

数据库系统之一

--基本知识与关系模型

战德臣

哈尔滨工业大学 教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

第4讲 关系模型之关系代数

战德臣

哈尔滨工业大学 教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

本讲学习什么？



基本内容

1. 关系代数之基本操作
2. 关系代数之扩展操作
3. 关系代数之组合与应用训练
4. 关系代数之复杂扩展操作(选学)

重点与难点

- 关系代数基本操作：并、差、积、选择、投影、(更名)。
- 关系代数扩展操作：交、 θ -连接、自然连接。
- 关系代数复杂扩展操作：除、外连接
- 书写关系代数的基本思维训练：“一个集合，施加一个操作得到一个集合，依次施加关系代数操作，进而得到所需结果” “以集合为中心”

关系代数概述

战德臣

哈尔滨工业大学 教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

- 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)、选择、投影和更名**等基本操作
- 以及**交、连接和关系除**等扩展操作，是一种集合思维的操作语言。
- 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，需要指明所用操作，具有一定的过程性。

$$\pi_{\text{姓名,课程名}}(\sigma_{\text{课程号}=c2}(R \bowtie S))$$

- 是一种抽象的语言，是学习其他数据库语言，如SQL等的基础

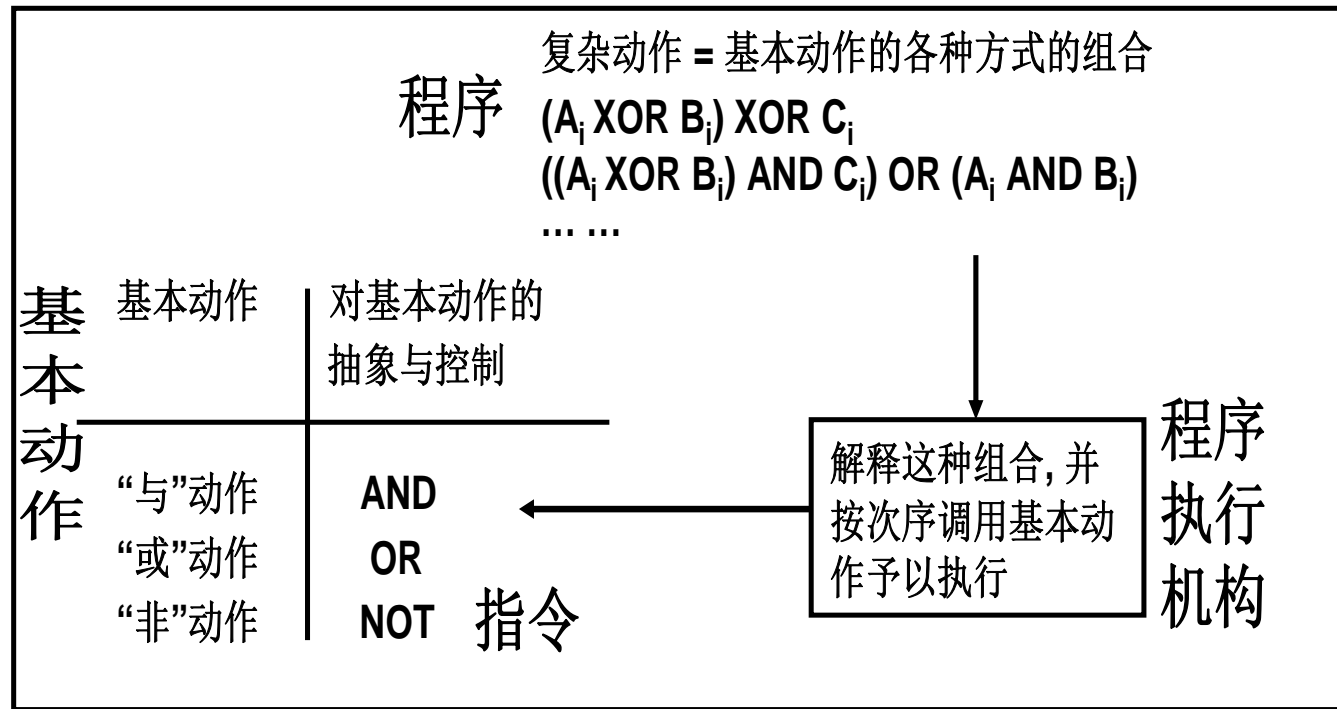
关系代数操作：集合操作和纯关系操作

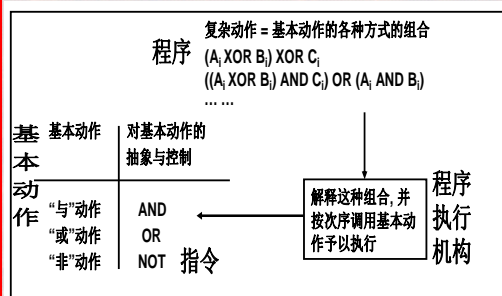
(1)集合操作

UNION (并)	R	S	$R \cup S$
INTERSECTION (交)	R	S	$R \cap S$
DIFFERENCE (差)	R	S	$R - S$
Cartesian PRODUCT (笛卡尔积)	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	$R \bowtie_{A \theta B} S$
DIVISION (除)	R	S	$R \div S$





程序

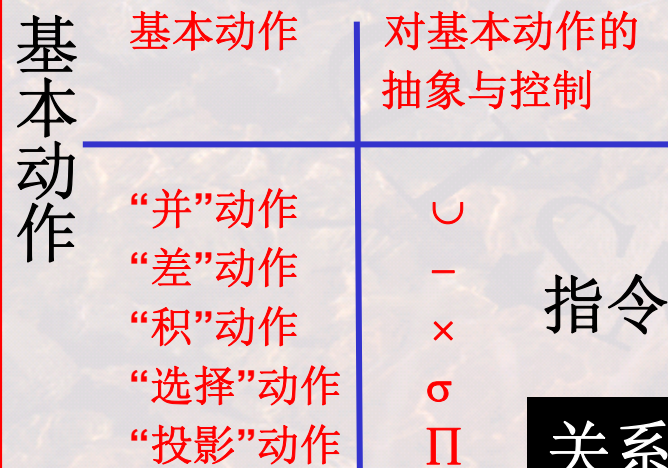
Select Sname From Student, SC
 Where Student.S# = SC.S# and SC.C# = '001'
 Order By Score DESC;

