

第2章关系模型与关系运算

2.1 关系模型与关系运算简述

2.2 关系与关系模型

2.3 关系代数运算

- 关系代数概述

- 关系代数的基本操作

- 用关系代数表达检索请求的示例

2.4 关系元组演算

2.5 关系域演算

2.3 关系代数运算

—关系代数概述

- 基于集合，提供了一系列的关系代数操作：并、差、笛卡尔积(广义积)、交、选择、投影、连接和关系除，是一种集合化的操作语言
- 关系代数操作以一个或多个关系为输入，结果是一个新的关系
- 用对关系的运算来表达查询，需要指明所用的操作，因此具有一定的过程性

$$\pi_{\text{姓名,课程名}}(\sigma_{\text{课程号}=c_2}(R \bowtie S))$$

- 是一种抽象的语言，是学习其他数据库语言，如SQL等的基础

2.3 关系代数运算

—关系代数基本操作概览

➤ 关系代数基本操作分为：集合操作和纯关系操作

(1) 集合操作

UNION(并)	R	S	$R \cup S$
INTERSECTION(交)	R	S	$R \cap S$
DIFFERENCE(差)	R	S	$R - S$
CARTESIAN PRODUCT(笛卡儿积)	R	S	$R \times S$

(2) 纯关系操作

PROJECT(投影)	R		$\pi_A(R)$
SELECT(选择)	R		$\sigma_{con}(R)$
JOIN(连接)	R	S	$R \bowtie S$
DIVISION(除)	R	S	$R \div S$

2.3 关系代数运算

—关系代数基本操作概览(续)

- 某些关系代数操作，如并、差、交等，需满足“并相容性”
- 并相容性
 - ❑ 参与运算的两个关系及其相关属性之间有一定的对应性、可比性或意义关联性
 - ❑ 定义：关系R与关系S存在相容性，当且仅当：
 - (1) 关系R和关系S的属性数目必须相同；
 - (2) 对于任意i，关系R的第i个属性的域必须和关系S的第i个属性的域相同
 - 假设： $R(A_1, A_2, \dots, A_n), S(B_1, B_2, \dots, B_m)$
 - R和S满足并相容性： $n = m$ 并且 $\text{Domain}(A_i) = \text{Domain}(B_i)$

2.3 关系代数运算

—关系代数基本操作概览(续)

➤ 并相容性的示例

STUDENT (SID char(10), Sname char(8), Age char(3))

PROFESSOR (PID char(10), Pname char(8), Age char(3))

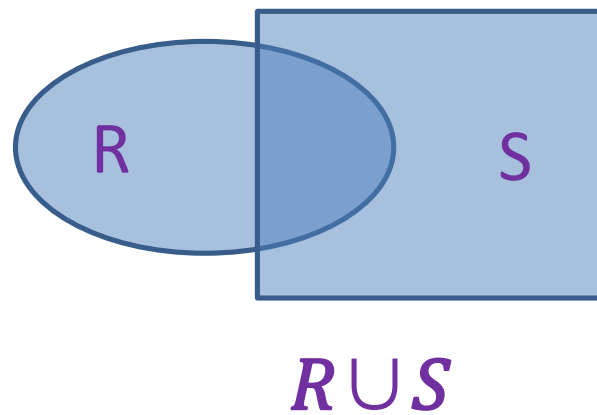
关系STUDENT与关系PROFESSOR是相容的，因为：

- (1) 关系R和关系S的属性数目都是3
- (2) 关系R的属性SID与关系S的属性PID的域都是char(10)
- (3) 关系R的属性Sname与关系S的属性Sname的域都是char(8)
- (4) 关系R的属性Age与关系S的属性Age的域都是char(3)

2.3 关系代数运算

—关系代数之集合操作：并(Union)

- **定义：**假设关系R和关系S是并相容的, 则关系R与关系S的并运算结果也是一个关系, 记作: $R \cup S$, 它由或者出现在关系R中, 或者出现在S中的元组构成。
- **数学描述:** $R \cup S = \{t | t \in R \vee t \in S\}$, 其中t是元组。
- 并运算是将两个关系的元组合并成一个关系, 在合并时去掉重复的元组。
- $R \cup S$ 与 $S \cup R$ 运算的结果是同一个关系。



2.3 关系代数运算

—关系代数之集合操作：并(Union) (续)

➤ 并操作的示例一 (抽象的)

· 假设R与S是并相容的两个关系

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

R∪S		
C1	C2	C3
a	b	c
a	d	g
f	b	e
a	b	e
h	d	g

2.3 关系代数运算

—关系代数之集合操作：并(Union) (续)

➤ 并操作的示例二 (语义的)

- 查询参加体育队或者参加文艺队，或者体育队和文艺队都参加学生的信息 ($R \cup S$)

R(参加体育
队的学生)

姓名	性别
张三	女
李四	男
王五	男

S(参加文艺
队的学生)

姓名	性别
张三	女
刘六	男
钱七	男

$R \cup S$

姓名	性别
张三	女
李四	男
王五	男
刘六	男
钱七	男



2.3 关系代数运算

—关系代数之集合操作：并(Union) (续)

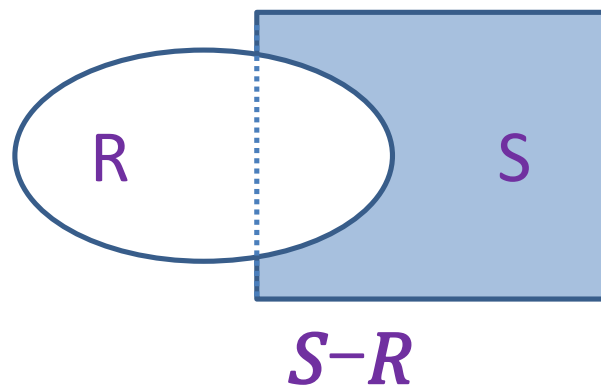
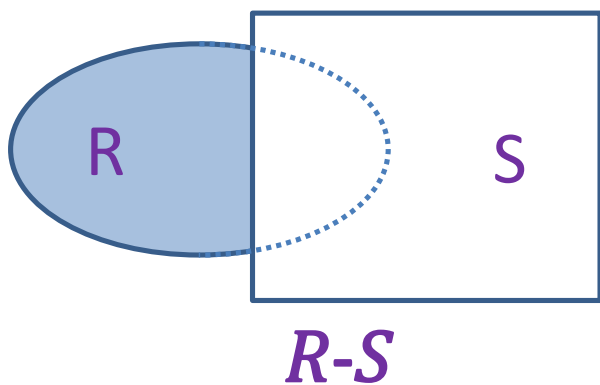
➤ 并操作的示例三(语义的)

- 若R为计算机学院的学生，S为材料学院的学生，则 $R \cup S$ 为两院所有的学生
- 若R为学过数据库课程的学生，S为学过自控理论课程的学生，则 $R \cup S$ 为学过两门课之一或者两门课都学过的所有学生
- 汉语中的“A或者B或者AB”通常意义是并运算的要求。首先要准确理解汉语的查询要求，然后再找到正确的操作

2.3 关系代数运算

—关系代数之集合操作：差(Difference)

