

关系代数

- 基本概念
- 集合运算
- 投影
- 选择
- 笛卡尔积
- 自然连接
- θ 连接
- 改名
- 除法
- 复合运算
- 对关系的约束
- 小结

基本概念

◆**关系代数**：是从抽象的观点出发学习数据库查询的问题，定义查询运算符。

关系代数中，**表达式**的**运算数**是关系，关系用关系名表示。

将要描述的任何运算符用于关系或用于关系的表达式，逐渐构造更为复杂的表达式。

一个查询就是关系代数的一个表达式。

◆关系代数运算包括：

- 集合运算符： \cup (并)， $-$ (差)， \cap (交)， \times ((广义)笛卡尔积)
- 专门的关系运算符： σ (选择)， π (投影)， θ (连接)， $|\times|$ (自然连接)， \div (除)
- 逻辑运算符： \wedge (与)， \vee (或)， \neg (非)
- 算术比较运算符： $>$ (大于)， \geq (大于等于)， $<$ (小于)， \leq (小于等于)， $=$ (等于)， \neq (不等于)



基本概念

- 并集、差集、选择、投影、改名运算不能由其他关系运算导出，称为**基本运算**。
- 交集、笛卡尔积、 θ 连接、自然连接可由其他关系运算导出，称为**导出运算**(非基本运算)。
- 设关系模式 $R(A_1, A_2, \dots, A_n)$ ，它的一个关系为 R ，则用 $t \in R$ 表示 t 是 R 的一个元组， $t[A_i]$ 表示元组 t 中相应于属性 A_i 的一个分量。
- **象集(image set)**: 给定一个关系 $R(X, Y)$ ， X 和 Y 为属性集，当 $t[X]=x$ 时， x 在 R 中的象集是
 R 中的属性集 X 上值为 x 的诸元组在 Y 上分量的集合，表示为 $Y_x = \{t[Y] | t \in R, t[X]=x\}$ 。

1 关系的基本集合运算

◆集合运算：并、交、差。对于关系R和S，

■ **$R \cup S$** R和S的并，是在R中或在S中的元组的集合，即R和S的所有元组合并，结果中有重复的元组，则只保存一次。

$$R \cup S = \{t | t \in R \vee t \in S\}$$

■ **$R \cap S$** R和S的交，是在R和S中都存在的元组的集合。

$$R \cap S = \{t | t \in R \wedge t \in S\}$$

■ **$R - S$** R和S的差，是在R中而不在S中的元组的集合。

$$R - S = \{t | t \in R \wedge \neg t \in S\}$$

交运算可以用差运算来表示： $R \cap S = R - (R - S)$

! **$R - S$** 不同于 **$S - R$** ，后者是在S中而不在R中的元组的集合。

! 这三种基本集合运算不改变关系模式。

? 是否任意两个关系都可以进行集合运算？



关系的集合运算

? 是否任意两个关系都可以进行集合运算?

需要R和S两个关系满足下面条件:

- R和S的模式必须具有相同个数的属性集。
- R中第i个属性和S中第i个属性必须来自同一个域（相同的数据类型）。
- 如果要对属性数相同但属性名不同的关系进行集合运算，则可以利用改名运算符来改变其中一个或两个关系的模式，使它们具有相同的属性集。



关系的集合运算 例

假定有两个关系R和S，都是关系MovieStar的实例。R和S的当前实例分别在下图给出。

关系R

name	address	gender	birthdate
<u>Carrie Fisher</u>	123 Maple St. Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd. Brentwood	M	8/8/88

关系S

name	address	gender	birthdate
<u>Carrie Fisher</u>	123 Maple St. Hollywood	F	9/9/99
Harrison Ford	678 Palm Dr. Beverly Hills	M	7/7/77

$R \cup S$

name	address	gender	birthdate
<u>Carrie Fisher</u>	123 Maple St. Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd. Brentwood	M	8/8/88
Harrison Ford	678 Palm Dr. Beverly Hills	M	7/7/77

$R \cap S$

name	address	gender	birthdate
Carrie Fisher	123 Maple St. Hollywood	F	9/9/99



关系的集合运算 例

关系R

name	address	gender	birthdate
Carrie Fisher	123 Maple St. Hollywood	F	9/9/99
Mark Hamill	456 Oak Rd. Brentwood	M	8/8/88