SQL语言

$\Pi_{\text{Sname}}(\sigma_{\text{student.s\#}=\text{sc.s\#}}(\text{Student} \times \text{SC}))$

复杂动作 = 基本动作的各种方式的组合

关系模型基本运算的各种组合



关系模型
基本运算

解释这种组合,并按次序调用基本动作予以执行

数据库管
理系统

程序
执行
机构

关系代数之基本操作

战德臣

哈尔滨工业大学 教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

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 $\text{Domain}(A_i) = \text{Domain}(B_i)$

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)

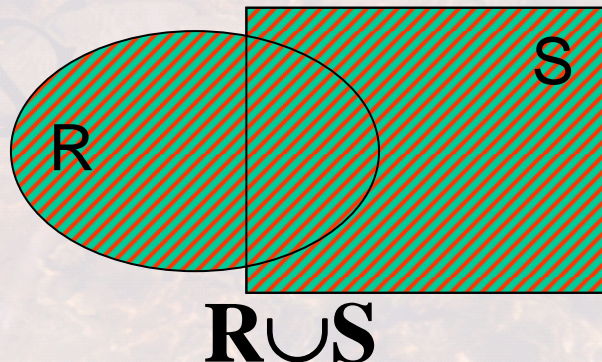
并(Union)

➤ **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**： $R \cup S = \{ t \mid t \in R \vee t \in S \}$ ，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



关系代数之基本操作

(1) “并”操作

- 并操作的示例一(抽象的)
- 假设**R**与**S**是并相容的两个关系，则 **$R \cup S$** 是几个元组呢？

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 \cup S		
C1	C2	C3
a	b	c
a	d	g
f	b	e
a	b	e
h	d	g

关系代数之基本操作

(1) “并”操作



- 并操作的示例二(语义的)
- 查询或者参加体育队或者参加文艺队所有学生的信息

R(参加体育队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

S(参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301

R ∪ S(或者参加体育队或者文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201

关系代数之基本操作

(1) “并”操作



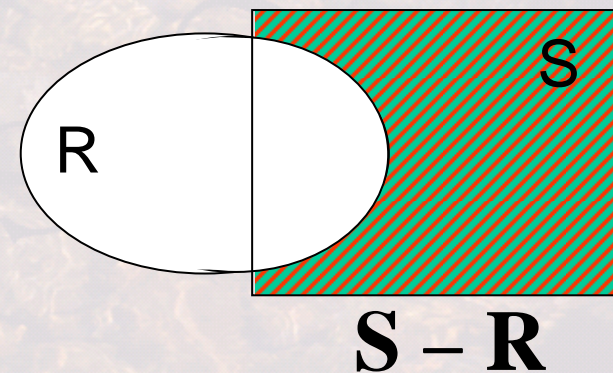
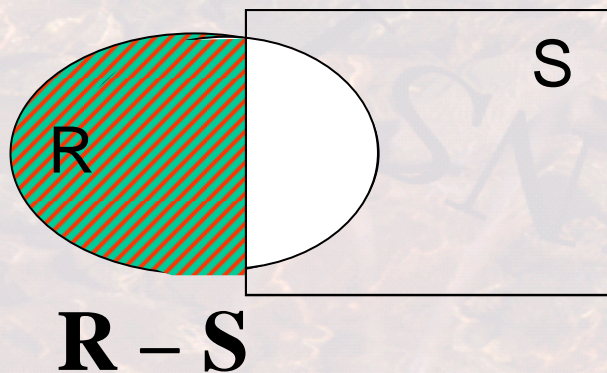
- 并操作的示例三(语义的)
- 若 R 为计算机学院的学生， S 为材料学院的学生
则： $R \cup S$ 为两院所有的学生
- 若 R 为学过数据库课程的学生， S 为学过自控理论课程的学生
则： $R \cup S$ 为学过两门课之一的所有学生
- 汉语中的“或者...或者...”通常意义是并运算的要求。
- 首先要准确理解汉语的查询要求，然后再找到正确的操作
- 同学可举出更多的示例...

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**： $R - S = \{ t \mid t \in R \wedge t \notin S \}$ ，其中**t**是元组

➤ $R - S$ 与 $S - R$ 是不同的



- 差操作的示例一(抽象的)
- 假设 R 与 S 是并相容的两个关系，则 $R-S$ 是？ $S-R$ 是？

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

- 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
- 查询只参加文艺队而未参加体育队的学生信息

R(参加体育队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

R-S(参加体育队而未参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

S(参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301

S-R(参加文艺队而未参加体育队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201

- 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，则：
 - $R - S$ 为 计算机学院非四年级的学生**
 - $S - R$ 为 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， 则：
 - $R - S$ 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
- 同学可举出更多的示例...

广义笛卡尔积 (Cartesian Product)

➤ 定义：关系 $R(<a_1, a_2, \dots, a_n>)$ 与关系 $S(<b_1, b_2, \dots, b_m>)$ 的广义笛卡尔积 (简称广义积,或 积 或笛卡尔积) 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：
$$R \times S = \{ <a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m> \mid <a_1, a_2, \dots, a_n> \in R \wedge <b_1, b_2, \dots, b_m> \in S \}$$

关系代数之基本操作

(3) “笛卡尔积”操作

- 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?

