
关系模型

3.2.1 关系模型概述

关系模型是目前数据库系统普遍采用的数据模型，也是应用最广泛的数据模型。关系模型通过二维表表示实体以及实体之间的联系。二维表并不一定是关系模型，只有具备下列特点的二维表才是关系模型：

- (1) 表格中的每一列都是不可再分的基本数据项；
- (2) 列和列之间的名字不同，每一列的数据类型相同或者兼容；
- (3) 行和行的顺序无关，列和列的顺序也无关；
- (4) 关系中不能存在完全相同的两行。

3.2.2 将 E-R 图转换为关系模型

在关系型数据库中，实体和实体之间的联系表示为表。将 E-R 图转换为关系模型的方法是将一个实体或联系转换为表；将实体或联系的属性转换为表的列；实体标识符就是表的主关键字。

把 E-R 图转换为关系模型可遵循如下原则：

1、对于 E-R 图中每个实体集，都应转换为一个关系，该关系应包括对应实体的全部属性，并应根据关系所表达的语义确定哪个属性或哪几个属性组作为“主关键字”，主关键字用来标识实体。

2、对于 E-R 图中的联系，情况比较复杂，要根据实体联系方式的不同，采取不同的手段加以实现。

将 E-R 图转换为关系模型，要分以下几种情况。

第一种情况，(1:1) 联系的 E-R 图转换：

1、联系单独对应关系模式，则由联系属性、参与联系的各实体集的主码属性构成关系模式，其主码可选参与联系的实体集的任一方的主码。参考图 3-，可以得到如下关系。

BJ (班级编号, 院系, 专业名, 人数)

BZ (学号, 姓名)

SY (学号, 班级编号)

2、联系不单独对应关系模式，联系的属性及一方的主码加入另一方实体集对应的关系模式中。参考图 3-，可以得到如下关系。

BJ (班级编号, 院系, 专业名, 人数)

BZ (学号, 姓名, 班级编号)

或者

BJ (班级编号, 院系, 专业名, 人数, 学号)

BZ (学号, 姓名)

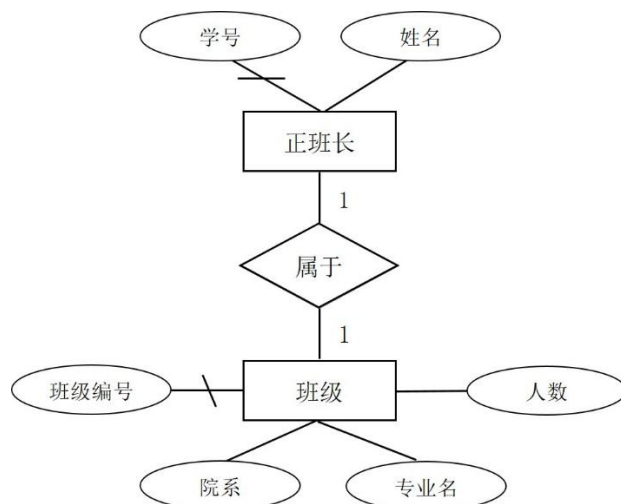


图 3-2 (1:1) 联系

第二种情况，(1:n) 联系的 E-R 图转换：

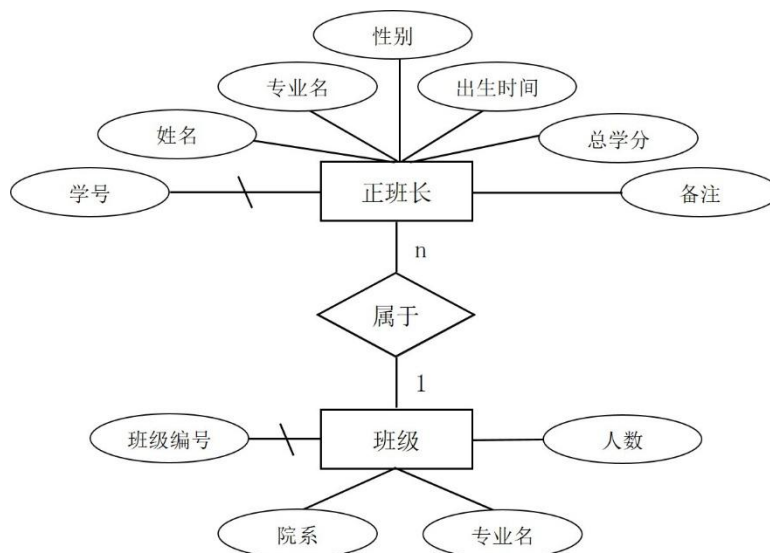


图 3-3 (1: n) 联系

1、联系单独对应关系模式，则由联系的属性、参与联系的各实体集的主码属性构成关系模式，n 端的主码作为该关系模式的主码。参考图 3-3，可以得到如下关系。

BJ (班级编号，院系，专业名，人数)

