

数据管理基础

第2章 关系数据库

(2.4 关系代数)

智能软件与工程学院



基本的关系操作

□ 常用的关系操作

➤ 集合操作：并、交、差、笛卡尔积

➤ 专门的关系操作

- 选择、投影

- 连接（包括 θ -连接，自然连接，外连接等）

- 除

□ 选择、投影、并、差、笛卡尔积是5种基本操作

□ 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

关系代数运算符

运 算 符		含 义
集合 运算符	\cup	并
	$-$	差
	\cap	交
	\times	笛卡尔积
专门的 关系 运算符	σ	选择
	π	投影
	\bowtie	连接
	\div	除

关系代数

□ 关系代数是一种抽象的查询语言，它用对关系的运算来表达查询

□ 关系代数

➤ 运算对象是 关系

➤ 运算结果亦为 关系

□ 关系代数的运算符有两类：集合运算符 和 专门的关系运算符

➤ 传统的集合运算是从关系的“水平”方向，即行的角度进行

➤ 专门的关系运算不仅涉及行而且涉及列

□ 设关系模式为 $R(A_1, A_2, \dots, A_n)$

➤ 它的一个关系设为 R

➤ $t \in R$ 表示 t 是 R 的一个元组

➤ $t[A_i]$ 则表示元组 t 中相应于属性 A_i 的一个分量（属性值）

➤ 若 $A = \{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$, 其中 $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ 是 A_1, A_2, \dots, A_n 中的一部分, 则 A 称为‘属性列’或‘属性组’或‘属性集’。

➤ $t[A] = (t[A_{i_1}], t[A_{i_2}], \dots, t[A_{i_k}])$ 表示元组 t 在属性列 A 上诸分量的集合。

➤ \bar{A} 则表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 $\{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$ 后剩余的属性组。

使用的记号 2

□ R 为 n 目关系， S 为 m 目关系。

➤ $t_r \in R$, $t_s \in S$, $\widehat{t_r t_s}$ 称为元组的连接。

➤ $\widehat{t_r t_s}$ 是一个 $n + m$ 列的元组，前 n 个分量来自于 R 中的一个元组，后 m 个分量来自于 S 中的一个元组。

➤ 为方便表示，以后用 (t_r, t_s) 来表示元组的连接。

□ 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：

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

➤ 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

象集的例子

□ 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

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

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

□并 (Union)

➤ R 和 S

- 具有相同的目 n (即两个关系都有 n 个属性)
- 相应的属性取自同一个域

➤ $R \cup S$ 的运算结果

- 仍为 n 目关系, 由属于 R 或属于 S 的元组组成

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

并 2

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cup S$

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_2

□ 差 (Difference)

➤ R 和 S

- 具有相同的目 n
- 相应的属性取自同一个域

➤ $R - S$ 的运算结果

- 仍为 n 目关系，由属于 R 而不属于 S 的所有元组组成

$$R - S = \{ t \mid t \in R \wedge t \notin S \}$$

差 2

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R - S$

A	B	C
a_1	b_1	c_1

□交 (Intersection)

➤ R 和 S

- 具有相同的目 n
- 相应的属性取自同一个域

➤ $R \cap S$ 的运算结果

- 仍为 n 目关系，由既属于 R 又属于 S 的元组组成

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

➤ 交不是一个基本运算符，其功能可以用差运算来实现

$$R \cap S = R - (R - S) = S - (S - R)$$

交 2

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

$R \cap S$

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

笛卡尔积 1

- ❑ 严格地讲应该是广义的笛卡尔积 (Extended Cartesian Product)
- ❑ R : n 目关系, k_1 个元组
- ❑ S : m 目关系, k_2 个元组
- ❑ 笛卡尔积 $R \times S$ 的结果是
 - 列: $(n + m)$ 列元组的集合
 - 元组的前 n 列来自于关系 R 的一个元组
 - 元组的后 m 列来自于关系 S 的一个元组
 - 行: $k_1 \times k_2$ 个元组
 - $R \times S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \}$

笛卡尔积 2

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \times S$

R.A	R.B	R.C	S.A	S.B	S.C
a_1	b_1	c_1	a_1	b_2	c_2
a_1	b_1	c_1	a_1	b_3	c_2
a_1	b_1	c_1	a_2	b_2	c_1
a_1	b_2	c_2	a_1	b_2	c_2
a_1	b_2	c_2	a_1	b_3	c_2
a_1	b_2	c_2	a_2	b_2	c_1
a_2	b_2	c_1	a_1	b_2	c_2
a_2	b_2	c_1	a_1	b_3	c_2
a_2	b_2	c_1	a_2	b_2	c_1