关系S

name	address	gender	birthdate
Carrie Fisher	123 Maple St. Hollywood	F	9/9/99
Harrison Ford	678 Palm Dr. Beverly Hills	M	7/7/77

R-S

name	address	gender	birthdate
Mark Hamill	456 Oak Rd. Brentwood	M	8/8/88

S-R

name	address	gender	birthdate
Harrison Ford	678 Palm Dr. Beverly Hills	M	7/7/77

2 投影

◆ **投影**运算符：从关系R产生一个只有R的某些列的新关系。
投影运算是单目运算。表示为 $\pi_{A_1 A_2 \dots A_n}(R)$

$$\pi_{A_1, A_2, \dots, A_n}(R) = \{t[A_1, A_2, \dots, A_n] \mid t \in R\}$$

- 结果关系的模式是属性集 $\{A_1, A_2, \dots, A_n\}$ ，并保持列出的顺序。
- 投影后有重复的元组，则结果中只保存一次。

! 投影运算可以改变关系的属性次序！

例 考虑关系模式Movie，将该关系投影到前三个属性上。

title	year	length	inColor	studioName	produc#
Star Wars	1977	124	true	Fox	11111
Mighty Ducks	1991	104	true	Disney	22222
Wayne's World	1992	95	true	Paramount	33333



投影 例

例 考虑关系模式Movie，将该关系投影到前三个属性上。

title	year	length	inColor	studioName	produc#
Star Wars	1977	124	true	Fox	11111
Mighty Ducks	1991	104	true	Disney	22222
Wayne's World	1992	95	true	Paramount	33333

$$\pi_{title, year, length}(Movie)$$

用上面投影表达式将该关系投影到前三个属性上，结果如下。

title	year	length
Star Wars	1977	124
Mighty Ducks	1991	104
Wayne's World	1992	95

3 选择

◆**选择运算符**：将关系**R**中满足某种条件**C**的元组组成一个新的关系，记为 $\sigma_C(\mathbf{R})$ 。选择运算是单目运算。

$\sigma_C(\mathbf{R}) = \{t | t \in \mathbf{R} \wedge C(t) \text{为真}\}$

- 运算结果关系的模式中属性的顺序应保持与**R**同样的顺序。
- 条件**C**由运算项和运算符组成。
 - 运算项是常量或**R**的属性（用属性名表示）。
 - 运算符有算术比较运算符（>，≥，<，≤，=，≠）和逻辑运算符（ \wedge \vee \neg ）。

条件**C**的结果为逻辑值“真”或“假”。

■ 运算方法：

把条件**C**用于**R**的每个元组**t**，计算是否满足条件。如果将条件**C**的每个属性用元组中的相应属性代替后条件**C**为**真**，那么，元组**t**就是出现在选择运算 $\sigma_C(\mathbf{R})$ 的结果中的元组之一，否则**t**不出现在结果中。



选择 例

title	year	length	inColor	studioName	produc#
Star Wars	1977	124	true	Fox	11111
Mighty Ducks	1991	104	true	Disney	22222
Wayne's World	1992	95	true	Paramount	33333

查询长度不少于100分钟的电影信息 $\sigma_{\text{length} \geq 100}(\text{Movie})$

title	year	length	inColor	studioName	produc#
Star Wars	1977	124	true	Fox	11111
Mighty Ducks	1991	104	true	Disney	22222

$\sigma_{(\text{length} \geq 100) \wedge (\text{studioName} \in \text{'Fox'})}(\text{Movie})$

单引号

title	year	length	inColor	studioName	produc#
Star Wars	1977	124	true	Fox	11111

◆选择是从关系的水平方向上进行运算的，而投影则是从关系的垂直方向上进行的。



4 笛卡尔积

◆**笛卡尔积**：关系R、S的笛卡尔积是两个关系的元组对的集合所组成的新关系，用 $R \times S$ 表示。该元素对是选择R的任何元素作为第一个元素，S的元素作为第二个元素构成的。

