微机数据库通用设计方法

科海培训中心
系列教材

敬喜　编

北京科海培训中心

序　　言

数据库是一门综合性的软件技术。其理论性和实践性都很强，是使用计算机进行各种信息管理的必备的技术。本书将以理论和实践并重，使读者既能掌握数据库系统的基本概念、基本原理和基础技术，又能掌握对所有部门和团体都适用的数据库设计的概念、特点、方法和步骤。

本书将系统地、深入浅出地阐述被公认为较好的分析和设计数据库的方法。这种方法是建立在牢固的理论基础之上的。并且是一种统一的设计方法，也就是说，无论现有的数据库管理系统(DBMS)是关系型的，或是层次型的，或是网型的，这种统一的设计方法都是适用的！我们力求准确地、简单明了的阐述这种方法的论点和技巧。在此基础上，本书专门开辟了一个附录A：数据库设计演习与指导。其是运用这种统一的设计方法，把数据库设计的全过程演示了一遍。这样就可以使广大计算机软件工作者在比较短的时间内掌握和运用这种统一的数据库设计方法。

为了使读者更好的掌握和学习本书的内容，下面就把本书的体系结构介绍给读者。

1975年，美国标准化组织ANSI/SPARC提出了三级数据库管理系统结构的第一个建议报告。此“三级”由下述三种数据模型来描述：

1. 外部数据模型
2. 概念数据模型
3. 内部数据模型

这个建议的核心是概念数据模型。概念数模型应对微观世界的逻辑数据结构提供一种完整的、无冗余的、中性的表示。这样就可以从概念数据模型出发，一方面将它映射到一系列描述物理结构的内部数据模型，另一方面又可以将它映射到一系列派生出来的外部数据模型(用户数据模型)。

图0.1表示的是数据库的整个结构。它是由相应数据模型来描述的三个范畴。

这种三级结构，当今已被广泛接受。因为它能够使人们达到高度的数据独立性，并且是用映射的办法取得的。这种方法就是，先把外部数据模型映射到近似于概念数据模型的总体逻辑模型。然后又把总体逻辑模型映射到物理模型。

