

## 第二章 数据模型

---

1

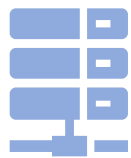
什么是数据模型

2

概念模型

3

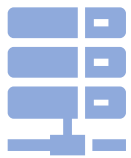
逻辑模型



# 关系模型

---

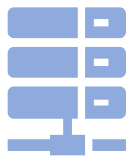
- 1 关系模型概述
- 2 关系数据结构 ( 结构 )
- 3 关系的完整性 ( 约束 )
- 4 关系代数 ( 操作 )



# 关系运算

---

- 关系模型与其他(层次、网状等)模型相比,最有特色的是它的数据库语言。这种语言灵活方便、表达能力和功能都很强。
- 目前关系数据库所使用的语言一般都具有定义、查询、更新和控制一体化的特点,而查询是最主要的部分。
- 查询的条件要使用关系运算表达式来表示。
- 因此,关系运算是设计关系数据语言的基础。
- 按表达查询的方法不同,关系运算可分为关系代数和关系演算两大类。



# 关系代数

---

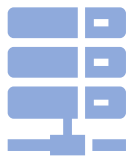
怎么理解关系代数？---以堆栈（Stack）的使用为例

■假设我们要在程序中使用堆栈数据结构。这时我们需要：

- 知道堆栈结构是什么样子
- 在堆栈上我们可以实施什么样的操作
  - ✓ PUSH 和 POP

■下一步，我们寻找一种堆栈结构的实现（为啥不自己写一个？）

■例如找到和使用 LEDA 开源算法库



# 关系代数

---

怎么理解关系代数？---以堆栈（Stack）的使用为例

■ LEDA 库中实现了对堆栈Push和Pop操作的封装，并提供了堆栈的操作语言L来方便的调用PUSH and POP操作

```
S = init_stack();
```

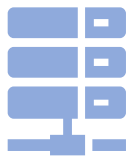
```
S.push(3); S.push(5);
```

```
int x = S.pop();
```

- 也可以组合复杂的堆栈操作表达式

```
T = init_stack(int);
```

```
T.push(S.pop());
```



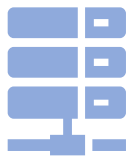
# 关系代数

---

怎么理解关系代数？---以堆栈（Stack）的使用为例

■ 综上，为使用堆栈，我们学习了：

- 堆栈的结构
- 堆栈的操作 (PUSH, POP)：可以称为“堆栈代数” stack algebra
- 堆栈的一种实现：例如LEAD
  - ✓ LEAD提供了堆栈操作语言L，我们通过L来调用堆栈的操作（PUSH 和POP）
- 我们可以通过堆栈的实现（LEAD）来操作堆栈
  - ✓ LEDA隐藏了堆栈使用的细节
  - ✓ LEDA 优化了PUSH和POP操作的性能



# 关系代数

怎么理解关系代数？---以堆栈（Stack）的使用为例

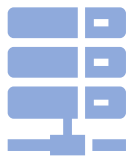
■ 综上，为使用堆栈，我们学习了

关系  
结构

关系  
代数

- 堆栈的结构
- 堆栈的操作 (PUSH, POP)：可以称为“堆栈代数” stack algebra
- 堆栈的一种实现：例如LEAD
  - ✓ LEAD提供了堆栈操作语言L，我们通过L来调用堆栈的操作（PUSH 和POP）
- 我们可以通过堆栈的实现（LEAD）来操作堆栈
  - ✓ LEDA隐藏了堆栈使用的细节
  - ✓ LEDA 优化了PUSH和POP操作的性能

SQL  
语言



# 关系代数

---

什么是代数 ( Algebra ) ?

■ 代数是一种研究数、数量、关系与结构的数学系统，包含：

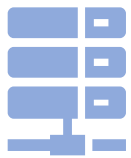
- 运算对象 ( Operands )

用于按照规则生成新的值的变量或值

- 运算符 ( Operators )

用于表示从旧值生成新值的过程 ( 规则 )



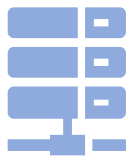


# 关系代数

---

什么是关系代数 ( Relational Algebra ) ?