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

$R \times S$					
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

关系代数之基本操作

(3) “笛卡尔积”操作

➤ 再看一个示例 (抽象的)

Relations r, s :

A	B
α	1
β	2

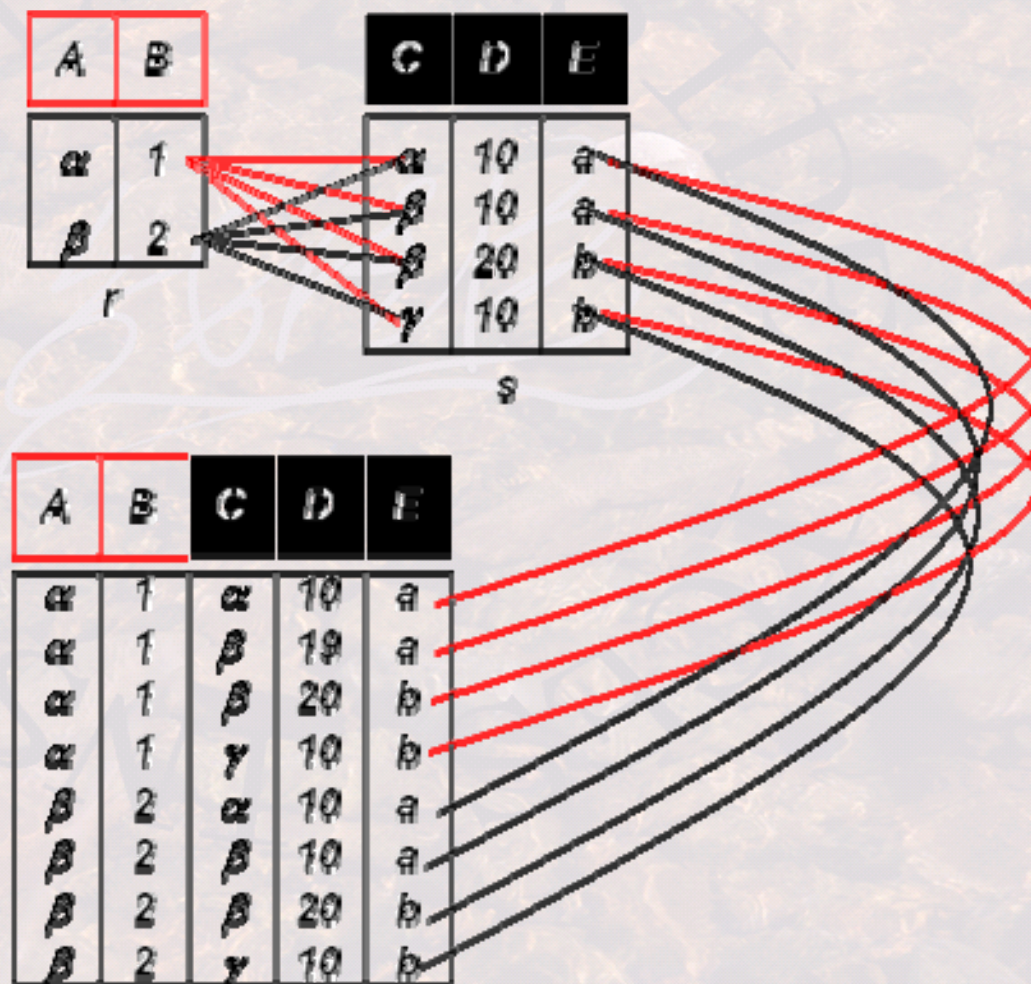
r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

$r \times s$:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b



关系代数之基本操作

(3) “笛卡尔积”操作

- 广义积操作的示例二(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

学生表				(所有学生)的(所有课程)							
学号	姓名	年龄	住址	学号	姓名	年龄	住址	课程号	课程名	教师	学时
981101	李四	22	3010	981101	李四	22	3010	C1	计算机	教师 1	52
981103	李三	21	3011	981101	李四	22	3010	C2	物理	教师 2	36
981105	李六	22	3011	981101	李四	22	3010	C3	高数	教师 5	40
				981103	李三	21	3011	C1	计算机	教师 1	52
				981103	李三	21	3011	C2	物理	教师 2	36
				981103	李三	21	3011	C3	高数	教师 5	40
				981105	李六	22	3011	C1	计算机	教师 1	52
				981105	李六	22	3011	C2	物理	教师 2	36
				981105	李六	22	3011	C3	高数	教师 5	40

课程表			
课程号	课程名	教师	学时
C1	计算机	教师 1	52
C2	物理	教师 2	36
C3	高数	教师 5	40

关系代数之基本操作

(3) “笛卡尔积”操作



- $R \times S = S \times R$: $R \times S$ 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 =
 - $n + m$ 。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 =
 - $x \times y$ 。($R \times S$ 的基数是 $x \times y$)。

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: $\sigma_{\text{con}}(R) = \{t \mid t \in R \wedge \text{con}(t) = \text{'真'}\}$,

□ 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由逻辑运算符连接比较表达式组成

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

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

[illegible]

- 选择操作的示例一(抽象的)
- 选择A3值大于0的元组?
- 选择A2值为a或者b的元组?
- 选择A3大于0且A1等于A2的元组

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

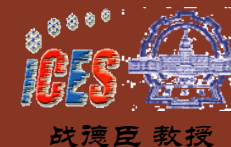
$\sigma_{A3>0}(R)$		
A1	A2	A3
a	a	10
f	b	5

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

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

关系代数之基本操作

(4) “选择”操作



➤ 选择操作的示例二(语义的)

R(学生表)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

查询所有年龄小于20同学的信息

$\sigma_{\text{Sage} < 20} (R)$

S#	Sname	Ssex	Sage	D#	Sclass
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98050104	孙六	女	19	05	980501

查询所有男同学的信息

$\sigma_{\text{Ssex} = \text{“男”}} (R)$

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402

查询所有3系或5系的同学信息

$\sigma_{\text{D\#} = \text{“03”} \vee \text{D\#} = \text{“05”}} (R)$

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98050104	孙六	女	19	05	980501

关系代数之基本操作

(4) “选择”操作



➤ 选择操作的示例三(语义的)

查询所有年龄大于20的3系同学的信息

$\sigma_{\text{Sage} > 20 \wedge \text{D\#} = \text{"03"}} (\text{R})$

S#	Sname	Ssex	Sage	D#	Sclass
98030102	张四	女	21	03	980301

R(学生表)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

查询不在(年龄大于20的3系同学)要求之内的所有其它同学的信息

$\sigma_{\neg(\text{Sage} > 20 \wedge \text{D\#} = \text{"03"})} (\text{R})$

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

- 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：
 $\text{Sage} < 20 \vee \text{Sage} > 18 \wedge \text{D\#} = \text{"03"}$
与
 $(\text{Sage} < 20 \vee \text{Sage} > 18) \wedge \text{D\#} = \text{"03"}$
- 同学可举出更多的示例...

