

# 第2章 关系数据库



# 主要内容

2.1 关系模型概述

2.2 关系的完整性

2.3 关系操作

2.4 关系代数

2.5 小结



# 关系模型

定义：用二维表格结构来表示实体及实体之间联系的数据模型称为关系模型。

## 数据结构及有关概念

关系模型的数据结构是一张规范化的二维表，它有表名、表头、表体。数据操作是集合操作，**操作对象和操作结果都是关系。**

部门关系

部门号	名称	地址	电话
011	市场部	公司办201	871
012	人事部	公司办202	781
014	财务部	公司办203	520
015	公司办	公司办204	356
...	...	...	...



- .....
- (1) 理解关系的数学定义(域、笛卡尔积、关系)及性质。
  - (2) 理解关系的键及三类完整性规则。
  - (3) 重点掌握关系代数中的各种运算（包括并、交、差、笛卡尔积及选择、投影、连接、除），能够熟练地对关系进行各种代数运算。



## 2.1关系模型的数据结构及其形式化定义

关系模型的数据结构只包含单一的数据结构——关系。在关系模型中，无论是实体还是实体间的联系，均由单一的结构——关系来表示。而关系模型是以集合代数理论为基础的，因此，用集合代数给出“关系”的形式化定义。

关系的形式化定义及其有关概念

1. 域 (Domain)

2. 笛卡尔积 (Cartesian Product)

3. 关系 (Relation)



## 2.1.1 关系的形式化定义及其有关概念

### 1. 域

域是一组具有相同数据类型的值的集合，又称值域，用**D**表示。例如整数、实数、**{0, 1, 2}**、**{a, b, c}**等都可以是域。域中所包含的值的个数称为域的基数，用**m**表示。在关系中用域来表示属性的取值范围。

例如：在教师关系中

姓名域 **D1={李力, 王平, 刘伟}**, **m=3**;

性别域 **D2={男, 女}={女, 男}**, **m=2**; 域中的值无排列次序。



## 2.笛卡尔积

给定一组域 $D_1, D_2, \dots, D_n$ , 这些域中可以存在相同的域。定义 $D_1, D_2, \dots, D_n$ 的笛卡尔积为:

$$D_1 \times D_2 \times \dots \times D_n = \{ (d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n \}.$$

由定义可见, 笛卡尔积也是一个集合, 其中:

(1) 每一个元素 $(d_1, d_2, \dots, d_n)$ 叫做一个 $n$ 元组或简称为元组。元素中的每一个值 $d_i$  ( $i=1, \dots, n$ ) 叫做一个分量,  $d_i$ 来自相应的域 $D_i$  ( $d_i \in D_i$ )。 (2) 若 $D_i$

( $i=1, \dots, n$ ) 为有限集, 笛卡尔积的基数 $M$ 就是构成该积的所有域的基数的乘积。即

$$M = \prod_{i=1}^n |D_i|$$



## 2.笛卡尔积

例如，上述教师关系中，姓名、性别两个域的笛卡尔积为： $D_1 \times D_2 = \{(李力, 男), (李力, 女), (王平, 男), (王平, 女), (刘伟, 男), (刘伟, 女)\}$ 。