❑ 学生-课程数据库:

学生关系 Student、课程关系 Course、选修关系 SC

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

基础关系 2

课程关系 **Course**

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

基础关系 3

选修关系 SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

选择 1

❑选择 (Selection) 又称为限制 (Restriction)

❑选择运算符的含义

➤在关系 R 中选择满足给定条件的诸元组

$$\sigma_F(R) = \{ t \mid t \in R \wedge F(t) = \text{'真'} \}$$

➤ F : 选择条件, 是一个逻辑表达式, 取值为“真”或“假”

●基本形式为: $X_1 \theta Y_1$

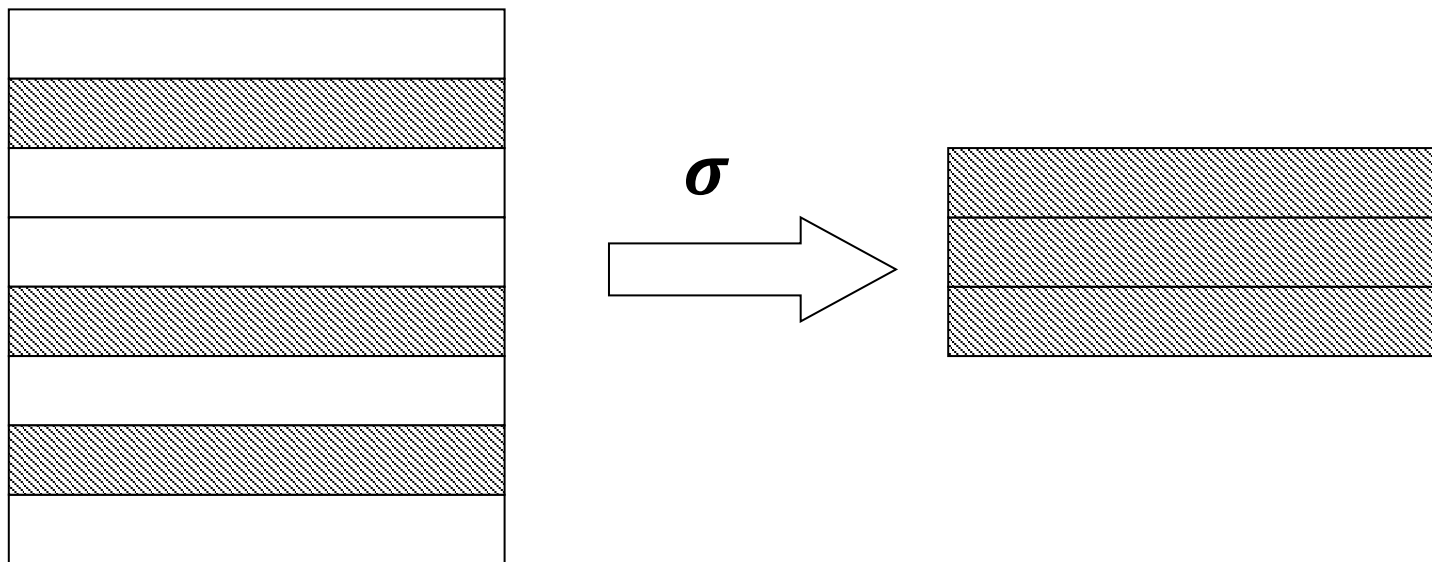
– θ 表示比较运算符, 它可以是 $>$, \geq , $<$, \leq , $=$, $<>$

(也可以用 \neq 来表示不等比较 $<>$)

●在基本的选择条件上可以进一步进行逻辑运算 (与, 或, 非)