- 关系代数是对关系的查询语言：以对关系的运算表达查询
- 以关系 ( relations ) 为运算对象
- 其操作符设计为在数据库中我们对于关系最常做的操作
- 其运算结果也是一个关系
- 关系代数运算的三个要素
  - 运算对象：关系
  - 运算结果：关系
  - 运算符：四类



# 关系代数

---

## ■ 四类关系运算符

### ● 集合运算符

- ✓ 将关系看成元组的集合
- ✓ 运算是从关系的“水平”方向即行的角度来进行

### ● 专门的关系运算符

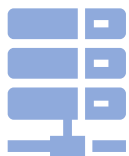
- ✓ 不仅涉及行而且涉及列

### ● 算术比较符

- ✓ 辅助专门的关系运算符进行操作

### ● 逻辑运算符

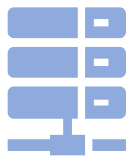
- ✓ 辅助专门的关系运算符进行操作



# 关系代数

## ■ 四类关系运算符

运算符		含义	运算符		含义
集合运算符	$\cup$	并	比较运算符	$>$	大于
	$\cap$	交		$\geq$	大于等于
	$-$	差		$<$	小于
	$\times$	广义笛卡尔积		$\leq$	小于等于
专门的关系运算符				$=$	等于
				$\neq$	不等于
	$\sigma$	选择	逻辑运算符	$\neg$	非
	$\Pi$	投影		$\wedge$	与
	$\bowtie$	连接		$\vee$	或
	$\div$	除			



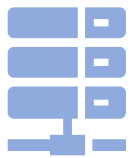
# 关系代数

---

## ■ 四类关系运算符

### ● 另一个概念：基本运算

- ✓ 集合并、集合差、广义笛卡儿积、选择、投影
- ✓ 由基本运算可以推导出其它所有运算

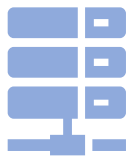


# 关系代数

---

## ■集合的并、交、差运算

- 两个关系R和S若进行并，交，差运算，则它们必须是相容的：
  - ✓ 关系R和S必须是同元的，即它们的属性数目必须相同。
  - ✓ 对于所有 $i$ ，R的第 $i$ 个属性的域必须和S的第 $i$ 个属性的域相同。



# 关系代数

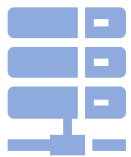
## ■ 并运算 (Union)

- 设R和S是n元关系，并且两者各对应属性的数据类型也相同。则R和S的并运算定义为：

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

- 其结果仍为n元关系，由属于R或属于S的元组构成。
- 运算结果中应消除重复行
- 应用示例：

ActiveTeachers  $\cup$  RetiredTeachers



# 关系代数

## ■ 并运算

R

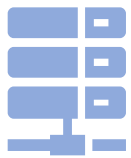
A	B	C
a	b	c
d	a	f
c	b	d

S

A	B	C
b	g	a
d	a	f

RUS

A	B	C
a	b	c
d	a	f
c	b	d
b	g	a



# 关系代数

## ■ 差运算 (Difference)

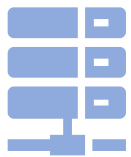
- 设R和S是n元关系，并且两者各对应属性的数据类型也相同。则R和S的差运算定义为：

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

- 其结果仍为n元关系，由属于R且不属于S的元组构成。
- 应用示例：

AllTeachers - RetiredTeachers





# 关系代数

## ■ 差运算

R

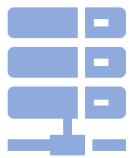
A	B	C
a	b	c
d	a	f
c	b	d

S

A	B	C
b	g	a
d	a	f

$R - S$

A	B	C
a	b	c
c	b	d



# 关系代数

## ■ 交运算 (Difference)

- 设R和S是n元关系，并且两者各对应属性的数据类型也相同。则R和S的交运算定义为：

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

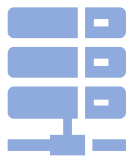
- 其结果仍为n元关系，由既属于R而又属于S的元组构成。
- 交运算可以通过差运算来重写：