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

R		
A1	A2	A3

➤ 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

关系代数之基本操作

(5) “投影”操作

- 投影操作的示例一(抽象的)
- 投影出A3列的元组?
- 投影出A3,A1两列的元组?

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

- 如果投影后有重复元组, 则应去掉

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

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

关系代数之基本操作

(5) “投影”操作

➤ 投影操作的示例二(语义的)

查询所有学生的姓名和年龄

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

Sname	Sage
张三	20
张四	21
张五	19
王三	18
王四	21
孙六	19

R (学生表)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

查询所有学生的姓名及其所在的系

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

Sname	D#
张三	03
张四	03
张五	03
王三	04
王四	04
孙六	05

关系代数之基本操作

(5) “投影”操作



➤ 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#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

用户可以根据需要通过投影、选择操作查询他所关心的数据信息。

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

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

S#	Sname
98030101	张三
98030102	张四

查询所有在4系就读的且男同学的学号和姓名

$\Pi_{S\#, Sname}(\sigma_{D\#="04" \wedge Ssex='男'}(R))$

S#	Sname
98040201	王三
98040202	王四



关系代数的基本书写思路:

- 选出将用到的关系/表
- 做“积”运算
- 做选择运算保留所需的行/元组
- 做投影运算保留所需的列/属性

关系代数之扩展操作

战德臣

哈尔滨工业大学 教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

交(Intersection)

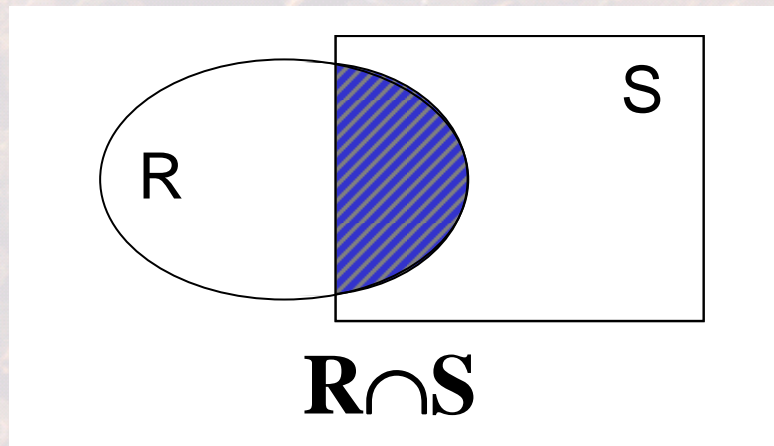
➤ **定义**：假设关系**R**和关系**S**是并相容的，则关系**R**与关系**S**的交运算结果也是一个关系，记作： $R \cap S$ ，它由**同时出现在**关系**R**和关系**S**中的元组构成。

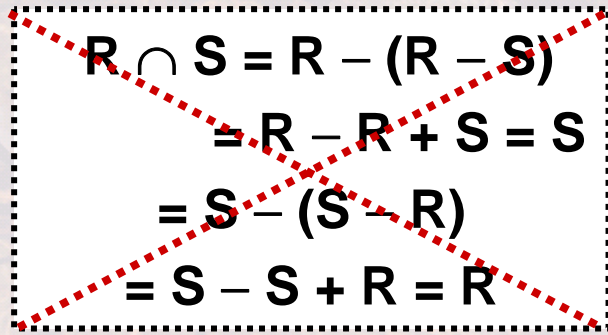
➤ **数学描述**： $R \cap S = \{ t \mid t \in R \wedge t \in S \}$ ，其中**t**是元组

➤ $R \cap S$ 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：

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




$$\begin{aligned} R \cap S &= R - (R - S) \\ &= R - R + S = S \\ &= S - (S - R) \\ &= S - S + R = R \end{aligned}$$

错在哪里呢？

关系代数之扩展操作

(1) “交”操作

- 交操作的示例一(抽象的)
- 假设**R**与**S**是并相容的两个关系，则 $R \cap S$?

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

关系代数之扩展操作

(1) “交”操作



战德臣 教授

- 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(参加体育队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

$R \cap S$ (既参加体育队又参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301

S(参加文艺队的学生)

S#	Sname	Ssex	Sage	D#	Sclass
98020101	孙三	女	18	02	980201
98020102	孙四	男	20	02	980201
98020103	孙五	女	19	02	980201
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301

- 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
- 同学可举出更多的示例...

θ -连接(θ -Join, theta-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, 这就需要 θ -连接操作
- 比如: 查询数据结构成绩在90分以上的学生姓名(涉及Student, Course, SC)

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

Course				
C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

SC		
S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

➤ **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 $R \bowtie_{A \theta B} S$ ，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**之间满足 θ 条件的元组构成。

➤ **数学描述**：

$$R \bowtie_{A \theta B} S = \sigma_{t[A] \theta s[B]} (R \times S)$$

□ 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

关系代数之扩展操作

(2) “ θ -连接”操作

➤ θ -连接(θ -Join)操作的示例一(抽象的)