- 结果关系模式是R和S的模式的并集，按代数表达式的顺序，R的分量在S的分量之前。
- 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 $k_1 * k_2$ 个元组。
- 如果R和S有同名属性，就需要为同名属性改名。

例如：R和S的模式中都有属性A，则在结果关系模式中对来自R的属性用R.A表示，对来自S的属性用S.A表示。



笛卡尔积 例

假设关系R和S具有如下给出的模式和元组。求乘积 $R \times S$ 包括的元组。

R		S		
A	B	B	C	D
1	2	2	5	6
3	4	4	7	8
		9	10	11

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

5 自然连接

◆ **自然连接**：两个关系R和S的自然连接是从 $R \times S$ 的结果集中，选取在公共属性上具有相同值的元组，组成新的关系。表示为 $R \bowtie S$ 。该运算是二目运算。

- R、S的公共属性是指属性集的交集（名称及类型相同）。
- 公共属性在结果关系模式中只出现一次。
- 假设 A_1, A_2, \dots, A_n 是R和S的公共属性，当且仅当R的元组r和S的元组s在 A_1, A_2, \dots, A_n 每个属性上都一致时，R的元组r和S的元组s才能成功地组成一对，作为自然连接的结果。此时构成一个**连接元组**。
- 没能在连接中和另一个关系的任何元组匹配的元组称为**悬挂元组**。
- 自然连接中可能会出现R中一个元组和S中几个元组连接的情况。



自然连接 例

A	B
1	2
3	4



B	C	D
2	5	6
4	7	8
9	10	11

=

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

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



B	C	D
2	3	4
2	3	5
7	8	10

=

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

关系U

关系V



自然连接

【分析】R和S的**自然连接**可看作是在广义笛卡尔积 $R \times S$ 中选出同名属性上符合相等条件元组，再进行投影，去掉重复的同名属性，组成新的关系。

即首先 $R \times S$ 。然后在如下条件C下进行选择

$$R.A_1=S.A_1 \text{ AND } R.A_2=S.A_2 \text{ AND } \dots \text{ AND } R.A_n=S.A_n$$

其中， A_1, A_2, \dots, A_n 是R和S的公共属性。最后再投影。

◆假设L是一个属性表，其中前面的属性是关系R的所有属性，后面的属性是除了共有属性以外的关系S的属性。则

$$R \bowtie S = \pi_L(\sigma_C(R \times S))$$



基本和导出运算 例

A	B	C		B	C	D
1	2	3	×	2	3	4
6	7	8		7	3	5
9	7	8		7	8	10

关系U

关系V

关系U和V的自然连接可以用乘积、选择和投影的形式写出：

$$\pi_{A,U.B,U.C,D}(\sigma_{(U.B=V.B) \wedge (U.C=V.C)}(U \times V)),$$

即首先得到乘积 $U \times V$ ，然后在每对同名属性（在这个例子中是B和C）之间进行等值选择。最后投影到所有的属性上，但只保留其中一个B和一个C。

6 θ 连接

◆ **θ 连接**，把两个关系的元组按指定条件匹配成对。**R和S基于条件C的 θ 连接表示为 $R \bowtie_C S$** 。该运算是二目运算。

■ 该运算的步骤如下：

➤ 获得R和S的笛卡尔积。

➤ 从乘积中选择满足条件C的元组。

■ 结果模式是R和S模式的并集，如果需要指出属性来自哪个模式，则可在属性前面加上前缀R.或S.。

■ θ 连接可以通过乘积和选择两个运算表达

$$R \bowtie_C S = \sigma_C(R \times S)$$



θ连接 例

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

B	C	D
2	3	4
2	3	5
7	8	10

=

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

关系 $U \bowtie_{A < D} V$

图中的结果模式包括所有6个属性，U和V的公共属性用U.和V.作为前缀加在相应的属性前面以区别它们。

? 若θ连接的条件是两个关系的公共属性相等，则是否等同于自然连接？

◆ 不同，自然连接的公共属性合并成一个副本。



θ连接 例

$U \bowtie_{(A < D) \wedge (U.B \neq V.B)} V$

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

关系U

B	C	D
2	3	4
2	3	5
7	8	10

关系V

$\sigma_{A < D \text{ AND } U.B \neq V.B}(U \bowtie V)$

A	U.B	U.C	V.B	V.C	D
1	2	3	7	8	10

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	2	3	4
6	7	8	2	3	5
6	7	8	7	8	10
9	7	8	2	3	4
9	7	8	2	3	5
9	7	8	7	8	10