其中(李力, 男)等是元组，李力、王平、刘伟、男、女都是分量，笛卡尔积基数 $M=3 \times 2=6$ ，即元组的个数为6。

(3) 笛卡尔积可用二维表的形式表示。

表 2.1 $D_1$ 和 $D_2$ 的笛卡尔积	
姓 名	性 别
李力	男
李力	女
王平	男
王平	女
刘伟	男
刘伟	女





# 笛卡尔积（续）

例如，给出**3**个域：

❖ **D1=导师集合SUPERVISOR= { 张清玫, 刘逸 }**

❖ **D2=专业集合SPECIALITY= { 计算机专业, 信息专业 }**

❖ **D3=研究生集合POSTGRADUATE= { 李勇, 刘晨, 王敏 }**

❖ **D1, D2, D3的笛卡尔积为**



# 笛卡尔积（续）

❖  $D1 \times D2 \times D3 = \{$

(张清玫, 计算机专业, 李勇), (张清玫, 计算机专业, 刘晨),  
(张清玫, 计算机专业, 王敏), (张清玫, 信息专业, 李勇),  
(张清玫, 信息专业, 刘晨), (张清玫, 信息专业, 王敏),  
(刘逸, 计算机专业, 李勇), (刘逸, 计算机专业, 刘晨),  
(刘逸, 计算机专业, 王敏), (刘逸, 信息专业, 李勇),  
(刘逸, 信息专业, 刘晨), (刘逸, 信息专业, 王敏) }

❖ 基数为  $2 \times 2 \times 3 = 12$



### 3.关系

笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的任一个子集称为定义在域 $D_1, D_2, \dots, D_n$ 上的一个 $n$ 元关系。表示为:

$$R(D_1, D_2, \dots, D_n)$$

其中,  $R$ 表示关系的名称,  $n$ 称为关系的目或度。如姓名、性别两个域的笛卡尔积 $D_1 \times D_2$ 的某个子集可构成教师关系 $T_1$ 。

表 2.2  $D_1 \times D_2$  笛卡尔积的子集 (关系  $T_1$ )

姓 名	性 别
李力	男
王平	女
刘伟	男



## 对关系的几点说明:

在数学上, 关系是笛卡尔积的任意子集, 但在实际应用中, 关系为笛卡尔积中所取的有意义的子集。如选取表**2.1**的一个子集构成表**2.4**, 表中关系显然不符合实际。

表 2.1 $D_1$ 和 $D_2$ 的笛卡尔积	
姓 名	性 别
李力	男
李力	女
王平	男
王平	女
刘伟	男
刘伟	女

表 2.4 不符合实际意义的关系	
姓 名	性 别
李力	男
李力	女



### 对关系的几点说明:

(1) 当 $n=1$ 时, 称该关系为单元关系; 当 $n=2$ 时, 称该关系为二元关系; 教师关系**T1**为二元关系。

(2) 关系中的每个元素( $d_1, d_2, \dots, d_n$ )称为关系的一个元组, 通常用**t**表示, 关系中元组的个数是关系的基数。如教师关系**T1**中有三个元组, 其基数为**3**。如果一个关系的元组的个数是无限的, 则称为**无限关系**; 如果一个关系的元组的个数是有限的, 则称为**有限关系**。由于计算机存储系统的限制, 一般不处理无限关系, 只考虑有限关系。

表 2.2 $D_1 \times D_2$ 笛卡尔积的子集 (关系 $T_1$ )	
姓 名	性 别
李力	男
王平	女
刘伟	男





## 2.1.2 关系的性质

- (1) 列是同质的，即每一列中的分量是同一类型的数据，来自同一个域。
- (2) 不同的列可出自同一个域，其中的每一列称为一个属性，要给予不同的属性名。
- (3) 列的次序可以任意交换。
- (4) 行的次序可以任意交换。
- (5) 任意两个元组不能完全相同。
- (6) 所有属性值都是原子，不允许属性又是一个二维关系。或者说所有属性值是一个确定的值，而不是值的集合。

表 2.5 一个关系的两个属性来自同一域		
姓 名	职 业	兼 职
张强	教师	辅导员
王丽	工人	教师
刘宁	教师	辅导员



## 2.1.3 关系模式

在数据库中要区分型和值。在关系数据库中，关系模式是型，关系是值。关系模式是对关系的描述。一个关系模式应当是一个五元组。它可以形式化表示为：

$$R = (U, D, DOM, F)$$

其中：**R**为关系名，**U**为组成该关系的属性名集合，**D**为属性组**U**中属性所来自的域，**DOM**为属性向域的映像集合，**F**为属性间数据的依赖关系集合。