R		S	
A	B	H	C
a	1	1	x
b	2	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

关系代数之扩展操作

(2) “θ-连接”操作



- **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)
竞聘的岗位必须由不低于其最低学历要求的人员担任,

worker				
W#	Wname	Wsex	Wage	Degree
01	张三	男	35	1
02	张四	女	49	3
03	张五	男	55	2

Position	
Type	Limited_Degree
组长	1
项目经理	2
部门经理	3

1: 本科
2: 硕士
3: 博士

- 找出所有员工的姓名及其可能竞聘职位的名称

$\Pi_{Wname, Type}(\text{worker} \bowtie_{\text{degree} \geq \text{limited_degree}} \text{position})$

关系代数之扩展操作

(2) “θ-连接”操作

第一步：对两个表进行广义笛卡尔积

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

(所有员工)的(所有职位) \leftarrow 广义笛卡尔积

W#	Wname	Wsex	Wage	Degree	Type	Limited_Degree
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	张四	女	49	3	组长	1
02	张四	女	49	3	项目经理	2
02	张四	女	49	3	部门经理	3
03	张五	男	55	2	组长	1
03	张五	男	55	2	项目经理	2
03	张五	男	55	2	部门经理	3

×
×
×

所有员工及其可以竞聘的职位

W#	Wname	Wsex	Wage	Degree	Type	Limited_Degree
01	张三	男	35	1	组长	1
02	张四	女	49	3	组长	1
02	张四	女	49	3	项目经理	2
02	张四	女	49	3	部门经理	3
03	张五	男	55	2	组长	1
03	张五	男	55	2	项目经理	2

第三步：在(Wname, Type)上进行投影操作，得到最终的结果

Wname	Type
张三	组长
张四	组长
张四	项目经理
张四	部门经理
张五	组长
张五	项目经理

关系代数之扩展操作

(2) “θ-连接”操作



- θ-连接操作的示例三(续)
- 关系与自身的θ-连接
- 查询至少98030101号同学和98040202号同学学过的所有课程号

$$\pi_{SC.C\#} (\sigma_{SC.S\#="98030101" \wedge SC1.S\#="98040202"} (SC \bowtie_{SC.C\# = SC1.C\#} \rho_{SC1}(SC)))$$

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#='98030101')			SC1(S#='98040202')		
S#	C#	Score	S#	C#	Score
98030101	001	92	98040202	001	55
98030101	002	85	98040202	002	90
98030101	003	88	98040202	003	80

SC		
S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

➤ 特别注意：

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。

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。

关系代数之扩展操作

(3) “等值-连接”操作

➤ 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 × 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

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

关系代数之扩展操作

(3) “等值-连接”操作

- 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) , 获奖类别表Honor(Type, Title)
- 找出所有获奖员工姓名、年龄及其获奖的名称

$\Pi_{Wname, Wage, Title} (worker \bowtie_{Honor_type = Type} Honor)$

worker				
W#	Wname	Wsex	Wage	Honor_type
01	张三	男	35	1
02	张四	女	49	3
03	张五	男	55	2

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

关系代数之扩展操作

(3) “等值-连接”操作

第一步：对两个表进行广义笛卡尔积

第二步：从广义笛卡尔积中选取符合(Honor_type=Type)条件的元组

(所有员工)的(所有获奖) ← 广义笛卡尔积

W#	Wname	Wsex	Wage	Honor_type	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	张四	女	49	3	1	全国劳模
02	张四	女	49	3	2	“五一”奖章获得者
02	张四	女	49	3	3	“三八”妇女红旗手
03	张五	男	55	2	1	全国劳模
03	张五	男	55	2	2	“五一”奖章获得者
03	张五	男	55	2	3	“三八”妇女红旗手

所有员工及其所获奖的信息

W#	Wname	Wsex	Wage	Honor_type	Type	Title
01	张三	男	35	1	1	全国劳模
02	张四	女	49	3	3	“三八”妇女红旗手
03	张五	男	55	2	2	“五一”奖章获得者

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

Wname	Wage	Title
张三	35	全国劳模
张四	49	“三八”妇女红旗手
张五	55	“五一”奖章获得者

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, ..., B_n**, 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

$R.B_1 = S.B_1$ and $R.B_2 = S.B_2$... and $R.B_n = S.B_n$ 才能连接

□ 要在结果中去掉重复的属性列(因结果中 **$R.B_i$** 始终是等于 **$S.B_i$** 所以可只保留一列即可)

关系代数之扩展操作

(4) “自然连接”操作

➤ 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 × 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 ⋈ S		
A	B	C
a	1	x
a	1	y

关系代数之扩展操作

(4) “自然连接”操作



- 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, T#)

SC		
S#	C#	Score
98030101	001	92.0
98030101	003	88.0
98040202	002	90.5

Course				
C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
002	高等数学	80	12	004

- 查询所有学生选课的成绩(包括学号, 课程名称, 成绩)

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

关系代数之扩展操作

(4) “自然连接”操作

第一步：对两个表进行广义笛卡尔积

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

广义笛卡尔积

S#	C#	Score	C#	Cname	Chours	Credit	T#
98030101	001	92.0	001	数据库	40	6	001
98030101	001	92.0	003	数据结构	40	6	003
98030101	001	92.0	002	高等数学	80	12	004
98030101	003	88.0	001	数据库	40	6	001
98030101	003	88.0	003	数据结构	40	6	003
98030101	003	88.0	002	高等数学	80	12	004
98040202	002	90.5	001	数据库	40	6	001
98040202	002	90.5	003	数据结构	40	6	003
98040202	002	90.5	002	高等数学	80	12	004

✓

✓

✓

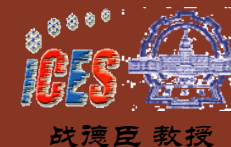
S#	C#	Score	C#	Cname	Chours	Credit	T#
98030101	001	92.0	001	数据库	40	6	001
98030101	003	88.0	003	数据结构	40	6	003
98040202	002	90.5	002	高等数学	80	12	004

第三步：去掉重复的列

S#	C#	Score	Cname	Chours	Credit	T#
98030101	001	92.0	数据库	40	6	001
98030101	003	88.0	数据结构	40	6	003
98040202	002	90.5	高等数学	80	12	004

