目录

文档目录内容

表目录

表目录内容

图目录

图目录内容

关系模型和关系数据库

本章讨论的内容是数据库逻辑设计中所使用的逻辑数据模型，是一种数据库模型，称为数据模型。数据模型是一种用来表达数据的工具。在计算机中表示数据的数据模型应该能够精确地描述数据的静态特性、数据的动态特性和数据完整性约束条件。因此数据模型通常是由数据结构、数据完整性规则和数据操作三部分内容构成。

数据结构用于描述数据的静态特性。关系数据模型的数据结构是关系，一种符合一定规则的二维表格。

数据的完整性规则是一组约束条件的集合。以保证数据正确、有效和一致。

数据操作用于描述数据的动态特性。数据操作是指对数据库中各类对象的实例（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更新（包括插入、删除、修改）两大类操作。

数据模型

1970年美国IBM公司的研究员E. F.Codd首次提出了数据库系统的关系模型。在此之前，计算机中使用的数据模型有层次模型和网状模型，20世纪70年代以后，关系模型逐渐地取代了这两种数据模型。

层次数据模型

层次数据模型（hierarchical data model）的基本结构是一种倒挂树状结构（如图2.1所示）。这种树结构司空见惯，例如，Windows操作系统中的文件夹和文件结构、一个组织的结构等等。

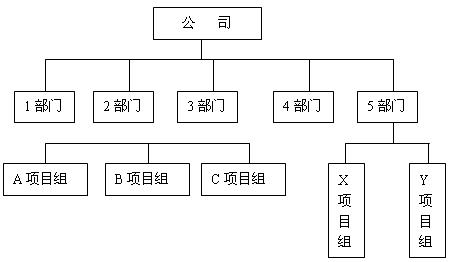


图2.1 层次数据模型示例

树结构具有如下的特征（或限制条件）：

1）有且仅有一个根结点，它是一个无父结点的结点。

2）除根结点以外的所有其他结点有且仅有一个父结点。

网状数据模型

取消层次数据模型的两个限制条件， 每一个结点可以有多个父结点便形成网状数据模型（network data model）。

关系数据模型

关系数据模型是一个满足一定条件的二维表格。 通俗地讲，满足关系模型的二维表格是个规则的二维表格，它的每一行是惟一的，每一列也是惟一的。在关系数据模型中，这样一个二维表格称为关系，表格的第一行是属性名，后续的每一行称为元组。每一列是一个属性，同一属性的取值范围相同。

关系模型的数据结构

一个关系模型的逻辑结构是一张二维表格，它由行和列组成，称之为关系，既关系是一个二维表格。在关系数据模型中，实体以及实体间的各种联系均用关系表示。下面介绍关系模型中使用的一些基本概念。

1．关系（relation）

一个二维表格。

2．属性（attribute）

表（关系）的每一列必须有一个名字称为属性。

3．元组（tuple）

表（关系）的每一行称为一个元组。

4．域（domain）

表（关系）的每一属性有一个取值范围，称为域。 域是一组具有相同数据类型的值的集合。

5．关键字（key）

又称主属性，可以惟一地标识一个元组（一行）的一个属性或多个属性的组合。可以起到这样作用的关键字有两类：候选关键字和主关键字。

１）候选关键字 （candidate key）

一个关系中可以惟一地标识一个元组（一行）的一个属性或多个属性的组合。一个关系中可以有多个候选关键字。

２）主关键字（primary key）

把关系中的一个候选关键字定义为主关键字。一个关系中只能有一个主关键字，用以惟一地标识元组，简称为关键字。

有的时候，关系中只有一个候选关键字，把这个候选关键字定义为主关键字后，关系中将没有候选关键字。

关系中不应该存在重复的元组（表中不能有重复的行），因此每个关系都至少有一个关键字。可能出现的一种极端情况是：关键字包含关系中的所有属性。

6．外部键（foreign key）

如果某个关系中的一个属性或属性组合不是所在关系的主关键字或候选关键字，但却是其他关系的主关键字，对这个关系而言，称其为外部关键字。

7．关系模式（relational schema）

关系模式是对关系数据结构的描述。 简记为：关系名（属性1，属性2，属性3，……属性n）。

表2.1是一个关系，关系名是仓库，此关系具有4个属性： 仓库号，仓库名，地点，面积。其关系模式是：仓库（仓库号，仓库名，地点，面积）。关系的关键字是仓库号，因此仓库号不能有重复值，同时不能为空。