属性的取值范围和属性到域的映像常常直接说明为属性的类型、长度；属性间的依赖关系**F**在后边介绍。因此，关系模式通常简记为：

$$R(U) \text{ 或 } R(A_1, A_2, \dots, A_n)$$

**A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>**为属性集中的属性名。



### 2.1.3 关系模式

由定义可见，关系模式是关系的型，是关系的框架，是对关系结构的描述，它是静态的、稳定的；关系是动态的、随时间不断变化的，它是关系模式在某一时刻的状态或内容。

例如，第**1**章教学数据库中，共有五个关系，其关系模式可分别表示为：

学生（学号，姓名，性别，年龄，系别）

教师（教师号，姓名，性别，年龄，职称，工资，岗位津贴，系别）

课程（课程号，课程名，课时）

选课（学号，课程号，成绩）

授课（教师号，课程号）





# 2.1 关系数据结构

## 2.1.1 关系

## 2.1.2 关系模式

## 2.1.3 关系数据库



## 2.1.3 关系数据库

### » 关系数据库

- 在一个给定的应用领域中，所有关系的集合构成一个关系数据库

### » 关系数据库的型与值

- 关系数据库的型: 关系数据库模式，是对关系数据库的描述
- 关系数据库的值: 关系模式在某一时刻对应的关系的集合，通常称为关系数据库



# 主要内容

2.1 关系模型概述

2.2 关系的完整性

2.3 关系操作

2.4 关系代数

2.5 小结



### 2.2.3 关系的完整性

为了维护关系数据库中数据与现实世界的一致性，对关系数据库的插入、删除和修改操作必须有一定的约束条件，这些约束条件实际是现实世界的要求。关系模型的完整性规则是对关系的某种约束条件，以保证关系数据库中数据的正确性、有效性和一致性。

关系模型中有三类完整性约束：实体完整性，参照完整性和用户定义完整性。



## 2.2关系的键与关系的完整性

### 1、候选键

**候选键**：能唯一标识关系中元组的一个属性或属性集合。也称为候选关键字或候选码。

**候选键的形式化定义**：设关系**R**有属性**A1,A2,...,An**，其属性集**K= (Ai, Aj, ..., Ak)**，当且仅当满足下列条件时，**K**被称为候选键。

(1) 唯一性，关系**R**的任意两个不同元组，其属性**K**的值是不同的。

(2) 最小性，组成关系键的属性集 (**Ai, Aj, ..., Ak**) 中，任一属性都不能从属性集**K**中删除，否则将破坏唯一性的性质。



## 2、主关系键

**主关系键：**如果一个关系中有多个候选键，可以选择一个作为查询、插入或删除元组的操作变量，被选用的候选键称为主关系键，或简称为主键、主码、主关键字。每个关系必须选择一个主关系键。

## 3、主属性

**主属性：**包含在候选键中的各个属性称为主属性。

**非码属性：**不包含在任何候选键中的属性称为非码属性。

**全码：**所有属性的组合是关系的候选键。

如：授课（教师号，课程号）



## (1) 实体完整性规则

关系中主码的值不能为空或部分为空。

因为关系中的元组是通过主码区分的，如果主码的值取空值，就说明存在某个不可标识的元组，这是不允许的。例如：

- ❖ 学生关系中的主码“学号”不能为空；
- ❖ 选课关系中的主码“学号+课程号”不能部分为空，即“学号”和“课程号”两个属性都不能为空。



## 2.2.2 外部关系键

**外部关系键**：如果关系**R2**的一个或一组属性**X**不是**R2**的主码，而是另一个关系**R1**的主码，则该属性或属性组**X**称为**R2**的外部关系键、外键或外码，并称关系**R2**为参照关系，关系**R1**为被参照关系。

