

数据模型

刘先锋

数据模型

[学习目的与要求]

深刻理解数据模型的内涵、数据库的三层模式结构与数据独立性的关系，理解数据从现实世界到计算机数据库中要经过三个范畴（现实世界、信息世界和机器世界），了解什么是实体属性，弄清楚实体和属性的“型”与“值”的概念，弄清实体间可能存在的不同联系方式，掌握用E-R图表示实体间联系的方式。

2.1 数据描述

2.2 概念数据模型与E-R方法

2.3 传统的三大数据模型

2.4 数据独立与三层结构

2.5 数据库管理系统

2.1.1 数据的三种范畴

数据不是直接从现实世界到计算机数据库中，它需要人们的认识、理解、整理、规范和加工，然后才能存放到数据库中。也就是说数据从现实生活进入到数据库实际上经历了若干个阶段。一般划分三个阶段，**即现实世界、信息世界和机器世界，称为数据的三种范畴。**

1. 现实世界

现实世界也叫客观世界。存在于人们头脑之外的客观事物及其相互联系就处在这个世界之中。

2. 信息世界（也叫观念世界）

信息世界又称观念世界，是现实世界在人们头脑中的反映；在进行现实世界管理时，客观事物必然在人们的头脑中产生反映，把这种反映称为信息。比如在日常的库存管理中，首先涉及的是仓库、货物的存放以及货物的进出库等，这种管理称为现实世界管理。

下面给出在信息世界中所涉及到的基本概念:

(1) 实体 (Entity)

实体是客观存在的事物在人们头脑中的反映,或者说,客观存在并可相互区别的客观事物或抽象事件称为实体。**实体可以指人**,如一名教师、一名护士等;**也可以指物**,如一把椅子、仓库、一个杯子等。**实体不仅可以指实际的事物,还可以指抽象的事物**,如一次访问、一次郊游、订货、演出、足球赛等;**甚至还可以指事物与事物之间的联系**,如“学生选课记录”和“教师任课记录”等。

(2) 属性 (Attribute)

在观念世界中,属性是一个很重要的概念。所谓属性是指实体所具有的某一方面的特性。一个实体可由若干个属性来刻画。例如,教师的属性有姓名、年龄、性别、职称等。

属性所取的具体值称作**属性值**。例如,某一教师的姓名为李辉,这是教师属性“姓名”的取值;该教师的年龄为45,这是教师属性“年龄”的取值,等等。

(3) 域 (Domain)

一个属性可能取的所有属性值的范围称为该属性的域。例如，教师属性“性别”的域为男、女；教师属性“职称”的域为助教、讲师、副教授、教授等。

由此可见，每个属性都是个变量，属性值就是变量所取的值，而域则是变量的变化范围。因此，属性是表征实体的最基本的信息。

(4) 码 (Key)

惟一标识实体的属性集称为码。例如学号是学生实体的码；姓名+出生年月等等

(5) 实体型 (Entity Type)

具有相同属性的实体必然具有共同的特性和性质。用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。例如，教师（姓名，年龄，性别，职称）就是一个实体型。

(6) 实体集 (Entity Set)

同一类型实体的集合。例如，某一学校中的教师具有相同的属性，他们就构成了实体集“教师”。

在信息世界中，一般就用上述这些概念来描述各种客观事物及其相互的区别与联系。

3. 机器世界（也叫数据世界）

当信息管理进入计算机后，就把它称为机器世界范畴或存储世界范畴。机器世界也称数据世界。

由于计算机只能处理数据化的信息，所以对信息世界中的信息必须进行数据化。信息经过加工、编码后即进入数据世界，利用计算机来处理它们。因此，数据世界中的对象是数据。现实世界中的客观事物及其联系在数据世界中是用数据模型来描述的。

