接、附件与查询向导。

(1) 文本。

文本类型字段允许最大 255 个字符或数字，Access 默认的大小是 50 个字符，而且系统只保存输入到字段中的字符，不保存文本字段中未用位置上的空字符。“字段大小”用于设置可输入的最大字符长度。

注意：在 Access 中，一个英文字符与一个汉字都是一个字符。

(2) 备注。

备注类型字段用来保存长度较长的文本及数字，它允许字段能够存储长达 64000 个字符的内容。但 Access 不能对备注字段进行排序或索引，却可以对文本字段进行排序和索引。在备注字段中虽然可以搜索文本，但却不如在有索引的文本字段中搜索得快。

（3）数字。

数字字段可以用来存储进行算术运算的数值数据，用户还可以设置“字段大小”属性定义一个特定的数字类型，任何指定为数字数据类型的字段可以设置成“字节”、“整数”、“长整数”、“单精度数”、“双精度数”、“小数”等类型。在 Access 中通常默认为“双精度数”。

（4）日期/时间。

日期/时间类型字段用来存储日期、时间或日期时间数据，每个日期/时间字段需要 8 个字节来存储空间。

（5）货币。

货币数据类型是数字数据类型的特殊形式，等价于具有双精度属性的数字类型。向货币字段输入数据时，不必输入人民币符号和千位处的逗号，Access 会自动显示人民币符号和逗号，并添加两位小数到货币字段。当小数部分多于两位时，Access 会对数据进行四舍五入。精确度为小数点左方 15 位数及右方 4 位数。

（6）自动编号。

这种数据类型较为特殊，它表示向表格添加记录时，Access 会自动插入唯一顺序或者随机编号，即在自动编号字段中指定某一数值。自动编号一旦被指定，就会永久地与记录连接。如果删除了表格中含有自动编号字段的一条记录后，Access 并不会为表格自动编号字段重新编号。当添加某一记录时，Access 不再使用已被删除的自动编号字段的数值，而是重新按递增的规律重新赋值。

（7）是/否。

这种数据类型是针对于某一字段中只包含两个不同的可选值而设立的字段，通过是/否数据类型的格式特性，用户可以对是/否字段进行选择。

（8）OLE 对象。

这种数据类型是指字段允许单独地“链接”或“嵌入”OLE（Object Linking and Embedding 对象连接与嵌入）对象。添加数据到 OLE 对象字段时，可以链接或嵌入 Access 表中的 OLE 对象。OLE 对象是指在其他使用 OLE 协议程序创建的对象，如 Word 文档、Excel 电子表格、图像、声音或其他二进制数据。OLE 对象字段最大可为 1GB，它主要受磁盘空间限制。

（9）超链接。

这种字段主要是用来保存超链接，该字段包含作为超链接地址的文本或以文本形式存储的字符与数字的组合。当单击一个超链接时，Web 浏览器或 Access 将根据超链接地址到达指定的目标。超链接最多可包含三部分：一是在字段或控件中显示的文本；二是到文件或页面的路径；三是在文件或页面中的地址。在这个字段或控件中插入超链接地址最简单的方法就是在“插入”菜单中单击“超链接”命令。

（10）附件。

附件表示将文件附加到在 Office Access 2007 中创建的并且使用的 .accdb 文件格式的数据库中。要存储附件必须在表中创建一个字段，且必须将该字段设置为附件数据类型。Access 2007 的附件数据类型可以将多个文件存储在单个记录中。例如，用户可以存储图像以及用文字处理和电子表格程序创建的文件。附件最多可以附加 2GB 的数据（Access 数据库的最大容量），单个文件的大小不得超过 256MB。将数据类型设置成附件数据类型之后，不能对其进行更改。

(11) 查询向导。

这种字段类型为用户提供了一个建立字段内容的列表，可以在列表中选择所列内容作为添入字段的内容。

4. 字段的属性

在 Access 的表中，每个字段都有一系列的属性描述。字段的属性表示字段所具有的特性，不同的字段类型有不同的属性，当选择某一字段时，表设计视图下部的“字段属性”区就会依次显示出该字段的相应属性，如图 6-22 所示。