表2.1 “仓库”关系

|  |  |  |  |
| --- | --- | --- | --- |
| 仓库号 | 仓库名 | 地点 | 面积 |
| WH1 | 兴旺 | 上海 | 390 |
| WH2 | 广发 | 长沙 | 460 |
| WH3 | 红星 | 昆明 | 500 |
| WH4 | 奥胜 | 兰州 | 280 |
| WH5 | 高利 | 长春 | 300 |
| WH6 | 中财 | 北京 | 600 |

综上所述，可以得出如下结论：

1）一个关系是一个二维表格。

2）二维表格的每一列是一个属性。每一列有惟一的属性名。属性在表中的顺序无关紧要。

3）二维表格的每一列数据的数据类型相同，数据来自同一个值域。不同列的数据也可以来自同一个值域。

4）二维表格中每一行（除属性名行）是一个元组，表中不能有重复的元组（元组是惟一的），用关键字（主关键字和候选关键字）来保证元组的惟一性，例如表2.1中的“仓库号”。元组在表中的顺序无关紧要。

同一事物在现实世界、信息世界和计算机世界中的称谓有所不同，对应关系如表2.2所示。

表2.2 数据模型中概念之间的对应关系

|  |  |  |  |
| --- | --- | --- | --- |
| 概念模型 | 关系模型 | DBMS | 用户 |
| 实体集 | 关系 | 数据库表 | 二维表格 |
| 实体 | 元组 | 记录 | 行 |
| 属性 | 属性 | 字段 | 列 |
| 键 | 关键字（主属性） | 主关键字 |  |
| 实体型 | 关系模式 |  |  |

关系数据库和关系数据库规范化

关系数据库

关系数据库是以关系模型为基础的数据库， 它利用关系描述现实世界中的对象。一个关系既可用来描述一个实体及其属性，也可用来描述实体间的联系。关系数据库是由一组关系组成的，针对一个具体问题，应该如何构造一个适合于它的数据模式，即应该构造几个关系？每个关系由那些属性组成？这就是关系数据库逻辑设计要研究的问题。

关系数据库规范化

关系数据库规范化（Normal Form）的目的是建立正确、合理的关系，规范化的过程是一个分析关系的过程。

实际上设计任何一种数据库应用系统，不论是层次、网状或关系， 都会遇到如何构造合适的数据模式即逻辑结构问题。由于关系模型有严格的数学理论基础，并且可以向其他数据模型转换，因此人们往往以关系模型为背景来讨论这一问题，形成了数据库逻辑设计的一个有力工具—关系数据库规范化理论。

函数依赖及其对关系的影响

函数依赖是属性之间的一种联系，普遍存在于现实生活中。例如，银行通过客户的存款帐号，可以查询到该帐号的余额。又例如，表2.3是描述学生情况的关系，用一种称为关系模式的形式表示为：

STUDENT1(学号，姓名，性别，出生日期，专业)

由于每个学生有惟一的学号，一个学号只对应一位学生，一个学生只就读于一个专业，因此当学号的值确定之后，姓名及其所就读专业的值也就被唯一地确定了。属性间的这种依赖关系类似于数学中的函数。因此称帐号函数决定 账户余额，或者称帐户余额函数地依赖于帐号；学号函数决定姓名和专业，或者说姓名和专业函数依赖于学号，记作：学号→姓名，学号→专业；同样有学号→性别，学号→出生日期。

表2.3 关系STUDENT1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 出生日期 | 专业 |
| 010001 | A | F | 01/01/82 | 会计 |
| 010002 | B | F | 04/11/83 | 注会 |
| 010003 | C | M | 05/18/81 | 会计 |
| 010004 | D | F | 09/12/82 | 会计 |