关系代数之扩展操作

(4) “自然连接”操作



第四步：在(S#, Cname, Score)上进行投影操作，得到最终结果

S#	Cname	Score
98030101	数据库	92.0
98030101	数据结构	88.0
98040202	高等数学	90.5

提问：如果查询所有学生选课的成绩(包括学生姓名，课程名称，成绩)

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

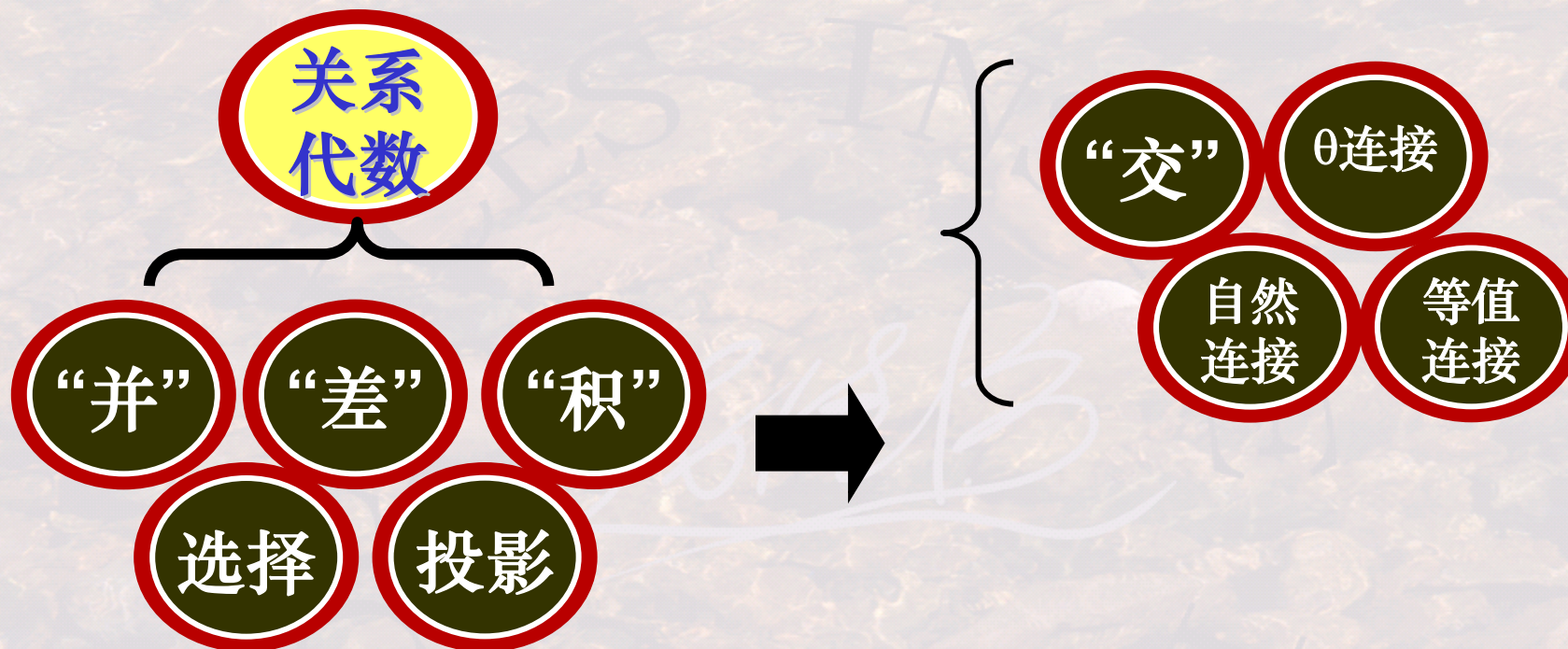
Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

Course				
C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
002	高等数学	80	12	004

SC		
S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

关系代数之扩展操作

(5) 小结



关系代数的基本书写思路:

- 选出将用到的关系/表
- 做“积”运算(可用连接运算替换)
- 做选择运算保留所需的行/元组
- 做投影运算保留所需的列/属性

关系代数操作之组合与应用训练

战德臣

哈尔滨工业大学 教授.博士生导师

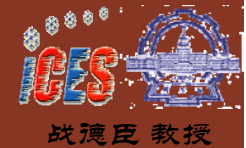
黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

关系代数操作之组合与应用训练

(1)集合操作思维训练



查询表达式

组合各种运算

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

SC		
S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

- 查询学习课程号为002的学生学号和成绩

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

- 查询学习课程号为001的学生学号、姓名

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

Course				
C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

- 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

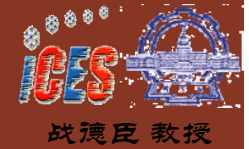
$Student \bowtie SC \bowtie Course$

$\sigma_{Cname = "数据结构"}(Student \bowtie SC \bowtie Course)$

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

关系代数操作之组合与应用训练

(1)集合操作思维训练



查询表达式

注意连接与积的差别

Student					
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

SC		
S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

➤ 查询学习课程号为001的学生学号、姓名

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

Course				
C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

$\pi_{S\#,Sname}(\sigma_{C\#='001' \wedge Student.S\# = SC.S\#}(Student \times SC))$

连接条件

➤ 查询学习课程号为**001**或**002**的学生的学号

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

➤ 查询至少学习课程号为**001**和**002**的学生的学号

□ 是否可写成如下形式呢？

$$\pi_{S\#}(\sigma_{C\#="001" \wedge C\#="002"}(SC))$$

➤ 查询至少学习课程号为**001**和**002**的学生的学号

$$\pi_{SC.S\#}(\sigma_{SC.C\#="001" \wedge SC1.C\#="002"}(SC \bowtie_{SC.S\# = SC1.S\#} \rho_{SC1}(SC)))$$

