# 第 2 章

# 数据库系统结构

本章主要介绍数据模型和数据库管理系统的体系结构，包括概念层数据模型和关系数据库采用的组织层数据模型，概念层数据模型即表示现实世界形象的模型，组织层数据模型是为方便计算机处理数据采用的逻辑模型。关系数据库将数据划分为不同的层次，使用户不需要关心数据的物理细节，从而简化用户对数据的访问，这些就是数据库系统结构要介绍的内容。理解本章的内容有助于读者后续章节的学习。本章的内容可能有些抽象和枯燥，但在学习完后续章节的内容后，再回顾这部分内容，就会有更好的理解。

## 2.1 数据和数据模型

数据是我们要处理的信息，数据模型是数据的组织方式。本节介绍数据和数据模型的基本概念。

### 2.1.1 数据与信息

为了了解世界、研究世界和交流信息，人们需要描述各种事物。用自然语言来描述虽然很直接，但过于繁琐，不便于形式化，而且也不利于用计算机来表达。为此，人们常常只抽取那些感兴趣的事物特征或属性来描述事物。例如，一名学生可以用信息“(张三，9912101，男，1981，计算机系，应用软件)”描述，这样的一行数据称为一条记录。单看这行数据我们很难知道其确切含义，但对其进行如下解释：张三是 9912101 班的男学生，1981 年出生，计算机系应用软件专业，其内容就是有意义的。我们将描述事物的符号记录称为数据，将从数据中获得的有意义的内容称为信息。数据有一定的格式，例如，姓名一般是长度不超过 4 个汉字的字符（假设不包括少数民族的姓名），性别是一个汉字的字符。这些格式的规定是数据的语法，而数据的含义是数据的语义。因此，数据是信息存在的一种形式，只有通过解释或处理才能成为有用的信息。

一般来说，数据库中的数据具有静态和动态两种特征：

1）静态特征。数据的静态特征包括数据的基本结构、数据间的联系以及对数据取值范围的约束。比如 1.2.1 节中给出的学生管理的例子。学生基本信息包含学号、姓名、性别、

出生日期、联系电话、所在系、专业、班号，这些都是学生所具有的基本性质，是学生数据的基本结构。学生选课信息包括学号、课程号和考试成绩等，这些是学生选课的基本性质。但学生选课信息中的学号与学生基本信息中的学号是有一定关联的，即学生选课信息中的“学号”能取的值必须在学生基本信息中的“学号”取值范围之内，因为只有这样，学生选课信息中所描述的学生选课情况才是有意义的（我们不会记录不存在的学生的选课情况），这就是数据之间的联系。最后我们看数据取值范围的约束。我们知道人的性别一项的取值只能是“男”或“女”、课程的学分一般是大于 0 的整数值、学生的考试成绩一般在 0 ～ 100 分之间等，这些都是对某个列的数据取值范围进行的限制，目的是在数据库中存储正确的、有意义的数据。

2）动态特征。数据的动态特征是指对数据可以进行符合一定规则的操作。对数据库数据的操作主要有查询数据和更改数据，更改数据一般又包括对数据的插入、删除和更新。

一般将对数据的静态特征和动态特征的描述称为数据模型三要素，即在描述数据时要包括数据的基本结构、数据的约束条件（这两个属于静态特征）和定义在数据上的操作（属于数据的动态特征）三个方面。

### 2.1.2　数据模型

对于模型，特别是具体的模型，人们并不陌生。一张地图、一组建筑设计沙盘、一架航模飞机等都是具体的模型。人们从模型可以联想到现实生活中的事物。模型是对事物、对象、过程等客观系统中感兴趣的内容的模拟和抽象表达，是理解系统的思维工具。数据模型（data model）也是一种模型，它是对现实世界数据特征的抽象。

数据库是企业或部门相关数据的集合，数据库不仅要反映数据本身的内容，而且要反映数据之间的联系。由于计算机不可能直接处理现实世界中的具体事物，因此，必须要把现实世界中的具体事物转换成计算机能够处理的对象。在数据库中用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息。通俗地讲，数据模型就是对现实世界数据的模拟。

现有的数据库系统均是基于某种数据模型的，因此，了解数据模型的基本概念是学习数据库的基础。

数据模型一般应满足三个要求：第一个是数据模型要能够比较真实地模拟现实世界；第二个是数据模型要容易被人们理解；第三个是数据模型要能够很方便地在计算机上实现。用一种模型来同时很好地满足这三方面的要求在目前是比较困难的。在数据库系统中可以针对不同的使用对象和应用目的，采用不同的数据模型来实现。

数据模型实际上是模型化数据和信息的工具。根据模型应用的不同目的，可以将这些模型分为两大类，它们分别属于两个不同的层次。

第一类是概念层数据模型，也称为概念模型或信息模型，它从数据的应用语义视角来抽取模型并按用户的观点来对数据和信息进行建模。这类模型主要用在数据库的设计阶段，它与具体的数据库管理系统无关。另一类是组织层数据模型，也称为组织模型，它从数据的组织方式来描述数据。所谓组织层就是指用什么样的数据结构来组织数据。数据库发展到现在主要包括如下几种组织方式（或叫组织模型）：层次模型（用树形结构组织数据）、网状模型（用图形结构组织数据）、关系模型（用简单二维表结构组织数据）以及对

象－关系模型（用复杂的表格以及其他结构组织数据）。组织层的数据模型主要是从计算机系统的观点对数据进行建模，它与所使用的数据库管理系统的种类有关，主要用于 DBMS 的实现。

为了把现实世界中的具体事物抽象、组织为某一具体 DBMS 支持的数据模型，人们通常首先将现实世界抽象为信息世界，然后再将信息世界转换为机器世界。即：首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统，而且也不与具体的 DBMS 相关，而是概念级的模型，也就是我们前面所说的概念层数据模型；然后再把概念级的模型转换为具体的 DBMS 支持的数据模型，也就是组织层数据模型。注意从现实世界到概念层数据模型使用的是“抽象”技术，从概念层数据模型到组织层数据模型使用的是“转换”，也就是说先有概念模型，然后再有组织模型。从概念模型到组织模型的转换应该是比较直接和简单的，因此使用合适的概念层模型就显得比较重要。这个过程如图 2-1 所示。