如果在关系STUDENT1的基础上增加一些信息，例如学生的“学院”及“院长”信息（如表2.4所示），有可能设计出如下关系模式：

STUDENT2(学号，姓名，性别，出生日期，专业，学院，院长)。

函数依赖关系是：学号→学院、学院→院长。

表2.4 关系STUDENT2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 出生日期 | 专业 | 院系 | 院长 |
| 010001 | A | F | 01/01/82 | 会计 | 会计 | Z |
| 010002 | B | F | 04/11/83 | 注会 | 会计 | Z |
| 010003 | C | M | 05/18/81 | 会计 | 会计 | Z |
| 010004 | D | F | 09/12/82 | 会计 | 会计 | Z |
| 010005 | E | M | 12/12/83 | 信管 | 信息 | W |
| 010006 | F | F | 10/11/82 | 信管 | 信息 | W |

上述关系模式存在如下四个问题：

1）数据冗余太大。例如院长的姓名会重复出现， 重复的次数与该学院学生的个数相同。

2）更新异常（update anomalies）。例如某学院更换院长后，系统必须修改与该学院学生有关的每一个元组。

3）插入异常（insertion anomalies）。如果一个学院刚成立，尚无学生，则这个学院及其院长的信息就无法存入数据库。

4）删除异常（deletion anomalies）。如果某个学院的学生全部毕业了，在删除该学院学生信息的同时，也把这个学院的信息全部删除了。

一个关系之所以会产生上述问题，是由于关系中存在某些函数依赖引起的。通常，当企图把太多的信息放在一个关系里时，出现的诸如冗余之类的问题称为“异常”（anomaly）。经常遇到的几种异常是：

1）冗余：数据在多个元组中不必要地重复。

2）更新异常：由于数据冗余，当更新数据库中数据时，系统要付出很大的代价来维护数据库的完整性。否则会面临数据不一致的危险。可能修改了一个元组中信息，但另一个元组中相同的信息却没有修改。

3）删除异常：如果删除一组属性，带来的副作用可能是丢失了一些其他信息。

规范化是为了设计出“好的”关系模型。规范化理论正是用来改造关系模式，通过分解关系模式来消除其中不合适的数据依赖，以解决更新异常、插入异常、删除异常和数据冗余问题。

规范化的本质

每个规范化的关系只有一个主题。如果某个关系有两个或多个主题，就应该分解为多个关系，每个关系只能有一个主题。规范化的过程就是不断分解关系的过程。

人们每发现一种异常，就研究一种规则防止异常出现。由此设计关系的准则得以不断改进。70年代初期，研究人员系统地定义了第一范式（Fist Normal Forms， 1NF），第二范式（Second Normal Form，2NF）和第三范式（Third Normal Form，3NF）。之后人们又定义了多种范式，但大多数简单业务数据库设计中只需要考虑第一范式、第二范式和第三范式。每种范式自动包含其前面的范式，各种范式之间的关系是：5NF ⊂ 4NF ⊂ BCNF ⊂ 3NF ⊂ 2NF ⊂ 1NF。因此符合第三范式的数据库自动符合第一、第二范式。

1）1NF

关系模式都满足第一范式，既符合关系定义的二维表格（关系）都满足第一范式。列的取值只能是原子数据；每一列的数据类型相同，每一列有惟一的列名（属性）；列的先后顺序无关紧要，行的先后顺序无关紧要。

2）2NF

关系的每一个非关键字属性都完全函数依赖于关键字属性， 则关系满足第二范式。

第二范式要求每个关系只包含一个实体的信息，所有非关键字属性依赖关键字属性。每个以单个属性作为主键的关系自动满足第二范式。

3）3NF

关系的所有非关键字属性相互独立，任何属性其属性值的改变不应影响其他属性，则该关系满足第三范式。一个关系满足第二范式，同时没有传递依赖，则该关系满足第三范式。

由1NF、2NF和3NF，总结出规范化的规则如下：

⑴ 每个关系只包含一个实体集，每个实体集只有一个主题

⑵ 每个关系有一个主键

⑶ 属性中只包含原子数据

⑷ 不能有重复属性

每个规范化的关系只有一个主题。如果某个关系有两个或多个主题，就应该分解为多个关系。规范化的过程就是不断分解关系模式的过程。经过不断地总结，人们归纳出规范化的规则如下：