(1) 字段大小。

通过“字段大小”属性，可以控制字段使用的空间大小。该属性只适用于数据类型为文本或数字的字段。对于一个文本类型的字段，其字段大小的取值范围是 0~255，默认值为 50，可以在该属性框中输入取值范围内的整数；对于一个数字类型的字段，可以单击“字段大小”属性框，然后单击右侧的下拉箭头按钮，并从下拉列表框中选择一种类型。

(2) 格式。

“格式”属性用来决定数据的打印方式和屏幕显示方式。不同数据类型的字段，其格式选择有所不同，这些类型字段的格式定义，读者可通过练习自己掌握。

文本与备注字段的数据格式一般要定义，其格式最多可分为三个区段，以分号“;”隔开，分别指定字段内的文字、零长度字符串与 Null 值的数据格式。用于创建字符串格式的字符如表 6-11 所示。例如，要使用“电话”字段的数据显示方式为：(0731)-8100988，则需要在“格式”属性框中输入(@@@@)-@@@@@。

表 6-11 控制字符串格式的字符

符号	使用说明
@	输入字符只能为文本或空格
&	不需要使用文本字符
<	输入的所有字母为小写
>	输入的所有字母为大写
!	强制由左向右填充占位符

(3) 默认值。

“默认值”是一个十分有用的属性。在一个数据库中，往往会有一些字段的数据内容相同或含有相同的部分。例如，性别字段只有“男”和“女”两种，这种情况就可以设置一个默认值为“男”OR“女”。

(4) 有效性规则。

“有效性规则”是 Access 中另一个非常实用的属性，利用该属性可以防止非法数据输入到表中。有效性规则的形式及设置目的随字段的数据类型不同而不同。对文本类型字段，可以设置输入的字符个数不能超过某一个值；对数字类型字段，可以让 Access 只接受一定范围内的数据；对日期/时间类型的字段，可以将数值限制在一定的月份或年份以内。有效性规则由运算符和比较符构成逻辑表达式表示。常见运算符表达式如表 6-12 所示。

表 6-12 常见运算符表达式

运算符	含义	使用举例
<	小于	<100
>	大于	>0 and <100
>=	大于或等于	<=100
<=	小于或等于	>=60 and <=100
=	等于	=100
<>	不等于	<>59
In(a1,a2,...,an)	属于括号内的某值	In(20,30,40,50)
Between a1 and a2	在 a1 与 a2 之间	Between 60 and 1000
like	模式字符串匹配	Like C*,like a??t

注意：*与? 为通配符，*代表任意字符串，? 代替任一字符。

(5) 输入掩码。

在输入数据时，如果希望输入的格式标准保持一致，或希望系统检查输入时的错误，可以使用 Access 提供的“输入掩码向导”来设置一个输入掩码，对于大多数数据类型，都可以定义一个输入掩码。用于定义掩码的字符如表 6-13 所示。例如：在输入“电话”字段数据时，要以 (0731)-8100988 格式输入数据，则可在“输入掩码”属性框中输入：(999)-0000000；要以 ABB-688 的形式输入车牌号数据，则在“输入掩码”属性框中输入：AAA-000。

表 6-13 输入掩码字符表

字符	使用说明
0	数字 0~9，必须，不允许加号和减号
9	数字 0~9，可选，不允许加号和减号
#	数字或空格，可选，不允许加号和减号
L	字母 A~Z，必须
?	字母 A~Z，可选
A	字母或数字，必须
&	任何字符或一个空格，必须
C	任何字符或一个空格，可选
::-/	小数点占位符及千位、日期与时间的分隔符

(6) 主键。

关系数据库系统（如 Microsoft Access）的强大功能来自于其可以使用查询（查询：有关表中所有数据的问题，或要对数据执行操作的请求。查询可以将多个表中的数据放在一起，以作为窗体、报表或数据访问页的数据源。）、窗体（Access 数据库对象之一，可以在这种对象上放置控件，用于执行操作，或在字段中输入、显示、编辑数据）和报表（Access 数据库对象的一种，用于按照用户的规格打印设定格式且组织好的信息。例如销售额汇总、电话列表以及邮件标签）快速地查找并组合存储在各个不同表中的信息。为了做到这一点，每个表