XS (学号，姓名，专业名，性别，出生时间，总学分，备注)

SY (学号，班级编号)

2、联系不单独对应一个关系模式，则将联系的属性及 1 端的主码加入 n 端实体集对应的关系模式中，主码仍为 n 端的主码。参考图 3-，可以得到如下关系。

BJ (班级编号，院系，专业名，人数)

XS (学号，姓名，专业名，性别，出生时间，总学分，备注，班级编号)

第三种情况，(m:n) 联系的 E-R 图转换：

对于 (m:n) 的联系，单独对应一关系模式，该关系模式包括联系的属性、参与联系的各实体集的主码属性，该关系模式的主码由各实体集的主码属性共同组成。参考图 3-，可

以得到如下关系。

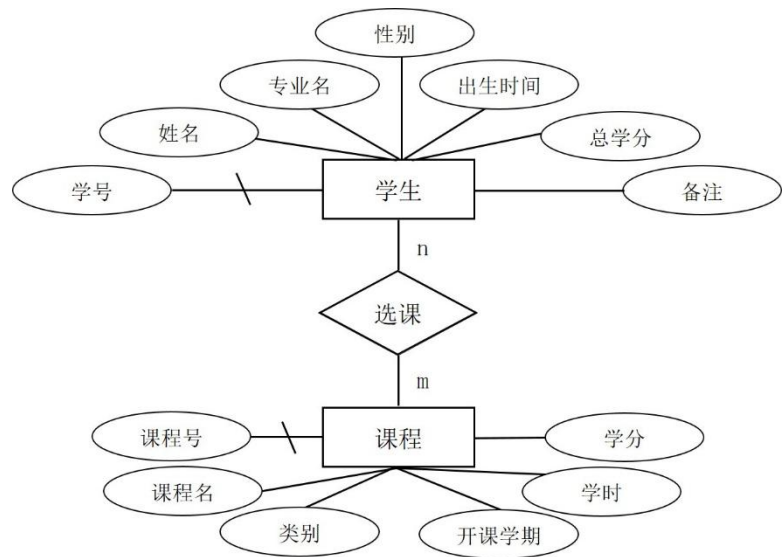


图 3-4 (m: n) 联系

XS(学号, 姓名, 专业名, 性别, 出生时间, 总学分, 备注),
KC(课程号, 课程名称, 类别, 开课学期, 学时, 学分),
XS_KC(学号, 课程号, 成绩)。