⑴ 每个关系只包含一个实体集；每个实体集只有一个主题， 一个实体集对应一个关系；

⑵ 属性中只包含原子数据既最小数据项；每个属性具有数据类型并取值于同一个值域；

⑶ 每个关系有一个主关键字， 用来惟一地标识关系中的元组；

⑷ 关系中不能有重复属性；所有属性完全依赖关键字（主关键字或候选关键字）；所有非关键字属性相互独立；

⑸ 元组的顺序无关；属性的顺序无关。

关系数据完整性规则

关系模型的完整性规则是对关系的某种约束条件。关系模型中的数据完整性规则包括：实体完整性规则、域完整性规则、参照完整性规则和用户定义完整性规则。

实体完整性规则是指保证关系中元组惟一的特性。通过关系的主关键字和候选关键字实现。

域完整性规则是指保证关系中属性取值正确、有效的特性。例如，定义属性的数据类型、设置属性的有效性规则。

参照完整性与关系之间的联系有关，包括插入规则、删除规则和更新规则。

用户定义完整性规则是指为满足用户特定需要而设定的规则。

在关系数据完整性规则中，实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称为是关系的两个不变性，由关系系统自动支持。

在以后的章节中将结合具体实例对数据库的数据完整性规则进行详细讨论。

E-R模型向关系模型的转换

E-R模型向关系模型转换要解决的问题是如何将实体以及实体之间的联系转换为关系模式，如何确定这些关系模式的属性和主关键字（这里所说的实体更确切地说是实体集）。注意，这里包含两个方面的内容，一是实体如何转换，二是实体之间的联系如何处理。

实体转换为关系模式

E-R模型的表现形式是E-R图，由实体、实体的属性和实体之间的联系三个要素组成。从E-R图转换为关系模式的方法是：为每个实体定义一个关系，实体的名字就是关系的名字；实体的属性就是关系的属性；实体的键是关系的主关键字。用规范化准则检查每个关系，上述设计可能需要改变，也可能不要改变。依据关系规范化准则，在定义实体时就应遵循每个实体只有一个主题的原则。实体之间的联系转换为关系之间的联系，关系之间的联系是通过外部关键字来体现的。

实体之间联系的转换

前面讨论过实体之间的联系通常有三种类型：一对一联系（1∶1）、一对多联系（1∶n）和多对多联系（m∶n）。下面从实体之间联系的角度来讨论三种常用的转换策略。

一对一联系的转换

两个实体之间的联系最简单的形式是一对一（1∶1）联系。1∶1联系的E-R模型转换为关系模型时，每个实体用一个关系表示，然后将其中一个关系的关键字置于另一个关系中，使之成为另一个关系的外部关键字。关系模式中带有下划线的属性是关系的主关键字。

**例2.1** 本例的需求分析和E-R模型见第1章例1.2。

根据转换规则，公司实体用一个关系表示；实体的名字就是关系的名字，因此关系名是“公司”；实体的属性就是关系的属性，实体的键是关系的关键字，由此得到关系模式：

公司（公司编号，公司名称，地址，电话）

同样可以得到关系模式：

总经理（经理编号，姓名，性别，出生日期，民族）

为了表示这两个关系之间具有一对一联系，可以把“公司”关系的关键字“公司编号”放入“总经理”关系，使“公司编号”成为“总经理”关系的外部关键字；也可以把“总经理”关系的关键字“经理编号”放入“公司”关系， 由此得到下面两种形式的关系模式。

关系模式一：

公司（公司编号，公司名称，地址，电话）

总经理（经理编号，姓名，性别，出生日期，民族，*公司编号*）

关系模式二：

公司（公司编号，公司名称，地址，电话，*经理编号*）

总经理（经理编号，姓名，性别，出生日期，民族）

其中斜体为外部关键字。

一对多联系的转换

一对多（1∶n）联系的E-R模型中，通常把“1”方（一方）实体称为“父”方，“n”方（多方）实体称为“子”方。1∶n联系的表示简单而且直观。一个实体用一个关系表示，然后把父实体关系中的关键字置于子实体关系中，使其成为子实体关系中的外部关键字。