举例：教学数据库中，学生关系、课程关系为被参照关系，选课关系为参照关系。

**学生** (学号，姓名，性别，年龄，系别)

**教师** (教师号，姓名，性别，年龄，职称，工资，岗位津贴，系别)

**课程** (课程号，课程名，课时)

**选课** (学号，课程号，成绩)

**授课** (教师号，课程号)





## (2) 参照完整性规则

如果关系 $R_2$ 的外码 $X$ 与关系 $R_1$ 的主码相对应，则在 $R_2$ 中，外码 $X$ 的每个值必须在关系 $R_1$ 主码的值中找到，或者为空值。参照完整性规则定义了外码与主码之间的应用规则。

职工关系S

编号	姓名	性别	部门号
2011	王萍	女	011
2012	李明	男	
...	...	...	...
2040	陈强	男	012
...	...	...	...

部门关系T

部门号	名称	地址	电话
011	市场部	公司办201	871
012	人事部	公司办202	781
014	财务部	公司办203	520
015	公司办	公司办204	356
...	...	...	...

关系S的外码“部门号”只能取：**空值**表示尚未给该职工分配部门；取**非空值**表示该值必须是关系T中某个元组的“部门号”的值，表示某职工不可能分配到一个不存在的部门中。



实体完整性规则和参照完整性规则是关系模型必须满足的完整性约束条件，由关系数据库系统自动支持。除此之外，不同的关系数据库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关系数据库的约束条件。



### （3）用户定义的完整性

用户自定义完整性是针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。

例如，属性值根据实际需要，要具备一定的约束条件，选课关系（学号，课程号，成绩），“成绩”不能为负数，取值范围**1~100**；性别属性必须取“男”，“女”等。

关系模型应提供定义和检验这类完整性的机制，以便使用统一的、系统的方法处理它们，而不是由应用程序承担这一功能。



2.1 关系模型概述

2.2 关系的完整性

2.3 关系操作

2.4 关系代数

2.5 小结



## 2.3.1 基本的关系操作

- 常用的关系操作
  - 查询操作：选择、投影、连接、除、并、差、交、笛卡尔积
    - 选择、投影、并、差、笛卡尔基是**5**种基本操作
  - 数据更新：插入、删除、修改
- 关系操作的特点
  - 集合操作方式：操作的对象和结果都是集合，**一次一集合**的方式



## 2.3.2 关系数据库语言的分类

- 关系代数语言
  - 用对关系的运算来表达查询要求
  - 代表：ISBL
- 关系演算语言：用谓词来表达查询要求
  - 元组关系演算语言
    - 谓词变元的基本对象是元组变量
    - 代表：APLHA, QUEL
  - 域关系演算语言
    - 谓词变元的基本对象是域变量
    - 代表：QBE
- 具有关系代数和关系演算双重特点的语言
  - 代表：SQL（Structured Query Language）



## 2.4 关系代数

关系代数是施加于关系上的一组集合代数运算。每个运算都以一个或多个关系作为运算对象，并生成另外一个关系作为该关系运算的结果。关系代数的运算按运算符的不同主要分为两类：**传统的集合运算(并、差、交、笛卡尔积等)和专门的关系运算(选择、投影、连接、除等)。**

注意：关系代数运算的基本运算是并、差、笛卡尔积、选择和投影，其它运算都可以从以上五个运算导出。



### 2.3.1 传统的集合运算

(1) 关系的并：关系R和关系S的所有元组合并，再删去重复的元组，组成一个新关系，称为R和S的并，记为 $R \cup S$ 。

(2) 关系的差：关系R和关系S的差是由属于R而不属于S的所有元组组成的集合，即关系R中删去与S关系中相同的元组，组成一个新的关系，记为 $R - S$ 。

(3) 关系的交：关系R和关系S的交是由既属于R又属于S的元组组成的集合，即在两个关系R和S中取相同的元组，组成一个新关系，记为 $R \cap S$ 。

注意：参与并、交、差运算的两个关系必须为同类型的（即两个关系具有相同的结构，或对应的属性名相同，且具有相同的域）





(4) 关系的笛卡尔积：两个分别为n目和m目的关系R和S的笛卡尔积记为 $R \times S$ ，是一个 $(n+m)$ 列关系，是所有 $(n+m)$ 列元组的集合，元组的前n列分量是R中的一个元组，后m列分量是S中的一个元组。若R有 $K_1$ 个元组，S有 $K_2$ 个元组，则关系R和关系S的笛卡尔积有 $K_1 \times K_2$ 个元组。