请问：上式使用的是等值连接，换成自然连接，写成如下形式是否正确？

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

- 前例我们也可以采用交运算来实现

$$\pi_{S\#}(\sigma_{C\#="001"}(SC)) \cap \pi_{S\#}(\sigma_{C\#="002"}(SC))$$

- 再举一个例子：查询不学习课程号为002的学生姓名和年龄
- 同学给出了如下的查询表达式, 这些表达式的结果是什么? 正确吗?

$$\pi_{Sname, Sage}(\sigma_{C\# \neq "002"}(S \bowtie SC))$$

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

➤ 再举一个例子：查询不学习课程号为**002**的学生姓名和年龄

$$\pi_{\text{Sname, Sage}}(\text{S}) - \pi_{\text{Sname, Sage}}(\sigma_{\text{C\#}='002'}(\text{S} \bowtie \text{SC}))$$

➤ 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

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

关系代数之复杂扩展操作

战德臣

哈尔滨工业大学 教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

除(Division)

➤ 除法运算经常用于求解 “查询... 全部的/所有的...” 问题

➤ **前提条件**: 给定关系 $R(A_1, A_2, \dots, A_n)$ 为 n 度关系, 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 $R \div S$, 分两部分来定义。

□ 先定义 $R \div S$ 结果的属性应有哪些?

□ 设属性集 $\{C_1, C_2, \dots, C_k\} = \{A_1, A_2, \dots, A_n\} - \{B_1, B_2, \dots, B_m\}$, 则有 $k = n - m$ 则 $R \div S$ 结果关系是一 k 度($n - m$ 度)关系, 由 $\{C_1, C_2, \dots, C_k\}$ 属性构成

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

S	
A2	A3
b	c

R÷S
A1
a
d

□再定义 $R \div S$ 的元组怎样形成?

□再设关系 $R (<a_1, \dots, a_n>)$ 和关系 $S (<b_1, \dots, b_m>)$, 那么 $R \div S$ 结果关系为元组 $<c_1, \dots, c_k>$ 的集合, 元组 $<c_1, \dots, c_k>$ 满足下述条件:

它与 S 中每一个元组 $<b_1, \dots, b_m>$ 组合形成的一个新元组都是 R 中的某一个元组 $<a_1, \dots, a_n>$ 。(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: $R \div S = \{ t \mid t \in \Pi_{R-S}(R) \wedge \forall u \in S (tu \in R) \}$
 $= \Pi_{R-S}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$

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

S	
A2	A3
b	c

R÷S
A1
a
d

关系代数之复杂扩展操作

(1) “除”操作

➤ 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	i

S	R ÷ S	
A3	A1	A2
c	a	b
	d	b
	a	e

(1)

S	R ÷ S	
A3	A1	A2
c	a	e
f		
g		
h		

(3)

S	R ÷ S	
A3	A1	A2
c	d	b
f	a	e

(2)

S	R ÷ S	
A2	A3	A1
b	c	a
		d

(4)

关系代数之复杂扩展操作

(1) “除”操作

➤ 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	i

S	R ÷ S	
A3	A1	A2
c	a	e
f		
g		
h		

(3)

$\Pi_{R-S}(R)$	
A1	A2
a	b
d	b
a	e

$\Pi_{R-S}(R) \times S$		
A1	A2	A3
a	b	c
a	b	f
a	b	g
a	b	h
d	b	c
d	b	f
d	b	g
d	b	h
a	e	c
a	e	f
a	e	g
a	e	h

$$R \div S = \Pi_{R-S}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$$

$(\Pi_{R-S}(R) \times S) - R$		
A1	A2	A3
a	b	f
a	b	g
a	b	h
d	b	g
d	b	h

$\Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$	
A1	A2
a	b
d	b

$\Pi_{R-S}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$	
A1	A2
a	e

(1) “除”操作

- 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

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

SC(学生选课表)

S#	C#	Score
98030101	001	92.0
98030101	002	88.0
98030101	003	90.0
98040202	001	90.5
98040202	002	88.0
98030201	001	93.0
98030201	002	95.0
98030202	001	89.0

Course (课程表)

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
002	高等数学	80	12	004

选修了全部课程的学生的学号

S#
98030101

(1) “除”操作

- 除(Division)操作的示例三(语义的)
- 查询选修了学号98030201学生所学全部课程的同学的姓名

$$\pi_{\text{Sname}}(S \bowtie (\pi_{\text{S\#, C\#}}(\text{SC}) \div \pi_{\text{C\#}}(\sigma_{\text{S\#}=\text{"98030201"}}(\text{SC}))))$$

(1) “除”操作

- 除(Division)操作的示例三(语义的)
- 查询选修了学号98030201学生所学全部课程的同学的姓名

$$\pi_{\text{Sname}}(S \bowtie (\pi_{\text{S\#, C\#}}(\text{SC}) \div \pi_{\text{C\#}}(\sigma_{\text{S\#}=\text{"98030201"}}(\text{SC}))))$$

- 请问下述写法与上有何不同? 结果是否一样

$$\pi_{\text{Sname}}(S \bowtie (\text{SC} \div \pi_{\text{C\#}}(\sigma_{\text{S\#}=\text{"98030201"}}(\text{SC}))))$$

关系代数之复杂扩展操作

(2) “外连接”操作

外连接(Outer-Join)

➤外连接问题的提出(示例)

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

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。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

teacher			Teach		Course						
T#	Tname	Salary	T#	C#	C#	Cname					
001	赵三	1200.00	001	001	001	数学					
002	赵四	1400.00	002	002	002	物理					
003	赵五	1000.00	004	002	003	化学					
004	赵六	1100.00									

T#	Tname	Salary	C#	Cname
001	赵三	1200.00	001	数学
002	赵四	1400.00	002	物理
004	赵六	1100.00	002	物理

关系代数之复杂扩展操作

(2) “外连接”操作

➤ **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	S#	City	S	P#	City	R 和 S 的外连接	S#	R.City	P#	S.City
	S7	London		P7	London		S7	London	P7	London
	S8	Paris		P8	NewYork		?	?	P8	NewYork
	S9	Harbin		P9	Belling	(City 值相等)	?	?	P9	Belling
							S8	Paris	?	?
							S9	Harbin	?	?