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

Derived  
Operation

- 应用示例：

ActiveTeachers  $\cap$  RetiredTeachers



# 关系代数

## ■ 交运算 (Difference)

R

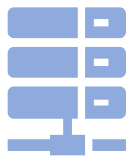
A	B	C
a	b	c
d	a	f
c	b	d

S

A	B	C
b	g	a
d	a	f

$R \cap S$

A	B	C
d	a	f



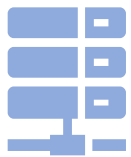
# 关系代数

## ■ 广义笛卡尔积 (Extended Cartesian Product)

- 笛卡尔积：集合运算 vs. 广义笛卡尔积：关系运算
- 两个关系R, S, 其度(degree)分别为n, m, 则它们的广义笛卡尔积是所有这样的元组集合：元组的前n个分量是R中的一个元组, 后m个分量是S中的一个元组。

$$R \times S = \{ t \mid t = \langle r, s \rangle \wedge r \in R \wedge s \in S \}$$

- ✓  $R \times S$ 的度为R与S的度之和,  $R \times S$ 的元组个数为R和S的元组个数的乘积。



# 关系代数

## ■ 广义笛卡尔积 (Extended Cartesian Product)

R

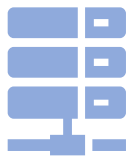
A	B	C
a	b	c
d	a	f
c	b	d

S

A	B	C
b	g	a
d	a	f

$R \times S$

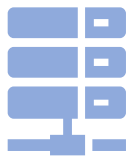
R.A	R.B	R.C	S.A	S.B	S.C
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f



# 关系代数

## ■ 专门的关系运算

- 并、交、叉和笛卡尔积运算只是从行的角度进行，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，从行和列的方向同时进行选择
- 专门的关系运算符包括：
  - 选择  $\sigma$ 、投影  $\Pi$ 、连接  $\bowtie$  和除  $\div$
  - 其中选择、投影是基本运算符



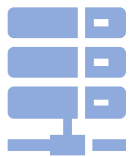
# 关系代数

## ■ 选择 ( selection )

- 选择运算是从指定的关系中选择某些元组形成一个新的关系，被选择的元组是用满足某个逻辑条件来指定的。
- 选择运算表示为：

$$\sigma_F(R) = \{t \mid t \in R \wedge F(t) = true\}$$

- 其中R是关系名， $\sigma$ 是选择运算符，F是逻辑表达式。



# 关系代数

## ■ 选择 ( selection )

R

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

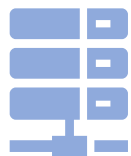
$\sigma_{B=b}(R)$

A	B	C
a	b	c
c	b	d

$\sigma_{B=b \wedge C=c}(R)$

A	B	C
a	b	c





# 关系代数

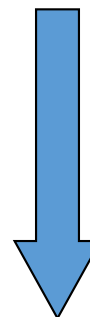
## ■ 选择 ( selection )

订购单关系

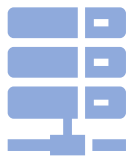
职工号	供应商号	订购单号	订购日期
E3	S7	OR67	2002/06/23
E1	S4	OR73	2002/07/28
E7	S4	OR76	2002/05/25
E6	S6	OR77	2002/06/19
E3	S4	OR79	2002/07/29
E1	S6	OR80	2002/06/22
E3	S6	OR90	2002/07/13
E3	S3	OR91	2002/10/27

从订购单关系中选择职工号为“E3”的元组构成新的关系

$\sigma_{\text{职工号}=\text{"E3"}}(\text{订购单})$



职工号	供应商号	订购单号	订购日期
E3	S7	OR67	2002/06/23
E3	S4	OR79	2002/07/29
E3	S6	OR90	2002/07/13
E3	S3	OR91	2002/10/27



# 关系代数

## ■ 选择 ( selection )

### Selection Example

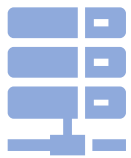
#### Employee

SSN	Name	DepartmentID	Salary
999999999	John	1	30,000
777777777	Tony	1	32,000
888888888	Alice	2	45,000

Find all employees with salary more than \$40,000.