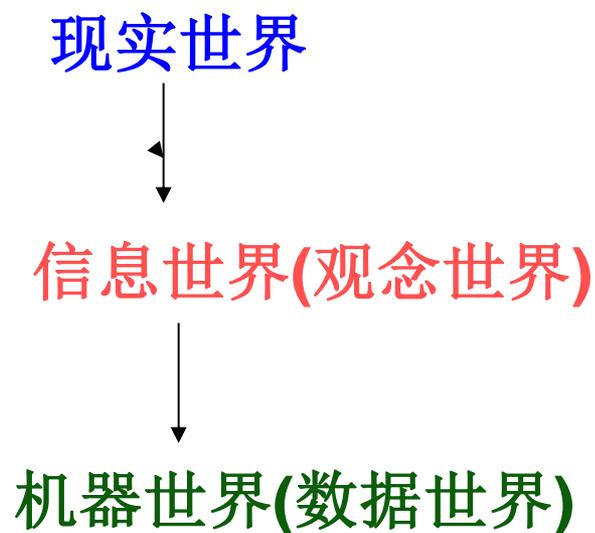
数据化后的信息称为数据，所以说数据是信息的符号表示。

与观念世界中的基本概念对应，在数据世界中也涉及到一些相关的基本概念：

- (1) 数据项（字段）（**field**）。对应于观念世界中的属性。例如，实体型“教师”中的各个属性中，姓名、性别、年龄、职称等就是数据项。
- (2) 记录（**record**）。每个实体所对应的数据。例如，对应某一教师的各项属性值为：李辉、45、男、副教授等就是一个记录。
- (3) 记录型（**record type**）。对应于观念世界中的实体型。
- (4) 文件（**file**）。对应于观念世界中的实体集。
- (5) 关键字（**key**）。能够惟一标识一个记录的字段集。

在数据世界中，就是通过上述这些概念来描述客观事物及其联系的。

上述信息是为了更好地处理信息，计算机所处理的信息形式是数据。因此，为了用计算机来处理信息，首先必须将现实世界中的客观事物转换为观念世界，然后将这些信息数据化。



2.1.2 实体间的联系

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中反映为实体（型）**内部**的联系和实体（型）**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。实体之间的联系通常是指不同实体集之间的联系。

一对一联系（1：1）

如果对于实体集A中的每一个实体，实体集B中至多有一个（也可以没有）实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系，记为1：1。

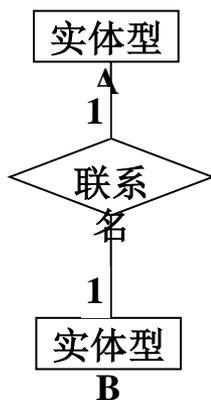


图2-1 1：1联系

例如，实体集学院与实体集院长之间的联系就是**1: 1**的联系。因为一个院长只领导一个学院，而且一个学院也只有一个院长。再如学校里，实体集班级与实体集班长之间的也具有**1: 1**联系，一个班级只有一个班长，而一个班长只在一个班中任职。

一对多联系 (1: n)

如果对于实体集A中的每一个实体，实体集B中有n个 ($n \geq 0$) 实体与之联系，反之，对于实体集B中的每一个实体，实体集A中至多有一个实体与之联系，则称实体集A与实体集B具有一对多联系，记为**1: n**，

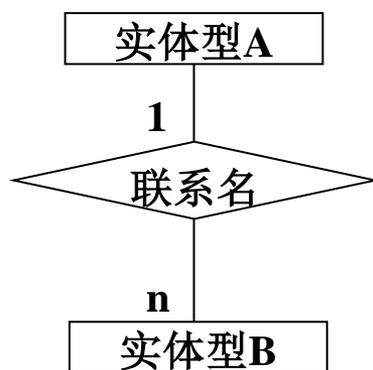


图2-2 1:n联系

例如，实体集班级与实体集学生就是一对多联系。因为一个班级中有若干名学生，而每个学生只在一个班级中学习。

多对多联系（ $m:n$ ）

如果对于实体集A中的每一个实体，实体集B中有 n 个（ $n \geq 0$ ）实体与之联系。反之，对于实体集B中的每一个实体，实体集A中也有 m （ $m \geq 0$ ）之联系，则称实体集A与实体集B具有多对多联系，记为 $m:n$ ，