**R**

A	B
a	d
b	a
c	c

**S**

A	B
d	a
b	a
d	c

**T**

B	C
b	b
c	d

**$R \cup S$**

A	B
a	d
b	a
c	c
d	a
d	c

**$R \cap S$**

A	B
b	a

**$R - S$**

A	B
a	d
c	c

**$R \times T$**

A	B	B	C
a	d	b	b
a	d	c	d
b	a	b	b
b	a	c	d
c	c	b	b
c	c	c	d



## 2.3.2 专门的关系运算

由于传统的集合运算，只是从行的角度对关系进行运算，而要灵活地实现关系数据库多样的查询操作，必须引入专门的关系运算。专门的关系运算不仅涉及行运算，也涉及列运算。



## (1) 选择

选取运算是单目运算，从关系中找出满足给定条件的所有元组称为选择。其中的条件是以逻辑表达式给出的。该逻辑表达式的值为真的元组被选取。

选择运算记为： $\sigma_F(R)$ 。

其中 $\sigma$ 为选择运算符， $R$ 为关系， $F$ 为布尔函数， $F$ 是由算术比较运算符（ $>$ ， $\geq$ ， $<$ ， $\leq$ ， $=$ ， $\neq$ ）、逻辑运算符（ $\vee$ ， $\wedge$ ， $\neg$ ）、属性名（属性序号）和常数连接起来的逻辑表达式。属性序号是对应属性在关系中的顺序编号。

选择运算从行的角度进行的运算。经过选择运算得到的关系，其关系模式不变，但其元组的数目小于等于原关系中元组的个数，它是原关系的一个子集。



S

B	C	D
2	3	2
5	6	3
9	8	5

$\sigma_{B=5}(S)$

结果:

B	C	D
5	6	3



## (2) 投影

从关系中挑选若干属性组成新的关系称为投影。投影运算也是单目运算，投影运算是从列的角度进行的运算，相当于对关系进行垂直分解。

投影运算记为： $\pi_x(R)$ 。

其中R为一个关系，x为一组属性名或属性序号组，各属性名或属性序号之间用逗号分开，属性序号是对应属性在关系中的顺序编号。

经过投影运算可以得到一个新关系，其所包含的属性个数往往比原关系少，或者属性的排列顺序不同。如果新关系中包含重复的元组，删去重复元组。



R

A	B	C
1	2	3
4	5	6
7	8	8

$\pi_{A, C}(R)$

结果:

A	C
1	3
4	6
7	8



### (3) 连接

连接运算是二目运算。连接是将两个关系所有属性名拼接在一起形成一个新的关系，生成的新关系中包含满足连接条件的元组。运算过程是通过连接条件控制的。是从两个关系的笛卡尔积中选取满足连接条件的元组组成新的关系。连接运算包括 $\theta$ 连接、F连接、自然连接。





## ➤ $\theta$ 连接

$\theta$ 连接操作是从关系**R**和关系**S**的笛卡尔积中选取属性值满足**某一 $\theta$ 操作**的元组，记为  $R \bowtie_{i\theta j} S$ 。