7 改名

◆**改名**：使用运算符 $\rho_s(A_1, A_2, \dots, A_n)(R)$ 把关系R改名。

结果关系和初始关系R有完全相同的元组，但关系名是S，并且结果关系S的属性从左至右依次命名为 A_1, A_2, \dots, A_n 。

如果只想把关系改名为S，而属性名和R中的一样，用 $\rho_s(R)$ 。

【例】求 $R \times \rho_s(X, C, D)(S)$ 。

A	B
1	2
3	4

关系R

B	C	D
2	5	6
4	7	8
9	10	11

关系S

X	C	D
2	5	6
4	7	8
9	10	11

A	B	X	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

8 除法

◆**除法**：是二目运算。设有关系 $R(X,Y)$ 与关系 $S(Y,Z)$ ，其中 X,Y,Z 为属性集合， R 中的 Y 与 S 中的 Y 可以有不同的属性名，但对应属性必须出自相同的域。

关系 R 除以关系 S 所得的商是一个新关系 $P(X)$ ， P 是 R 中满足下列条件的元组在 X 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：

$$R \div S = \{tr[X] | tr \in R \wedge \pi_y(S) \subseteq Y_x\}$$

其中， Y_x 为 x 在 R 中的象集， $x=tr[X]$ 。

■除法运算为非基本运算，可以表示为：

$$R \div S = \pi_x(R) - \pi_x(\pi_x(R) \times \pi_y(S) - R)$$



除法 例

A	B	C	D	E
a1	b2	c3	d5	e1
a1	b2	c4	d6	e1
a2	b4	c1	d3	e3
a3	b5	c2	d8	e4

R

C	D	F
c3	d5	f3
c4	d6	f4

S

A	B	E
a1	b2	e1

R ÷ S



除法 例

sno	pno
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4

A

pno
P2

B1

sno
S1
S2
S3
S4

A/B1

pno
P2
P4

B2

sno
S1
S4

A/B2

pno
P1
P2
P4

B3

sno
S1

A/B3

9 查询中的复合运算

◆可以通过将运算符用于子表达式，必要时用括号表明运算项分组，来构造关系代数表达式。

【例】查询“由Fox公司制作的至少有100分钟长的电影的名称和年份是什么?”。

Movie(title,year,length,filmType,studioName)

【解】要计算这个查询的答案，可以按下面步骤进行。

- 1)选择长度length≥100的Movie元组。
- 2)选择StudioName='Fox'的Movie元组。
- 3)计算(1)和(2)的交集。
- 4)将来自(3)的关系投影到属性title和year上。

$$\pi_{title, year} \left(\sigma_{length \geq 100} (Movie) \cap \sigma_{studioName = 'Fox'} (Movie) \right)$$

$$\pi_{title, year} \left(\sigma_{length \geq 100 \wedge studioName = 'Fox'} (Movie) \right)$$



复合运算

◆ 自然连接运算的一个用途是重组被分解的关系。

【例】 设Movie1具有模式 {title,year,length,filmType,studioName}, Movie2具有模式 {title,year,starName}, 查询“主演的电影至少100分钟长的影星的名字”。

【解】 该查询首先用自然连接 \bowtie 把在title和year上一致的元组成功地匹配成对,也就是把同一部电影的元组匹配成对。

然后使用一个选择,其条件是要求电影的长度至少是100分钟。然后投影到所要求的属性starName上。表达式如下:

$$\pi_{starName} \left(\sigma_{length \geq 100} (Movie1 \bowtie Movie2) \right)$$



复合运算

◆除法运算同时从行和列的角度进行运算，适合于包含“全部”之类的短语的查询。

【例】SCore(SNO,CNO,grade),Course(CNO,Cname,Ccredit),
Student(SNO,Sname,Ssex,Sbirthday)