在本书中，我们将始终使用SPARC的术语，并把“正则结构”称为“概念模型。”图0.2说明了本书的内容及如何和SPARC方法联系起来的。

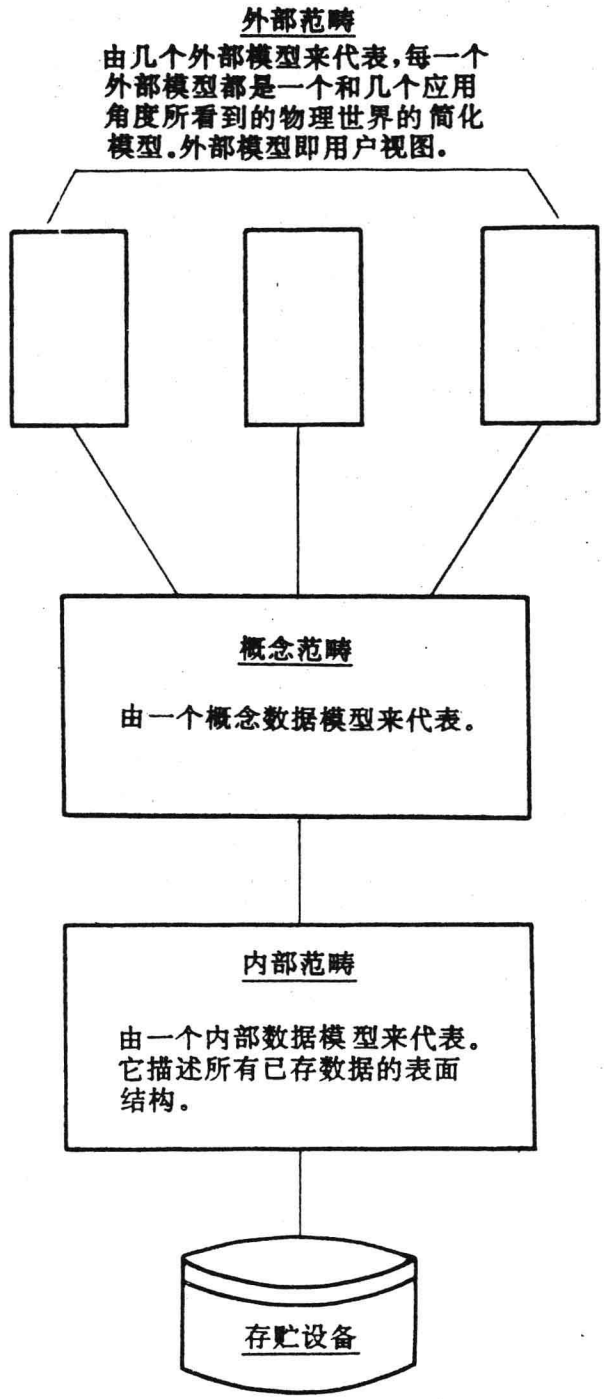


图 0.1　ANSI/SPARC 数据库系统结构

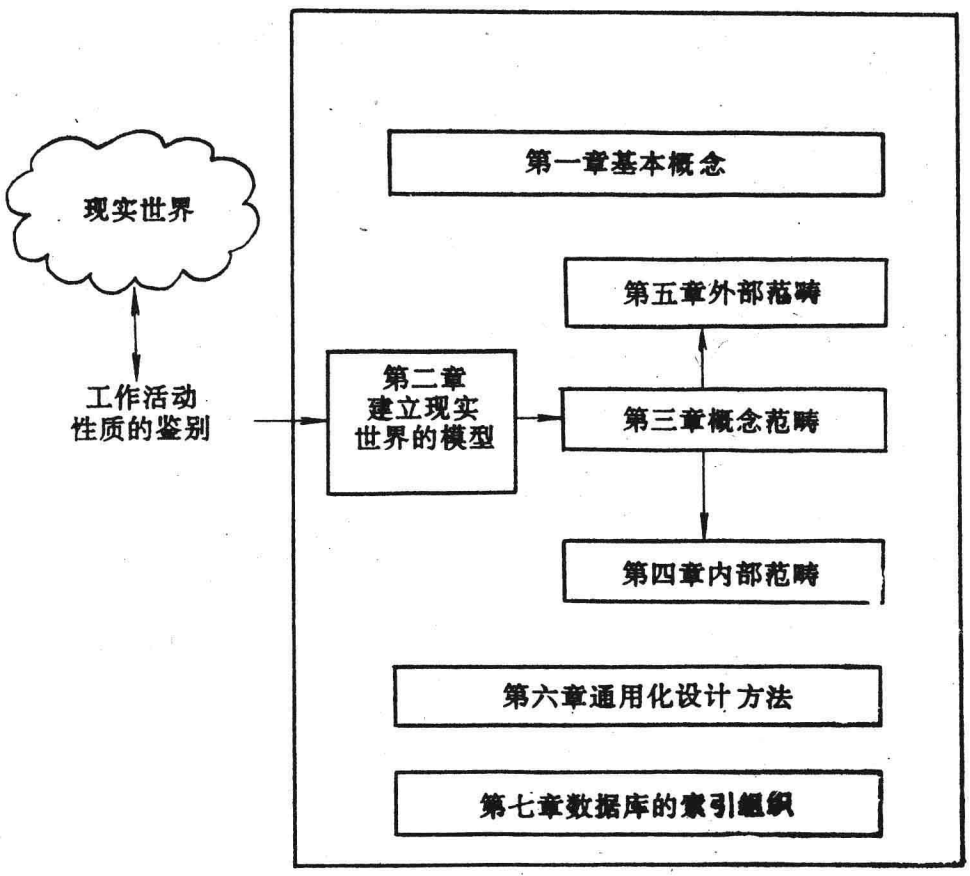


图 0.2. 本书各章所论述的范围

本书不论述主题鉴别阶段，即对于“一个部门应该为数据库选择什么样的信息主题范围”这一问题，我们不去讨论。读者自己会做出答案的。本书力图使读者“在知道了该部门想要在数据库中反映的那部分现实世界之后，知道如何定义数据、建立数据结构，储存和提供数据。”将在这方面向读者提供严密而可行的方法。

我们把整个数据库设计过程划分为五个阶段：

1. 定义阶段（第二章）

2. 概念化阶段（第三章）

3. 计算机效率优化阶段，即关系的规范化阶段（第四章）

4. 人工效率优化阶段（第五章）

5. 巩固阶段（第六章）

定义阶段（第二章）所讨论的内容是，一旦分析人员知道了“要考虑的现实世界的那部分之后，如何去确定和定义一个部门的信息资源。”因此，我们先介绍现实世界的原始概念的实体、实体性质、性质值，以及实体关联等概念。在鉴别阶段所得到的就是有关一个部门的实体、实体关联等这些原始概念供定义阶段再作进一步的分析处理。然后介绍实体集合、关联集合、实体属性及关联属性的概念，即所谓“概念体”的概念。这样，设计人员就可以对现实世界的概念进行分类，并把设计过程建立在坚实的理论基础之上。