- 定义：假设关系R和关系S是并相容的，则关系R与关系S的差运算结果也是一个关系，记作： $R-S$ ，它由出现在关系R中但不出现在关系S中的元组构成。
- 数学描述： $R - S = \{t | t \in R \wedge t \notin S\}$ ，其中t是元组
- $R-S$ 与 $S-R$ 是不同的



2.3 关系代数运算

—关系代数之集合操作：差(Difference) (续)

➤ 差操作的示例一(抽象的)

· 假设R与S是并相容的两个关系

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

R-S		
D1	D2	D3
f	b	e

S-R		
E1	E2	E3
a	b	e
h	d	g

2.3 关系代数运算

—关系代数之集合操作：差(Difference) (续)

➤ 差操作的示例二(语义的)

- 查询只参加体育队而未参加文艺队的学生信息 ($R-S$)
- 查询只参加文艺队而未参加体育队的学生信息 ($S-R$)

R(参加体育
队的学生)

姓名	性别
张三	女
李四	男
王五	男

S(参加文艺
队的学生)

姓名	性别
张三	女
刘六	男
钱七	男

$R-S$

姓名	性别
李四	男
王五	男

$S-R$

姓名	性别
刘六	男
钱七	男

2.3 关系代数运算

—关系代数之集合操作：差(Difference) (续)

➤ 差操作的示例三(语义的)

· 若R为计算机学院的学生, S为四年级的学生,

$R-S$ 为计算机学院但不是四年级的所有学生

(计算机学院非四年级的学生)

$S-R$ 为四年级但不是计算机学院的所有学生

(四年级非计算机学院的学生)

· 若R为学过数据库课程的学生, S为学过自控理论课程的学生,

$R-S$ 为学过数据库课程但没学过自控理论课程的所有学生

· 汉语中的“是…但不含…”通常意义是差运算的要求。首先要准确理解汉语的查询要求, 然后再找到正确的操作

2.3 关系代数运算

—关系代数之集合操作：交 (Intersection)

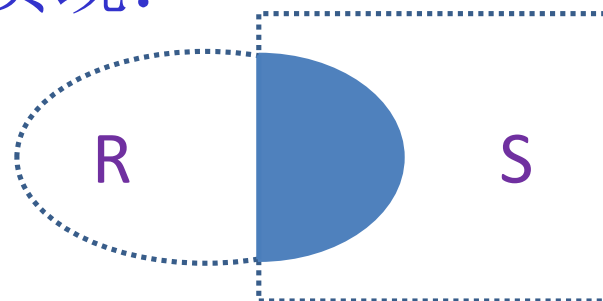
➤ **定义：** 假设关系R和关系S是并相容的，则关系R与关系S的交运算结果也是一个关系，记作： $R \cap S$ ，它由同时出现在关系R和关系S中的元组构成。

➤ **数学描述：** $R \cap S = \{t | t \in R \wedge t \in S\}$ ，其中t是元组

➤ $R \cap S$ 和 $S \cap R$ 运算的结果是同一个关系

➤ 交运算可以通过差运算来实现：

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



$$R \cap S = S \cap R$$

2.3 关系代数运算

—关系代数之集合操作：交 (Intersection) (续)

➤ 交操作的示例一 (抽象的)

· 假设R与S是并相容的两个关系

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

$R \cap S$		
F1	F2	F3
a	b	c
a	d	g

2.3 关系代数运算

—关系代数之集合操作：交 (Intersection) (续)

➤ 交操作的示例二 (语义的)

· 查询既参加体育队又参加文艺队的学生信息 ($R \cap S$)

R (参加体育
队的学生)

姓名	性别
张三	女
李四	男
王五	男

S (参加文艺
队的学生)

姓名	性别
张三	女
刘六	男
钱七	男

$R \cap S = S \cap R$

姓名	性别
张三	女



2.3 关系代数运算

—关系代数之集合操作：交 (Intersection) (续)

➤ 交操作的示例三 (语义的)

- 若R为年龄小于20岁的学生, S为计算机学院的学生, 则 $R \cap S$ 为计算机学院并且年龄小于20岁的所有学生
- 若R为学过数据库课程的学生, S为学过自控理论课程的学生, 则 $R \cap S$ 为既学过数据库课程又学过自控理论课程的所有学生
- 汉语中的“既…又…”, “…, 并且…”通常意义是交运算的要求, 首先要准确理解汉语的查询要求, 然后再找到正确的操作

2.3 关系代数运算

—关系代数之集合操作：广义笛卡尔积
(Cartesian Product)

➤ 定义：关系 $R(\langle a_1, a_2, \dots, a_n \rangle)$ 与关系 $S(\langle b_1, b_2, \dots, b_m \rangle)$ 的广义笛卡尔积(简称广义积)运算结果也是一个关系，记作： $R \times S$ ，它由关系 R 中的元组与关系 S 的元组进行所有可能的拼接(或串接)构成。

➤ 数学描述： $R \times$

$$S = \left\{ \langle a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m \rangle \mid \langle a_1, a_2, \dots, a_n \rangle \in R \wedge \langle b_1, b_2, \dots, b_m \rangle \in S \right\}$$

2.3 关系代数运算

—关系代数之集合操作：

广义笛卡尔积(续)

➤ 广义积操作的示例一(抽象的)

· 关系R的元组数目是3, 度数是3;

关系S的元组数目是4, 度数是3;

则 $R \times S$ 的元组数目是12, 度数是6

2.3 关系代数运算

—关系代数之集合操作：
广义笛卡尔积(续)