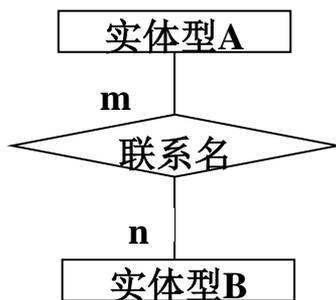


图2-3 $m:n$ 联系

例如，实体集课程与实体集学生之间的联系是多对多联系（ $m:n$ ）。因为一个课程同时有若干名学生选修，而一个学生可以同时选修多门课程。

实体型之间的这种一对一、一对多、多对多联系不仅存在于两个实体型之间，也存在于两个以上的实体型之间。**例如**，对于课程、教师与参考书三个实体型，如果一门课程可以有若干个教师讲授，使用若干本参考书，而每一个教师只讲授一门课程，每一本参考书只供一门课程使用，则课程与教师、参考书之间的联系是一对多的，

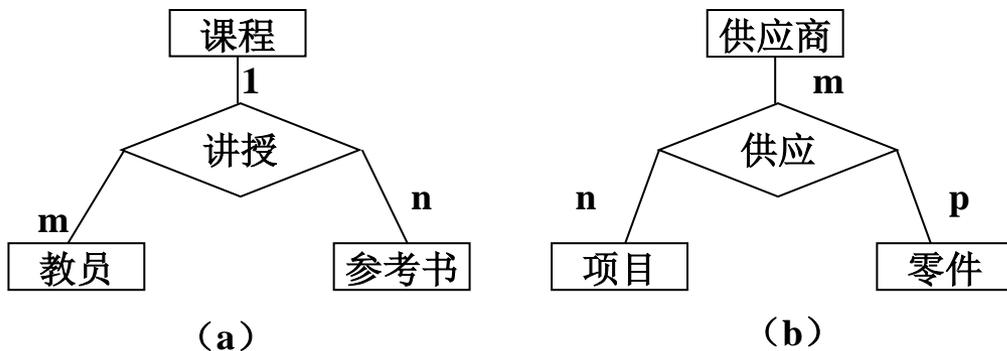
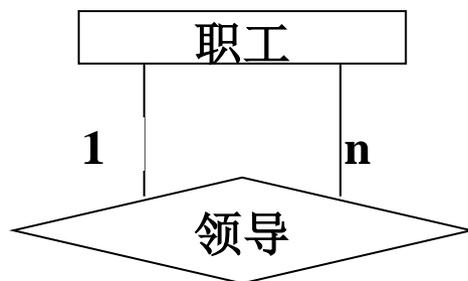


图2-4 三个实体型之间的联系

又如，三个实体型：供应商、项目、零件，一个供应商可以供给多个项目多种零件，而每个项目可以使用多个供应商供应的零件，每种零件可由不同供应商供给，由此可见，供应商、项目、零件三个实体之间是多对多的联系，

同一个实体集内的各实体之间也存在一对一、一对多、多对多的联系。例如职工实体集内部具有领导与被领导的联系，即某一职工（干部）“领导”若干名职工，而一个职工仅被另外一个职工直接领导，因此这是同一实体集一对多的联系。



一个实体型之间的一对多联系

描述信息是为了更好地处理信息，计算机所处理的信息形式是数据。因此，为了用计算机来处理信息，首先必须将现实世界中的客观事物转换为信息世界，然后将这些信息**数据化**。

2.2 概念数据模型与E-R方法(Entity Relation)

2.2.1 数据模型概述

为了用计算机处理现实世界中的具体事物，人们必须事先对具体事物加以抽象，提取主要特征，归纳形成一个简单清晰的轮廓，转换成计算机能够处理的数据，这就是“数据建模”。通俗地讲数据模型就是现实世界的模型。表示实体类型及实体之间联系的模型称为“数据模型”（Data Model）。