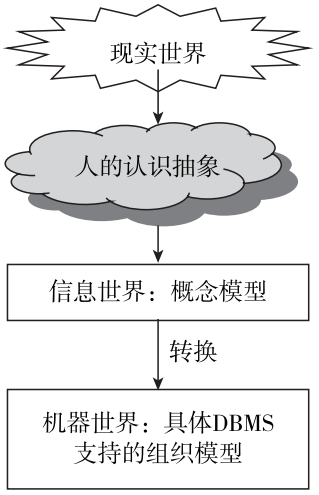


图 2-1 从现实世界到机器世界的过程

## 2.2 概念层数据模型

从图 2-1 可以看出，概念层数据模型实际上是现实世界到机器世界的一个中间层次。本节介绍概念层数据模型的基本概念及构建方法。

### 2.2.1 基本概念

概念层数据模型：抽象现实系统中有应用价值的元素及其关联关系，反映现实系统中有应用价值的信息结构，并且不依赖于数据的组织层数据模型。

概念层数据模型用于对信息世界进行建模，是现实世界到信息世界的第一层抽象，是数据库设计人员进行数据库设计的工具，也是数据库设计人员和用户之间进行交流的工具，因此，该模型一方面应该具有较强的语义表达能力，能够方便、直接地表达应用中的各种语义知识；另一方面它还应该简单、清晰和易于被用户理解。

概念层数据模型是面向用户、面向现实世界的数据模型，它与具体的 DBMS 无关。采用概念层数据模型，设计人员可以在设计的开始把主要精力放在了解现实世界上，而把涉及 DBMS 的一些技术性问题推迟到后面再考虑。

常用的概念层数据模型有实体－联系（Entity-Relationship，E-R）模型、语义对象模型。我们这里只介绍实体－联系模型。

### 2.2.2 实体－联系模型

如果直接将现实世界的信息按具体数据模型的要求进行组织，必须同时考虑很多因素，设计工作非常复杂，并且效果也不一定理想，因此需要一种方法来对现实世界的信息结构进行描述。事实上在这方面已经有了一些方法，我们要介绍的是 P. P. S. Chen 于 1976 年提出的实体－联系（Entity-Relationship）方法，即通常所说的 E-R 方法。这种方法由于简单、实用，因此得到了广泛的应用，也是目前描述信息结构最常用的方法。

E-R 方法使用的工具称为 E-R 图，它所描述的现实世界的信息结构称为企业模式（Enterprise Schema），也把这种描述结果称为 E-R 模型。

实体 – 联系方法试图定义许多数据分类对象，然后数据库设计人员就可以将数据项归类到已知的类别中。第 8 章将介绍如何将 E-R 模型转换为组织层数据模型。下面介绍几个基本概念：

### 1. 实体

实体是具有公共性质并可相互区分的现实世界对象的集合。实体是具体的，例如：职工、学生、教师、课程都是实体。

在 E-R 图中用矩形框表示具体的实体，把实体名写在框内。如图 2-2a 中的“经理”和“部门”实体。

实体中的每个具体的记录值（一行数据），比如学生实体中的每个具体的学生，称为实体的一个实例。

**注意**

有些书也将实体称为实体集或实体类型，而将每行具体的记录称为实体。

### 2. 属性

每个实体都具有一定的特征或性质，这样才能根据实体的特征来区分一个个实例。属性就是描述实体或者联系的性质或特征的数据项，属于一个实体的所有实例都具有相同的性质，在 E-R 模型中，这些性质或特征就是属性。

比如学生的学号、姓名、性别等都是学生实体具有的特征，这些特征就构成了学生实例的属性。实体所具有的属性的多少是由用户对信息的需求决定的。例如，假设用户还需要学生的民族信息，则可以在学生实例中加一个“民族”属性。

属性在 E-R 图中用圆角矩形表示，在矩形框内写上属性的名字，并用连线将属性框与它所描述的实体联系起来。如图 2-2c 所示。

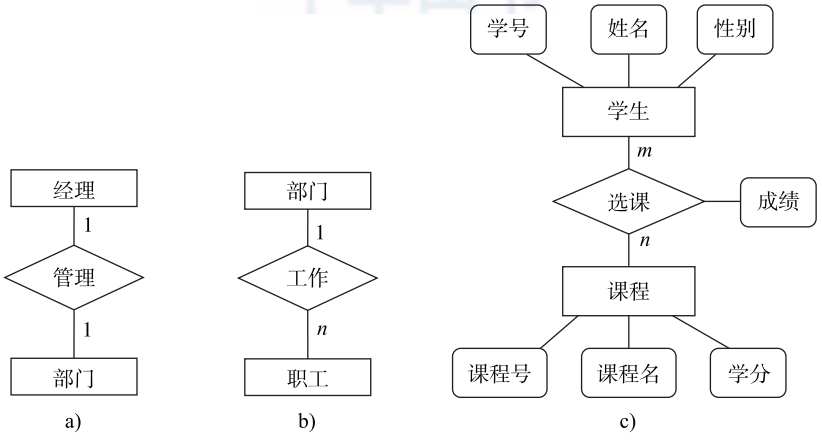


图 2-2　联系的示例

### 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一个实体内部各属性之间的

联系，实体之间的联系通常是指不同实体之间的联系。比如在职工实体中，假设有职工号、姓名、所在部门和部门经理号等属性，其中部门经理号描述的是管理这个职工的部门经理的职工号。一般来说，部门经理也属于单位的职工，因此，部门经理号和职工号通常采用的是同一套编码方式，因此部门经理号与职工号之间有一种关联的关系，即部门经理号的取值在职工号取值范围内。这就是实体内部的联系。而学生和课程之间的关联关系是通过学生选课体现的，在学生选课中至少会包含学生的学号以及学生所选的课程号，而且学生选课中的学号必须是学生实体中已经存在的学号，因为我们不允许为不存在的学生记录选课情况。同样，学生选课中的课程号也必须是课程实体中存在的课程号。这种关联到两个不同实体的联系就是实体之间的联系。通常情况下的联系都是实体之间的联系。

联系是数据之间的关联集合，是客观存在的应用语义链。在 E-R 图中联系用菱形框表示，框内写上联系名，并用连线将联系框与它所关联的实体连接起来。如图 2-2c 中的“选课”联系。