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

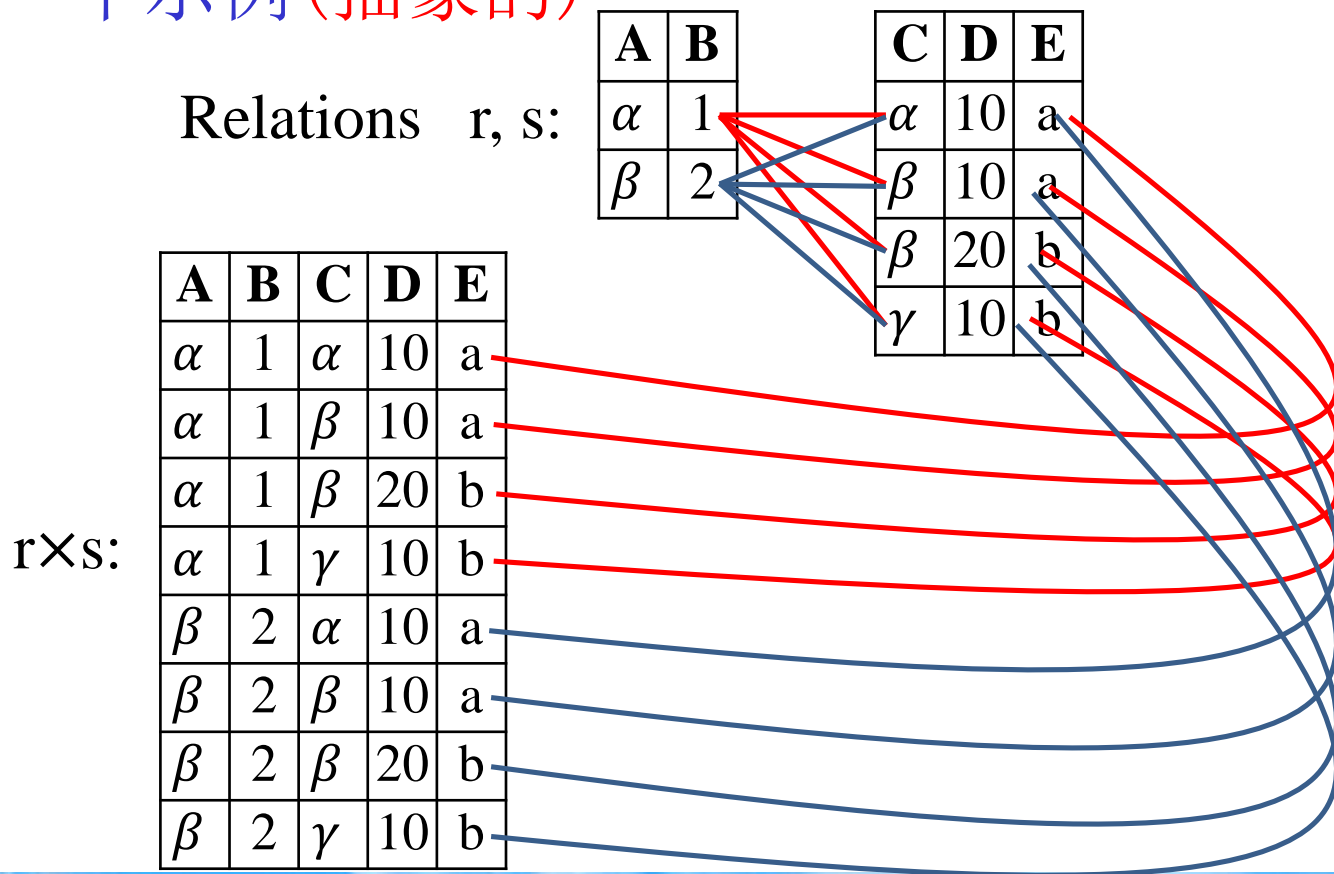
S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

R×S					
A1	A2	A3	B1	B2	B3
a	b	c	a	b	c
a	b	c	a	b	e
a	b	c	a	d	g
a	b	c	h	d	g
a	d	g	a	b	c
a	d	g	a	b	e
a	d	g	a	d	g
a	d	g	h	d	g
f	b	e	a	b	c
f	b	e	a	b	e
f	b	e	a	d	g
f	b	e	h	d	g

2.3 关系代数运算

—关系代数之集合操作：广义笛卡尔积(续)

➤再看一个示例(抽象的)



2.3 关系代数运算

—关系代数之集合操作：广义笛卡尔积(续)

➤ 广义积操作的示例之二(语义的)

- 当一个检索涉及到多个表时(如学生表和课程表)，便需要将这些表串接或拼接起来，然后才能检索，这时，就要使用广义笛卡尔积运算
- 是后面学习各种连接运算的基础



2.3 关系代数运算

—关系代数之集合操作：广义笛卡尔积(续)

➤ 广义积操作的示例之二(语义的)

学生表 (所有学生) 的课程表 (所有课程)

学号	姓名	年龄	宿舍
01	李三	21	101
02	李四	21	102
03	李五	20	102

课程表

课程号	课程名	教师	学时
C1	几何	A	40
C2	物理	B	40
C3	代数	C	60

学号	姓名	年龄	住址	课程号	课程名	教师	学时
01	李三	21	101	C1	几何	A	40
01	李三	21	101	C2	物理	B	40
01	李三	21	101	C3	代数	C	60
02	李四	21	102	C1	几何	A	40
02	李四	21	102	C2	物理	B	40
02	李四	21	102	C3	代数	C	60
03	李五	20	102	C1	几何	A	40
03	李五	20	102	C2	物理	B	40
03	李五	20	102	C3	代数	C	60

2.3 关系代数运算

—关系代数之集合操作：广义笛卡尔积(续)

- $R \times S = S \times R$: $R \times S$ 为R中的每一个元组都和S中的所有元组进行串接。 $S \times R$ 为S中的每一个元组都和R中的所有元组进行串接。结果是相同的。
- 两个关系R和S，它们的属性个数分别为n和m(R是n度关系，S是m度关系)，则笛卡尔积 $R \times S$ 的属性个数为 $n + m$ ，即元组的前n个分量是R中元组的分量，后m个分量是S中元组的分量($R \times S$ 是 $n + m$ 度关系)。
- 两个关系R和S，它们的元组个数分别为x和y(关系R的基数x，S的基数y)，则笛卡尔积 $R \times S$ 的元组个数为 $x \times y$ ($R \times S$ 的基数是 $x \times y$)。

2.3 关系代数运算

—关系代数之纯关系操作：选择(Select)

➤ 定义：给定一个关系R, 同时给定一个选择的条件condition(简记con), 选择运算结果也是一个关系, 记作 $\sigma_{con}(R)$, 它从关系R中选择出满足给定条件condition的元组构成。

➤ 数学描述： $\sigma_{con}(R) = \{t | t \in R \wedge con(t) = '真'\}$

□ 设 $R(A_1, A_2, \dots, A_n)$, t 是R的元组, t 的分量记为 $t[A_i]$, 或简写为 A_i

□ 条件con由逻辑运算符连接算术表达式组成

□ 逻辑运算符： \wedge, \vee, \neg 或写为and, or, not

□ 算术表达式： $X \theta Y$, 其中 X, Y 是 t 的分量、常量或简单函数, θ 是比较运算符, $\theta \in \{>, \geq, <, \leq, =, \neq\}$

R		
A1	A2	A3

2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

➤ 选择操作的示例一(抽象的)

R		
A1	A2	A3
a	a	10
b	d	-4
f	b	5

$\sigma_{A3>0}(R)$		
A1	A2	A3
a	a	10
f	b	5

$\sigma_{A1=a \vee A2=b}(R)$		
A1	A2	A3
a	a	10
f	b	5

$\sigma_{A3>0 \wedge A1=A2}(R)$		
A1	A2	A3
a	a	10



2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

➤ 选择操作的示例二(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

查询所有男同学的信息 $\sigma_{Ssex="男"}(R)$

S#	Sname	Ssex	Sage	D#
1110102	李四	男	20	02
1110103	王五	男	19	03

2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

➤ 选择操作的示例二(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

查询所有年龄小于20同学的信息 $\sigma_{Sage < 20}(R)$

S#	Sname	Ssex	Sage	D#
1110103	王五	男	19	03

2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

➤ 选择操作的示例二(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

查询所有1系或3系的同学信息 $\sigma_{D\#="01" \vee D\#="03"}(R)$

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110103	王五	男	19	03



2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

➤ 选择操作的示例三(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

查询所有年龄大于20的1系同学的信息 $\sigma_{Sage>20 \wedge D\#="01"}(R)$

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01

2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

► 选择操作的示例三(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

查询不在(年龄大于20的1系同学)要求之内的
所有其它同学的信息 $\sigma_{\neg(Sage>20 \wedge D\#="01")}(R)$

S#	Sname	Ssex	Sage	D#
1110102	李四	男	20	02
1110103	王五	男	19	03

2.3 关系代数运算

—关系代数之纯关系操作：选择(Select) (续)