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；

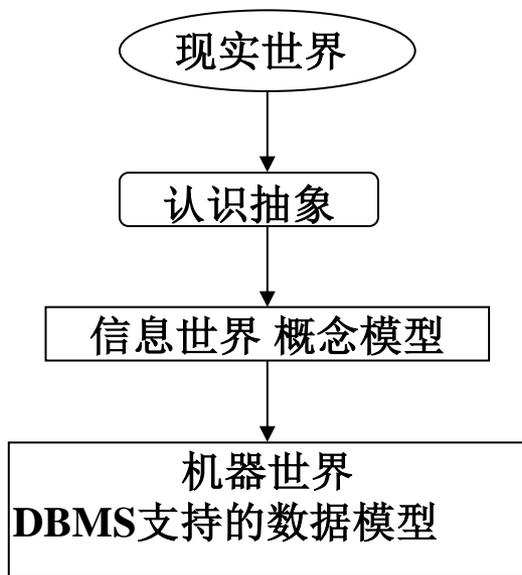
三是便于在计算机上实现。

在数据库系统中针对不同的使用对象和应用目的，采用不同的数据模型。

根据模型的应用的不同目的，可以将这些模型划分为两类，它们分属于不同的层次。

第一类模型是概念数据模型，也称信息模型，它是按用户的观点来对数据和信息建模，主要用于**数据库设计**。

另一类模型是基本数据模型，主要包括网状模型、层次模型、关系模型等，它是按计算机系统的观点数据建模，主要用于**DBMS的实现**。



现实世界中客观对象的抽象过程

过程说明: 首先把现实世界中的客观对象抽象为某种信息结构，这种信息结构并不依赖于具体的计算机系统，不是某一个**DBMS** 支持的数据模型，而是概念级的模型；然后再把概念模型转换为计算机上某一**DBMS**支持的数据模型，

数据模型的三要素

1. 数据结构

数据结构是所研究的对象类型的集合。这些对象是数据库的组成部分，它们包括两类，一类是与数据类型、内容、性质有关的对象，例如网状模型中的数据项、记录，关系模型中的域、属性、关系等；一类是与数据之间联系有关的对象，例如网状模型中的系型（Set Type）。

数据结构用于描述系统的静态特性。

2. 数据操作

数据操作是指对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有检索和修改（包括插入、删除、更新）两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级）以及实现操作的语言。

数据操作用于描述系统的动态特征。

3. 数据完整性约束

数据完整性约束是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和储存规则，用以限制符合数据模型的数据库状态以及状态的变化，用以确保数据的正确、有效和相容。

数据模型之概念数据模型

概念数据模型，有时也简称**概念模型**。概念数据模型是按用户的观点对现实世界数据建模，是一种独立于任何计算机系统的模型，完全不涉及信息在计算机系统上的表示，也不依赖于具体的数据库管理系统。只是用来描述某个特定组织所关心的信息结构。它是对现实世界的第一层抽象，是用户和数据库设计人员之间交流的工具。

概念数据模型是理解数据库的基础，也是设计数据库的基础。

1. 概念数据模型的基本概念

概念数据模型所涉及的主要基本概念有：实体（**Entity**）、属性（**Attribute**）、域（**Domain**）、码（**Key**）、实体型（**Entity Type**）和实体集（**Entity Set**）。

