



第二章 关系数据库

- 2.1 概述
- 2.2 关系基本概念
- 2.3 关系模型
- 2.4 关系代数
- 2.5 关系演算



第二章 学习目标

- 关系模型的数据表示
- 完整性约束的表达
- 数据操纵的实现
 - 关系代数
 - 关系演算



2.1 概述

- 一九七〇年，IBM公司的E.F.Codd发表论文,首先提出了关系数据模型。随后他又发表一系列论文，阐述了关系规范化的概念。（A Relational Mode of Data for Large Shared Data Banks)
- The purpose of models is not to fit the data but to sharpen the question(Samuel Karlin)
- 早期代表系统
 - System R：由IBM研制
 - INGRES：由加州Berkeley分校研制
- 目前主流的商业数据库系统
 - Oracle, Informix, Sybase, DB2 , SQL Server
 - Access, Foxpro, Foxbase
 - Postgres SQL,Mysql



Codd的十二条准则

Rule 1: The Information Rule

Rule 2: Guaranteed Access Rule

Rule 3: Systematic Treatment of Null Values

Rule 4: Dynamic On-line Catalog Based on the Relational Model

Rule 5: Comprehensive Data Sub language Rule

Rule 6: View Updating Rule

Rule 7: High-level Insert, Update, and Delete

Rule 8: Physical Data Independence

Rule 9: Logical Data Independence

Rule 10: Integrity Independence

Rule 11: Distribution Independence

Rule 12: Non subversion Rule



2.2 关系基本概念

- 域 (Domain)
- 笛卡尔积 (Cartesian Product)
- 关系 (Relation)
- 候选码 (Candidate Key)
- 主码 (Primary Key)
- 主属性 (Primary Attribute)
- 外部码 (Foreign Key)



2.2 关系基本概念(续)

- 域

- 一组值的集合，这组值具有相同的数据类型。

- 笛卡尔积

- 一组域 D_1, D_2, \dots, D_n 的笛卡尔积为:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, \dots, n\}$$

- 笛卡尔积的每个元素 (d_1, d_2, \dots, d_n) 称作一个n-元组 (n-tuple)
 - 元组的每一个值 d_i 叫做一个分量 (component)

- 若 D_i 的基数为 m_i ，则笛卡尔积的基数为 $\prod_{i=1}^n m_i$



2.2 关系基本概念(续)

• 例： 设

D_1 为教师集合 $(T) = \{t_1, t_2\}$

D_2 为学生集合 $(S) = \{s_1, s_2, s_3\}$

D_3 为课程集合 $(C) = \{c_1, c_2\}$

则 $D_1 \times D_2 \times D_3$ 是个三元组集合，元组个数为 $2 \times 3 \times 2$ ，是所有可能的（教师，学生，课程）元组集合

2.2 关系基本概念(续)

- 笛卡儿积可表示为一个二维表

T	S	C
t ₁	s ₁	c ₁
t ₁	s ₁	c ₂
t ₁	s ₂	c ₁
...
t ₂	s ₃	c ₂

元组

域(课程
集合)



2.2 关系基本概念(续)

学号	姓名	性别	年龄	住址	班级
0012011	张三	男	21	武汉	001
0012012	李四	女	18	北京	001
0012013	王五	男	18	长沙	002
0012012	李四	男	18	北京	002
。 。 。	。 。 。				

2.2 关系基本概念(续)

• 关系

- 笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的子集叫做在域 D_1, D_2, \dots, D_n 上的关系, 用 $R(D_1, D_2, \dots, D_n)$ 表示

- R 是关系的名字, n 是关系的度或目

单元关系 (unary relation)

二元关系 (binary relation)

- 关系是笛卡尔积中有意义的子集, 关系也可以表示为二维表

关系TEACH(T, S, C)