概念化阶段（第三章）是本书的重点。主要是使设计人员会分析各种信息类型之间的联系，如何表示概念体，使得每一个事实最多表示一次。这不仅为达到高度的数据独立性所必需，而且是，要把设计过程建立在坚实理论的基础上必然就会得出的结论。

图 0.3 说明了讲述这一章的方法和实际的作用。这一章安排了如下三节：

3.3 确定传递闭包

3.4 确定最小覆盖

3.5 缩减基本关系的数目

实际上是设计过程的三个步骤。这三个步骤不是非要不可，可以沿着虚线方向直接跳到第四章去讲。然而，想要得到一个最佳的设计方案，设计人员应该进行上面所说的三个步骤。

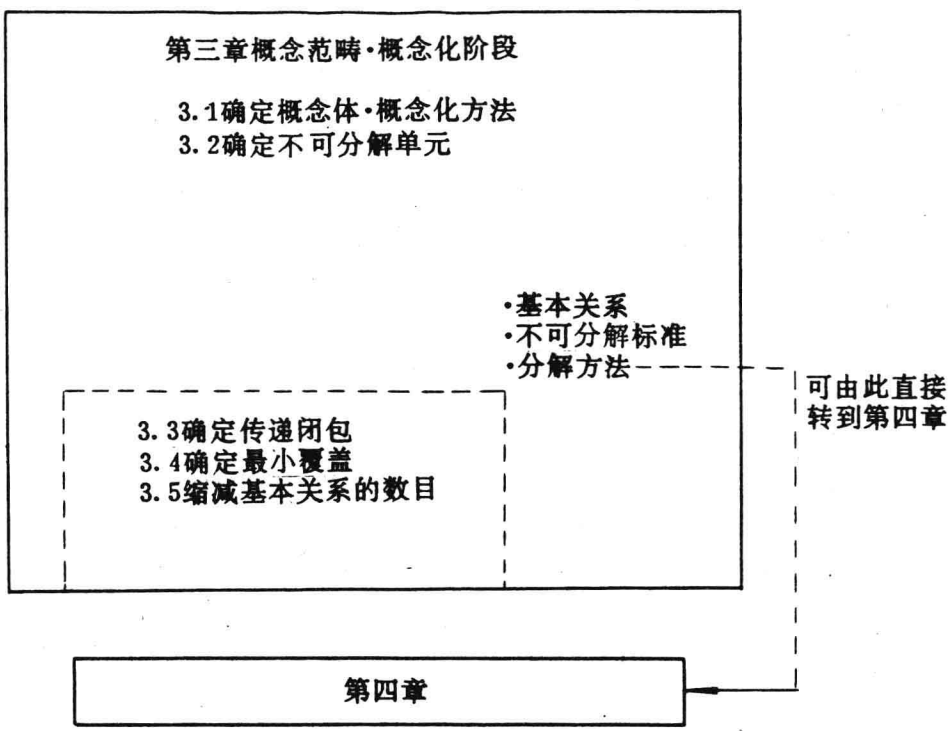

第三章概念范畴·概念化阶段
3.1确定概念体·概念化方法
3.2确定不可分解单元
·基本关系
·不可分解标准
·分解方法
可由此直接
转到第四章
3.3确定传递闭包
3.4确定最小覆盖
3.5缩减基本关系的数目
第四章

图 0.3

计算机效率优化阶段(第四章),涉及到关系的规范化问题,以及讨论如何来解释用以CODASYL为基础的DBMS来实现由n元关系描述的内部模型。

人工效率优化阶段(第五章),涉及的是,要以一种为用户所熟悉又便于数据问题的解决的形式来向用户提供数据。为了便于数据的使用,人们应该能够把某一种表示法转换成另一种形式的表示方法。这一章主要介绍如何把前四章推导出来的、独立于系统的数据模型映射到具体的由DBMS所支持的数据模型。

巩固阶段(第六章通用化设计方法),是一个数据库设计过程的综合性练习,以便巩固所学的内容。

第七章是向读者介绍数据库的物理组织方法中的部分内容,即索引的组织方法,其目的在于介绍如何提高数据库使用的效率。

附录A是通过一个实例向读者展示设计数据库的全过程,以便使读者知道怎样把学过的数据库设计方法运用到实践中去。

另外,为了便于自学,在附录E中给出了全书所有习题的答案,供读者参考。

最后,我们向读者说明设计一个数据库的要求是什么?概括起来说有以下要求:

(1)要为多种多样的应用项目服务;

(2)因此要能很容易增加新的数据类型和新的应用项目;

(3)要做到随着用途和使用方式的演变,存在磁盘上的数据结构也要能加以改变,而又不影响应用程序;

(4)因此所有数据类型都可以存入数据,虽然任何一个用户都不知道这种数据类型;

(5)该数据库要和已建立起来的标准相一致。

(6)该数据库使用的数据库管理系统是可以在某个计算机上运行的DBMS。

因此,设计的目的就是要建立这样一种数据库结构,它体现了下列两点并以其作为设计的基础:

(1)由上述六条要求而得到的各种结论(或原则);

(2)一个部门所具有的长期不变的特点(或具体情况)。

设计时,就是要把这些原则和具体情况有机地结合起来。在某种意义上来说,分析与设计的目的就是鉴别出对于一个部门来说是长期不变的特点及根据上面的六条要求所得到的结论(或原则)来设计数据库。这样的要求和原则对所有工作部门都是适用的。

目　　录

(205)

第一章　基本概念

1.1 联系的含义

在实际生活中，两个事物总是可以把它们联系起来的。譬如说，某个老师和某个同学是同乡，某个教师教这个同学的数学，这样，在师生间便有了联系，这种联系也就是一种关系。所以，关系也是表现客观事物之间的联系的一种方法。

为了能够区别各种不同的信息，可以把联系加以分类。这就是简单型联系，条件型联系和复杂型联系。

现在先看一个订货单的例子，如表 1.1.1 所示。

表 1.1.1　　定单

定单号	交货日期	订货日期	金额	订货人
201	89.10.2	88.12.1	12.35	甲
202	89.11.2	88.10.2	14.50	乙
203	90.1.2	88.9.2	12.35	丙

同这个定单有关的信息类型是：定单号，交货日期，订货日期，金额，订货人等，我们把它们叫做这个定单的属性。这些属性名下面就是各属性的值。每一行的值就是一张特定的定单所具有的值，即每一行代表一张定单。这张定单里含有各种信息类型以及它们间的联系。

1.1.1 简单联系

我们看一下表 1.1.1 中“定单号”同“金额”间的联系：

定单号	金额
201	12.35
202	14.50
203	12.35

可见，每一个定单都有一个金额与之联系，不可能有两个金额。也就是说，一个定单就决定了一个金额，我们说这是“对一”的联系，即简单联系，写成：

定单号——→金额

但是，从金额到定单号就不是“对一”的联系了，因为同一个金额 12.35 同 201 和 203 两个定单号相联系。所以说，联系是有方向性的。

作为“定单号”是值的集合，“金额”也是值的集合。这两个集合间的联系（从定单号到金额）形成了有序对，例如，<201,12.35>，<202,14.50>，<203,12.35>。于是可以画成下面的图形（图 1.1.1）。这是作为信息类型的二个集合之间的联系的一种表示。

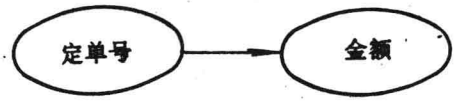


图 1.1.1

这样，我们就可以说：如果集合 A(如，定单号)的每一个成员只和集合 B(如，金额)的一个成员有联系(如，上面的有序对)，那么就说这样的联系是简单联系(也称 1 型联系)。并记作，

$$a:A\longrightarrow B$$

读成“a 是从 A 到 B 的简单联系”。

集合 A 叫做定义域，集合 B 叫做陪域，对于某个 $a\in A$，与之对应的 $b\in B$ 叫做 a 的象，用 f(a)表示。

上面所说的联系都是一个实体(定单)的各个属性之间的联系。事实上，在实体集合之间也可以存在简单联系。譬如说，

$$职工=\{甲,乙\}$$

$$房间号=\{301,302,303\}$$

职工住房情况可以如下表示：

$$住房情况=\{(甲,301),(乙,302)\}$$

若表示成一种联系，则为

$$职工\longrightarrow房间号$$

在此，每个职工只在一个有序对当中出现，因此是 1 型联系。但这里并没有说明从房间号到职工的反向联系的情况。

如果用有序对的集合来表示，则为

$$住房情况=(职工,房间号)$$

显然是一个二元关系。

到目前为止，我们所说的都是从一个属性到另一个属性或从一个实体集合到另一个实体集合之间的联系，这个概念可以推广到多个属性情况中去。为此，我们看一表 1.1.2，这是一张不同的供应者供应零件，价格等情况的清单。

表 1.1.2.