查询选修了全部课程的学生们的学号和姓名。

【解】首先需要获得全部课程信息，
然后利用除法得到具有全部课程成绩的学生们的学号，
再利用自然连接将学生的姓名获得。

$(\pi_{SNO,CNO}(SCore) \div \pi_{CNO}(Course)) \bowtie \pi_{SNO,SName}(Student)$



关系代数习题注意事项

- 1、问题要求回答的内容才要显示，其他内容不能有。别忘了用投影显示要查询的属性。
- 2、笛卡尔积、自然连接和 θ 连接要分清。
 - 笛卡尔积是两个关系的元组的排列组合，没有任何条件。
 - 自然连接要求元组中相同属性有相同的值。
 - θ 连接是在做了笛卡尔积之后再选择符合条件的组合。
- 3、复杂的查询可以首先将它分解成一步一步的简单的查询。



关系代数练习1

【例】有关系模式如下，写出关系代数表达式回答下列查询：

Product(maker(制造商),model,producttype(产品类型))

PC(model(型号),speed,ram(内存),hd(硬盘),cd(光驱),price)

- 1、找出PC速度至少为150的型号。
- 2、列出制造商制造的所有的PC的型号、速度、价格和相应的制造商名。
- 3、找出价格高于1000美元的PC的制造商。
- 4、找出速度大于133或者内存大于256的PC的型号。

$$1. \pi_{model} \left(\sigma_{speed \geq 150} (PC) \right)$$

$$2. \pi_{model, speed, price, maker} (Product[X]PC)$$

$$3. \pi_{maker} \left(\sigma_{price > 1000} (Product[X]PC) \right)$$

$$4. \pi_{model} \left(\sigma_{speed > 133 \vee ram > 256} PC \right)$$



关系代数练习2

【例】有关系模式如下，写出关系代数表达式回答下列查询：

Product(maker(制造商),model,producttype(产品类型))

PC(model(型号),speed,ram(内存),hd(硬盘),cd(光驱),price)

Laptop(model,speed,ram,hd,screen,price)

Printer(model,color(颜色),type(类型),price)

1、找出在两个或两个以上PC中出现的硬盘容量。

【解】 $R1 = \rho_{PC1}(PC)$

$R2 = \rho_{PC2}(PC)$

$R3 = R1 \bowtie_{PC1.hd=PC2.hd \wedge PC1.model \neq PC2.model} R2$

$R4 = \pi_{PC1.hd}(R3)$



关系代数练习2

【例】有关系模式如下，写出关系代数表达式回答下列查询：

Product(maker(制造商),model,producttype(产品类型))

PC(model(型号),speed,ram(内存),hd(硬盘),cd(光驱),price)

Laptop(model,speed,ram,hd,screen,price)

Printer(model,color(颜色),type(类型), price)

2、找出速度相同且RAM相同的成对的PC型号。

【解】 $R1 = \rho_{PC1}(PC)$

$R2 = \rho_{PC2}(PC)$

$R3 =$

$R1 \bowtie_{PC1.speed=PC2.speed \wedge PC1.ram=PC2.ram \wedge PC1.model \neq PC2.model} R2$

$R4 = \pi_{PC1.model, PC2.model}(R3)$

? 如何消除重复的PC对?



关系代数练习2

【例】有关系模式如下，写出关系代数表达式回答下列查询：

Product(maker(制造商),model,producttype(产品类型))

PC(model(型号),speed,ram(内存),hd(硬盘),cd(光驱),price)

Laptop(model,speed,ram,hd,screen,price)

Printer(model,color(颜色),type(类型),price)

3、找出至少生产两种不同的计算机且机器速度至少为133的厂商。

【解】 $R1 = \pi_{model, speed} (PC) \cup \pi_{model, speed} (Laptop)$

$$R2 = \pi_{maker, model} (\sigma_{speed \geq 133} (R1)[X]Product)$$

$$R3 = \rho_{R3}(maker\ 2, model\ 2)(R2)$$

$$R4 = R2[X]_{maker=maker\ 2 \wedge model \neq model\ 2} R3$$

$$R5 = \pi_{maker} R4$$

10 对关系的约束

◆如果R和S是关系代数表达式，则

- $R=\emptyset$ 是一个约束，它表明“R的值必须为空”，即“在R的结果中没有元组”。
- $R\subseteq S$ (R属于S)是一个约束。它表明“R中的每个元组也必须在S中”。而S可能包括不是R的元组。
- 约束 $R\subseteq S$ 也可以写为 $R-S=\emptyset$ ， $R=\emptyset$ 也可以写为 $R\subseteq\emptyset$ 。