这里**i**和**j**分别是关系**R**和关系**S**中第**i**个和第**j**个属性的序号， $\theta$ 为算术运算符。

如果 $\theta$ 是等号“=”，该 $\theta$ 连接操作称为“等值连接”。

$\theta$ 连接运算可以用选取运算和笛卡尔积运算来表示：

$$R \bowtie_{i\theta j} S = \sigma_{i\theta j}(R \times S)$$



R

A	B	C
1	2	3
4	5	6
7	8	9

S

B	C	D
2	3	2
5	6	3
9	8	5

求:  $R \bowtie S$   
 $[3] = [2]$

$R \bowtie S$   
 $C = C$

R×S

A	B	C	B	C	D
1	2	3	2	3	2
1	2	3	5	6	3
1	2	3	9	8	5
4	5	6	2	3	2
4	5	6	5	6	3
4	5	6	9	8	5
7	8	9	2	3	2
7	8	9	5	6	3
7	8	9	9	8	5

$R \bowtie S$ 的结果:  
 $[3] = [2]$

A	B	C	B	C	D
1	2	3	2	3	2
4	5	6	5	6	3

- $\theta$ 连接的结果中，所有元组的属性值都满足 $\theta$ 操作。
- 等值连接中参加等值操作的属性名可以不同，必须具有相同的域；
- $\theta$ 连接结果可以包括多个同名属性。



## ➤ F连接

F连接操作是从关系R和关系S的笛卡尔积中选取属性值满足某一公式F的元组，记为： $R \bowtie_F S$ 。这里F是形为 $F_1 \wedge F_2 \wedge \dots \wedge F_n$ 的公式，每个 $F_p$ 是形为 $i \theta j$ 的式子，而i和j应分别为R和S的第i、第j个分量的序号。

F连接可以用选取运算和笛卡尔积运算来表示：

$$R \bowtie_F S = \sigma_F(R \times S)$$



R

A	B	C
1	2	3
4	5	6
7	8	9

S

B	C	D
2	3	2
5	6	3
9	8	5

求  $R \bowtie S$   
 $(A > D) \wedge (C = C)$

R×S

A	B	C	B	C	D
1	2	3	2	3	2
1	2	3	5	6	3
1	2	3	9	8	5
4	5	6	2	3	2
4	5	6	5	6	3
4	5	6	9	8	5
7	8	9	2	3	2
7	8	8	5	6	3
7	8	9	9	8	5

$R \bowtie S$  结果:

$(A > D) \wedge (C = C)$

A	B	C	B	C	D
4	5	6	5	6	3

F连接的结果中，所有元组的属性值都满足F公式。F连接结果可以包括多个同名属性。



## ➤ 自然连接

自然连接是基于同名属性的等值连接，且结果除去重复属性，它是连接运算的一个特例，是最常用的连接运算。

自然连接记为： $R \bowtie S$ 。

其中R和S是两个关系，并且具有一个或多个同名属性。自然连接是在笛卡尔积 $R \times S$ 中选出同名属性上符合相等条件元组，去掉重复的同名属性，组成新的关系。

自然连接要求两个关系中，进行等值运算的属性必须同名。如果两个关系进行自然连接，若没有同名属性，则自然连接的结果等于两个关系的笛卡尔积。



R

A	B	C
1	2	3
4	5	6
7	8	9

S

B	C	D
2	3	2
5	6	3
9	8	5

求  $R \bowtie S$  $R \times S$ 

A	B	C	B	C	D
1	2	3	2	3	2
1	2	3	5	6	3
1	2	3	9	8	5
4	5	6	2	3	2
4	5	6	5	6	3
4	5	6	9	8	5
7	8	9	2	3	2
7	8	8	5	6	3
7	8	9	9	8	5

 $R \bowtie S$  结果: $(B = B) \wedge (C = C)$ 

A	B	C	B	C	D
1	2	3	2	3	2
4	5	6	5	6	3

 $R \bowtie S$  结果:

A	B	C	D
1	2	3	2
4	5	6	3

自然连接与等值连接区别:

- 自然连接要求相等属性值的属性名必须相同; 等值连接下不同属性名, 相同域的属性可进行等值运算。
- 自然连接结果去掉重复属性列, 自然连接是去掉重复列的等值连接。等值连接保留重复属性列。