选择 2

□ 选择运算是从关系 R 中选取使逻辑表达式 F 为真的元组，是从行的角度进行的运算



选择 3

例：查询信息系（IS系）全体学生。

$$\sigma_{Sdept = 'IS'}(Student)$$

结果：

Sno	Sname	Ssex	Sage	Sdept
201215125	张立	男	19	IS

选择 4

例：查询年龄小于20岁的学生。

$$\sigma_{Sage < 20} (Student)$$

结果：

Sno	Sname	Ssex	Sage	Sdept
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

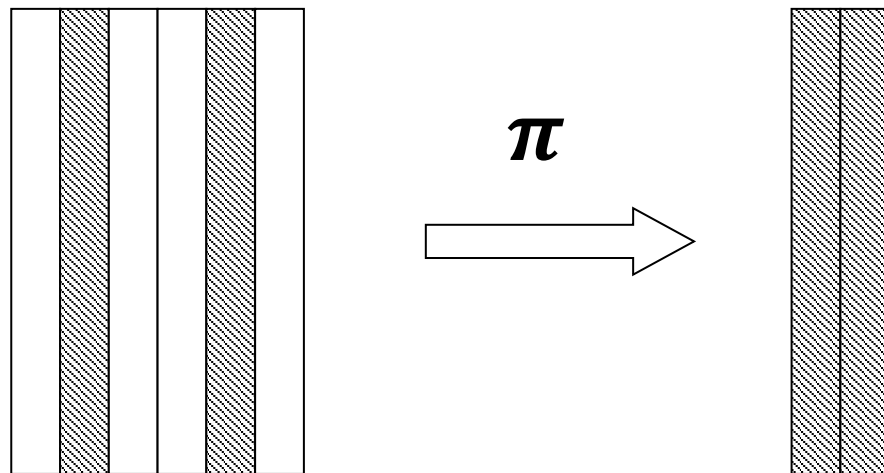
投影 1

□投影 (Projection)

➤从 R 中选择出若干属性列组成新的关系 (A 是 R 中的属性列)

$$\pi_A(R) = \{ t[A] \mid t \in R \}$$

➤投影操作主要是从列的角度进行运算



➤投影之后不仅过滤了原关系中的某些列，而且还可能消除某些元组 (避免重复行)，最后得到计算的结果关系

投影 2

□例：查询学生的姓名和所在系。

即求Student关系上学生姓名和所在系两个属性上的投影

$$\pi_{Sname, Sdept}(Student)$$

结果：

Sname	Sdept
李勇	CS
刘晨	CS
王敏	MA
张立	IS

例：查询学生关系Student中都有哪些系。

$$\pi_{Sdept}(Student)$$

结果：

Sdept
CS
IS
MA

连接 1

□ 连接 (Join) 也称为 θ 连接

□ 连接运算的含义

从两个关系的笛卡尔积中选取属性间满足一定条件的元组

$$R \underset{A\theta B}{\bowtie} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge (t_r[A] \theta t_s[B]) \}$$

● A 和 B ：分别为 R 和 S 上度数相等且可比的属性组

● θ ：比较运算符

➤ 连接运算从 R 和 S 的广义笛卡尔积 $R \times S$ 中选取 R 关系在 A 属性组上的值与 S 关系在 B 属性组上的值满足比较关系 θ 的元组。

□ 等值连接 (equi join)

- θ 为 “=” 的连接运算称为等值连接
- 从关系 R 与 S 的广义笛卡尔积中选取 A 、 B 属性值相等的那些元组，即等值连接为：

$$R \underset{A=B}{\bowtie} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B] \}$$

连接 3

□ 自然连接 (Natural join)