**例2.2** 本例的需求分析和E-R模型见第1章例1.4。

在这个E-R模型中，仓库实体是“一方”父实体，员工实体是“多方”子实体。每个实体用一个关系表示，然后把仓库关系的主关键字“仓库号”放入员工关系中，使之成为员工关系的外部关键字。于是得到关系模式。

关系模式：

仓库（仓库号，仓库名，地点，面积）

员工（员工号，姓名，性别，出生日期，工资，*仓库号*）

**例2.3** 考虑学生毕业设计中的指导教师和学生的情况。

1）需求分析

学校使用数据库来管理学生毕业设计时的教师和学生数据。毕业设计时，一名教师指导多位学生，每位学生必须有一名教师指导其毕业设计论文。

2）E-R模型

E-R图如图2.2所示。实体型如下：

教师（教师号，姓名，院系，电话）

学生（学号，姓名，性别，出生日期，所属院系）

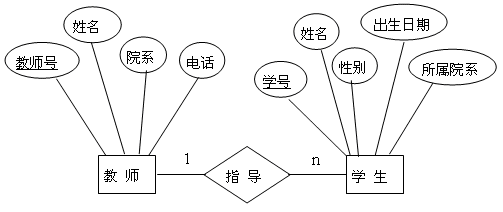


图2.2 教师实体与学生实体的一对多联系

3）关系模型的表示方法关系模式如下：

教师（教师号，姓名，院系，电话）

学生（学号，姓名，性别，出生日期，所属院系，*教师号*）

注意：1∶n联系的E-R数据模型转换为关系数据模型时，一定是父实体关系中的关键字置于子实体关系中。反之不可。

多对多联系的转换

多对多（m∶n）联系的E-R数据模型转换为关系数据模型的转换策略是把一个m∶n联系分解为两个1∶n联系，分解的方法是建立第三个关系（称为“纽带”关系）。原来的两个多对多实体分别对应两个父关系，新建立第三个关系，作为两个父关系的子关系，子关系中的必有属性是两个父关系的关键字。

**例2.4** 学生和社团问题。需求分析和E-R模型见第1章例1.5。

1）对应社团实体和学生实体分别建立社团关系和学生关系

社团（编号，名称，地点，电话）

学生（学号，姓名，性别，出生日期，所属院系）

2）建立第三个关系表示社团关系与学生关系之间具有m∶n联系

为了表示社团关系和学生关系之间的联系是多对多联系， 建立第三个关系“成员”，把“社团”关系和“学生”关系的主关键字放入“成员”关系中，用关系“成员”表示“社团”关系与“学生”关系之间的多对多联系。“成员”关系的主关键字是编号+学号，同时编号和学号又是这个关系的外部关键字。

成员（编号，学号）

综上所述得到的关系模型的关系模式：

社团（编号，名称，地点，电话）

学生（学号，姓名，性别，出生日期，所属院系）

成员（编号，学号）

上述转换过程实际上是把一个多对多联系拆分为两个一对多联系。社团关系与成员关系是一个1∶n联系；学生关系与成员关系也是一个1∶n联系。成员关系有两个父关系：社团和学生，同样成员关系同时是学生和社团关系的子关系。子关系的关键字是父关系关键字的组合“编号+学号”；学号和编号又分别是子关系的两个外部关键字。

**例2.5** 学生与选修课程之间的情况。每个学生会选择多门课程，每门课程也对应多名学生选修。需求分析和E-R模型见第1章例1.6和图1.9。

转换多对多（m∶n）联系的策略是首先为学生实体和课程实体分别建立对应的关系，然后建立第三个关系“学生成绩”，用第三个关系表示“学生”与“课程”之间多对多的联系。第三个关系“学生成绩”中必须具有的属性是学生关系的关键字“学号”和课程关系的关键字“课程号”。根据具体情况还可能有其他属性，例如成绩。由此得到关系模式：