联系也可以有自己的属性，比如图 2-2c 中“选课”联系有“成绩”属性。

两个实体之间的联系可以分为三类：

（1）一对一联系（1:1）

如果实体 A 中的每个实例在实体 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，部门和经理（假设一个部门只有一个经理，一个人只能担任一个部门的经理）、系和正系主任（假设一个系只有一个主任，一个人只能担任一个系的主任）都是一对一联系。一对一联系如图 2-2a 所示。

（2）一对多联系（1:$n$）

如果实体 A 中的每个实例在实体 B 中有 $n$ 个实例（$n \geqslant 0$）与之联系，而实体 B 中每个实例在实体 A 中最多只有一个实例与之联系，则称实体 A 与实体 B 是一对多联系，记作 1:$n$。

例如，假设一个部门有若干职工，而一个职工只在一个部门工作，则部门和职工之间就是一对多联系。又比如，假设一个系有多名教师，而一个教师只在一个系工作，则系和教师之间也是一对多联系。一对多联系如图 2-2b 所示。

（3）多对多联系（$m$:$n$）

如果对于实体 A 中的每个实例，实体 B 中有 $n$ 个实例（$n \geqslant 0$）与之联系，而对实体 B 中的每个实例，在实体 A 中也有 $m$ 个实例（$m \geqslant 0$）与之联系，则称实体 A 与实体 B 的联系是多对多的，记为 $m$:$n$。

比如学生和课程，一个学生可以选多门课程，一门课程也可以被多个学生选，因此学生和课程之间是多对多的联系，如图 2-2c 所示。

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。实体之间联系的种类与语义直接相关。

例如部门和经理，如果一个部门只有一个经理，一个人只担任一个部门的经理，则部门和经理之间是一对一联系；如果一个部门可以有多个经理，而一个人只担任一个部门的经理，则部门和经理之间就是一对多联系；如果一个部门可以有多个经理，而且一个人也可以担任多个部门的经理，则部门和经理之间就是多对多联系。

E-R 图不仅能描述两个实体之间的联系，而且还能描述两个以上实体之间的联系。比如

有顾客、商品、售货员三个实体，并且有语义：每个顾客可以从多个售货员那里购买商品，并且可以购买多种商品；每个售货员可以向多名顾客销售商品，并且可以销售多种商品；每种商品可以由多个售货员销售，并且可以销售给多名顾客。描述顾客、商品和售货员之间的关联关系的 E-R 图如图 2-3 所示，这里联系被命名为“销售”。

**注意**

如果将顾客、商品和售货员之间的关联关系描述成如图 2-4 所示的形式，则其描述的是售货员、顾客和商品两两之间的联系，因此不符合语义上所要求的三者之间共同有联系的要求。

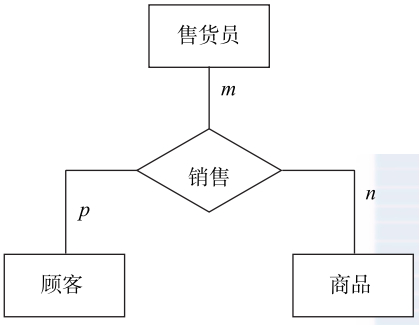


图 2-3　多个实体之间的联系示例

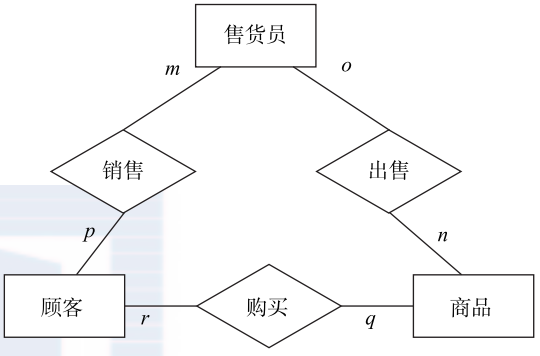


图 2-4　不符合语义要求的联系

## 2.3　组织层数据模型

组织层数据模型是从数据的组织方式的角度来描述信息，目前，在数据库技术的发展过程中用到的组织层数据模型有四种，它们是：层次模型、网状模型、关系模型和面向对象模型。组织层数据模型是按组织数据的逻辑结构来命名的，比如，层次模型采用树形结构。目前使用最普遍的是关系数据模型。关系数据模型技术从 20 世纪 70 ～ 80 年代开始到现在已经发展得非常成熟，因此，我们重点介绍关系数据模型。

关系数据模型（或称为关系模型）是目前最重要的一种数据模型。关系数据库就是采用关系模型作为数据的组织方式。20 世纪 80 年代以来，计算机厂商推出的数据库管理系统几乎都支持关系模型，非关系系统的产品也大都加上了关系接口。下面从数据模型的三要素角度来介绍关系数据模型的特点。

### 2.3.1　关系模型的数据结构

关系数据模型源于数学，它用二维表来组织数据，而这个二维表在关系数据库中称为关系。关系数据库是表（或者说是关系）的集合。

关系数据库要求让用户所感觉的数据库就是一张张表。在关系数据库中，表是逻辑结构而不是物理结构。实际上，系统在物理层可以使用任何有效的存储结构来存储数据，比如，顺序文件、索引、散列表、指针等。因此，表是对物理存储数据的一种抽象表示——对很多存储细节的抽象，如存储记录的位置、记录的顺序、数据值的表示以及记录的访问结构，如索引等，对用户来说都是不可见的。

用关系表示实体以及实体之间联系的模型称为关系数据模型，简称为关系模型。表 2-1 所示的是学生基本信息的关系模型。

表 2-1 学生基本信息

| 学 号 | 姓 名 | 性 别 | 年 龄 | 所在系 |
|---|---|---|---|---|
| 9512101 | 李勇 | 男 | 19 | 计算机系 |
| 9512102 | 刘晨 | 男 | 20 | 计算机系 |
| 9512103 | 王敏 | 女 | 20 | 计算机系 |
| 9521101 | 张立 | 男 | 22 | 信息系 |
| 9521102 | 吴宾 | 女 | 21 | 信息系 |