➤ 一种特殊的等值连接

- 两个关系中进行比较的分量必须是相同的属性组
- 在结果中把重复的属性列去掉

➤ 自然连接的含义

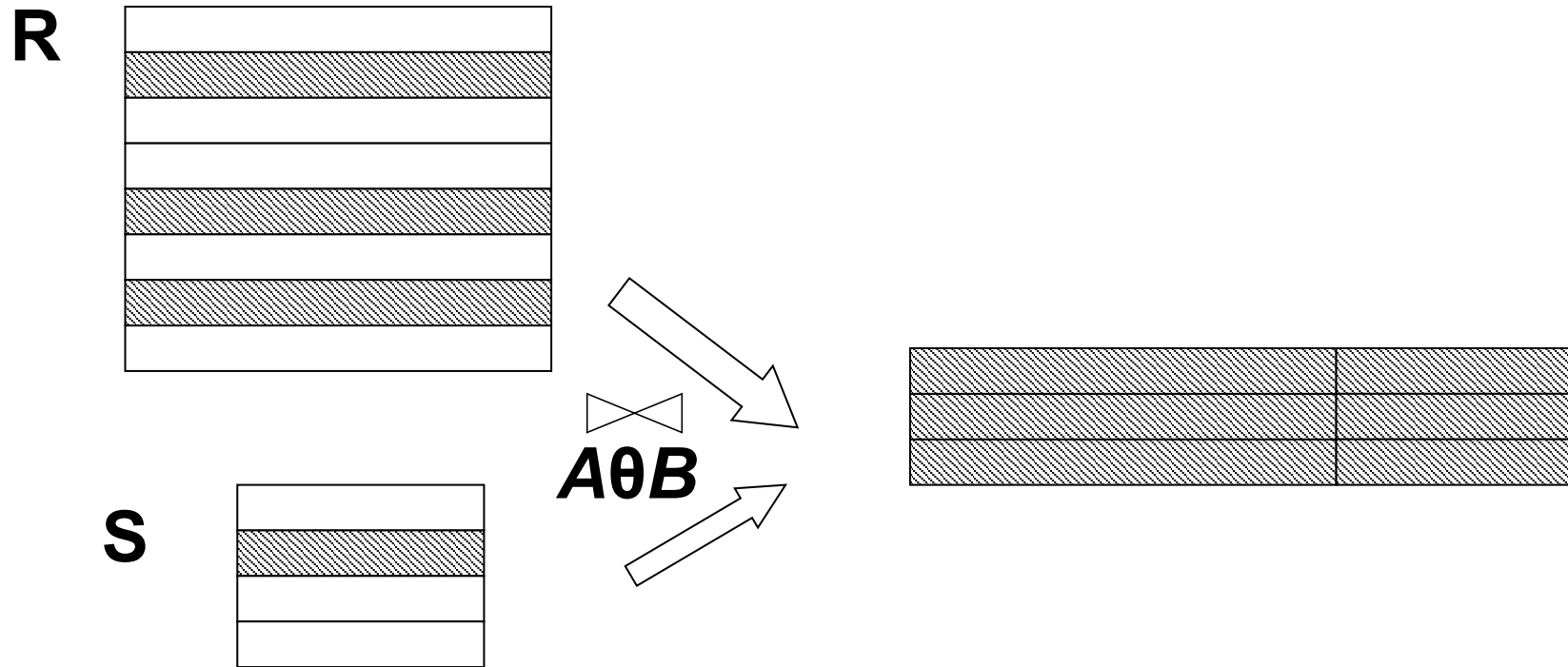
- R 和 S 含有相同的属性组 B

$$R \bowtie S = \{ \widehat{t_r t_s} [U - B] \mid t_r \in R \wedge t_s \in S \wedge t_r[B] = t_s[B] \}$$

其中： U 是两个关系所有属性的并集（包括相同的属性组 B ）

连接 4

□一般的连接操作是从行的角度进行运算。



自然连接还需要取消重复列，所以是同时从行和列的角度进行运算。

□例：关系 R 和关系 S 如下所示

R		
A	B	C
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

S	
B	E
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

连接 6

一般连接 $R \bowtie_{C < E} S$ 的结果如下:

A	R.B	C	S.B	E
a_1	b_1	5	b_2	7
a_1	b_1	5	b_3	10
a_1	b_2	6	b_2	7
a_1	b_2	6	b_3	10
a_2	b_3	8	b_3	10

连接 7

等值连接 $R \bowtie_{R.B=S.B} S$ 的结果如下:

A	R.B	C	S.B	E
a_1	b_1	5	b_1	3
a_1	b_2	6	b_2	7
a_2	b_3	8	b_3	10
a_2	b_3	8	b_3	2

连接 8

自然连接 $R \bowtie S$ 的结果如下:

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

□ 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。

□ 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

外连接 2

下图是关系*R*和关系*S*的外连接的结果: ***R* outer join *S***

A	B	C	E
<i>a</i> ₁	<i>b</i> ₁	5	3
<i>a</i> ₁	<i>b</i> ₂	6	7
<i>a</i> ₂	<i>b</i> ₃	8	10
<i>a</i> ₂	<i>b</i> ₃	8	2
<i>a</i> ₂	<i>b</i> ₄	12	NULL
NULL	<i>b</i> ₅	NULL	2

外连接 3

图(b)是关系*R*和关系*S*的左外连接, 图(c)是右外连接

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL

图(b) *R* left join *S*

A	B	C	E
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
NULL	b_5	NULL	2

图(c) *R* right join *S*

除运算 1

□除运算 (Division)

- 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:

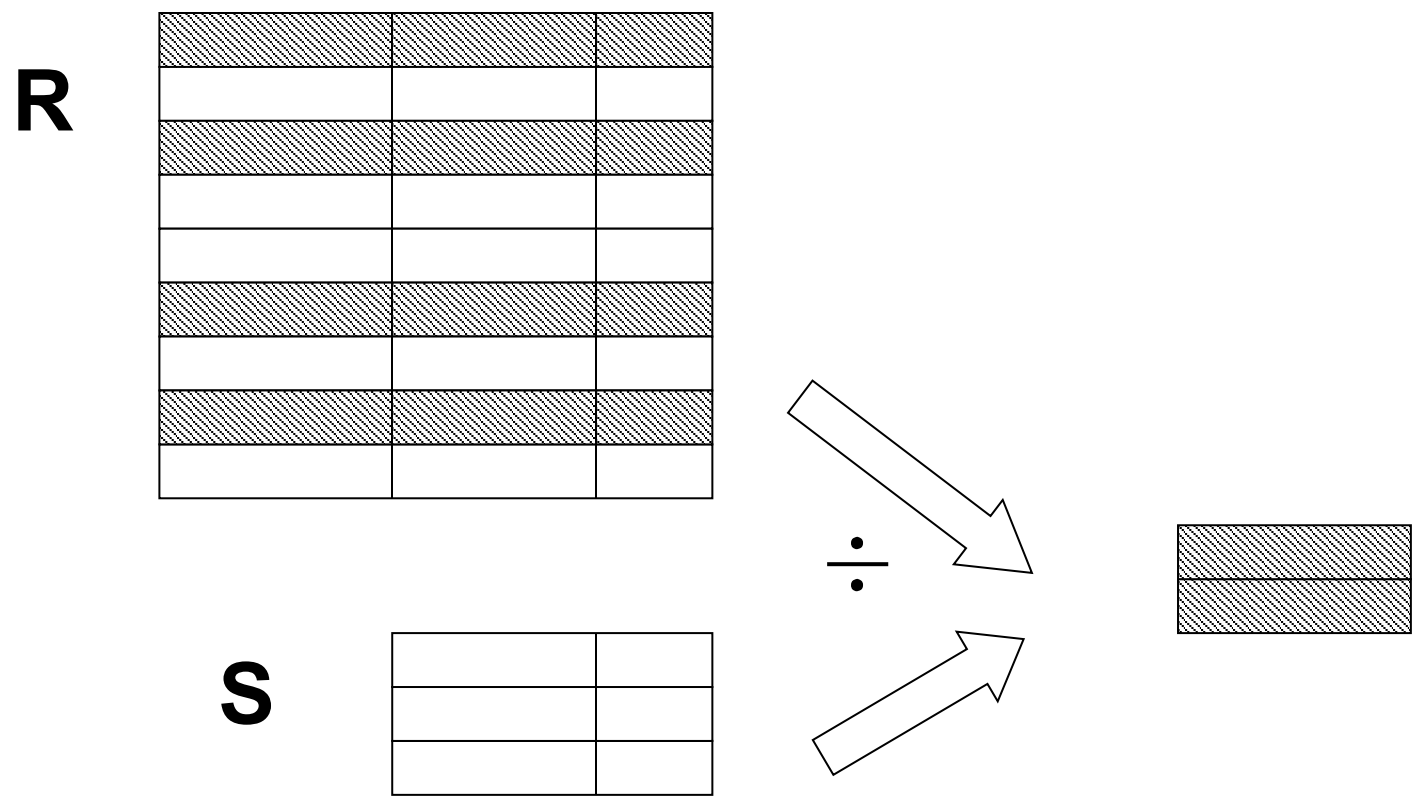
元组在 X 上分量值 x 的象集 Y_x 包含 S 在 Y 上投影的集合, 记作:

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

其中: Y_x 是 x 在 R 中的象集, $x = t_r[X]$

除运算 2

❑ 除操作是同时从行和列角度进行运算



除运算 3

例：设关系 R 、 S 分别为下图的(a)和(b)， $R \div S$ 的结果为图(c)

R

A	B	C
a_1	b_1	c_2
a_2	b_3	c_7
a_3	b_4	c_6
a_1	b_2	c_3
a_4	b_6	c_6
a_2	b_2	c_3
a_1	b_2	c_1

S

B	C	D
b_1	c_2	d_1
b_2	c_1	d_1
b_2	c_3	d_2

$R \div S$

A
a_1

除运算 4

- ❑ 在关系 R 中, A 可以取四个值 $\{a_1, a_2, a_3, a_4\}$
 - 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:
 $\{(b_1, c_2), (b_2, c_1), (b_2, c_3)\}$
- ❑ 只有 a_1 的象集包含了 S 在 (B, C) 属性组上的投影
所以 $R \div S = \{a_1\}$