➤ 选择操作的示例四(语义的)

➤ 选择操作从给定的关系中选出满足条件的行，条件的书写很重要，尤其是当不同运算符在一起时，要注意运算符的优先次序，

➤ 优先次序自高至低为{括弧； θ ； \neg ； \wedge ； \vee }

· 例如：

$\text{Sage} < 20 \vee \text{Sage} > 18 \wedge D\# = "03"$

与

$(\text{Sage} < 20 \vee \text{Sage} > 18) \wedge D\# = "03"$

2.3 关系代数运算

—关系代数之纯关系操作：投影(Project)

- 定义：给定一个关系R, 投影运算结果也是一个关系, 记作 $\Pi_A(R)$, 它从关系R中选出属性包含在A中的列构成。
- 数学描述：

$$\Pi_{A_{i1}, A_{i2}, \dots, A_{ik}}(R) = \{ \langle t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}] \rangle \mid t \in R \}$$

- 设 $R(A_1, A_2, \dots, A_n)$,
- $\{A_{i1}, A_{i2}, \dots, A_{ik}\} \subseteq \{A_1, A_2, \dots, A_n\}$
- $t[A_i]$ 表示元组t中相应于属性 A_i 的分量
- 投影运算可以对原关系的列在投影后重新排列
- 投影操作从给定关系中选出某些列组成新的关系, 而选择操作是从给定关系中选出某些行组成新的关系。

R		
A1	A2	A3

2.3 关系代数运算

—关系代数之纯关系操作：投影(Project) (续)

➤ 投影操作的示例一(抽象的)

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

$\Pi_{A3}(R)$
A3
c
g
e

$\Pi_{A3, A1}(R)$	
A3	A1
c	a
g	a
e	f

2.3 关系代数运算

—关系代数之纯关系操作：投影(Project) (续)

➤ 如果投影后有重复元组，则应去掉

R		
A1	A2	A3
a	a	c
a	d	c
f	b	c

$\Pi_{A1, A3}(R)$	
A3	A1
a	c
f	c

2.3 关系代数运算

--关系代数之纯关系操作：投影(Project) (续)

➤ 投影操作的示例二(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

$\Pi_{\text{Sname, Sage}}(R)$

查询所有学生的姓名和年龄

$\Pi_{\text{Sname, D\#}}(R)$

查询所有学生的姓名及其所在的系

2.3 关系代数运算

--关系代数之纯关系操作：投影(Project) (续)

➤ 投影操作的示例二(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03

查询所有学生的姓名和年龄

$\Pi_{Sname, Sage}(R)$	
Sname	Sage
张三	21
李四	20
王五	19

查询所有学生的姓名及其所在的系

$\Pi_{Sname, D\#}(R)$	
Sname	D#
张三	01
李四	02
王五	03

2.3 关系代数运算

--关系代数之纯关系操作：投影(Project) (续)

➤ 投影与选择操作一起使用的示例(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03
1110104	刘六	女	21	03

查询所有在3系就读的且年龄
大于19的学生的学号和姓名

$\Pi_{S\#, Sname}(\sigma_{D\#="03" \wedge Sage > 19}(R))$

$\Pi_{Sname, Sage}(R)$

2.3 关系代数运算

--关系代数之纯关系操作：投影(Project) (续)

➤ 投影与选择操作一起使用的示例(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03
1110104	刘六	女	21	03

查询所有在3系就读的且年龄大于19的学生的学号和姓名

$\Pi_{S\#,Sname}(\sigma_{D\#="03" \wedge Sage > 19}(R))$

$\Pi_{Sname, Sage}(R)$	
S#	Sname
1110104	刘六

2.3 关系代数运算

--关系代数之纯关系操作：投影(Project) (续)

➤ 投影与选择操作一起使用的示例(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03
1110104	刘六	女	21	03

查询所有在3系就读的
且男同学的学号和姓名

$$\Pi_{Sname, Sage}(R)$$
$$\Pi_{S\#, Sname}(\sigma_{D\#="03" \wedge Ssex="男"}(R))$$

2.3 关系代数运算

—关系代数之纯关系操作：投影(Project) (续)

➤ 投影与选择操作一起使用的示例(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03
1110104	刘六	女	21	03

查询所有在3系就读的
且男同学的学号和姓名

$\Pi_{S\#,Sname}(\sigma_{D\#="03" \wedge Ssex="男"}(R))$

$\Pi_{Sname, Sage}(R)$	
S#	Sname
1110103	王五

2.3 关系代数运算

—关系代数之纯关系操作： θ -连接 (θ -Join)

- 投影与选择操作只是对单个关系(表)进行操作, 而实际应用中往往涉及多个表之间的操作
- 比如: 查询数据结构成绩在80分及以上的学生姓名 (涉及学生, 科目, 成绩), 这就需要 θ -连接操作

数据库系统

查询数据结构成绩在
80分及以上的学生姓名

学生表

S#	Sname	Ssex
1001	张三	女
1002	李四	男
1003	王五	男
1004	刘六	女

科目表

C#	Cname	Chours	Credit
001	数据库	40	4
002	数据结构	60	4
003	C语言	40	2
004	编译原理	60	4

成绩表

S#	C#	Score
1001	001	80
1001	002	85
1001	003	83
1001	004	88
1002	001	78
1002	002	80
1002	003	82
1002	004	85
1003	001	82
1003	002	68
1003	003	79
...



2.3 关系代数运算

—关系代数之纯关系操作： θ -连接(θ -Join) (续)

➤ 定义：给定关系R和关系S，R与S的 θ 连接运算结果也是一个关系，记作 $R \bowtie_{\theta} S$ ，它由关系R和关系S的笛卡尔积中，选取R中属性A与S中属性B之间满足条件的元组构成。

➤ 数学描述： $R \bowtie_{\theta} S = \sigma_{t[A] \theta s[B]}(R \times S)$

- 设 $R(A_1, A_2, \dots, A_n)$, $A \in \{A_1, A_2, \dots, A_n\}$
- $S(B_1, B_2, \dots, B_m)$, $B \in \{B_1, B_2, \dots, B_m\}$
- t 是关系R中的元组， s 是关系S中的元组
- 属性A和属性B具有可比性
- θ 是比较运算符， $\theta \in \{>, \geq, <, \leq, =, \neq\}$

➤ 在实际应用中， θ -连接操作经常与投影、选择操作一起使用

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接 (θ -Join) 操作的示例一 (抽象的)

R	
A	B
a	1
b	2

S	
H	C
1	x
1	y
3	z

$R \times S$			
A	B	H	C
a	1	1	x
a	1	1	y
a	1	3	z
b	2	1	x
b	2	1	y
b	2	3	z

$R \bowtie_{B \leq H} S$			
A	B	H	C
a	1	1	x
a	1	1	y
a	1	3	z
b	2	3	z



2.3 关系代数运算

—关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例二 (语义的)

- 员工表Worker (W#, Wname, Wsex, Wage, Degree),
- 职位限定表Position (Type, Limited_Degree (LD))

员工表Worker

W#	Wname	Wsex	Wage	Degree
01	张三	男	35	1
02	李四	男	49	3
03	王五	女	45	2

职位限定表Position

Type	LD
组长	1
项目经理	2
部门经理	3

1:本科 2:硕士 3:博士

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例二 (语义的) (续)