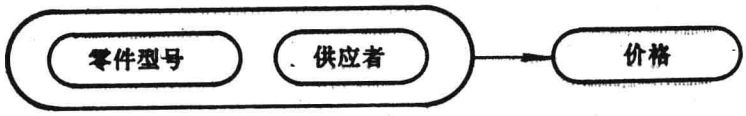
零件型号	供应者	包装量(个数/包)	价格(元/包)
101	S1	100	500
102	S2	10	70
103	S3	50	400
104	S4	20	200

可见，凡指出“零件型号和供应者”值的正确组合，就得到一个“包装量”的值，即“包装量”依赖于“零件型号和供应者”，这是一种简单联系或 1 型联系。可表示成图 1.1.2(a)形式

(a)

图 1.1.2(a)

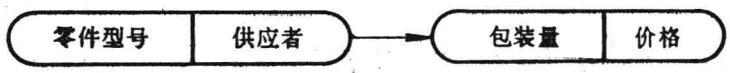
同理,“价格”也依赖于“零件型号和供应者”,如图 1.1.2(b)。



(b)

图 1.1.2(b)

此种情况,定义域是一个有序对的集合,陪域(或函数)是单个的属性的值。类似地,陪域(或函数),也可以是一个有序对,例如,“包装量和价格”的组合,于是有图 1.1.2(c)。



(c)

图 1.1.2(c)

当然这也是一种 1 型联系。因为,一般地说,1 型联系是定义域的每一个成员,在陪域中有且仅有一个成员(又称象)与之对应。

1.1.2 条件联系

〔定义〕如果集合 A 的每一个成员最多只和 B 中的一个成员有联系,或者也可能跟 B 的任何成员没联系,则称这种联系为条件联系(即 C 型联系)。记作

$$a:A\longrightarrow B$$

读做“a 是从 A 到 B 的条件联系”。

例如,图 1.1.3 说明:定义域的每个成员或与陪域一个成员有联系,或根本没有联系,即是一种 C 型联系。

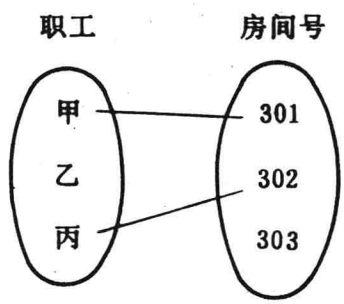


图 1.1.3

此时,

住房情况:职工——→房间号

就是一种C型联系。也可以有一个用序对的集合来表示住房情况:

{(甲,301),(丙,302)}

因为这里假定,一个职工最多住一个房间(号)(也许没住任何一间房子),所以从职工到房间号的联系是条件(或C型)联系。

1.1.3 复杂联系

〔定义〕如果集合A的每一个成员和B的若干个成员相联系(可能根本没有联系),则这种联系称为复杂联系,即M型联系。记作

a:A——→→B

读做"a是从A到B的复杂联系"。

上述定义的复杂联系,有时又称"对多"联系。例如,多个学生,学习多种课程,那么从"学生"到"课程"就是一种复杂联系。如图1.1.4所示。

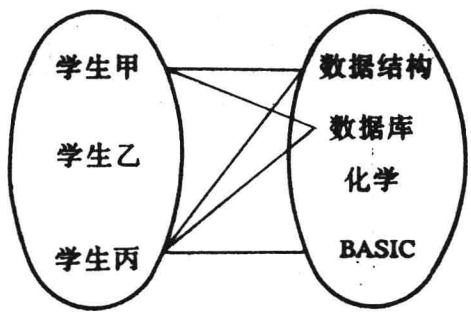


图1.1.4

于是,选修课程情况就可以用有序对的集合来表示:

选修情况={(学生甲,数据结构),(学生甲,数据库),(学生丙,数据结构),(学生丙,数据库),(学生丙,BASIC)}

简单地又可以表示成:

选修情况:学生——→→课程

若把这种复杂联系写成一种关系表达式,也可以写成为如下的关系:

选修情况(学生,课程)

它也是一个二元关系。

1.1.4 集合内的联系

一个集合内的某些成员同另外一些成员也可能有联系,这种联系叫做集合内的联系。例如,在一个公司内的职员间的隶属关系(图1.1.5):