都应该包含主键。指定了表的主键之后，Access 将阻止在主键字段中输入重复值或 Null 值（Null：可以在字段中输入或用于表达式和查询，以标明丢失或未知的数据。有些字段（如主键字段）不可以包含 Null 值）为数据表设定主键既可加快数据的查询速度，也是建立表之间关系的要求。

设置主键的方法非常简单，在表设计视图中，首先选中要设为主键的字段或字段组（单击行选定器，可选定一个字段；先按 Ctrl 键，再依次单击行选定器，可选择多个字段），然后，单击数据库工具栏中的图标（也可通过在选定的行上右击，在弹出的菜单中单击“主键”命令），即可将选定的字段设为主键。此时，字段行的选定器上出现钥匙符号。

（7）索引。

在记录数据较多的表中查找、排序数据时，利用索引可以极大地加快操作速度，如果经常需要在某个字段进行查找、排序，建议对该字段设置“索引”。

“索引”值默认为“无”，表示不建立索引，“索引”值设定为“有（有重复）”，表示对该字段建立索引，并且允许字段的值重复；“索引”值设定为“有（无重复）”，表示对该字段建立唯一索引，并且不允许字段的值重复。

注意：设置一个表的主键后，Access 系统会自动在该主键上创建无重复的唯一索引，也称之为索引。因此，对主键不应重复设置索引。另外，备注、OLE 对象或超链接类型的字段也不能设置索引。

（8）标题。

默认情况下，以窗体、报表和查询的形式显示此字段的标签文本。如果此属性为空，则会使用字段的名称。允许使用任何的文本字符串。表 6-14 列出了字段属性设定的作用。

表 6-14 字段的属性及该属性的作用


字段属性	属性的作用
字段大小	设置存储为文本、数字或自动编号数据类型的数据的最大宽度
格式	自定义显示或打印时字段的显示方式
小数位数	指定显示数字时使用的小数位数
新值	设置自动编号数据类型字段是递增的还是为其指定随机值
输入掩码	显示指导数据输入的编辑字符
标题	设置默认情况下在表单、报表和查询的标签中显示的文本
默认值	添加新记录时为字段自动指定默认值
有效性规则	提供在此字段中添加或更改值时必须为真的表达式
有效性文本	输入当值与有效性规则表达式冲突时显示的文本
必填	要求在字段中输入数据
允许零长度字符串	允许在文本或备注类型字段中输入（通过设置为“是”）零长度字符串（" "）
索引	通过创建和使用索引来加速对此字段中数据的访问
Unicode 压缩	存储大量文本（大于 4096 个字符）时压缩此字段中存储的文本
输入法模式	控制 Windows 亚洲语言版本中的字符
IME 语句模式	控制 Windows 亚洲语言版本中的字符转换

续表

字段属性	属性的作用
智能标记	对此字段附加智能标记
仅追加	允许（通过设置为“是”）对备注类型字段执行版本控制
文本格式	选择“格式文本”，将按 HTML 格式存储文本，并允许设置多种格式。选择“纯文本”将只存储文本
文本对齐	指定控件中文本的默认对齐方式
精度	指定允许的数字总位数，包括小数点左右两侧的位数
数值范围	指定可在小数分隔符右侧存储的最大位数

6.6.3 数据表的使用

1. 数据表的打开与关闭

在 Access 2007 中，打开数据库后，在左边的导航窗口中有数据库的表对象，直接在导航窗口中双击表名，就可以打开表。当然也可以单击“Office 按钮”，在弹出的菜单中单击“打开”命令来打开表。

表的关闭方法更为简单，只需单击表编辑窗口中的“关闭”按钮就可以。

2. 数据表的编辑

在实际应用中，数据表结构建立完成以后，有时要对数据表进行编辑，主要包括表结构的修改，在表中追加记录、修改与删除数据等操作。

（1）修改字段定义。

修改字段主要包括字段名、字段数据类型与字段属性的修改。修改过程与定义过程完全一致，可参照字段定义的内容进行，在此不再赘述。

（2）添加字段。

添加字段包括在数据表结构中插入新字段和在表结构末尾追加新字段。插入字段的过程是：选择要插入字段的后一个字段，右击，在弹出快捷菜单中单击“插入行”命令，然后，定义插入的字段。