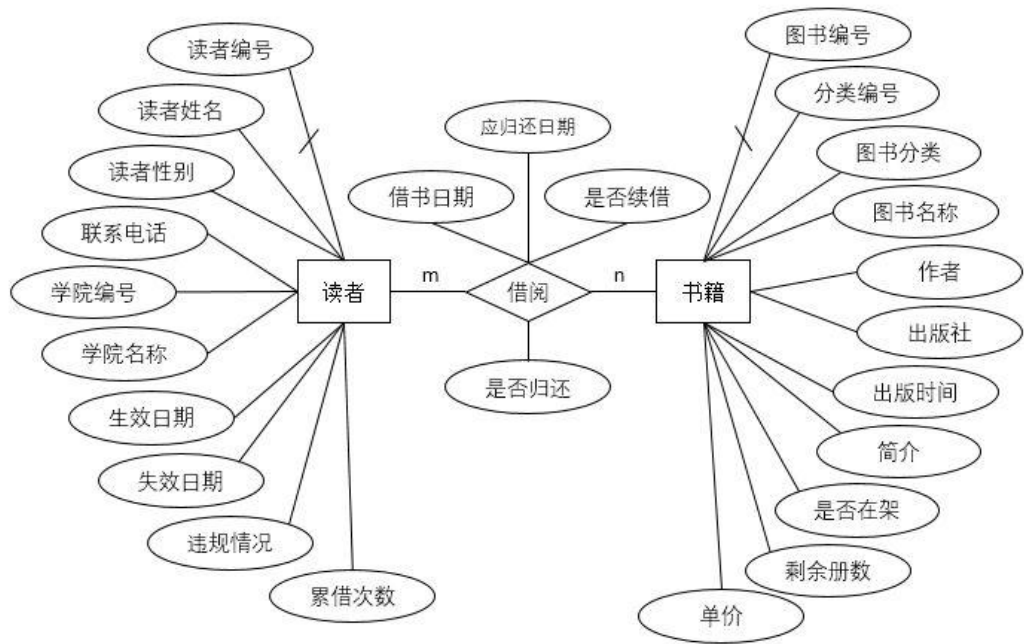


图 3-5 图书馆信息系统 E-R 图

E-R 模型到关系模型的转换总结：

假设 A 实体集与 B 实体集是 (1:1) 的联系，联系的转换有三种方法：

- 1、把 A 实体集的主关键字加入到 B 实体集对应的关系中，如果联系有属性也一并加入；
- 2、把 B 实体集的主关键字加入到 A 实体集对应的关系中，如果联系有属性也一并加入；

3、建立第三个关系，关系中包含两个实体集的主关键字，如果联系有属性也一并加入。

两实体集间（1:n）联系，联系的转换方法如下：

两实体集间（1:n）联系，可将“一方”实体的主关键字纳入“n方”实体集对应的关系中作为“外部关键字”，同时把联系的属性也一并纳入“n方”对应的关系中。

两实体集间（m:n）联系，联系的转换方法如下：

对于两实体集间（m:n）联系，必须对“联系”单独建立一个关系，用来联系双方实体集。该关系的属性中至少要包括被它所联系的双方实体集的“主关键字”，并且如果联系有属性，也要归入这个关系中。

根据图 3-5，图书馆信息系统则可以转换成如下关系：

读者（读者编号，读者姓名，读者性别，联系电话，学院编号，学院名称，生效日期，失效日期，违规情况，累借次数）；

书籍（图书编号，分类编号，图书分类，图书名称，作者，出版社，出版时间，简介，是否在架，单价，剩余册数）；

借阅（读者编号，图书编号，借书日期，应归还日期，是否续借，是否归还）。

3.2.3 关系规范化

设计关系数据库时，遵从不同的规范要求，设计出合理的关系型数据库，确保数据库的数据的可靠性和准确性。这些不同的规范要求被称为不同的范式，各种范式呈递次规范，越高的范式数据库冗余越小。

Dr E. F. Codd 最初定义了规范化的三个级别，范式是具有最小冗余的表结构。这些范式是：第一范式、第二范式和第三范式。目前关系数据库有六种范式：第一范式（1NF—First Normal Form）、第二范式（2NF）、第三范式（3NF）、巴斯-科德范式（BCNF）、第四范式（4NF）和第五范式（5NF，又称完美范式）。

关系规范化的目的是为了消除存储异常、减少数据冗余，以保证数据完整性（数据的正确性、一致性）和存储效率，一般将关系规范到第三范式即可。

3.2.4 函数依赖

在关系中，一个列中任意行的值能唯一地标识出其他列对应的值，则该列是主关键字，也被称为主属性，其他的列被称为非主属性。这种决定现象，被称为非主属性完全函数依赖于主属性。如果除主属性外，非主属性的部分列不能被主属性决定，被称为非主属性部分函数依赖于主属性。

在图书馆信息系统中，“读者”中的主属性是读者编号，它的值确定后，读者姓名、读者性别等其他属性都可以得到与之对应的确定值。即称读者姓名、读者性别等其他属性都完全函数依赖于读者编号。

3.2.5 函数传递依赖

在关系中，属性（列）A 决定了属性（列）B，属性（列）B 又能决定列属性（列）C，则存在属性（列）C 通过属性（列）B 而依赖于属性（列）A。这种情况，称属性（列）A 和属性（列）C 之间存在函数传递依赖。

例如，在图书馆信息系统中，“读者”中的主属性是读者编号，其他属性完全函数依赖于读者编号。观察读者编号和学院名称，是不是存在如果读者编号确定了，就知道对应的学院名称？而学院名称又可以被学院编号决定。现在，读者编号决定了学院编号，学院编号决