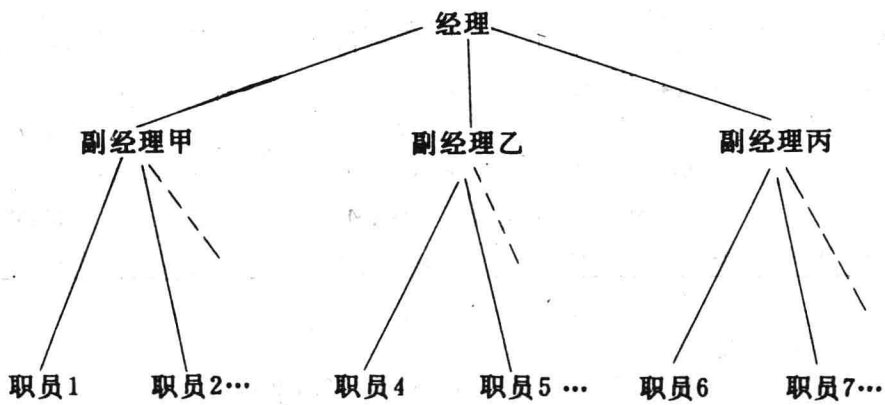


图 1.1.5

若用有序对来表示上图中的隶属关系，即表示“某人是某人的经理”，则有：

某人的经理＝{＜经理，副经理甲＞，＜经理，副经理乙＞，……，＜副经理乙，职员 4＞，＜副经理乙，职员 5＞，……}

如果我们把公司内的所有成员都视为“职员”，那么，图 1.1.5 中的职员的隶属结构有下面的复杂联系(即 M 型联系)：

某人的经理：职员——→→职员

1.1.5 映射

映射是数据库领域里经常使用的一个术语。映射是两个集合之间，定义域和陪域之间联系的规则。它们可以是也可以不是同一个集合。映射涉及到联系和反向联系。每种联系和每种反向联系都可能是 1 型的，或者是 C 型的，或者是 M 型的联系。因此，三种联系和三种反向联系就构成了 3×3＝9 种映射。

我们把两个集合 A 和 B 间的映射类型定义为：

(反向联系：正向联系)

用符号记作：

m：A　Tr——Tf　B

其中，Tr 和 Tf 表示联系类型，即 Tr 可以是←(1 型联系)，或(——(C 型联系)，或是←←(M 型联系)；同理，Tf 可以是→，或是—)，或是→→。例如，

m：A←←——→B

表示 A 和 B 间具有(M：1)型映射。读做“m 是 A 和 B 之间的‘多对一’的映射”。

注意，映射是对称的。即把分别在(1：C)，(1：M)，(C：M)映射中涉及到的集合的先后顺序颠倒过来，就可得到映射(C：1)，(M：1)和(M：C)。所以，在九种映射类型中，只有六种类型才是真正不同的。表 1.1.3 说明了这个道理。

表 1.1.3　　映射类型

a(正向联系)	a 的反向联系		
	1 型 a^{-1}:B→A	C 型 a^{-1}:B——)A	M 型 a^{-1}:B→→A
1 型: A→B	(1:1) m:A←——→B	(C:1) m:A(——→B	(M:1) m:A←←——→B
C 型 a:A—)B	(1:C) m:A←——)B	(C:C) m:A(———)B	(M:C) m:A←←——)B
M 型 a:A→→B	(1:M) m:A←——→→B	(C:M) m:A(——→→B	(M:M) m:A←←→→B

下面用例子加以说明。

〔例 1.1.1〕设有一个饭店,有营业用房间={101,102,201,202,301,302,…}。租出房间={101,201,301,302}。想来租房间的人(简记来客)={甲,乙,丙,丁,午,己,庚},已决定租房间的人(简称房客)={甲,乙,丙,丁}。现在分几种情况讨论。

(1)假定每个房间至多住一位房客,每个房客只能租一间房间。于是有映射:

住房情况(1):房客←→租出房间

表示(1:1)的映射。其结构也可以表示成(图 1.1.6):

图 1.1.6

(2)假定的条件同(1),那么看房客与营业性用房间之间的映射又怎样呢?从图 1.1.7 可以看到,有的房间根本没被租用,可见房客和营业性用房之间的映射是一种(C:1)型映射,即

住房情况(2):房客(——→营业性用房间

或者表示成(图 1.1.7):

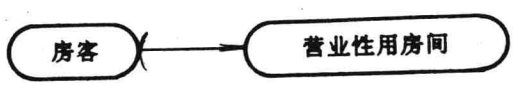


图 1.1.7