2. 概念数据模型中的基本关系

实体间一对一、一对多和多对多三类基本联系是概念数据模型的基础。

实体之间的联系类型并不取决于实体本身，而是取决于现实世界的管理方法，或者说取决于语义，即同样两个实体，如果有不同的语义，则可以得到不同的联系类型。 P24实例

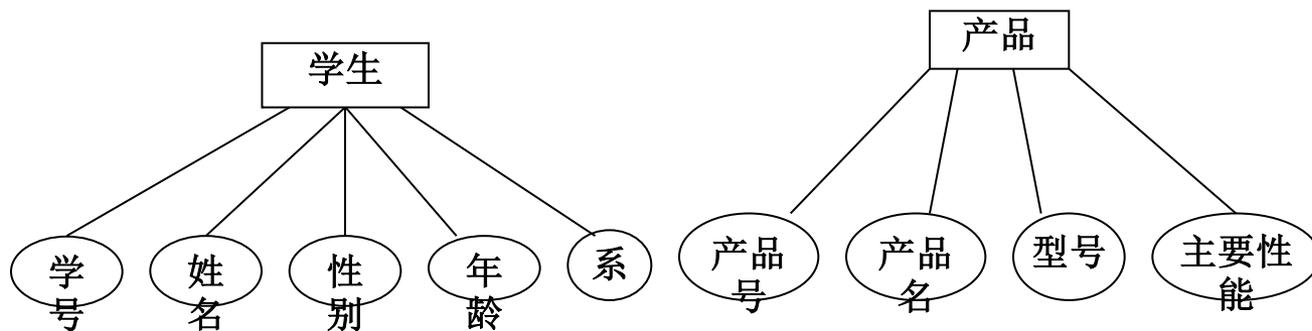
概念数据模型的E-R表示方法

E-R图提供了表示实体型、属性和联系的方法：

实体型：用矩形表示，矩形框内写明实体名。

属性：用椭圆表示，椭圆形框内写明属性名，并用无向边将其与相应的实体连接起来。

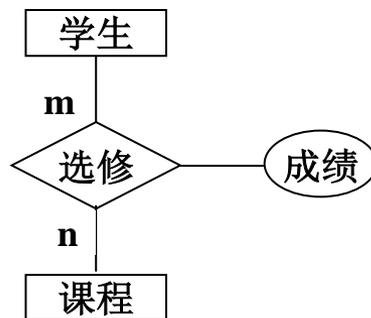
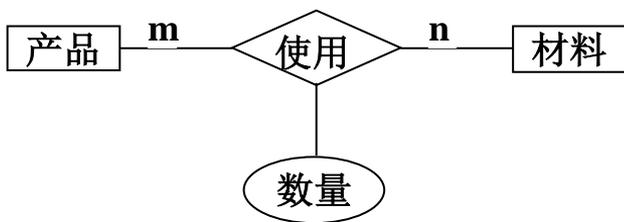
例如，学生实体具有学号、姓名、性别、年龄、系等属性，产品实体具有产品号、产品名、型号、主要性能等属性。



实体及属性

联系：用菱形表示，菱形框内写联系名，并用无向边分别与有关实体连接起来，同时，在无向边旁标注联系的类型（1: 1, 1: n或m: n）。

联系也可以有自己的属性，**需要注意的是**，如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来。



联系及属性

数据模型之传统的三大数据模型

目前常用的数据模型有三种：层次模型、网状模型和关系模型。其中层次模型和网状模型统称为非关系模型。

1. 层次模型

用树型结构来表示实体之间联系的模型称为层次模型。

构成层次模型的树是由结点和连线组成的，结点表示实体集（文件或记录型），连线表示相连两个实体之间的联系，这种联系只能是一对多的。通常把表示“一”的实体放在上方，称为父结点；而把表示“多”的实体放在下方，称为子结点。根据树结构的特点，建立数据的层次模型需要满足下列两个条件：

- (1) 有且仅有一个结点没有父结点，这个结点即为树根结点。
- (2) 其他数据记录有且仅有一个父结点。

现实世界中许多实体之间的联系本身就呈现一种很自然的层次关系。例如，一个学院下属有若干个系、处和研究所：每个系下属有若干个教研室和办公室；每个处下层有若干个科室，每个研究所下属有若干个教研室和办公室；等等。这样一个学校的行政机构就明显地有层次关系，

层次模型的一个基本的特点是，任何一个给定的记录值只有按其路径查看时，才能现出它的全部意义，没有一个子女记录值能够脱离双亲记录值而独立存在。

层次模型最明显的特点是层次清楚、构造简单以及易于实现，它可以很方便地表示出一对一和一对多这两种实体之间的联系。

层次模型的主要优点有：

(1)层次数据模型本身比较简单。

(2)对于实体间联系是固定的，且预先定义好的应用系统，采用层次模型来实现，其性能优于关系模型，不低于网状模型。

(3)层次数据模型提供了良好的完整性支持。

层次模型的主要缺点有：

(1)现实世界中很多联系是非层次性的，如多对多联系、一个结点具有多个双亲等，层次模型表示这类联系的方法很笨拙，只能通过引入冗余数据（易产生不一致性）或创建非自然组织（引入虚结点）来解决。