下面介绍一些关系模型中的基本术语：

### 1. 关系

关系就是二维表，它满足如下条件：

1）关系中的每一列都是不可再分的基本属性。如表 2-2 所示的表就不是关系，因为“出生日期”列不是基本属性，它包含了子属性“年”、“月”、“日”。

2）一个关系中的各属性不能重名。

3）关系中的行、列次序并不重要，即交换列的前后顺序，比如在表 2-1 中，将“性别”放置在“年龄”的后边，不影响其表达的语义。

表 2-2 包含复合属性的表

| 学 号 | 姓 名 | 性 别 | 年 龄 | 所在系 | 出生日期 | | |
|---|---|---|---|---|---|---|---|
| | | | | | 年 | 月 | 日 |
| 9512101 | 李勇 | 男 | 19 | 计算机系 | 1984 | 4 | 6 |
| 9512102 | 刘晨 | 男 | 20 | 计算机系 | 1984 | 12 | 15 |
| 9512103 | 王敏 | 女 | 20 | 计算机系 | 1983 | 8 | 21 |
| 9521101 | 张立 | 男 | 22 | 信息系 | 1983 | 6 | 3 |

### 2. 元组

关系中的每一行数据称为一个元组，它相当于一个记录值。

### 3. 属性

关系中的每一列是一个属性值的集合，列可以命名，称为属性名。例如，表 2-1 中有 5 个属性。属性与前面介绍的实体的属性（特征）或记录的字段意义相当。

因此，关系是元组的集合，如果关系有 $n$ 个列，则称该关系是 $n$ 元关系。关系中的每一列都是不可再分的基本属性，而且关系中的每一行数据应该不允许完全相同，因为存储值完全相同的两行或多行数据并没有实际意义。

因此，在数据库中有两套标准术语，一套用的是表、行、列；而另外一套即关系（对应表）、元组（对应行）和属性（对应列）。

### 4. 主码

主码（primary key）也称为主键或主关键字，是关系中用于唯一确定一个元组的一个属性或最小的属性组。主码可以由一个属性组成，也可以由多个属性共同组成。例如，表 2-1 所示的例子中，学号就是此“学生基本信息”关系的主码，因为它可以唯一地确定一个具体

的学生（一个元组）。而表 2-3 所示关系的主码就由学号和课程号共同组成。因为一个学生可以选修多门课程，而且一门课程也可以有多个学生选修，因此，只有将学号和课程号组合起来才能共同确定一行数据。我们称由多个属性共同组成的主码为复合主码。当某个关系是由多个属性共同作为主码时，就用圆括号将这些属性括起来，表示共同作为主码。比如，表 2-3 的主码是（学号，课程号）。

表 2-3　学生选课信息

| 学　号 | 课程号 | 成　绩 |
|---|---|---|
| 9512101 | c01 | 90 |
| 9512101 | c02 | 86 |
| 9512101 | c06 | |
| 9512102 | c02 | 78 |
| 9512102 | c04 | 66 |
| 9521102 | c01 | 82 |
| 9521102 | c02 | 75 |
| 9521102 | c04 | 92 |
| 9521102 | c05 | 50 |

注意，我们不能根据表在某时刻所存储的数据来决定哪些列是主码，这样做只能是猜测，是不可靠的。关系的主码与其实际的应用语义有关、与关系设计者的意图有关。例如，对于表 2-3，用（学号，课程号）作为主码，在一个学生对一门课程只能有一次考试的前提下是成立的，如果实际允许一个学生对一门课程可以有多次考试，则用（学号，课程号）作为主码就不够了，因为一个学生对一门课程有多少次考试，则其（学号，课程号）的值就会重复多少遍。如果是这种情况，可以为这个关系添加一个“考试次数”属性，同时用（学号，课程号，考试次数）作为主码。

有时一个关系中可能存在多个可以做主码的属性，比如，对于“学生基本信息”，如果能够保证姓名肯定不重复的话，那么“姓名”也可以作为“学生基本信息”关系的主码。如果一个关系中存在多个可以作为主码的属性，则称这些属性为候选码属性，相应的码称为候选码。从候选码中选取哪一个作为主码都可以，因此，主码是从候选码中选取出来做主码的属性。

**5. 域**

属性的取值范围称为域。例如，大学生的年龄假设在 14～40 岁之间，因此“年龄”属性的域就是（14～40），而人的性别只能是“男”和“女”两个值，因此，“性别”属性的域就是（男，女）。

**6. 关系模式**

二维表的结构称为关系模式，或者说，关系模式就是二维表的表框架或表头结构。

关系模式一般表示为：关系名（属性 1，属性 2，…，属性 $n$）

例如，表 2-1 所示关系的关系模式为：

学生（学号，姓名，性别，年龄，所在系）

如果将关系模式理解为数据类型，则关系就是该数据类型的一个具体值。关系模式是对关系的“型”或元组的结构共性的描述。

关系、关系模式、元组以及属性之间的关系如图 2-5 所示。


图 2-5　各概念之间的关系

### 2.3.2 关系模型的数据操作

关系模型的操作对象是集合（也就是关系），而不是单个的行，也就是操作的数据以及操作的结果都是完整的表（是包含行集的表，而不只是单行，当然，只包含一行数据的表是合法的，空表或不包含任何数据行的表也是合法的）。而在非关系型数据库系统中，典型的操作是一次一行或一次一个记录。因此，集合处理能力是关系系统区别于其他系统的一个重要特征。

关系数据模型的数据操作主要包括四种：查询、插入、删除和修改数据。关系数据库中的信息只有一种表示方式，就是表中的行列位置有明确的值。这种表示是关系系统中唯一可行的方式（当然，这里指的是逻辑层）。特别地，关系数据库中没有连接一个表到另一个表的指针。在表 2-1 和表 2-3 中，表 2-1 的“学生基本信息”的第一行数据与表 2-3 的“学生选课信息”中的第一行有联系（当然也与第二行和第三行有联系），因为学生 9512101 选了课程。但在关系数据库中这种联系不是通过指针来实现的，而是通过学生基本信息的“学号”属性中的值与“学生选课”中“学号”属性的值联系的（学号值相等）。但在非关系系统中，这些信息一般由指针来表示，这种指针对用户来说是可见的。

需要注意的是，当我们说关系数据库中没有指针时，并不是指在物理层没有指针，实际上，在关系数据库的物理层也使用指针，但所有这些物理层的存储细节对用户来说都是不可见的，用户所看到的物理层是没有指针的。

### 2.3.3 关系模型的数据完整性约束

数据完整性是指数据库中存储的数据是有意义的或正确的。关系模型中的数据完整性规则是对关系的某种约束条件。它的数据完整性约束主要包括三大类：实体完整性、参照完整性和用户定义的完整性。