- 竞聘的岗位必须由不低于其最低学历要求的人员担任，找出所有员工的姓名及其可能竞聘职位的名称)

$\Pi_{\text{wname,type}}(\text{worker} \bowtie_{\text{Degree} \geq \text{LD}} \text{position})$

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例二 (续)

第一步：对两个表进行广义笛卡尔积

(所有员工)的(所有职位) ← 广义笛卡尔积

W#	Wname	Wsex	Wage	Degree	Type	LD
01	张三	男	35	1	组长	1
01	张三	男	35	1	项目经理	2
01	张三	男	35	1	部门经理	3
02	李四	男	49	3	组长	1
02	李四	男	49	3	项目经理	2
02	李四	男	49	3	部门经理	3
03	王五	女	45	2	组长	1
03	王五	女	45	2	项目经理	2
03	王五	女	45	2	部门经理	3

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例二 (续)

第二步：从广义笛卡尔积中选取符合条件的元组
(所有员工)的(所有职位)  广义笛卡尔积

W#	Wname	Wsex	Wage	Degree	Type	LD
01	张三	男	35	1	组长	1
01	张三	男	35	1	项目经理	2
01	张三	男	35	1	部门经理	3
02	李四	男	49	3	组长	1
02	李四	男	49	3	项目经理	2
02	李四	男	49	3	部门经理	3
03	王五	女	45	2	组长	1
03	王五	女	45	2	项目经理	2
03	王五	女	45	2	部门经理	3

Degree \geq LD

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例二 (续)

第二步：从广义笛卡尔积中选取符合
(degree \geq limited_degree)条件的元组

所有员工及其可以竞聘的职位

W#	Wname	Wsex	Wage	Degree	Type	LD
01	张三	男	35	1	组长	1
02	李四	男	49	3	组长	1
02	李四	男	49	3	项目经理	2
02	李四	男	49	3	部门经理	3
03	王五	女	45	2	组长	1
03	王五	女	45	2	项目经理	2

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例二 (续)

第三步：在进行投影操作，得到最终的结果

$$\Pi_{\text{wname,type}}(\text{worker} \bowtie_{\text{Degree} \geq \text{LD}} \text{position})$$

最终想要得到的结果

Wname	Type
张三	组长
李四	组长
李四	项目经理
李四	部门经理
王五	组长
王五	项目经理

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例三 (续)

- 关系与自身的 θ -连接
- 查询1001号同学和1002号同学都学过的所有课程的课程号

$$\Pi_{sc.c\#}(\sigma_{sc.s\#="1001" \wedge sc1.s\#"1002"}(sc \bowtie_{sc.c\# = SC1.C\#} \rho_{sc1}(sc)))$$

- 注：上式 $\rho_{sc1}(sc)$ 表更名操作，即将表SC更名为SC1，当一个表需要和其自身进行连接运算时，通常要使用更名操作

SC

S#	C#	Score
1001	001	80
1001	002	85
1001	003	83
1002	001	78
1002	004	80
1002	005	82
1003	005	81
1003	006	68
1003	007	79
...

$$SC \bowtie \rho_{SC1}(SC)$$

SC.C# = SC1.C#

SC. S#	SC. C#	SC. Score	SC1. S#	SC1. C#	SC1. Score
1001	001	80	1001	001	80
1001	001	80	1002	001	78
1002	001	78	1001	001	80
1002	001	78	1002	001	78
1002	005	82	1002	005	82
1002	005	82	1003	005	81
1003	005	81	1002	005	82
1003	005	81	1003	005	81
...

$$SC \quad \Pi_{sc.c\#}(\sigma_{sc.s\#="1001" \wedge sc1.s\#"1002"}(SC \bowtie \rho_{sc1}(SC)))$$

$SC.C\# = SC1.C\#$

S#	C#	Score
1001	001	80
1001	002	85
1001	003	83
1002	001	78
1002	004	80
1002	005	82
1003	005	81
1003	006	68
1003	007	79
...

SC. S#	SC. C#	SC. Score	SC1. S#	SC1. C#	SC1. Score
1001	001	80	1002	001	78
...

2.3 关系代数运算

--关系代数之纯关系操作： θ -连接 (θ -Join) (续)

➤ θ -连接操作的示例三 (续)

- 关系与自身的 θ -连接
- 查询1001号同学和1002号同学都学过的所有课程的课程号

$$\Pi_{sc.c\#}(\sigma_{sc.s\#="1001" \wedge sc1.s\#"1002"}(sc \bowtie_{sc.c\# = SC1.C\#} \rho_{sc1}(sc)))$$

两个同学都学过的
所有课程的课程号：

SC.C#
001

2.3 关系代数运算

—关系代数之纯关系操作： θ -连接 (θ -Join) (续)

- 虽然我们在讲解 θ -连接操作时，使用笛卡尔积然后再进行选择来得到 θ -连接结果。这主要是方便大家理解。但当引入连接操作后，DBMS可直接进行连接操作，而不必先形成笛卡尔积

2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

➤ **定义**：给定关系R和关系S, R与S的等值连接运算结果也是一个关系,记作 $R \bowtie_{A=B} S$ ，它由关系R和关系S的笛卡尔积中选取R中属性A与S中属性B上值相等的元组所构成。

➤ **数学描述**： $R \bowtie_{A=B} S = \sigma_{t[A]=s[B]}(R \times S)$

➤ 当 **θ -连接**中运算符为“=”时，就是等值连接，等值连接是 **θ -连接**的一个特例；

➤ 广义积的元组组合并不是都有意义的,另广义积的元组组合数目也非常庞大,因此采用 **θ -连接/等值连接**运算可大幅度降低中间结果的保存量,提高速度。

2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

➤ 等值连接(Equi-Join)操作的示例一(抽象的)

R	
A	B
a	1
b	2

S	
H	C
1	x
1	y
3	z

$R \times S$			
A	B	H	C
a	1	1	x
a	1	1	y
a	1	3	z
b	2	1	x
b	2	1	y
b	2	3	z

$R \bowtie_{B=H} S$			
A	B	H	C
a	1	1	x
a	1	1	y

2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

➤ 等值连接(Equi-Join)操作的示例二(语义的)

· 员工表

Worker(W#, Wname, Wsex, Wage, Honor_type(Ht)),

· 获奖类别表Honor(Type, Title)

员工表Worker

W#	Wname	Wsex	Wage	Ht
01	张三	男	35	1
02	李四	女	49	3
03	王五	女	45	2

获奖类别表Honor


Type	Title
1	全国劳模
2	“五一”奖章获得者
3	“三八”妇女红旗手

2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

- 找出所有获奖员工姓名、年龄及其获奖的名称

$\Pi_{\text{wname}, \text{Wage}, \text{Title}}(\text{Worker} \bowtie_{\text{Ht}=\text{Type}} \text{Honor})$