$\sigma_{Salary > 40000}(\text{Employee})$

SSN	Name	DepartmentID	Salary
888888888	Alice	2	45,000



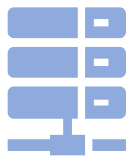
# 关系代数

## ■ 投影 ( projection )

- 投影运算对指定的关系进行投影操作，根据该关系分两步产生一个新关系：
  - ✓ 选择指定的属性，形成一个可能含有重复行的表格
  - ✓ 删除重复行，形成新的关系
- 投影运算定义为：

$$\pi_A(R) = \{r.A \mid r \in R\}$$

- ✓ 其中R是关系名， $\pi$ 是投影运算符，A是被投影的属性或属性集



# 关系代数

## ■ 投影 ( projection )

订购单关系

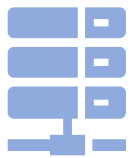
职工号	供应商号	订购单号	订购日期
E3	S7	OR67	2002/06/23
E1	S4	OR73	2002/07/28
E7	S4	OR76	2002/05/25
E6	S6	OR77	2002/06/19
E3	S4	OR79	2002/07/29
E1	S6	OR80	2002/06/22
E3	S6	OR90	2002/07/13
E3	S3	OR91	2002/10/27

选取职工号和供应商号两列构成新的关系

职工号	供应商号
E3	S7
E1	S4
E7	S4
E6	S6
E3	S4
E1	S6
E3	S6
E3	S3



$\pi$ 职工号, 供应商号(订购单)



# 关系代数

## ■ 投影 ( projection )

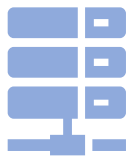
- 选择运算是从关系的水平方向上进行运算的，而投影运算则是从关系的垂直方向上进行的。
- 投影运算可以改变关系的属性次序
- 投影后取消了某些属性列，就可能出现重复行，应该取消这些完全相同的行。所以投影之后，不但减少了属性，元组也可能减少，新关系与原关系不相容。

R

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

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

B	A
b	a



# 关系代数

## ■ 连接 (join)

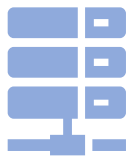
- 连接运算是两个表之间的运算，这两个表通常是具有一对多联系的父子关系。所以连接过程一般是由参照关系的外部关键字和被参照关系的主关键字来控制的，这样的属性通常也称为连接属性
- 定义：

从两个关系的广义笛卡儿积中选取给定属性间满足 $\theta$ 操作的元组。

$$R \bowtie_{A \theta B} S = \{ t \mid t = \langle r, s \rangle \wedge r \in R \wedge s \in S \wedge r[A] \theta s[B] \}$$

$\theta$ 为算术比较符，当 $\theta$ 为等号时称为等值连接； $\theta$ 为“<”时，称为小于连接； $\theta$ 为“>”时，称为大于连接等等。

此类连接也被统称为**Theta连接**( $\theta$ 连接)



# 关系代数

## ■ 连接 (join)

R

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

S

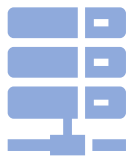
D	E
3	1
6	2

$R \times S$

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

$R \bowtie_{B < D} S$

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



# 关系代数

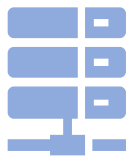
## ■ 连接 ( join )

Sells(	store	beer	price	)	Stores(	name	addr	)
	Walm	Bud	2.50			Walm	Xuey Rd.	
	Walm	Snow	2.75			Caref	Chan St.	
	Caref	Bud	2.50					
	Caref	Yanj	3.00					

StoreInfo := Sells JOIN<sub>Sells.store = Stores.name</sub> Stores

StoreInfo(	Store	beer	price	name	addr	)
	Walm	Bud	2.50	Walm	Xuey Rd.	
	Walm	Snow	2.75	Walm	Xuey Rd.	
	Caref	Bud	2.50	Caref	Chan St.	
	Caref	Yanj	3.00	Caref	Chan St.	





# 关系代数

## ■ 连接 (join)

- 自然连接 (natural join)

在连接运算中最常用的连接是自然连接。

定义:

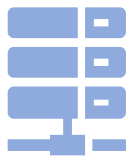
从两个关系的广义笛卡尔积中选取在相同属性列上取值相等的元组，并去掉重复的属性列。

$$R \bowtie S = \{ t \mid t = \langle r, s \rangle \wedge r \in R \wedge s \in S \wedge r[B] = s[B] \}$$

自然连接是特殊的等值连接

等值连接：从两个关系(R、S)的笛卡尔积中选取属性(A、B)值相等的元组

自然连接：在等值连接中取消重复属性



# 关系代数

## ■ 连接 (join)

- 自然连接 (natural join)

R

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

S

B	D	E
b	c	d
b	c	e
a	d	b

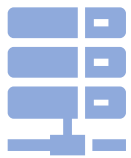
计算广义笛卡尔积  $R \times S$ ;

选择满足条件  $r[B]=s[B]$  的所有元组;