学生（学号，姓名，性别，出生日期，院系）

课程（课程号，课程名，开课单位，学时数，学分）

学生成绩（学号，课程号，成绩）

上述转换过程也是把一个多对多联系拆分为两个一对多联系。 学生关系与学生成绩关系是一个1∶n联系；课程关系与学生成绩关系也是一个1∶n联系。学生成绩关系有两个父关系：学生和课程，同样学生成绩关系同时是学生和课程的子关系。子关系的关键字是父关系关键字的组合“学号+课程号”；学号和课程号又分别是子关系的两个外部关键字。

综上所述，E-R模型中的联系转换为关系数据模型的方法如表2.5所示。

表2.5 E-R模型中联系转换为关系的方法

|  |  |
| --- | --- |
| 联系类型 | 方法描述 |
| 1∶1 | 一个关系的主关键字置于另一个关系中。 |
| 1∶n | 父关系（一方）的主关键字置于子关系（多方）中。 |
| m∶n | 分解成两个1∶n关系。  建立“纽带关系”，两个父关系的关键字置于纽带关系中，纽带关系是两个父关系的子关系。 |

多元联系E-R模型转换为关系模型

**例2.6** 仓库—员工—订单—供应商。 需求分析和E-R模型见第1章例1.7和图1.10。

本例中E-R数据模型转换为关系数据模型的步骤如下。

1）首先为每个实体建立与之相对应的关系

仓库（仓库号，仓库名，地点，面积）

员工（员工号，姓名，性别，出生日期，婚否，工资）

订单（订购单号，订购日期，金额）

供应商（供应商号，供应商名，地址）

2）分别处理每两个关系之间的联系

⑴ 仓库关系与员工关系之间具有一对多联系（见E-R模型），应该把仓库关系（父关系）的关键字“仓库号”放入员工关系（子关系），员工关系有了外部关键字“仓库号”，以此表示仓库关系与员工关系之间1∶n的联系。

⑵ 员工关系与订单关系之间同样具有一对多联系，员工关系的关键字“员工号”放入订单关系，使订单关系有外部关键字“员工号”，以此表示员工关系与订单关系之间1∶n的联系。

⑶ 供应商关系与订单关系之间也具有一对多联系，供应商关系的关键字“供应商号”放入订单关系，使订单关系有外部关键字“供应商号”，以此表示供应商关系与订单关系之间1∶n的联系。

综上所述，得到关系数据模型：

仓库（仓库号，仓库名，地点，面积）

员工（员工号，姓名，性别，出生日期，婚否，工资，仓库号）

订单（订购单号，订购日期，金额，*员工号*，*供应商号*）

供应商（供应商号，供应商名，地址）

关系数据操作基础

关系是集合，关系中的元组可以看作是集合的元素。 因此，能在集合上执行的操作也能在关系上执行。

关系代数是一种抽象的查询语言，是关系数据操纵语言的一种传统表达方式，它是用对关系的运算来表达查询的。关系代数是封闭的，也就是说一个或多个关系操作的结果仍然是一个关系。关系运算分为传统的集合运算和专门的关系运算。

集合运算

传统的集合运算包括并、差、交、广义笛卡尔积四种运算。

设关系A和关系B都具有n个属性，且相应属性值取自同一个值域，则可以定义并、差、交和积运算如下。

1. 并运算

|  |
| --- |
| 公式1 |

关系A和关系B的并是指把A的元组与B的元组加在一起构成新的关系C。元组在C中出现的顺序无关紧要，但必须去掉重复的元组。 既关系A和关系B并运算的结果关系C由属于A和属于B的元组构成，但不能有重复的元组，并且仍具有n个属性。关系A和关系B并运算记作A∪B或A+B。