#### 1. 实体完整性

实体完整性指的是关系数据库中所有的表都必须有主码，而且表中不允许存在如下的记录：

- 无主码值的记录
- 主码值相同的记录

因为若记录没有主码值，则此记录在表中一定是无意义的。前面介绍过，关系模型中的每一行记录都对应客观存在的一个实例或一个事实。比如，一个学号唯一地确定了一个学生。如果关系中存在没有学号的学生记录，则此学生一定不属于正常管理的学生。另外，如果关系中存在主码值相等的两个或多个记录，则这两个或多个记录会对应同一个实例。这会出现两种情况，第一，若表中的其他属性值也完全相同，则这些记录就是重复的记录，存储重复的记录是无意义的；第二，若其他属性值不完全相同则会出现语义矛盾，比如同一个学生（学号相同），而其名字不同或性别不同，这显然不可能。

关系模型中使用主码作为记录的唯一标识，主码所包含的属性称为关系的主属性，其他的非主码属性称为非主属性。在关系数据库中主属性不能取空值。关系数据库中的空值是特殊的标量常数，它代表未定义的（不适用的）或者有意义但目前还处于未知状态的值。比如当向表 2-3 所示的“学生选课信息”中插入一行数据时，在学生还没有考试之前，其成绩是不确定的，因此，此列上的值即为空。空值用“NULL”表示。

**2. 参照完整性**

参照完整性有时也称为引用完整性。现实世界中的实体之间往往存在着某种联系，在关系模型中，实体以及实体之间的联系都是用关系表示的，这样就自然存在着关系（表）与关系（表）之间的引用关系。参照完整性就是描述实体与实体之间的联系的。

参照完整性一般是指多个实体或表之间的关联关系。比如表 2-3 中，“学生选课信息”所描述的学生必须受限于表 2-1 学生基本信息表中已有的学生，不能在“学生选课信息”中描述一个根本就不存在的学生，也就是“学生选课信息”中学号的取值必须在“学生基本信息”中学号的取值范围内。这种限制一个关系中某属性的取值受另一个关系的某属性取值范围约束的特点就称为参照完整性。在关系数据库中用外码（foreign key，有时也称为外部关键字或外键）来实现参照完整性。例如，只要将“学生选课信息”中的“学号”定义为引用“学生基本信息”的“学号”的外码，就可以保证“学生选课信息”中的“学号”的取值在“学生基本信息”的已有“学号”范围内。

外码一般出现在联系所对应的关系中，用于表示两个或多个实体之间的关联关系。外码实际上是关系中的一个（或多个）属性，它引用某个其他关系（特殊情况下，也可以是外码所在的关系）的主码，当然，也可以是候选码，但多数情况下是主码。

下面举例说明如何指定外码。

【例 1】 设有“学生”和“专业”两个关系模式，其中主码用下划线标识。

学生（学号，姓名，性别，专业号，出生日期）

专业（专业号，专业名）

这两个关系模式之间存在属性引用关系，即“学生”中的“专业号”属性引用了“专业”中的“专业号”属性，显然，“学生”关系模式中的“专业号”属性的取值必须是确实存在的专业的专业号。也就是说，“学生”关系模式中的“专业号”参照了“专业”关系模式中的“专业号”，即“学生”关系模式中的“专业号”是引用了“专业”关系模式中的“专业号”的外码。

【例 2】 学生、课程以及学生与课程之间的选课关系可以用如下三个关系模式表示，其中主码用下划线标识：

学生（学号，姓名，性别，专业号，出生日期）

课程（课程号，课程名，学分）

选课（学号，课程号，成绩）

在这三个关系模式中，“选课”中的“学号”必须是“学生”中已有的学生，因此“选课”中的“学号”属性引用了“学生”中的“学号”属性。同样“选课”中的“课程号”的取值也必须是“课程”中已有的课程，即“选课”中的“课程号”属性引用了“课程”中的“课程号”属性。因此，“选课”关系模式中的“学号”是引用了“学生”关系模式中的“学号”的外码，而“选课”关系模式中的“课程号”是引用了“课程”关系模式中的“课程号”的外码。

主码要求必须是非空且不重复的，但外码无此要求。外码可以有重复值，这点从表 2-3 可以看出。外码也可以取空值，例如职工与其所在的部门可以用如下两个关系模式表示：

职工（职工号，职工名，部门号，工资级别）

部门（部门号，部门名）

其中，“职工”关系模式中的“部门号”是引用“部门”关系模式的“部门号”的外码，如果某新来职工还没有被分配到具体的部门，则其“部门号”就为空值；如果职工已经被分配到了某个部门，则其部门号就有了确定的值（非空值）。

**3. 用户定义的完整性**

用户定义的完整性也称为域完整性或语义完整性。任何关系数据库管理系统都应该支持实体完整性和参照完整性，除此之外，不同的数据库应用系统根据其应用环境的不同，往往还需要一些特殊的约束条件，用户定义的完整性就是针对某一具体应用领域定义的数据约束条件，它反映某一具体应用所涉及的数据必须满足应用语义的要求。

用户定义的完整性实际上就是指明关系中属性的取值范围，也就是属性的域，这样可以限制关系的属性的取值类型及取值范围，防止属性的值与应用语义矛盾。例如，学生考试成绩的取值范围为 0～100，或取｛优、良、中、及格、不及格｝。

## 2.4　数据库系统的结构

本节介绍的数据库系统结构是为后续章节介绍的数据库概念建立一个框架结构，这个框架用于描述一般数据库系统的概念，但并不是说所有的数据库管理系统都一定使用这个框架，这个框架结构在数据库中并不是唯一的，特别是一些“小”的数据库管理系统将难以支持这个体系结构的所有方面。但这里介绍的数据库系统的体系结构基本上能很好地适应大多数数据库管理系统，而且，它基本上与 ANSI/SPARC DBMS 研究组提出的数据库管理系统的体系结构（称作 ANSI/SPARC 体系结构）是相同的。对本部分内容的理解有助于对现代数据库系统的结构和功能有一个较全面的认识。

### 2.4.1　三级模式结构

数据模型（组织层数据模型）是描述数据的一种形式，模式是用给定的数据模型描述具体的数据（就像用某一种编程语言编写具体应用程序一样）。

在 2.3.1 节已介绍模式是数据库中全体数据的逻辑结构和特征的描述，它仅仅涉及型的描述，不涉及具体的值。模式的一个具体值称为模式的一个实例，比如表 2-1 中的每一行数据就是其表头结构（模式）的一个具体实例。一个模式可以有多个实例。模式是相对稳定的（结构不会经常变动），而实例是相对变动的（具体的数据值可以经常变化）。数据模式描述一类事物的结构、属性、类型和约束，实质上是用数据模型对一类事物进行模拟，而实例则反映的是某类事物在某一时刻的当前状态。