去掉重复的属性。

$R \bowtie S$

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



# 关系代数

## ■ 连接 (join)

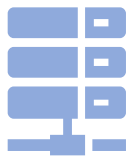
- 自然连接 (natural join)

仓库号	城市	面积
WH1	北京	370
WH2	上海	500
WH3	广州	200
WH5	合肥	130

仓库号	职工号	工资
WH2	E1	1220
WH1	E3	1210
WH2	E4	1250
WH3	E6	1230
WH1	E7	1250



仓库号	城市	面积	职工号	工资
WH2	上海	500	E1	1220
WH1	北京	370	E3	1210
WH2	上海	500	E4	1250
WH3	广州	200	E6	1230
WH1	北京	370	E7	1250

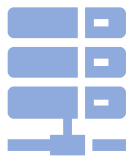


# 关系代数

## ■ 连接 ( join )

- 自然连接 ( natural join )

- ✓ 给定关系模式  $R(A, B, C, D)$ ,  $S(A, C, E)$ , 那  $R \bowtie S$  的关系模式是?
- ✓ 给定  $R(A, B, C)$ ,  $S(D, E)$ , 那么  $R \bowtie S$  是?
- ✓ 给定  $R(A, B)$ ,  $S(A, B)$ , 那么  $R \bowtie S$  是?



# 关系代数

## ■ 关系代数的综合演算

仓库

仓库号	城市	面积
WH1	北京	370
WH2	上海	500
WH3	广州	200
WH5	合肥	130

职工

仓库号	职工号	工资
WH2	E1	1220
WH1	E3	1210
WH2	E4	1250
WH3	E6	1230
WH1	E7	1250

根据以上关系求出在上海工作的职工的工资值都有哪些?

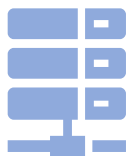
$$R_1 = \sigma_{\text{城市}="上海"}(\text{仓库})$$

$$R_2 = R_1 \bowtie \text{职工}$$

$$R_3 = \pi_{\text{工资}}(R_2)$$

**关系代数语句为：**

$$R = \pi_{\text{工资}}(\sigma_{\text{城市}="上海"}(\text{仓库}) \bowtie \text{职工})$$



# 关系代数

## 关系代数的综合演算

查询过程示意

仓库号	职工号	工资
WH2	E1	1220
WH1	E3	1210
WH2	E4	1250
WH3	E6	1230
WH1	E7	1250

职工

仓库号	城市	面积
WH1	北京	370
WH2	上海	500
WH3	广州	200
WH5	合肥	130

仓库

选择运算

仓库号	城市	面积
WH2	上海	500

R<sub>1</sub>

自然连接运算

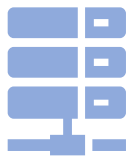
仓库号	城市	面积	职工号	工资
WH2	上海	500	E1	1220
WH2	上海	500	E4	1250

R<sub>2</sub>

投影运算

工资
1220
1250

R<sub>3</sub>



# 关系代数

## ■ 连接 (join)

- 外连接 (outer join)

例：列出老师的有关信息，包括姓名、工资、所教授的课程。

$$\Pi_{P\#, PN, SAL, C\#, CN}((PROF) \bowtie PC \bowtie C)$$

PROF:

P#	PN	SAL
P01	赵明	800
P02	钱广	700
P03	孙立	600
P04	李三	500

PC:

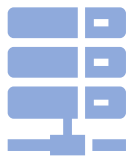
C#	P#
C01	P01
C02	P02
C02	P04

C:

C#	CN
C01	物理
C02	数学
C03	化学

问题：有关P03号职工的姓名和工资信息没有显示出来。

P#	PN	SAL	C#	CN
P01	赵明	800	C01	物理
P02	钱广	700	C02	数学
P04	李三	500	C02	数学



# 关系代数

## ■ 连接 ( join )

- 外连接 ( outer join )

为避免自然连接时因失配而发生的信息丢失，可以假定往参与连接的一方表中附加一个取值全为空值的行，它和参与连接的另一方表中的任何一个未匹配上的元组都能匹配，称之为外连接。