定了学院名称，读者编号和学院名称之间存在函数传递依赖。

同样的，在“书籍”中的主属性是图书编号，其他属性完全函数依赖于图书编号。再来观察图书编号和图书分类，当图书编号确定了，就可以知道对应的分类编号。分类编号决定了图书分类，图书编号和图书分类之间存在函数传递依赖。

3.2.6 范式定义

1、第一范式

所谓第一范式（1NF）是指在关系模型中，对于添加的一个规范要求，所有的域都应该是原子性的，即数据库表的每一列都是不可分割的原子数据项，而不能是集合，数组，记录等非原子数据项。即实体中的某个属性有多个值时，必须拆分为不同的属性。在符合第一范式（1NF）表中的每个域值只能是实体的一个属性或一个属性的一部分。简而言之，第一范式就是无重复的原子域。

在 3.2.2 中，图书馆信息系统则可以转换成读者、书籍和借阅三个关系。

如果在“读者”关系中，属性“联系电话”是下表的形式，则需要将“联系电话”进行进一步细分为两个属性。

表 3-1 现实世界中读者的联系电话可以有两种情况

	联系电话		
	手机	座机	
	

上述的各个关系中的各个属性都是不可再分的，所以上述关系符合第一范式。

2、第二范式

第二范式首先是第一范式，然后关系中的每一个非主属性完全函数依赖于主关键字，则该关系是第二范式。单个属性作为主关键字的情况比较简单，因为主关键字的作用就是能唯一标识表中的每一行，关系中的非主属性都能完全函数依赖于主关键字，所以这样的关系是第二范式。对于组合属性作为主关键字的那些关系，通常要判断每一个非主属性是完全函数依赖还是部分函数依赖于主关键字。

将非第二范式规范为第二范式的方法是：将部分函数依赖关系中的主属性和非主属性从关系中提取出来，单独构成一个关系；将关系中剩余的其他属性加上主关键字，构成关系。

当前的图书馆信息系统中的“读者”和“书籍”这个关系，非主属性都完全函数依赖于主属性，符合第二范式。以“读者”关系为例：

读者（读者编号，读者姓名，读者性别，联系电话，学院编号，学院名称，生效日期，失效日期，违规情况，累借次数）。

“读者”关系中，所有属性都不可再进行分割，因此符合第一范式；主属性是“读者编号”，该属性的值一旦确定，其他属性的值就能确定。即关系中的每一个非主属性完全函数依赖于主关键字。

“借阅”关系中，由于是读者和书籍两个实体集发生的业务联系而产生的，所以其主属性有读者和书籍两个实体中各自的主属性进行组合而成，构成组合主属性。其他的非主属性都完全函数依赖于组合主属性。但是，存在这样一个问题：按照目前的组合主属性，某个读者借阅某本书只能一次；如果借第二次，则读者编号和图书编号的组合要有 2 次，不能做主属性。因此，需要引入一个单独的借阅编号来做主属性。

借阅的关系则改成：

借阅（借阅编号，读者编号，图书编号，借书日期，应归还日期，是否续借，是否归还）。

第二范式的关系模式依然存在数据冗余、数据不一致的问题，需要进一步将其规范为第三范式。

3、第三范式

第三范式首先是第二范式，且关系中的任何一个非主属性都不函数传递依赖于主关键字，则此关系是第三范式。

在关系中，首先需要找出关系中的主关键字，然后判断任何一个非主属性和主关键字之间是否存在函数传递。如果存在，则要消除函数传递依赖关系。例如“读者”和“书籍”这两个关系，前面讨论过，存在传递依赖，需要进行消除。在“读者”关系中，将“学院编号”和“学院名称”提出来组成独立的关系，在“读者”关系中保留“学院编号”；在“书籍”关系中，将“分类编号”和“图书分类”提出来组成独立的关系，在“书籍”关系中保留“分类编号”；

读者（读者编号，读者姓名，读者性别，联系电话，学院编号，生效日期，失效日期，违规情况，累借次数），

学院（学院编号，学院名称）；

书籍（图书编号，分类编号，图书名称，作者，出版社，出版时间，简介，是否在架，剩余册数，单价），

图书分类（分类编号，图书分类）。

第三范式的关系中的数据基本独立，关系和关系之间通过外键进行联系，从根本上消除了数据冗余、数据不一致的问题。