追加字段的过程与表结构定义过程一致。

（3）字段删除。

要想删除表中的某个字段，可以先将鼠标移动到这个字段的标题处，这时鼠标变成向下的箭头，右击这个字段，整个字段都变成黑色并弹出一个菜单，单击菜单上的“删除列”命令，这时屏幕上弹出“是否确定要删除这个字段及其中的数据”的对话框。单击“是”按钮可以将字段删除。如果是在表设计视图中，选择字段，右击，弹出快捷菜单，单击“删除行”命令就可删除字段。

注意：在删除一个字段的同时也会将这个字段中的数值全部删除，所以当执行这个操作时，一定要小心，以免造成由于误删使有用的数据丢失。

3. 输入数据

数据表结构定义好后，就可以在数据表视图中输入数据，但在输入数据之前先要打开数据表。

(1) 输入文本、数字与货币型数据。

这三种类型数据输入过程较简单，直接在单元格中输入数据即可。输入的数据要受到“有效性规则”等字段属性的限制。

(2) 输入是/否型数据。

如果字段的数据类型为是/否型，则在该字段中输入数据时会出现一个复选框。选中表示输入“是(-1)”，否则表示输入“否(0)”。应用查阅向导功能可直观地显示选择的结果，但存入数据库中仍是-1或0。

(3) 输入日期/时间型数据。

输入该类型数据时，要注意格式，如 2009-02-26 等，不能输入不存在格式的数据。

(4) 输入 OLE 对象型数据。

将插入点移到要输入数据的单元格，右击，弹出一个快捷菜单，在菜单中单击“插入对象”命令，这时出现“插入对象”窗口，在窗口中选择要插入的对象的类型或要插入的对象的文件名，如果要插入图片就保持“新建”单选框被选，并在“对象类型”列表框中选择一种图片格式；如果要插入声音就在“对象类型”列表框中选择“声音”；如果要插入影像就在列表框中选择“影像剪辑”，然后单击“确定”按钮。如果要在这个值中加入一个特定的图片、声音或影像文件，就需要在“插入对象”窗口中选择“由文件创建”单选框，并通过单击“浏览”按钮，在浏览窗口中找到需要放置的文件，双击将它选定，单击“确定”按钮退出“插入对象”窗口，这时表的相应位置就出现了要插入的文件的文件名，双击文件名，就可以看到或听到相应的文件了。

(5) 输入超链接数据。

在表的数据表视图中单击要建立超链接的单元格，再右击，在弹出的菜单中单击“超链接”→“编辑超链接”命令。这时弹出“插入超链接”对话框，在“请输入文件名称和 Web 页名称”下面的文本框中输入想要链接的文件名称和 Web 页的名称。单击“确定”按钮就可以在表中建立一个超链接了。这时单击超链接的名称，系统就会调出 Internet 浏览器访问 Web 页面了。

(6) 查阅向导的应用。

在 Access 中，表字段的值可以直接输入，也可以通过从一组固定数据或其他表的字段中选择来实现数据的输入。在 Access 中，数据的选择输入，可以通过字段的查询功能来实现。

6.6.4 建立数据表的关系

1. 表之间关系的概念

Access 与其他数据库一样，每个表都是数据库中一个独立的部分，但是每个表又不是完全孤立的部分，表与表之间可能存在着相互的联系。表之间有 3 种关系，分别为：一对一关系、一对多关系、多对多关系。这些概念在前面已作介绍，请参照前面相关的内容。

2. 参照完整性

参照完整性是一个规则系统，能确保相关表行之间关系的有效性，并且确保不会在无意之中删除或更改相关数据。

当实施参照完整性时，必须遵守以下规则：

(1) 如果在相关表的主键中没有某个值，则不能在相关表的外部键列中输入该值。但是，

可以在外部键列中输入一个 Null 值；

(2) 如果某行在相关表中存在相匹配的行，则不能从一个主键表中删除该行；

(3) 如果主键表的行具有相关性，则不能更改主键表中的某个键的值。

当符合下列所有条件时，才可以设置参照完整性：

(1) 主表中的匹配列是一个主键或者具有唯一约束；

(2) 相关列具有相同的数据类型和大小；

(3) 两个表属于相同的数据库。

3. 表之间的关系

当想让两个表共享数据时，可以创建两个表之间的关系。在数据库中为每个主题创建表后，必须为 Office Access 2007 提供在需要时将这些信息重新组合到一起的方法。具体方法是在相关的表中放置公共字段，并在表之间定义表关系。然后，可以创建查询、窗体和报表，以同时显示几个表中的信息。