(所有员工)的(所有获奖)  广义笛卡儿积

W#	Wname	Wsex	Wage	Ht	Type	Title
01	张三	男	35	1	1	全国劳模
01	张三	男	35	1	2	“五一”奖章获得者
01	张三	男	35	1	3	“三八”妇女红旗手
02	李四	女	49	3	1	全国劳模
02	李四	女	49	3	2	“五一”奖章获得者
02	李四	女	49	3	3	“三八”妇女红旗手
03	王五	女	45	2	1	全国劳模
03	王五	女	45	2	2	“五一”奖章获得者
03	王五	女	45	2	3	“三八”妇女红旗手


第一步：
对两个表
进行广义
笛卡尔积

2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

- 找出所有获奖员工姓名、年龄及其获奖的名称

$\Pi_{\text{wname}, \text{Wage}, \text{Title}}(\text{Worker} \bowtie_{\text{Ht}=\text{Type}} \text{Honor})$

(所有员工)的(所有获奖)  广义笛卡儿积

W#	Wname	Wsex	Wage	Ht	Type	Title
01	张三	男	35	1	1	全国劳模
01	张三	男	35	1	2	“五一”奖章获得者
01	张三	男	35	1	3	“三八”妇女红旗手
02	李四	女	49	3	1	全国劳模
02	李四	女	49	3	2	“五一”奖章获得者
02	李四	女	49	3	3	“三八”妇女红旗手
03	王五	女	45	2	1	全国劳模
03	王五	女	45	2	2	“五一”奖章获得者
03	王五	女	45	2	3	“三八”妇女红旗手

Ht=Type

第二步：
从广义笛
卡尔积中
选取出符
合条件的
元组

2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

- 找出所有获奖员工姓名、年龄及其获奖的名称

$\Pi_{\text{wname, Wage, Title}}(\text{Worker} \bowtie_{\text{Ht=Type}} \text{Honor})$

第二步：从广义笛卡尔积中选出符合(Ht=Type)条件的元组

所有员工及其所获奖的信息

W#	Wname	Wsex	Wage	Ht	Type	Title
01	张三	男	35	1	1	全国劳模
02	李四	女	49	3	3	“三八” 妇女红旗手
03	王五	女	45	2	2	“五一” 奖章获得者



2.3 关系代数运算

—关系代数之纯关系操作：等值连接(Equi-Join)

- 找出所有获奖员工姓名、年龄及其获奖的名称

$\Pi_{\text{wname, Wage, Title}}(\text{Worker} \bowtie_{\text{Ht=Type}} \text{Honor})$

第三步：在(Wname, Wage, Title)上
进行投影运算，得到最终结果

所有获奖员工
姓名、年龄及
其获奖的名称：

Wname	Wage	Title
张三	35	全国劳模
李四	49	“三八” 妇女红旗手
王五	45	“五一” 奖章获得者

2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 定义：给定关系R和关系S, R与S的自然连接运算结果也是一个关系，记作 $R \bowtie S$, 它由关系R和关系S的笛卡尔积中选取相同属性组B上值相等的元组所构成。

➤ 数学描述： $R \bowtie S = \sigma_{t[B]=s[B]}(R \times S)$

- ❑ 自然连接是一种特殊的等值连接
- ❑ 要求关系R和关系S必须有相同的属性组B(如R, S共有一个属性B1, 则B是B1; 如R, S共有一组属性B1, B2, ..., Bn, 则B是这些共有的所有属性)
- ❑ R, S属性相同, 值必须相等才能连接, 即 $R.B1=S.B1$ and $R.B2=S.B2 \cdots$ and $R.Bn=S.Bn$ 才能连接
- ❑ 要在结果中去掉重复的属性列(因结果中 $R.Bi$ 始终是等于 $S.Bi$ 所以可只保留一列即可)

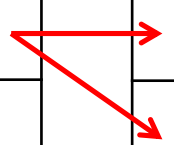
2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 自然连接(Natural-Join)操作的示例一(抽象的)

R	
A	B
a	1
b	2

S	
B	C
1	x
1	y
3	z



$R \times S$			
A	B	B	C
a	1	1	x
a	1	1	y
a	1	3	z
b	2	1	x
b	2	1	y
b	2	3	z

$R \bowtie S$		
A	B	C
a	1	x
a	1	y



2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 自然连接(Natural-Join)操作的示例二(语义的)

- 学生选课表SC(S#, C#, Score)
- 课程表Course(C#, Cname, Chours, Credit)

学生选课表SC

S#	C#	Score
1001	001	80
1002	002	85
1003	003	83

课程表Course

C#	Cname	Chours	Credit
001	数据库	40	4
002	数据结构	60	4
003	C语言	40	2

2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 自然连接(Natural-Join)操作的示例二(语义的)

· 查询所有学生选课的成绩(包括学号, 课程名称, 成绩)

$\Pi_{S\#,Cname,Score}(SC \bowtie Course)$

广义笛卡尔积

S#	C#	Score	C#	Cname	Chours	Credit
1001	001	80	001	数据库	40	4
1001	001	80	002	数据结构	60	4
1001	001	80	003	C语言	40	2
1002	002	85	001	数据库	40	4
1002	002	85	002	数据结构	60	4
1002	002	85	003	C语言	40	2
1003	003	83	001	数据库	40	4
1003	003	83	002	数据结构	60	4
1003	003	83	003	C语言	40	2

第一步：
对两个表
进行广义
笛卡尔积

2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 自然连接(Natural-Join)操作的示例二(语义的)

· 查询所有学生选课的成绩(包括学号, 课程名称, 成绩)

$\Pi_{S\#,Cname,Score}(SC \bowtie Course)$

广义笛卡尔积

S#	C#	Score	C#	Cname	Chours	Credit
1001	001	80	001	数据库	40	4
1001	001	80	002	数据结构	60	4
1001	001	80	003	C语言	40	2
1002	002	85	001	数据库	40	4
1002	002	85	002	数据结构	60	4
1002	002	85	003	C语言	40	2
1003	003	83	001	数据库	40	4
1003	003	83	002	数据结构	60	4
1003	003	83	003	C语言	40	2

第二步:

从广义笛卡尔积中选取
在相同列
(C#)上值相
同的元组

2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 自然连接(Natural-Join)操作的示例二(语义的)

· 查询所有学生选课的成绩(包括学号, 课程名称, 成绩)

$\Pi_{S\#,Cname,Score}(SC \bowtie Course)$

第二步：从广义笛卡尔积中选取在相同列(C#)上值相同的元组

S#	C#	Score	C#	Cname	Chours	Credit
1001	001	80	001	数据库	40	4
1002	002	85	002	数据结构	60	4
1003	003	83	003	C语言	40	2

第三步：去掉重复的列

S#	C#	Score	Cname	Chours	Credit
1001	001	80	数据库	40	4
1002	002	85	数据结构	60	4
1003	003	83	C语言	40	2

2.3 关系代数运算

—关系代数之纯关系操作：自然连接(Natural-Join)

➤ 自然连接(Natural-Join)操作的示例二(语义的)

· 查询所有学生选课的成绩(包括学号, 课程名称, 成绩)

$\Pi_{S\#,Cname,Score}(SC \bowtie Course)$

第四步：在(S#, Cname, Score)上进行投影操作，得到最终结果：

S#	Score	Cname
1001	80	数据库
1002	85	数据结构
1003	83	C语言

提问：如何查询所有学生选课的成绩(包括学生姓名, 课程名称, 成绩)