ANSI ／ SPARC 体系结构将数据库划分为三层结构：即内模式、概念模式和外模式（参见图 2-6）。

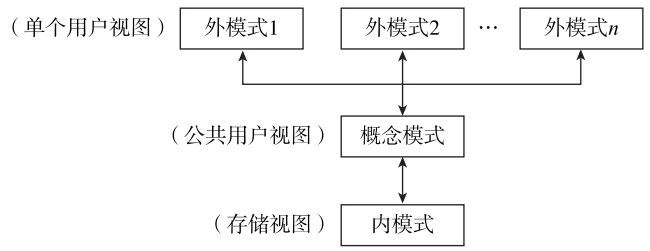


图 2-6　数据库系统的三级模式结构

这三级模式结构的含义如下：

- 内模式：最接近物理存储，也就是数据的物理存储方式。
- 外模式：最接近用户，也就是用户所看到的数据视图。
- 概念模式：介于内模式和外模式之间的中间层次，也称为模式。

在图 2-6 所示的三级模式中，外模式是面向每类用户的数据需求的视图，而模式描述的是一个企业或公司的全部数据。换句话说，外模式可以有许多，每一个都或多或少地抽象表示整个数据库的某一部分数据；而模式只有一个，它是对包含现实世界业务中的全体数据的抽象表示，注意这里的抽象指的是记录和字段这些更加面向用户的概念，而不是位和字节那些面向机器的概念。内模式也只有一个，它表示数据库的物理存储。

这里所讨论的内容与数据库系统是不是关系型的没有直接关系，但简单说明一下关系系统中的三级体系结构，有助于理解这些概念。

第一，关系系统的概念模式一定是关系的，在该层可见的实体是关系的表和关系的操作符。

第二，外模式也是关系的或接近关系的，它们的内容来自概念模式。例如，可以定义两个外模式，一个记录学生的学号、姓名、性别（表示为：学生基本信息 1（学号，姓名，性别）），另一个记录学生的姓名和所在系（表示为：学生基本信息 2（姓名，所在系）），这两个外模式的内容均来自学生基本信息表。“外模式”在关系数据库中对应的是“视图”，它在关系数据库中有特定的含义，我们将在第 5 章详细讨论视图的概念。

第三，内模式不是关系的，因为该层的实体不是关系的原样照搬。其实，不管是什么系统，其内模式都是一样的，都是存储记录、指针、索引、散列表等。事实上，关系模型与内模式无关，它关心的是用户的数据视图。

下面我们从外模式开始进一步详细讨论这三层结构。整个讨论过程都以图 2-7 为基础。该图显示了数据库系统结构的主要组成部分和它们之间的联系。

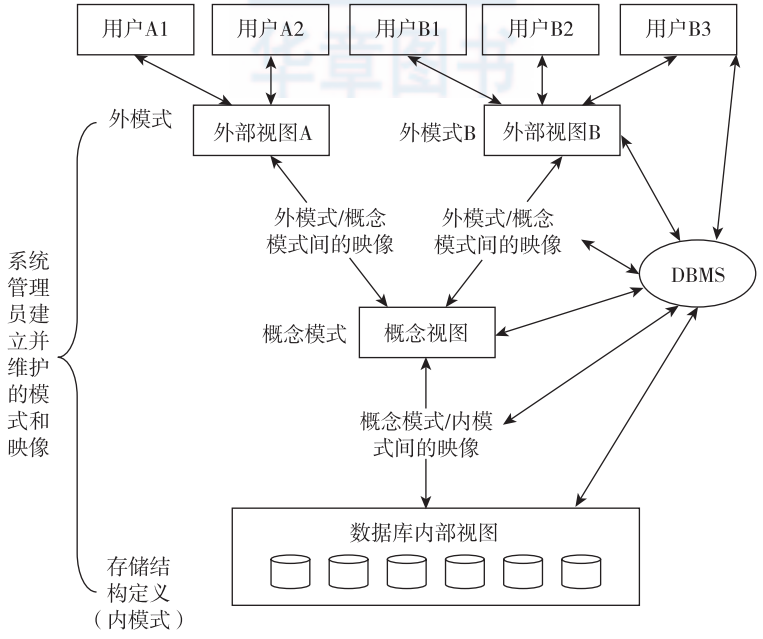


图 2-7　数据库系统的三层结构

### 1. 外模式

外模式也称为用户模式或子模式，它是对现实系统中用户感兴趣的整体数据结构的局部描述，用于满足不同数据库用户需求的数据视图，是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，是对数据库整体数据结构的子集或局部重构。

外模式通常是模式的子集。一个数据库可以有多个外模式。由于外模式是各个用户的数据视图，如果不同的用户在应用需求、看待数据的方式、对数据保密的要求等方面存在差异，则其外模式描述就不相同。模式中同样的数据，在外模式中的结构、类型、长度等都可以不同。

例如，对于表 2-1 所示的学生基本信息，分配宿舍部门关心的信息可能是学号、姓名和性别，学院管理人员关心的信息可能是学号、姓名、所在系。

因此，可以分别为这两类用户建立外模式：

宿舍部门（学号，姓名，性别）

院部（学号，姓名，所在系）

外模式同时也是保证数据库安全的一个措施。每个用户只能看到和访问其所对应的外模式中的数据，并屏蔽其不需要的数据，因此保证不会出现由于用户的误操作和有意破坏而造成数据损失。例如，假设有“职工”关系模式：

职工（职工号，姓名，所在部门，基本工资，职务工资，奖励工资）

如果不希望一般职工看到每个职工的“奖励工资”，则可生成一个包含一般职工可以看的信息的外模式，结构如下：

职工信息（职工号，姓名，所在部门，基本工资，职务工资）

这样就可保证一般用户不会看到“奖励工资”项。

ANSI / SPARC 将用户视图称为外部视图。外部视图就是特定用户所看到的数据库的内容（对那些用户来说，外部视图就是数据库）。

### 2. 概念模式

概念模式也称为逻辑模式或模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。概念模式表示数据库中的全部信息，其形式要比数据的物理存储方式抽象。它是数据库系统结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序、所使用的应用开发工具和环境（比如，Visual Basic、PowerBuilder、C#、Java 等）无关。