(2)对插入和删除操作的限制比较多。

(3)查询子结点必须通过双亲结点。

(4)由于结构严密，层次命令趋于程序化。

用层次模型设计出来的数据库称为层次数据库。

层次模型主要用于表示一对一、一对多的关系。

2. 网状模型

网状模型和层次模型在本质上是一样的，从逻辑上看它们都是用连线表示实体之间的联系，用结点表示实体集；从物理上看，层次模型和网络模型都是用指针来实现两个文件之间的联系，其差别仅在于网状模型中的连线或指针更加复杂，更加纵横交错，从而使数据结构更复杂。

在数据库中，把满足以下两个条件的基本层次联系集合称为**网状模型**：

- (1) 允许一个以上的结点无双亲；
- (2) 一个结点可以有多个的双亲。

网状模型是一种比层次模型更具普遍性的结构，它去掉了层次模型的两个限制，允许多个结点没有双亲结点，允许结点有多个双亲结点，此外它还允许两个结点之间有多种联系（称之为复合联系）。

由于网状模型所描述的数据之间的关系要比层次模型复杂得多，在层次模型中子结点与双亲结点的联系是唯一的，而在网状模型中这种联系可以不唯一。因此，为了描述网状模型的记录之间的联系，引进了

“**系**（set）”概念。所谓“系”可以理解为命名了的联系，它由一个父记录型和一个或多个子记录型构成。每一种联系都用“系”来表示，并将其标以不同的名称，以便相互区别。

用网状模型设计出来的数据库称为**网状数据库**。网状数据库是目前应用较为广泛的一种数据库，它不仅具有层次模型数据库的一些特点而且也能方便地描述较为复杂的数据关系。可以看出，网状模型是层次模型的一般形式，层次模型则是网状模型的特殊情况。

网状模型可以直接表示实体之间多对多的联系。

网状数据模型的优点主要有：

- 能够更为直接地描述现实世界，如一个结点可以有多个双亲。
- 具有良好的性能，存取效率较高。

网状数据模型的缺点主要有：

- 结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于用户最终掌握。
- 其DDL，DML语言复杂，用户不容易使用。

3. 关系模型

关系模型是用表格数据来表示实体本身及其相互之间的联系的，在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。

在关系模型中，把数据看成一个**二维表**，每一个二维表称为一个**关系**。关系表中的每一列称为**属性**，相当于记录中的一个**数据项**，对属性的命名称为**属性名**；表中的一行称为一个**元组**，相当于**记录值**。

对于表示关系的二维表，**其最基本的要求是**，表中元组的每一个分量必须是不可分的数据项，即不允许表中再有表。关系是关系模型中最基本的概念。

关系模型较之格式化模型有以下几个方面的优点：

(1) 数据结构比较简单。在关系模型中，对实体的描述以及对实体之间联系的描述，都采用关系这个单一的结构来表示。因此，数据的结构比较简单、清晰。

(2) 具有很高的数据独立性。在关系模型中，用户完全不涉及数据的物理存放，只与数据本身的特性发生关系。因此，数据独立性很高。

(3) 可以直接处理多对多的联系。在关系模型中，由于使用表格数据来表示实体之间的联系，因此，可以直接描述多对多的实体联系。 **P33**

(4) 坚实的理论基础。

在关系模型中，一个n元关系有n个属性，属性的取值范围称为值域。

一个关系属性名的表称为关系模式，也就是二维表的框架，相当于记录型。若某一关系的名称为R，其属性名为A1, A2, ..., An，则该关系的模式记为：

R (A1, A2, ..., An) P34

现在耳闻目睹的数据库管理系统，全部都是关系数据库管理系统，像 Sybase、Oracle、MS SQL Server以及FoxPro和Access等。

当然，关系数据模型也有缺点，其中**最主要的缺点**是，查询效率往往不如非关系数据模型。