$\Pi_{Sname,Cname,Score}(SC \bowtie Course \bowtie Student)$

2.3 关系代数运算

—组合操作示例

➤ 多种操作结合的示例 (语义的)

- 查询学习课程号为002的学生学号和成绩

$$\Pi_{S\#,Score}(\sigma_{C\#="002"}(SC))$$

学生选课表SC

S#	C#	Score
1001	001	80
1002	002	85
1003	003	83



2.3 关系代数运算

—组合操作示例

➤ 多种操作结合的示例 (语义的)

- 查询学习课程号为001的学生学号、姓名

$$\Pi_{S\#,Sname}(\sigma_{C\#="001"}(Student \bowtie SC))$$

学生选课表SC

S#	C#	Score
1001	001	80
1002	002	85
1003	003	83

学生表Student

S#	Sname	Ssex	Sage	D#
1001	张三	女	21	01
1002	李四	男	20	02
1003	王五	男	19	03
1004	刘六	女	21	03



2.3 关系代数运算

—组合操作示例

➤ 多种操作结合的示例 (语义的)

- 查询学习课程名称为数据结构的学生学号、姓名和这门课程的成绩

$\Pi_{S\#,Sname,Score}(\sigma_{Cname="数据结构"}(Student \bowtie SC \bowtie Course))$

学生选课表SC

S#	C#	Score
1001	001	80
1002	002	85
1003	003	83

课程表Course

C#	Cname	Chours	Credit
001	数据库	40	4
002	数据结构	60	4
003	C语言	40	2

学生表Student

S#	Sname	Ssex	Sage	D#
1110101	张三	女	21	01
1110102	李四	男	20	02
1110103	王五	男	19	03
1110104	刘六	女	21	03

2.3 关系代数运算

—组合操作示例(续)

➤ 多种操作结合的示例(语义的)

- 查询学习课程号为001或002的学生的学号

$$\Pi_{S\#}(\sigma_{C\#="001" \vee C\#="002"}(SC))$$

学生选课表SC

S#	C#	Score
1001	001	80
1002	002	85
1003	003	83



2.3 关系代数运算

—组合操作示例(续)

➤ 多种操作结合的示例(语义的)

- 查询至少学习课程号为001和002的学生的学号

学生选课表SC

S#	C#	Score
1001	001	80
1002	002	85
1003	003	83

- 是否可写成如下形式呢? ~~$\Pi_{S\#}(\sigma_{C\#="001" \wedge C\#="002"}(SC))$~~

$$\Pi_{SC.S\#}(\sigma_{SC.C\#="001" \vee SC1.C\#="002"}(SC \bowtie \rho_{SC1}(SC)))$$

$SC.S\#=SC1.S\#$

- 我们也可以采用交运算来实现 $\Pi_{S\#}(\sigma_{C\#="001"}(SC) \cap \Pi_{S\#}(\sigma_{C\#="002"}(SC)))$

2.3 关系代数运算

—组合操作示例(续)

➤ 书写关系代数表达式的基本思路

- 检索是否涉及多个表，如不涉及，则可直接采用并、差、交、选择与投影，只要注意条件书写正确与否即可
- 如涉及多个表，则检查
 - ▣ 能否使用自然连接，将多个表连接起来(多数情况是这样的)
 - ▣ 如不能，能否使用等值或不等值连接(θ -连接)
 - ▣ 还不能，则使用广义笛卡尔积，注意相关条件的书写
- 连接完后，可以继续使用选择、投影等运算，即所谓数据库的“**选投联**”操作

$$\Pi_{\text{Sname,Sage}}(\sigma_{C\#="002"}(S \bowtie SC))$$

2.3 关系代数运算

--关系代数之纯关系操作：除(Division)

- 下面介绍关系的除法，除法运算经常用于求解“查询… 全部的…”问题
- **前提条件：**给定关系 $R(A_1, A_2, \dots, A_n)$ 为 n 度关系和关系 $S(B_1, B_2, \dots, B_m)$ 为 m 度关系，如果可以进行关系 R 与关系 S 的除运算，**当且仅当：**属性集 $\{B_1, B_2, \dots, B_m\}$ 是属性集 $\{A_1, A_2, \dots, A_n\}$ 的**真子集**，即 $m < n$ 。
- **定义：**关系 R 和关系 S 的除运算结果也是一个关系，记作 $R \div S$

2.3 关系代数运算

--关系代数之纯关系操作：除(Division)(续)

- 我们先看 $R \div S$ 结果的属性应有哪些，然后再看 $R \div S$ 的元组怎样形成。
- 设属性集 $\{C1, C2, \dots, Ck\} = \{A1, A2, \dots, An\} - \{B1, B2, \dots, Bm\}$ ，则有 $k=n-m$ 则 $R \div S$ 结果关系是一 k 度($n-m$ 度)关系，由 $\{C1, C2, \dots, Ck\}$ 属性构成

R		
A1	A2	A3
a	b	c
b	b	c
a	e	c
b	e	f

S	
A2	A3
b	c

$R \div S$
A1
a
b



2.3 关系代数运算

--关系代数之纯关系操作：除(Division) (续)

- 再设关系 $R(\langle a_1, \dots, a_n \rangle)$ 和关系 $S(\langle b_1, \dots, b_m \rangle)$, 那么 $R \div S$ 结果关系为元组 $\langle c_1, \dots, c_k \rangle$ 的集合, 元组 $\langle c_1, \dots, c_k \rangle$ 满足下述条件: 它与 S 中每一个元组 $\langle b_1, \dots, b_m \rangle$ 组合形成的一个新元组都是 R 中的某一个元组 $\langle a_1, \dots, a_n \rangle$

(其中, $a_1, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_k$ 分别是属性 $A_1, \dots, A_n, B_1, \dots, B_m, C_1, \dots, C_k$ 的值)

- 数学描述:

$$R \div S = \{t | t \in \prod_{R-S}(R) \wedge \forall u \in S (tu \in R)\}$$

$R \div S$: 在 R 中, 找出与 S 中所有的元组有关系的 R 元组

2.3 关系代数运算

—关系代数之纯关系操作：除(Division) (续)

➤ 除(Division)操作的示例一(抽象的)

R		
A1	A2	A3
a	b	c
d	b	c
a	e	c
a	e	f
d	b	f
a	e	g
a	e	h
a	b	l

S	$R \div S$
A3	A1 A2
c	a b
	d b
	a e

(1)

S	$R \div S$
A3	A1 A2
c	d b
f	a e

(2)

2.3 关系代数运算

—关系代数之纯关系操作：除(Division) (续)

➤ 除(Division)操作的示例一(抽象的)

R		
A1	A2	A3
a	b	c
d	b	c
a	e	c
a	e	f
d	b	f
a	e	g
a	e	h
a	b	l

S	$R \div S$
A3	A1 A2
c	a e
f	
g	
h	

(3)

S	$R \div S$
A2 A3	A1
b c	a
	d

(4)



2.3 关系代数运算

--关系代数之纯关系操作：除(Division)

➤ 除(Division)操作的示例二(语义的)

- 查询选修了全部课程的学生的学号

$$\Pi_{S\#,C\#}(SC) \div \Pi_{C\#}(Course)$$

学生选课表SC

S#	C#	Score
1001	001	80
1001	002	85
1001	003	83
1002	001	78
1002	002	83
1003	001	87
1003	002	83
1004	001	79