概念视图由许多概念记录类型的值构成概念记录。例如，概念视图可以包含学生记录值的集合，课程记录值的集合，选课记录值的集合，等等。它既不与外部记录相同，也不与存储记录相同。

概念视图是用概念模式定义的。概念模式实际上是数据库数据在逻辑层次上的视图。一个数据库只有一种模式。数据库模式以某种数据模型为基础，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。定义数据库模式时不仅要定义数据的逻辑结构，比如，数据记录由哪些数据项组成，数据库项的名字、类型、取值范围等，而且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。

概念模式不涉及存储字段的表示，也不涉及存储记录对列、索引、指针或其他存储的访问细节。如果概念视图以这种方式真正地实现数据独立性，那么根据这些概念模式定义的外模式也会有很强的独立性。

数据库管理系统提供了模式定义语言（DDL）来定义数据库的模式。

3. 内模式

内模式也称为存储模式。内模式是对整个数据库的底层表示，它描述了数据的存储结构，比如数据的组织与存储。注意，内模式与物理层是不一样的，内模式不涉及物理记录的形式（即物理块或页），也不考虑具体设备的柱面或磁道大小。换句话说，内模式假定了一个无限大的线性地址空间，地址空间到物理存储的映射细节是与特定系统有关的，这些并不反映在体系结构中。

内模式用另一种数据定义语言——内部数据定义语言来描述。本书通常使用更直观的“存储结构”来代替“内部视图”，用“存储结构定义”代替“内模式”。

### 2.4.2　模式映像与数据独立性

数据库系统的三级模式（或三层结构）是对数据的三个抽象级别，它把数据的具体组织留给 DBMS 管理，使用户能逻辑、抽象地处理数据，而不必关心数据在计算机中的具体表示方式与存储方式。为了能够在内部实现这三个抽象层的联系和转换，数据库管理系统在三个模式之间提供了以下两级映像（参见图 2-7）：

- 外模式 / 模式映像。
- 模式 / 内模式映像。

正是这两级映像功能保证了数据库中的数据能够具有较高的逻辑独立性和物理独立性，使数据库应用程序不随数据库数据的逻辑或存储结构的变动而变动。

1. 外模式 / 模式映像

模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。对应于同一个模式可以有多个外模式。对于每个外模式，数据库管理系统都有一个外模式到模式的映像，它定义了该外模式与模式之间的对应关系，即如何从外模式找到其对应的模式。这些映像定义通常包含在各自的外模式描述中。

当模式发生变化时（比如，增加新的关系、对某个关系增加新的属性、改变属性的数据类型等），可由数据库管理员用外模式定义语句，调整外模式到模式的映像，从而保持外模式不变。由于应用程序一般是依据数据的外模式编写的，因此也不必修改应用程序，从而保证了程序与数据的逻辑独立性。

例如，设有学生关系模式：

学生（学号，姓名，性别，所在系，专业）

假设在此关系模式上建有外模式：

女学生（学号，姓名，专业）

则当“学生”关系模式发生变化，比如，增加一个“班号”属性，则外模式“女学生”不需要有任何变化，因为该外模式的用户并不关心“班号”属性。另一种情况是，假设“学生”关系模式的“专业”属性改名为“所学专业”，这时就需要数据库管理员调整“女学生”外模式的“专业”属性，使其从原来与“学生”模式的“专业”属性对应，改为与现在的“所学专业”属性对应，这些变化对使用外模式的用户是不可见的，这就是逻辑独立性的含义。

2. 模式 / 内模式映像

模式 / 内模式映像定义了数据库的逻辑结构与物理存储之间的对应关系，该映像关系通常被保存在数据库的系统表（由数据库管理系统自动创建和维护，用于存放维护系统正常运

行的表）中。当数据库的物理存储改变了，比如选择了另一个存储位置，只需要对模式 / 内模式映像做相应的调整，就可以保持模式不变，从而也不必改变应用程序。因此，保证了数据与程序的物理独立性。

在数据库系统的三级模式结构中，模式（即全局逻辑结构）是数据库的中心和关键，它独立于数据库系统的其他层。设计数据库时也是首先设计数据库的逻辑模式。

数据库的内模式依赖于数据库的全局逻辑结构，它独立于数据库的用户视图（也就是外模式），也独立于具体的存储设备。内模式将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织，以达到较好的时间与空间效率。

数据库的外模式面向具体的用户需求，它定义在逻辑模式之上，独立于存储模式和存储设备。当应用需求发生变化，相应的外模式不能满足用户的要求时，就需要对外模式做相应的修改以适应这些变化。因此设计外模式时应充分考虑到应用的扩充性。

原则上，应用程序都应该在外模式描述的数据结构上进行编写，而且它应该只依赖于数据库的外模式，并与数据库的模式和存储结构独立，但目前很多应用程序都是直接针对模式进行编写的，因为外模式到模式的映像有一些局限性，我们将在第 5 章介绍。不同的应用程序有时可以共用同一个外模式。数据库的两级映像保证了数据库外模式的稳定性，从而从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改。

数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出来。另外，由于数据的存取由 DBMS 负责管理和实施，因此，用户不必考虑存取路径等细节，从而简化了应用程序的编制，减少了对应用程序的维护和修改工作。

## 2.5　数据库管理系统

数据库管理系统（DBMS）是处理数据库访问的系统软件，从概念上讲，它包括以下处理过程（参见图 2-8 ）：

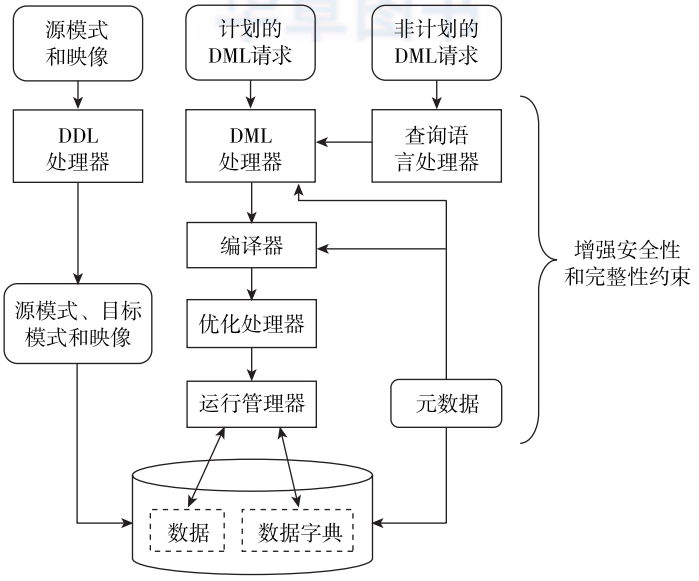


图 2-8　DBMS 的功能和组成

- 用户使用数据库语言（比如 SQL）发出一个访问请求。
- DBMS 接受请求并分析。
- 然后 DBMS 检查用户外模式、相应的外模式 / 概念模式间的映像、概念模式、概念模式 / 内模式间的映像和存储结构定义。