(1) 创建表的关系。

创建表关系可以在“关系”窗口中进行。在表之间创建关系时，不要求公共字段具有相同的名称，但实际情况往往是这样。不过，公共字段必须具有相同的数据类型。但是，如果主键字段为自动编号数据类型字段，并且主键字段与外键字段的 FieldSize 属性相同，则外键字段也可以为数字字段。例如，如果自动编号字段与数字字段的 FieldSize 属性都是“长整型”，则可以将这两个字段相匹配。即使两个公共字段都是数字字段，它们也必须具有相同的 FieldSize 属性设置。具体操作过程如下：

1) 单击“Microsoft Office 按钮”，然后单击“打开”命令。在“打开”对话框中，选择并打开数据库。

2) 在“数据库工具”选项卡上的“显示/隐藏”命令组中，单击“关系”命令。如图 6-24 所示。

3) 如果用户尚未定义过任何关系，则会显示“显示表”对话框。如果未出现该对话框，可在“设计”选项卡上的“关系”命令组中单击“显示表”命令，如图 6-25 所示。



图 6-24 “数据库工具”选项卡上的“显示/隐藏”命令组

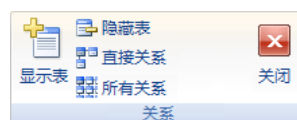


图 6-25 “设计”选项卡上的“关系”命令组

“显示表”对话框会显示数据库中的所有表和查询。如果只查看表，可单击“表”按钮。若只查看查询，可单击“查询”按钮。若要同时查看表和查询，单击“两者”按钮。

4) 选择一个或多个表或查询，然后单击“添加”按钮。将表和查询添加到“关系”文档选项卡之后，单击“关闭”按钮。

5) 将字段（通常为键）从一个表拖至另一个表中的公共字段（外键）。要拖动多个字段，可按 Ctrl 键，单击每个字段，然后拖动这些字段。

6) 此时系统将显示如图 6-26 所示“编辑关系”对话框。

验证显示的字段名称是否是关系的公共字段。如果字段名称不正确，可单击该字段名称并从列表中选择合适的字段。

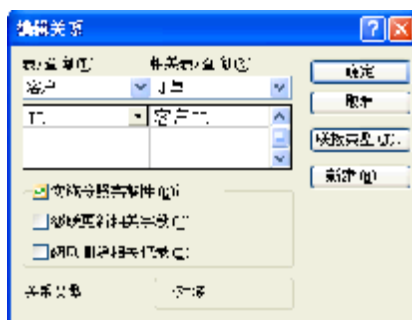


图 6-26 “编辑关系”对话框

要对此关系实施参照完整性，可选中“实施参照完整性”复选框。有关参照完整性的详细信息，已在前面作了介绍，请参照相关内容。

7) 单击“确定”按钮，Access 会在两个表之间绘制一条关系线。如果已选中“实施参照完整性”复选框，则该线在每端都显示为较粗。此外，仅当再次选中“实施参照完整性”复选框后，数字 1 才会出现在关系线一侧较粗的部分之上，无穷大符号 (∞) 将出现在该线另一侧较粗的部分之上，如图 6-27 所示。

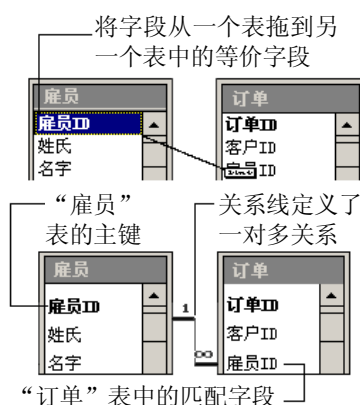


图 6-27 表之间的关系

(2) 编辑表关系。

更改表关系的方法是在“关系”文档选项卡中选择表关系，然后对其进行编辑。编辑表关系的方法是：

移动鼠标指针指向关系线，双击该线。或在“设计”选项卡上的“工具”命令组中，单击“编辑关系”命令。系统弹出“编辑关系”对话框，如图 6-26 所示。此时可以对关系进行编辑了。