## 连接（续）

### » 悬浮元组（Dangling tuple）

- 两个关系 $R$ 和 $S$ 在做自然连接时，关系 $R$ 中某些元组有可能在 $S$ 中不存在公共属性上值相等的元组，从而造成 $R$ 中这些元组在操作时被舍弃了，这些被舍弃的元组称为悬浮元组。



# 连接（续）

## » 外连接（Outer Join）

- 如果把悬浮元组也保存在结果关系中，而在其他属性上填空值(Null)，就叫做外连接
- 左外连接(LEFT OUTER JOIN或LEFT JOIN)
  - 只保留左边关系 $R$ 中的悬浮元组
- 右外连接(RIGHT OUTER JOIN或RIGHT JOIN)
  - 只保留右边关系 $S$ 中的悬浮元组





## 连接（续）

下图是例2.8中关系*R*和关系*S*的外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL
NULL	b5	NULL	2



## 连接（续）

图(b)是例2.8中关系*R*和关系*S*的左外连接,图(c)是右外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL

图(b)

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
NULL	b5	NULL	2

图(c)



#### (4) 除

设有关系 $R(X, Y)$ 与关系 $S(Z)$ ，其中， $X, Y, Z$ 为属性集合。假设 $Y$ 和 $Z$ 具有相同的属性个数，且对应属性出自同域。

关系 $R(X, Y)$ 除以关系 $S(Z)$ 所得的商关系是关系 $R$ 在属性 $X$ 上投影的一个子集，该子集和 $S(Z)$ 的笛卡尔积必须包含在 $R(X, Y)$ 中。记为 $R \div S$ 。

$X=(A,B)$   $Y=(C,D)$   $Z=(C,D)$   $Y$ 和 $Z$ 属性名可以相同，也可以不同。

A	B	C	D
a	b	c	d
a	b	e	f
c	a	c	d

C	D
c	d
e	f

$R$ 在 $A, B$ 上的投影 $T$ :

A	B
a	b
c	a

$R \div S$ 结果:

A	B
a	b

$T \times S$

A	B	C	D
a	b	c	d
a	b	e	f
c	a	c	d
c	a	e	f



# 专门的关系运算（续）

象集 $Z_x$

给定一个关系 $R(X, Z)$ ， $X$ 和 $Z$ 为属性组。

当 $t[X]=x$ 时， $x$ 在 $R$ 中的象集（Images Set）为：

$$Z_x = \{t[Z] | t \in R, t[X]=x\}$$

它表示 $R$ 中属性组 $X$ 上值为 $x$ 的诸元组在 $Z$ 上分量的集合



# 专门的关系运算（续）

$R$

$x_1$	$Z_1$
$x_1$	$Z_2$
$x_1$	$Z_3$
$x_2$	$Z_2$
$x_2$	$Z_3$
$x_3$	$Z_1$
$x_3$	$Z_3$

象集举例

- $x_1$ 在 $R$ 中的象集

$$Z_{x_1} = \{Z_1, Z_2, Z_3\},$$

- $x_2$ 在 $R$ 中的象集

$$Z_{x_2} = \{Z_2, Z_3\},$$

- $x_3$ 在 $R$ 中的象集

$$Z_{x_3} = \{Z_1, Z_3\}$$



## 除运算（Division）（续）

给定关系 $R(X, Y)$ 和 $S(Y, Z)$ ，其中 $X, Y, Z$ 为属性组。

$R$ 中的 $Y$ 与 $S$ 中的 $Y$ 可以有不同的属性名，但必须出自相同的域集。

$R$ 与 $S$ 的除运算得到一个新的关系 $P(X)$ ,

$P$ 是 $R$ 中满足下列条件的元组在  $X$  属性列上的投影：

元组在 $X$ 上分量值 $x$ 的象集 $Y_x$ 包含 $S$ 在 $Y$ 上投影的集合，记作：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$$