通常在检索数据时，从概念上讲，DBMS 首先检索所有要求的存储记录的值，然后构造所要求的概念记录值，最后再构造所要求的外部记录值。每个阶段都可能需要数据类型或其他方面的转换。当然，这个描述是简化了的，非常简单。但这也说明了整个过程是解释性的，因为它表明分析请求的处理、检查各种模式等都是在运行时进行的。

下面简单地解释一下 DBMS 的功能。DBMS 的功能包括以下几项：

**1. 数据定义**

DBMS 必须能够接受数据库定义的源形式，并把它们转换成相应的目标形式。即 DBMS 必须包括支持各种数据定义语言（DDL）的 DDL 处理器或编译器。

**2. 数据操纵**

DBMS 必须能够检索、更新或删除数据库中已有的数据，或向数据库中插入数据。即 DBMS 必须包括数据操纵语言（DML）的 DML 处理器或编译器。

**3. 优化和执行**

计划（在请求执行前就可以预见到的请求）的或非计划（不可预知的请求）的数据操纵语言请求必须经过优化器的处理，通过优化器来决定执行请求是必要的过程。

**4. 数据安全和完整性**

DBMS 要监控用户的请求，拒绝那些会破坏 DBA 定义的数据库安全性和完整性的请求。在编译或运行时都会执行这些任务。实际操作中，运行管理器调用文件管理器来访问存储的数据。

**5. 数据恢复和并发**

DBMS 或其他相关的软件（通常称为“事务处理器”或“事务处理监控器”）必须保证有恢复和并发控制功能。

**6. 数据字典**

DBMS 包括数据字典。数据字典本身也可以看作一个数据库，只不过它是系统数据库，而不是用户数据库。“字典”是“关于数据的数据”(有时也称为数据的描述或元数据)。特别地，在数据字典中，也保存各种模式和映像的各种安全性和完整性约束。

有些人也把数据字典称为目录或分类，有时也称为数据存储池。

**7. 性能**

DBMS 应尽可能高效地完成全部任务。

总而言之，DBMS 的目标就是提供数据库的用户接口。用户接口可定义为系统的边界，在此之下的数据对用户来说是不可见的。

## 2.6　小结

本章首先介绍了数据库中数据模型的概念。数据模型根据其应用的对象分为两个层次：概念层数据模型和组织层数据模型。概念层数据模型是对现实世界信息的第一次抽象，它与

具体的数据库管理系统无关，是用户与数据库设计人员的交流工具。因此概念层数据模型一般采用比较直观的模型，本章主要介绍的是应用范围很广泛的实体－联系模型。

组织层数据模型是对现实世界信息的第二次抽象，它与具体的数据库管理系统有关，也就是与数据库管理系统采用的数据的组织方式有关。从概念层数据模型到组织层数据模型的转换一般是很方便的。本章主要介绍了目前应用范围最广、技术发展非常成熟的关系数据模型。

最后本章从体系结构角度分析了数据库系统，介绍了三个模式和两个映像。三个模式分别为：内模式、概念模式和外模式。内模式最接近物理存储，它考虑数据的物理存储；外模式最接近用户，它主要考虑单个用户看待数据的方式；概念模式介于内模式和外模式之间，它提供数据的公共视图。两个映像分别是概念模式与内模式间的映像和外模式与概念模式间的映像，这两个映像是提供数据的逻辑独立性和物理独立性的关键。最后介绍了数据库管理系统的功能，DBMS 主要负责执行用户的数据定义和数据操纵语言的请求，同时也负责提供数据字典的功能。

## 习题

1. 解释数据模型的概念，为什么要将数据模型分成两个层次？
2. 概念层数据模型和组织层数据模型分别是针对什么进行的抽象？
3. 实体之间的联系有哪几种？请为每一种联系列举一个例子。
4. 说明实体－联系模型中的实体、属性和联系的概念。
5. 指明下列实体间联系的种类：
   （1）教研室和教师（假设一个教师只属于一个教研室，一个教研室可有多名教师）。
   （2）商店和顾客。
   （3）国家和首都。
6. 解释关系模型中的主码、外码、主属性、非主属性的概念，并说明主码、外码的作用。
7. 指出下列关系模式的主码：
   （1）考试情况（课程号，考试性质，考试日期，考试地点）。假设一门课程在不同的日期可以有多次考试，但在同一天只能考一次。多门不同的课程可以同时进行考试。
   （2）教师授课（教师号，课程号，授课时数，学年，学期）。假设一名教师在同一个学年和学期可以讲授多门课程，也可以在不同学年和学期多次讲授同一门课程，对每门课程的讲授都有一个授课时数。
   （3）图书借阅（书号，读者号，借书日期，还书日期）。假设一个读者可以在不同的日期多次借阅同一本书，一个读者可以同时借阅多本不同的图书，一本书可以在不同的时间借给不同的读者。但一个读者不能在同一天对同一本书借阅多次。
8. 设有如下两个关系模式，试指出每个关系模式的主码、外码，并说明外码的引用关系。
   产品（产品号，产品名，价格），其中“产品名”可能有重复。
   销售（产品号，销售时间，销售数量），假设可同时销售多种产品，但同一产品在同一时间只销售一次。
9. 关系模型的数据完整性包含哪些内容？分别说明每一种完整性的作用。
10. 数据库包含哪三级模式？试分别说明每一级模式的作用？
11. 数据库管理系统提供的两级映像是什么？它带来了哪些功能？
12. 数据库三级模式划分的优点是什么？它能带来哪些数据独立性？
13. 简单说明数据库管理系统包含的功能。