本章小结

- I 信息就是客观事物属性的反映，数据是对信息的符号描述。计算机中的数据一般分临时性数据与永久性数据，数据库中的数据就是永久性数据之一。计算机数据处理是利用计算机对各种类型的数据进行收集、存储、分类、计算、加工、检索及传输的过程。

- I 数据库是长期存放在计算机内，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，具有较小的冗余度、较高的数据独立性和易扩展性，并可为不同的用户共享。数据库管理系统是指对数据库进行管理的软件系统，它是用户与数据库的接口，有非常明确的功能与组成。数据库系统是由计算机硬件、操作系统、数据库管理系统，以及在它支持下建立起来的数据库、应用程序、用户和数据库管理员组成的一个整体。
- I 在数据库设计发展过程中，曾使用过层次模型、网状模型和关系模型三种。层次模型用一棵倒置的树来描述数据之间的关系，树的结点表示实体集（多条记录），结点之间的连线表示相连两实体集之间的联系，这种联系只能是“1—M（多）”。网状模型与层次结构不同，网状模型的结点间可以任意发生联系，可表示各种复杂的关系。网状模型的优点是可以更直接地去描述现实世界，可以避免数据的重复性，缺点是关联性比较复杂。关系数据模型用二维表格结构来表示实体以及实体间联系的数据模型在关系模型中，实体和实体间的联系都用关系表示，它是目前应用最广、理论最成熟的一种数据模型。
- I 一个关系就是一个二维表，每个关系有一个关系名。二维表中的列称为属性（也称字段），表中的每一列在关系范围内唯一。在数据库管理系统中，每个字段须定义名称、数据类型与数据宽度等属性。表中的行称为元组，元组也称为记录。
- I 关系数据库管理系统定义了一些操作来通过已知的关系创建出新的关系，这些操作主要有插入、删除、更新、选择、投影、连接操作。这些操作可以通过命令来实现，也可以通过 SQL 语句来完成。
- I 一般来说，数据库的设计都要经历需求分析、概念设计、实现设计和物理设计 4 个阶段。需求分析的目的是分析系统的需求。该过程的主要任务是从数据库的所有用户那里收集对数据的需求和对数据处理的要求，并把这些需求写成用户和设计人员都能接受的说明书。
- I 概念设计是整个数据库设计的关键。它的目的是将需求说明书中关于数据的需求，综合为一个统一的 DBMS 概念模型。设计过程是：首先根据单个应用的需求，画出能反映每一应用需求的局部 E-R 模型，然后将这些 E-R 模型图合并起来，消除冗余和可能存在的矛盾，得出系统总体的 E-R 模型。
- I 实现设计的目的是将 E-R 模型转换为某一特定的 DBMS 能够接受的逻辑模式，也就是说，把 E-R 图中的实体与实体之间的联系用关系来描述。对关系数据库，主要是完成表的关联和结构的设计。
- I 物理设计的目的在于确定数据库的存储结构。其主要任务包括：确定数据库文件和索引文件的记录格式和物理结构，选择存取方法，决定访问路径和外存储器的分配策略等。
- I 在逻辑设计阶段，常常使用关系规范化理论来指导关系数据库设计。其基本思想为：每个关系都应该满足一定的规范，从而使关系模式设计合理，达到减少冗余，提高查询效率的目的。规范被划分为若干等级，每一级称为一个“范式”。在关系数据库规范中已建立了一个范式系列：1NF、2NF、3NF、BCNF、4NF 和 5NF，这 6 种范式一级比一级有更严格的要求。一般数据库的设计至少要符合第三范式。

- I 结构化查询语言是为关系数据库管理系统开发的一种查询语言，由数据定义语言、数据操纵语言、数据查询语言与数据控制语言 4 部分组成。现代的数据库管理系统都支持该语言。
- I Access 2007 是微软公司最新发布的 Office 2007 办公软件的重要组成部分之一，是一款重要的关系数据库产品。我们必须做到能使用 Access 2007 创建与维护数据库，创建与使用表，创建数据表之间的关系。