$Y_x$ :  $x$ 在 $R$ 中的象集,  $x = t_r[X]$



# 除运算（续）

[例2.9]设关系*R*、*S*分别为下图的(a)和(b)，*RS*的结果为图(c)

**R**

A	B	C
a1	b1	c2
a2	b3	c7
a3	b4	c6
a1	b2	c3
a4	b6	c6
a2	b2	c3
a1	b2	c1

**S**

B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

**$R \div S$**

A
a1



## 除运算（续）

- 在关系R中，A可以取四个值{a1, a2, a3, a4}  
 $a_1$ 的象集为  $\{(b_1, c_2), (b_2, c_3), (b_2, c_1)\}$   
 $a_2$ 的象集为  $\{(b_3, c_7), (b_2, c_3)\}$   
 $a_3$ 的象集为  $\{(b_4, c_6)\}$   
 $a_4$ 的象集为  $\{(b_6, c_6)\}$
- S在(B, C)上的投影为  
 $\{(b1, c2), (b2, c1), (b2, c3)\}$
- 只有 $a_1$ 的象集包含了S在(B, C)属性组上的投影

所以  $R \div S = \{a_1\}$





# 综合举例

以学生-课程数据库为例

[例2.10] 查询至少选修1号课程和3号课程的学生号码。

首先建立一个临时关系 $K$ :

Cno
1
3

然后求:  $\pi_{Sno,Cno}(SC) \div K$



## 综合举例（续）

- [例2.10]续

$\pi_{Sno,Cno}(SC)$

201215121象集{1, 2, 3}

201215122象集{2, 3}

$K=\{1, 3\}$

于是：

$\pi_{Sno,Cno}(SC) \div K = \{201215121\}$

Sno	Cno
201215121	1
201215121	2
201215121	3
201215122	2
201215122	3



## 综合举例（续）

[例2.11] 查询选修了2号课程的学生学号。

$$\pi_{\text{Sno}}(\sigma_{\text{Cno}='2'}(\text{SC}))=\{201215121,201215122\}$$

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$$\pi_{\text{Sname}}(\sigma_{\text{Cpno}='5'}(\text{Course} \bowtie \text{SC} \bowtie \pi_{\text{Sno},\text{Sname}}(\text{Student})))$$

或

$$\pi_{\text{Sname}}(\pi_{\text{Sno}}(\sigma_{\text{Cpno}='5'}(\text{Course}) \bowtie \text{SC}) \bowtie \pi_{\text{Sno},\text{Sname}}(\text{Student}))$$

[例2.13] 查询选修了全部课程的学生号码和姓名。

$$\pi_{\text{Sno},\text{Cno}}(\text{SC}) \div \pi_{\text{Cno}}(\text{Course}) \bowtie \pi_{\text{Sno},\text{Sname}}(\text{Student})$$



## 2.5 小结

- 关系数据库系统是目前使用最广泛的数据库系统
- 关系数据库系统与非关系数据库系统的区别：
  - 关系系统只有“表”这一种数据结构
  - 非关系数据库系统还有其他数据结构，以及对这些数据结构的操作



# 小结（续）

- 关系数据结构
  - 关系
    - 域
    - 笛卡尔积
    - 关系
      - 关系，属性，元组
      - 候选码，主码，主属性
      - 基本关系的性质
  - 关系模式
  - 关系数据库
  - 关系模型的存储结构



# 小结（续）

- 关系操作

- 查询

- 选择、投影、连接、除、并、交、差

- 数据更新

- 插入、删除、修改



## 小结（续）

- 关系的完整性约束
  - 实体完整性
  - 参照完整性
    - 外码
  - 用户定义的完整性



# 小结（续）

- 关系数据语言
  - 关系代数语言
  - 关系演算语言
    - 元组关系演算语言    ALPHA
    - 域关系演算语言        QBE