$R-R=\emptyset$ 。

◆**参照完整性约束**：如果实体A与实体B相关，那么B必须实际存在。

在关系模型中，如果一个关系R的一个元组有一个分量值为v，那么根据用户的设计意图，可以期望v将出现在另一个关系S的某个元组的特定分量中。



参照完整性约束 例

考虑两个关系

Movie(title,year,length,inColor,studioName,producerC#)

MovieExec(name,address,cert#,networth)

如果要求它们之间存在参照完整性约束，即假定每部电影的制片人必须出现在MovieExec关系中，

也就是每个Movie元组的producerC#分量也必须出现在某个MovieExec元组的cert#分量中。

则可以用集合的包含关系来表达这个约束：

$$\pi_{producerC\#}(Movie) \subseteq \pi_{cert\#}(MovieExec)$$

$$\pi_{producerC\#}(Movie) - \pi_{cert\#}(MovieExec) = \Phi$$



参照完整性约束 例

如果某个参照完整性约束涉及到多个属性，也可以类似地表达。例如，要求在关系

starsIn(movieTitle,movieYear,starName)

中涉及到的任何电影也出现在下面的关系中

Movie(title,year,length,inColor,studioName,producerC#)

在两个关系中电影都用title-year对表示。

则用下面表达式表达这样的参照完整性约束。

$$\pi_{movieTitle, movieYear}(starsIn) \subseteq \pi_{title, year}(Movie)$$

! 注意前后投影顺序要一致。



附加约束

◆ **域约束**就是要求属性的值有特定的数据类型，例如整型或长度为30的字符串。

如果域约束涉及到属性所需要的特定值，属性可取值的集合可以用选择条件语言表达，那么这个域约束就可以用代数约束语言表达。

【例】假定希望MovieStar“性别”属性仅有的合法值为‘F’和‘M’。则可以用如下的代数方法表示这个约束。

$$\sigma_{gender \neq 'F' \text{ AND } gender \neq 'M'}(MovieStar) = \Phi$$



附加约束 例

有关系MovieExec(name,address,cert#,networth)和Studio(name,address,presC#), 假定要求一个人必须至少有净资产\$10000000才可以做电影制片公司的总裁。试将该约束用关系代数表示。

首先用 θ 连接把两个关系连起来, θ 连接中利用的条件是来自studio的presC#和来自MovieExec的cert#相等。该连接把包含制片公司和行政长官的两个元组组合成对。

如果从这个关系中选择净资产少于\$10000000的元组, 则要求它必须为空。这样, 可以把该约束表达为

$$\sigma_{netWorth < 10000000} (studio \bowtie_{presC\# = cert\#} MovieExec) = \Phi$$

$$\pi_{presC\#} (studio) \subseteq \pi_{cert\#} (\sigma_{networth \geq 10000000} (MovieExec))$$

关系代数小结

- 1、并 $R \cup S$ 、交 $R \cap S$ 、差 $R - S$ 。
- 2、**投影** 从关系 R 产生一个只有 R 的某些列的新关系。
- 3、**选择** $\sigma_C(R)$ 保留满足条件 C 的 R 中的元组。
- 4、**笛卡尔积** $R \times S$ 是两个关系 R 和 S 的元素对的集合。
- 5、**自然连接** $R \bowtie S$ ，结果关系的元组是 R 和 S 元组在公共属性上取值一致的所有元组的组合。
- 6、 **θ 连接** $R \bowtie_C S$ ，结果关系是首先获得 R 和 S 的积(笛卡尔积)，然后从乘积中选择满足条件 C 的元组。
- 7、**改名** $\rho_s(A_1, A_2, \dots, A_n)(R)$ ，结果关系名是 S ，属性从左至右依次命名为 A_1, A_2, \dots, A_n 。
- 8、**除法** $R \div S$ 是 R 中满足下列条件的元组在 X 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。