外连接 = 自然连接 + 失配的元组。

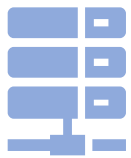
外连接的形式：左外连接、右外连接、全外连接。

$R \bowtie S$  左外连接 = 自然连接 + 左侧表中失配的元组。

$R \bowtie S$  右外连接 = 自然连接 + 右侧表中失配的元组。

$R \bowtie S$  全外连接 = 自然连接 + 两侧表中失配的元组。





# 关系代数

## ■ 连接 (join)

### ● 外连接 (outer join)

R

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

S

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

$R \bowtie S$

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

$R \bowtie_{\text{left}} S$

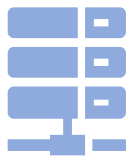
A	B	C	D
a	b	c	d
a	b	c	e
c	a	d	b
b	b	f	null
null	e	f	g

$R \bowtie_{\text{right}} S$

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

$R \bowtie_{\text{full}} S$

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



# 关系代数

## ■ 除 ( divide )

- 除运算解决什么问题？

示例：

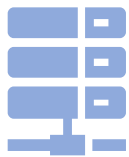
**R**

姓名	课程
张军	物理
王红	数学
张军	数学

**S**

课程
数学
物理

求选修了全部课程的学生

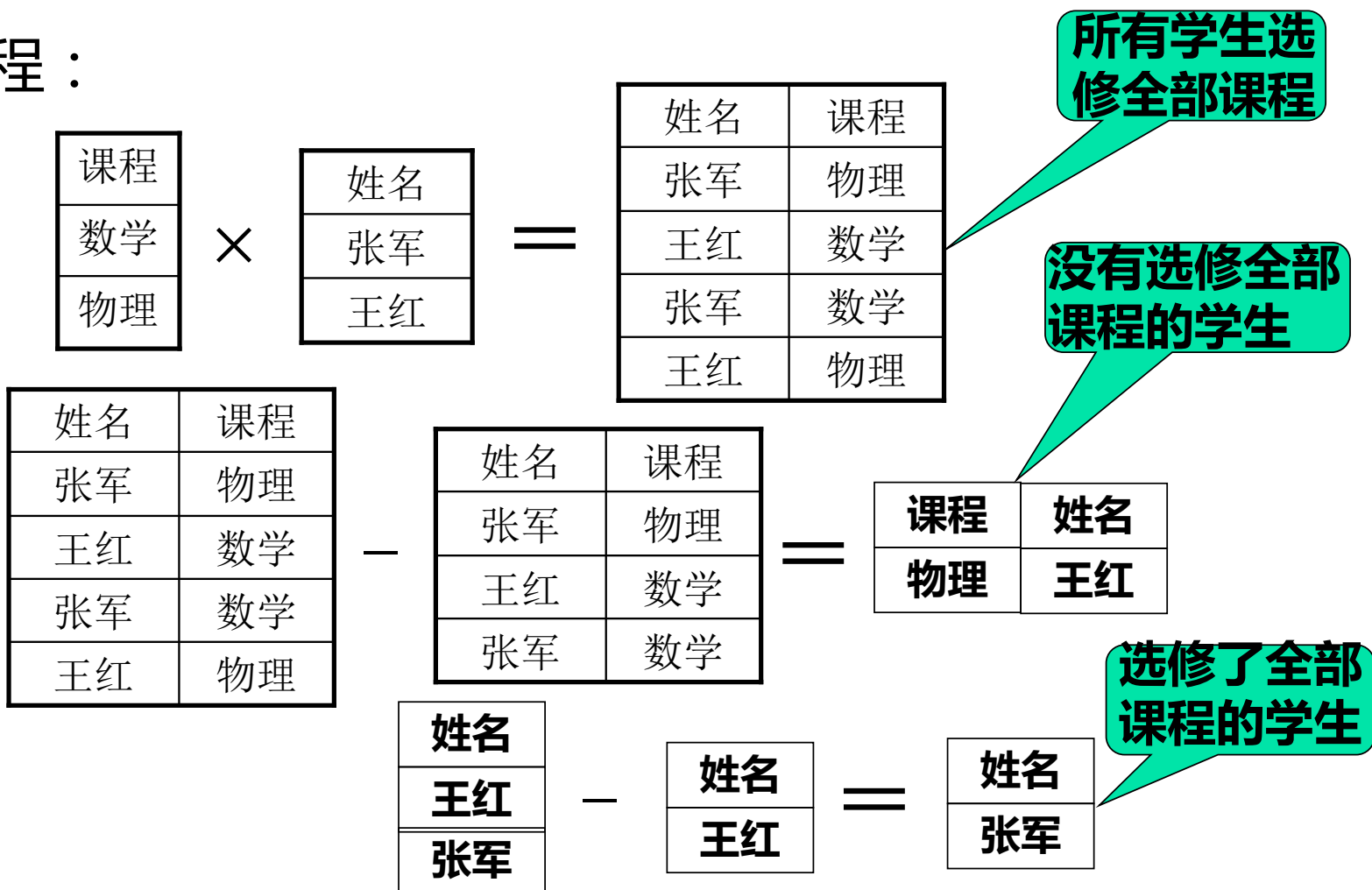


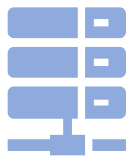
# 关系代数

## ■ 除 ( divide )

- 除运算解决什么问题？

解题过程：





# 关系代数

## ■ 除 ( divide )

- 除运算是上述运算过程的等价算法，常用于包含语义“至少...”的语义的查询或运算