2. 差运算

|  |
| --- |
| 公式2 |

关系A和关系B差运算的结果关系C仍为n目关系，由只属于A而不属于B的元组构成。关系A和关系B差运算记作A-B。注意，A-B与B-A的结果是不同的。

3. 交运算

|  |
| --- |
| 公式3 |

关系A和关系B交运算形成新的关系C，关系C由既属于A同时又属于B的元组构成并仍为n个属性。关系A和关系B交运算记作A∩B。

**例2.7** 设有关系R1和R2。

|  |  |
| --- | --- |
| R1中是K社团学生名单。 | R2中是L社团学生名单。 |
| |  |  |  | | --- | --- | --- | | 关系R2 | | | | 学号 | 姓名 | 性别 | | 001 | A | F | | 101 | C | F | | 909 | E | M | | |  |  |  | | --- | --- | --- | | 关系R2 | | | | 学号 | 姓名 | 性别 | | 001 | A | F | | 101 | C | F | | 909 | E | M | |

（1）R1+R2的结果是K社团和L社团学生名单。

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 性别 |
| 001 | A | F |
| 008 | B | M |
| 101 | C | F |
| 600 | D | M |
| 909 | E | M |

（2）R1-R2的结果是只参加K社团而没有参加L社团的学生名单（比较R2-R1）。

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 性别 |
| 008 | B | M |
| 600 | D | M |

（3）R1∩R2的结果是同时参加了K社团和L社团的学生名单。

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 性别 |
| 001 | A | F |
| 101 | C | F |

4．积运算

如果关系A有m个元组，关系B有n个元组，关系A与关系B的积运算是指一个关系中的每个元组与另一个关系中的每个元组相联接形成新的关系C。 关系C中有m×n个元组。关系A和关系B积运算记作A×B。

关系运算

专门的关系操作包括投影、选择和联接。

1．投影

投影操作是指从一个或多个关系中选择若干个属性组成新的关系。投影操作取得垂直方向上关系的子集（列），既投影是从关系中选择列。投影可用于变换一个关系中属性的顺序。

2．选择

选择操作是指从关系中选择满足一定条件的元组。选择操作取得的是水平方向上关系的子集（行）。

**例2.8** 关系的投影和选择操作。

student关系内容如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 出生日期 | 党员否 | 出生地 |
| 993501438 | 刘昕 | 女 | 02/28/81 | .T. | 北京 |
| 993501437 | 颜俊 | 男 | 08/14/81 | .F. | 山西 |
| 993501433 | 王倩 | 女 | 01/05/80 | .F. | 黑龙江 |
| 993506122 | 李一 | 女 | 06/28/81 | .F. | 山东 |
| 993505235 | 张舞 | 男 | 09/21/79 | .F. | 北京 |
| 993501412 | 李竟 | 男 | 02/15/80 | .F. | 天津 |
| 993502112 | 王五 | 男 | 01/01/79 | .T. | 上海 |
| 993510228 | 赵子雨 | 男 | 06/23/81 | .F. | 河南 |

从student关系中选择部分属性构成如下新的关系，此操作称为投影：

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 出生日期 | 出生地 |
| 993501438 | 刘昕 | 02/28/81 | 北京 |
| 993501437 | 颜俊 | 08/14/81 | 山西 |
| 993501433 | 王倩 | 01/05/80 | 黑龙江 |
| 993506122 | 李一 | 06/28/81 | 山东 |
| 993505235 | 张舞 | 09/21/79 | 北京 |
| 993501412 | 李竟 | 02/15/80 | 天津 |
| 993502112 | 王五 | 01/01/79 | 上海 |
| 993510228 | 赵子雨 | 06/23/81 | 河南 |

从student关系中选择部分元组构成如下新的关系，此操作称为选择：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 出生日期 | 党员否 | 出生地 |
| 993501437 | 颜俊 | 男 | 08/14/81 | .F. | 山西 |
| 993505235 | 张舞 | 男 | 09/21/79 | .F. | 北京 |
| 993501412 | 李竟 | 男 | 02/15/80 | .F. | 天津 |
| 993502112 | 王五 | 男 | 01/01/79 | .T. | 上海 |
| 993510228 | 赵子雨 | 男 | 06/23/81 | .F. | 河南 |

3．联接

联接操作是两个关系的积、选择和投影的组合。有等值联接和自然联接。

在以后的章节中将结合具体实例讨论与关系数据操作有关的命令。