- 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: $R \bowtie S$
- 右外连接(**Right Outer Join**)记为: $R \ltimes S$
- 全外连接(**Full Outer Join**)记为: $R \ltimes\bowtie S$

关系代数之复杂扩展操作

(2) “外连接”操作

➤ 外连接操作示例

$R \bowtie S$

R	S#	City	S	P#	City	R 和 S 的左外连接 (City 值相等)	S#	R.City	P#	S.City
	S7	London		P7	London		S7	London	P7	London
	S8	Paris		P8	NewYork		S8	Paris	?	?
	S9	Harbin		P9	Beijing		S9	Harbin	?	?

$R \ltimes S$

R	S#	City	S	P#	City	R 和 S 的右外连接 (City 值相等)	S#	R.City	P#	S.City
	S7	London		P7	London		S7	London	P7	London
	S8	Paris		P8	NewYork		?	?	P8	NewYork
	S9	Harbin		P9	Beijing		?	?	P9	Beijing

关系代数之复杂扩展操作

(2) “外连接”操作

➤ 外连接操作示例

$R \bowtie S$

R	S#	City	S	P#	City	R 和 S 的外连接	S#	R.City	P#	S.City
	S7	London		P7	London		S7	London	P7	London
	S8	Paris		P8	NewYork		?	?	P8	NewYork
	S9	Harbin		P9	Beijing	(City 值相等)	?	?	P9	Beijing
							S8	Paris	?	?
							S9	Harbin	?	?

$R \ltimes S$

R	S#	City	S	P#	City	R 和 S 的等连接	S#	R.City	P#	S.City
	S7	London		P7	London		S7	London	P7	London
	S8	Paris		P8	NewYork					
	S9	Harbin		P9	Beijing	(City 值相等)				

关系代数之复杂扩展操作

(2) “外连接”操作

外连接(Outer-Join)操作的示例

- 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	1200.00
002	赵四	1400.00
003	赵五	1000.00
004	赵六	1100.00



Teach	
T#	C#
001	001
002	002
004	002



Course	
C#	Cname
001	数学
002	物理
003	化学

=

T#	Tname	Salary	C#	Cname
001	赵三	1200.00	001	数学
002	赵四	1400.00	002	物理
004	赵六	1100.00	002	物理
003	赵五	1000.00	null	null

关系代数之复杂扩展操作

(2) “外连接”操作

外连接(Outer-Join)操作示例

➤ 查询所有课程的信息

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

teacher		
T#	Tname	Salary
001	赵三	1200.00
002	赵四	1400.00
003	赵五	1000.00
004	赵六	1100.00



Teach	
T#	C#
001	001
002	002
004	002



Course	
C#	Cname
001	数学
002	物理
003	化学

=

T#	Tname	Salary	C#	Cname
001	赵三	1200.00	001	数学
002	赵四	1400.00	002	物理
004	赵六	1100.00	002	物理
null	null	null	003	化学

关系代数之复杂扩展操作

(2) “外连接”操作

外连接(Outer-Join)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	1200.00
002	赵四	1400.00
003	赵五	1000.00
004	赵六	1100.00



Teach	
T#	C#
001	001
002	002
004	002



Course	
C#	Cname
001	数学
002	物理
003	化学

=

T#	Tname	Salary	C#	Cname
001	赵三	1200.00	001	数学
002	赵四	1400.00	002	物理
004	赵六	1100.00	002	物理
003	赵五	1000.00	null	null
null	null	null	003	化学

回顾本讲学了什么？

战德臣

哈尔滨工业大学 教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on **I**ntelligent
Computing for **E**nterprises & **S**ervices,
Harbin **I**nstitute of **T**echnology

回顾本讲学习了什么？



关系代数的基本书写思路：

- 选出将用到的关系/表
- 做“积”运算(可用连接运算替换)
- 做选择运算保留所需的行/元组
- 做投影运算保留所需的列/属性

回顾本讲学习了什么？



关系代数

Π 列名, ..., 列名 (σ 检索条件 (表名1 \times 表名2 \times ...))

Π S#, Sname, Score (σ Cname="数据结构" \wedge Student.S#=SC.S# \wedge Course.C#=SC.C# (Student \times SC \times Course))

数据库语言SQL

Select 列名 [[, 列名] ...]

From 表名1 [[, 表名2], ...]

[**Where** 检索条件] ;

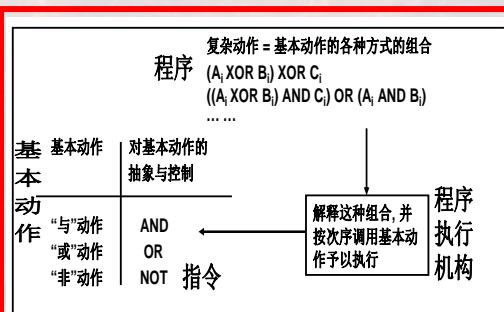
语义：将From后面的所有表串接起来，检索出满足“检索条件”的元组，并按给定的列名及顺序进行投影显示。

Select S#, Sname, Score

From Student, SC, Course

Where Cname= '数据结构' **and** Student.S#=SC.S# **and** Course.C#=SC.C#;

回顾本讲学习了什么？



程序

Select Sname From Student, SC
 Where Student.S# = SC.S# and SC.C# = '001'
 Order By Score DESC;

SQL语言

$\Pi_{\text{Sname}}(\sigma_{\text{student.s\#}=\text{sc.s\#}}(\text{Student} \times \text{SC}))$

复杂动作 = 基本动作的各种方式的组合

关系模型基本运算的各种组合

基本动作

基本动作

对基本动作的抽象与控制

“并”动作
“差”动作
“积”动作
“选择”动作
“投影”动作

\cup
—
 \times
 σ
 Π

指令

关系模型
基本运算

解释这种组合, 并按次序调用基本动作予以执行

数据库管
理系统

程序
执行
机构