课程表Course

C#	Cname	Chours	Credit
001	数据库	40	4
002	数据结构	60	4
003	C语言	40	2

选修了全部课程
的学生的学号

S#
1001

2.3 关系代数运算

—关系代数之纯关系操作：除(Division)

➤ 除(Division)操作的示例三(语义的)

- 查询选修了学号1001学生所学全部课程的同学的姓名

$$\Pi_{\text{Sname}} \left(s \bowtie \left(\Pi_{\text{S\#,C\#}}(sc) \div \Pi_{\text{C\#}}(\sigma_{\text{S\#}="1001"}(sc)) \right) \right)$$

□ 请问下述写法与上有何不同？结果是否一样 $\Pi_{\text{Sname}} \left(s \bowtie \left(sc \div \Pi_{\text{C\#}}(\sigma_{\text{S\#}="1001"}(sc)) \right) \right)$ 不一样

学生表 S

S#	Sname
1001	张三
1002	李四
1003	王五
1004	刘六

S#	C#	Score
1001	001	80
1001	002	85
1001	003	83
1002	001	78
1002	002	83
1003	001	87
1003	002	83
1003	003	79

R		
A1	A2	A3
a	b	c
d	b	c
a	e	c
a	e	f
d	b	f
a	e	g
a	e	h
a	b	l

S	R ÷ S	
A3	A1	A2
c	a	b
	d	b
	a	e

(1)

S	R ÷ S	
A3	A1	A2
c	d	b
f	a	e

(2)

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 下面介绍关系的外连接操作

➤ 外连接问题的提出：先看例子

- Teacher (T#, Tname, Salary),
- Teach (T#, C#),
- Course (C#, Cname)

Teacher

T#	Tname	Salary
001	赵三	1200
002	赵四	1400
003	赵五	1000
004	赵六	1100

Teach

T#	C#
001	001
002	002
004	002

Course

C#	Cname
001	物理
002	数学
003	化学

数据库系统

- Teacher(T#,Tname,Salary),Teach(T#,C#),Course(C#,Cname)

T#	Tname	Salary	C#	Cname
001	赵三	1200	001	物理
002	赵四	1400	002	数学
004	赵六	1100	003	数学

- 请列出所有老师的有关信息,包括姓名,工资,所教课程等
 $\pi_{T\#,Tname,Salary,C\#,Cname}(Teacher \bowtie Teach \bowtie Course)$
- 按上式连接的结果,003号教师的姓名和工资信息丢失了
- 因为在Teach表中没有和003号教师相匹配的元组,元组003号教师(又称为失配元组)不能和其他表的元组形成连接元组,信息因而丢失。
- 怎样保证使003号教师信息仍旧出现在结果关系中呢?
----这就需要外连接----

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 定义：两个关系R与S进行连接时, 如果关系R(或S)中的元组在S(或R)中找不到相匹配的元组, 则为了避免该元组信息丢失, 从而将该元组与S(或R)中假定存在的全为空值的元组形成连接, 放置在结果关系中, 这种连接称之为外连接(Outer Join)。

R和S的外连接(city值相等)

R	
S#	City
S7	威海
S8	烟台
S9	青岛

S	
P#	City
P7	威海
P8	北京
P9	上海

S#	R.City	P#	S.City
S7	威海	P7	威海
?	?	P8	北京
?	?	P9	上海
S8	烟台	?	?
S9	青岛	?	?

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

- 外连接 = 自然连接(或 θ 连接)+失配的元组(与全空元组形成的连接)
- 外连接的形式：左外连接、右外连接、全外连接
 - ▣ 左外连接 = 自然连接(或 θ 连接)+左侧表中失配的元组
 - ▣ 右外连接 = 自然连接(或 θ 连接)+右侧表中失配的元组
 - ▣ 全外连接 = 自然连接(或 θ 连接)+两侧表中失配的元组
- 左外连接(Left Outer Join)记为： $R \bowtie_{\text{L}} S$
- 右外连接(Right Outer Join)记为： $R \bowtie_{\text{R}} S$
- 全外连接(Full Outer Join)记为： $R \bowtie_{\text{F}} S$

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 外连接操作示例

R

S#	City
S7	威海
S8	烟台
S9	青岛

S

P#	City
P7	威海
P8	北京
P9	上海

R和S的左外连接(city值相等)

S#	R.City	P#	S.City
S7	威海	P7	威海
S8	烟台	?	?
S9	青岛	?	?

R和S的右外连接(city值相等)

S#	R.City	P#	S.City
S7	威海	P7	威海
?	?	P8	北京
?	?	P9	上海



2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 外连接操作示例

R

S#	City
S7	威海
S8	烟台
S9	青岛

S

P#	City
P7	威海
P8	北京
P9	上海

R和S的全外连接(city值相等)

S#	R.City	P#	S.City
S7	威海	P7	威海
?	?	P8	北京
?	?	P9	上海
S8	烟台	?	?
S9	青岛	?	?

R和S的等连接(city值相等)

S#	R.City	P#	S.City
S7	威海	P7	威海

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 外连接操作示例

· 前面问题例子的解决方案：查询所有老师的信息

$\pi_{T\#,Tname,Salary,C\#,Cname}(Teacher \bowtie Teach \bowtie Course)$

Teacher

T#	Tname	Salary
001	赵三	1200
002	赵四	1400
003	赵五	1000
004	赵六	1100

Teach

T#	C#
001	001
002	002
004	002

Course

C#	Cname
001	物理
002	数学
003	化学

\bowtie

\bowtie

=

T#	Tname	Salary	C#	Cname
001	赵三	1200	001	物理
002	赵四	1400	002	数学
003	赵五	1000	null	null
004	赵六	1100	003	数学

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 外连接操作示例

- 查询所有课程的信息

$\pi_{T\#,Tname,Salary,C\#,Cname}(Teacher \bowtie Teach \bowtie Course)$

T#	Tname	Salary
001	赵三	1200
002	赵四	1400
003	赵五	1000
004	赵六	1100

\bowtie

Teach

T#	C#
001	001
002	002
004	002

\bowtie

Course

C#	Cname
001	物理
002	数学
003	化学

=

T#	Tname	Salary	C#	Cname
001	赵三	1200	001	物理
002	赵四	1400	002	数学
004	赵六	1100	003	数学
null	null	null	003	化学

2.3 关系代数运算

—关系代数之纯关系操作：外连接(Outer-Join)

➤ 外连接操作示例

- 查询所有老师和所有课程的信息

$\pi_{T\#,Tname,Salary,C\#,Cname}(Teacher \bowtie Teach \bowtie Course)$

Teacher

T#	Tname	Salary
001	赵三	1200
002	赵四	1400
003	赵五	1000
004	赵六	1100

Teach

T#	C#
001	001
002	002
004	002

Course

C#	Cname
001	物理
002	数学
003	化学

\bowtie

\bowtie

=

T#	Tname	Salary	C#	Cname
001	赵三	1200	001	物理
002	赵四	1400	002	数学
003	赵五	1000	002	数学
004	赵六	1100	null	null
null	null	null	003	化学