习题六

一、选择题

1. 数据 (Data)、数据库 (DB)、数据库管理系统 (DBMS) 与数据库系统 (DBS) 之间是一种包含关系，下面 () 能正确描述这种包含关系。
A. DBMS\DBS\DB\Data B. DBS\DBMS\DB\Data
C. Data\DBMS\DBS\DB D. DBMS\Data\DB\DBS
2. 下列第 () 项，对数据库特征的描述是错误的。
A. 数据具有独立性 B. 可共享
C. 消除了冗余 D. 数据集中控制
3. 由于数据库是为多用户共享，因此，就需要特殊的用户对数据库进行规划、设计、协调、维护和管理。这个特殊用户被称为 ()。
A. 用户 B. 程序员 C. 工程师 D. 数据库管理员
4. 按使用的数据模型来分，数据库可分为 () 三种模型。
A. 网状、环状和链状 B. 层次、关系和网状
C. 大型、中型和小型 D. 独享、共享和分时
5. 二维表的一行对应 ()，二维表的一列对应 ()。
A. 字段 B. 记录 C. 关系 D. 主键
6. 关系数据模型 ()。
A. 只能表示实体间 1:1 联系 B. 只能表示实体间 $n:m$ 联系
C. 只能表示实体间 1: n 联系 D. 可以表示实体间的上述三种联系
7. 对于二维表的关键字来讲，不一定存在的是 ()。
A. 主关键字 B. 外部关键字 C. 超关键字 D. 候选关键字
8. 在数据库中，数据模型是 () 的集合。
A. 文件 B. 记录 C. 数据 D. 记录及其联系
9. 关系数据库管理系统管理的系统是 ()。
A. 一个二维表 B. 若干个二维表
C. 一个数据库 D. 若干个 DBC 文件
10. 对表进行垂直方向的分割用的运算是 ()。
A. 连接 B. 选择 C. 交 D. 投影
11. 一般数据库的设计至少要符合 ()。

- A. 第一范式 B. 第二范式 C. 第三范式 D. 第四范式

二、简答题

1. 在为学校的学生成绩管理系统设计数据库时，通过对学校学生成绩进行调查确定学生成绩管理系统中的实体有学生、班级、课程、专业与系等。在该系统中对这些实体要求的属性分别是：

班级：班级编号，班级名称。

课程：课程编号，课程名称，周学时，课程性质（考试/考查），类型（公共基础/专业基础/专业课/公共选修/专业选修）。

学生：学号，姓名，性别，出生日期，联系方式。

专业：专业编号，专业名称。

系：系编号，系名称，系办地址。

请画出学生与课程的 E-R 图，然后对该 E-R 图进行数据库的逻辑设计。

2. 请指出下列二维表违背了哪些范式设计，请阐述理由。

学号	姓名	性别	成绩	
			语文	数学
101	bert	男	90	84
.....

表 A

学号	姓名	性别	课程号	课程名
2008001	刘林	女	07	英语
2008005	王海波	男	06	动漫设计
2008001	刘林	女	01	语文
2008126	易杨	女	03	机械系

表 B

3. 请写出下列 SQL 语句（关系如图 6-4 所示）。

- (1) 利用 SQL 语句生成一个仅含有学生 ID 和 Name 的关系。
- (2) 利用 SQL 语句生成一个包含 ID=2010 的学生所上课程的关系。
- (3) 利用 SQL 语句生成一个包含 Blake 教授所教课程的关系。
- (4) 利用 SQL 语句生成一个包含 Units=3 的课程的关系。

4. 请为实际的图书管理系统设计一个关系数据库。

5. 请在 Access 2007 中创建如图 6-5 所示的数据库。

6. 何为 Data、DB、DBMS 与 DBS？它们之间有什么关系？

7. 在数据库设计发展过程中，曾使用过哪三种模型？每种模型有何特点？

8. 关系的基本特点有哪些？关系有哪些基本操作？

9. 数据库的设计包括哪四个阶段，每个阶段产生的主要文档是什么？

10. 什么是 E-R 图？

11. 什么是范式？请简述第一范式、第二范式与第三范式的要求。

12. SQL 的全称是什么？这种语言的主要功能是什么？