设 $R(X,Y)$ 和 $S(Y)$ 是两个关系，则

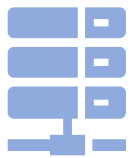
$$R \div S = \Pi_X(R) - \Pi_X((\Pi_X(R) \times S) - R)$$

Derived  
operation

- 除运算定义：

设有关系 $R(X,Y)$ 和 $S(Y)$ ，其中 $X$ 、 $Y$ 可以是单个属性或属性集，则：

$$R \div S = \{r.X \mid r \in R \wedge Y_x \supseteq S\}$$

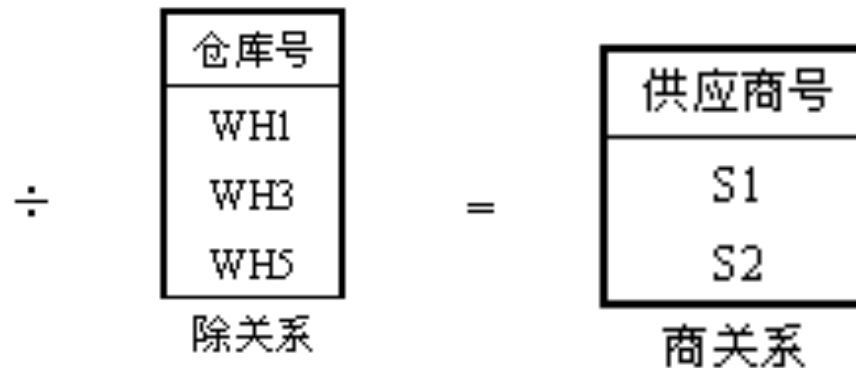


# 关系代数

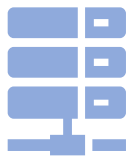
## ■ 除 ( divide )

### ● 理解除运算

仓库号	供应商号
WH1	S1
WH3	S1
WH5	S1
WH1	S2
WH3	S2
WH5	S2
WH6	S2
WH1	S3
WH2	S3
WH5	S4



它的含义是：至少向**WH1、WH3、WH5**供货的供应商号。



# 关系代数

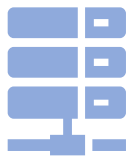
- 基本运算 ( basic operation ) 与派生运算 ( derived operation )

引进派生运算并不增加语言的能力，但是可以简化表达

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

$$R \bowtie S = \pi_X (\sigma_{r[A_i]=s[B_j]} (R \times S))$$

$$R \div S = \pi_X (R) - \pi_X ((\pi_X (R) \times S) - R)$$



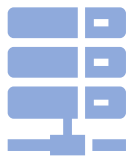
# 关系代数

## ■ 关系代数的能力限制

- 关系代数并不能完成所有想要的查询运算
- 例如：不能计算传递闭包 ( transitive closure)

Name1	Name2	Relationship
Fred	Mary	Father
Mary	Joe	Cousin
Mary	Bill	Spouse
Nancy	Lou	Sister