R

A	B	C
a_1	b_1	c_2
a_2	b_3	c_7
a_3	b_4	c_6
a_1	b_2	c_3
a_4	b_6	c_6
a_2	b_2	c_3
a_1	b_2	c_1

S

B	C	D
b_1	c_2	d_1
b_2	c_1	d_1
b_2	c_3	d_2

综合举例 1

□ 以学生-课程数据库为例：

选修关系 SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

[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: $\pi_{Sno, Cno}(SC) \div K$

(计算过程见下一页)

综合举例 2

□ 先执行查询 $\pi_{Sno, Cno}(SC)$, 结果如右表所示. 其中:

➤ 201215121 象集为 $\{1, 2, 3\}$

➤ 201215122 象集为 $\{2, 3\}$

➤ $K = \{1, 3\}$

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

□ 于是:

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

[例2.11] 查询选修了2号课程的学生学号。

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

选修关系 SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

综合举例 4

[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

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

或

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

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

课程关系 Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

选修关系 SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

综合举例 5

[例2.13] 查询选修了全部课程的学生号码和姓名。

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

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

课程关系 Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

选修关系 SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80



复习思考题 1

1. 在关系代数中:

- ① 关系是如何表示的? 关系上的数据操纵是如何表示的?
- ② 在关系模型上进行数据操纵, 其结果是什么?

2. 关于两个关系之间的并、交、差运算

- ① 什么样的两个关系, 可以在它们之间执行并、交或差运算?
- ② 如果在两个关系之间执行并、交或差运算, 那么结果关系的关系模式和结果元组集合分别是什么?
- ③ 为什么说‘交’运算不是关系代数中的基本运算?
- ④ 在什么情况下需要使用‘差’运算? 在使用‘差’运算时需要注意什么?
- ⑤ 请比较‘差’运算和‘ \neq ’ (不等) 比较运算之间的区别。

3. 关于两个关系之间的‘笛卡尔积’ (product) 运算

- ① 什么样的两个关系, 可以在它们之间执行笛卡尔积运算?
- ② 在两个关系之间执行笛卡尔积运算, 结果关系的关系模式和结果元组集合分别是什么?
- ③ 笛卡尔积运算在关系数据库访问中有什么实际作用?

复习思考题 2

4. 设有一个公司产品零售数据库，其关系模式如下(带下划线的属性是码)：

关系名	属性集	关系模式
顾客	<u>顾客编号</u> , 姓名, 居住城市, 折扣	customers (<u>cid</u> , cname, city, discent)
供应商	<u>供应商编号</u> , 名称, 所在城市, 佣金比例	agents (<u>aid</u> , aname, city, percent)
商品	<u>商品编号</u> , 名称, 库存城市, 库存数量, 单价	products (<u>pid</u> , pname, city, stqty, price)
订单	<u>订单编号</u> , 订购日期, 顾客编号, 供应商编号, 商品编号, 订购数量, 销售金额	orders(<u>ordno</u> , orddate, cid, aid, pid, qty, dols)

在下述六组关系代数查询表达式中，请分析他们各自查询的目标对象、查询语义和结果集的相互关系。

第1组	$Q_1: \sigma_{pid='p01'}(orders)$	$Q_2: \sigma_{pid \neq 'p01'}(orders)$
第2组	$Q_3: \pi_{cid}(\sigma_{pid='p01'}(orders))$	$Q_4: \pi_{cid}(\sigma_{pid \neq 'p01'}(orders))$
第3组	$Q_5: \pi_{cid}(orders) - \pi_{cid}(\sigma_{pid='p01'}(orders))$	$Q_6: \pi_{cid}(orders) - \pi_{cid}(\sigma_{pid \neq 'p01'}(orders))$
第4组	$Q_7: \pi_{cid}(customers) - \pi_{cid}(\sigma_{pid='p01'}(orders))$	$Q_8: \pi_{cid}(customers) - \pi_{cid}(\sigma_{pid \neq 'p01'}(orders))$
第5组	$Q_9: orders - \sigma_{pid='p01'}(orders)$	$Q_{10}: orders - \sigma_{pid \neq 'p01'}(orders)$
第6组	$Q_{11}: \pi_{cid}(orders - \sigma_{pid='p01'}(orders))$	$Q_{12}: \pi_{cid}(orders - \sigma_{pid \neq 'p01'}(orders))$