无法用关系代数从上表中找出Fred所有的亲属（有些亲属是非直系的）  
这类查询需用高级编程语言编程实现



# 关系代数

## ■练习

### STUDENT

Sno	Sname	Sex	Sage	Sdept
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

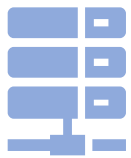
### SC

Sno	Cno	Grade
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80

### COURSE

Cno	Cname	Cpno	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





# 关系代数

## ■练习

□查找学号为95001的学生姓名和年龄

$$\Pi_{Sname, Sage} ( \sigma_{Sno='95001'} (STUDENT))$$

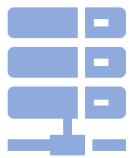
□查找信息系的男学生的学号和姓名

$$\Pi_{Sno, Sname} ( \sigma_{Sdept='IS' \wedge Ssex='男'} (STUDENT))$$

□查找选修了1号课程或3号课程的学生学号

$$\Pi_{Sno} ( \sigma_{Cno=1 \vee Cno=3} (SC))$$

$$\text{或：} \Pi_{Sno} ( \sigma_{Cno=1} (SC)) \cup \Pi_{Sno} ( \sigma_{Cno=3} (SC))$$



# 关系代数

## ■练习

□求选修了1号而没有选2号课程的学生号。

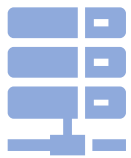
$$\Pi_{Sno}(\sigma_{Cno = 1}(SC)) - \Pi_{Sno}(\sigma_{Cno = 2}(SC))$$

□求选修了1号和2号课程的学生号。

$$\Pi_{Sno}(\sigma_{Cno = 1}(SC)) \cap \Pi_{Sno}(\sigma_{Cno = 2}(SC))$$

□求选修了1号课程的学生号和姓名。

$$\Pi_{Sno, Sname}(\sigma_{Cno = 1}(Student \bowtie SC))$$



# 关系代数

## ■练习

- 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。

$$\Pi_{\text{Sname}}(\sigma_{\text{Cpno} = 5}(\text{Course}) \bowtie \text{SC} \bowtie \Pi_{\text{Sno}, \text{Sname}}(\text{STUDENT}))$$

- 列出所有学生的选课情况

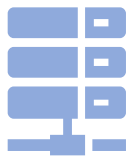
$$\text{STUDENT} \bowtie \text{SC}$$

- 查询选修了全部课程的学生号和姓名

$$((\Pi_{\text{Sno}, \text{Cno}}(\text{SC}) \div \Pi_{\text{Cno}}(\text{Course})) \bowtie \Pi_{\text{Sno}, \text{sname}}(\text{STUDENT}))$$

- 思考题：查询最多选修了一门其直接先行课为5号课程的学生姓名

学生表和学生表进行连接，Cno不相同的连在一起！！！！



# 关系代数

Agents(代理商)

aID	aname	City
a001	Huang	CQ
a002	Yao	NJ
a003	Cai	DL
a004	liu	Shanghai

## ■作业

Customers

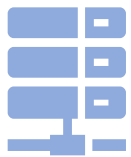
CID	Cname	City
C001	Zhang	BJ
c002	Li	TJ
c003	Zhao	Hangzhou
c004	Wang	Shanghai

products

Pid	Pname	City	price
P01	Comb	Shangha i	0.5
p02	brush	bj	0.5
p03	pencil	hz	1

Orders

Ordno	month	cid	Aid	Pid	Qty
1	Jan	c003	A01	p03	1000
2	Jan	C002	A02	P02	1000
3	feb	C001	a03	P01	100

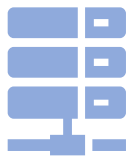


# 关系代数

---

## ■作业

- 找出所有顾客、代理商和商品都在同一个城市的三元组(cid,aid,pid)。
- 找出所有顾客、代理商和商品两两不在同一个城市的三元组(cid,aid,pid)。
- 取出至少被一个在杭州的顾客通过位于上海的代理商订购的商品的名字。列出所有在同一个城市的代理商的aid对。

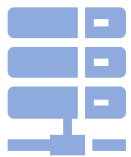


# 关系代数

---

## ■作业

- 取出销售过所有曾被顾客c002定购过的商品的代理商的名字。
- 取出所有的三元组 ( cid , aid , pid ) , 要求对应的顾客, 代理商和商品中至少有两者是位于同一座城市。
- 取出接受过上海的顾客一笔总额超过 ¥ 500的订单的代理商的aid值。
- 取出只从一家代理商处定购过商品的顾客的cid值。



# 关系代数

---

## ■作业（接上）

- 课本70页第6题（只做关系代数），第7、8题
- 提交时间：下一次上课前