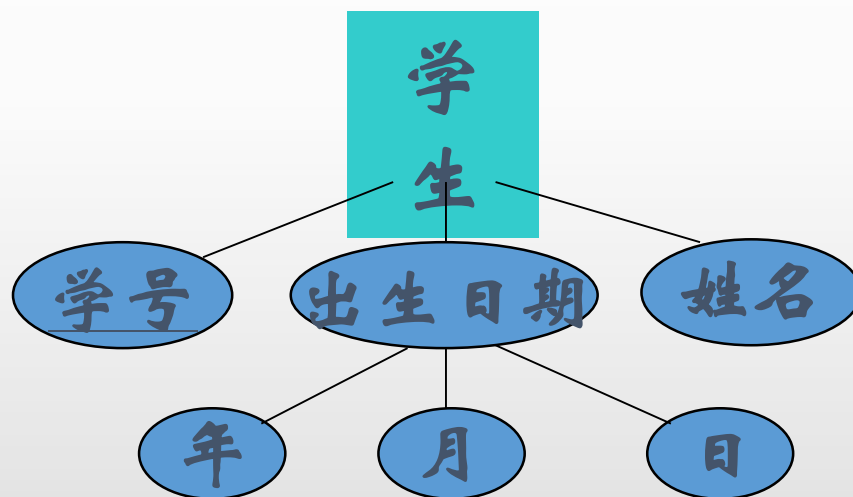
T	S	C
t_1	s_1	c_1
t_1	s_1	c_2
t_1	s_2	c_1
t_2	s_3	c_2

元组 (行)

属性 (列)

2.2 关系基本概念(续)

- 关系的性质
 - 列是同质的
 - 行列的顺序无关紧要
 - 任意两个元组不能完全相同
 - 每一分量必须是不可再分的数据。(原子特性)
 - 不同的属性，属性名不能相同



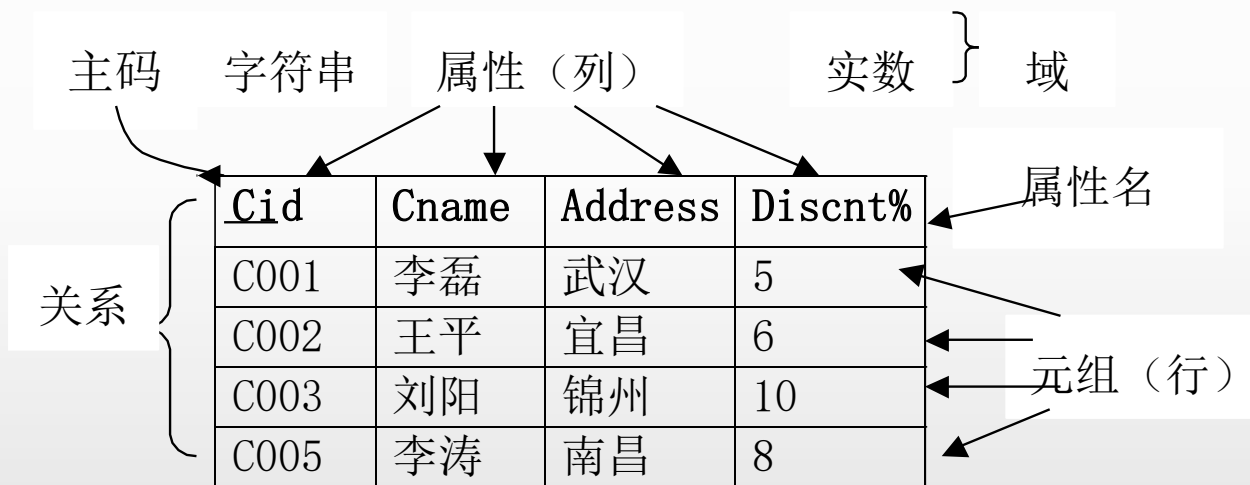


2.2 关系基本概念(续)

- 候选码
 - 关系中的一个属性组，其值能唯一标识一个元组。若从属性组中去掉任何一个属性，它就不具有这一性质了，这样的属性组称作候选码。
 - 任何一个候选码中的属性称作**主属性**
- 主码
 - 进行数据库设计时，从一个关系的多个候选码中选定一个作为主码
 - 主码的选择问题
- 外部码
 - 关系R中的一个属性组，它不是R的码，但它与另一个关系S的码相对应，则称这个属性组为R的外部码。



2.2 关系基本概念(续)





2.2 关系基本概念(续)

• 补充说明

笛卡儿积不满足交换律，因此，按照数学定义，
 $(d_1, d_2, \dots, d_n) \neq (d_2, d_1, \dots, d_n)$ 。当关系作为关系
数据模型的数据结构时，我们需要给予如下的限定和扩充：

(1) 限定关系数据模型中的关系必须为有限集合

(2) 关系元组的无序性：

$$(d_1, d_2, \dots, d_n) = (d_2, d_1, \dots, d_n)$$

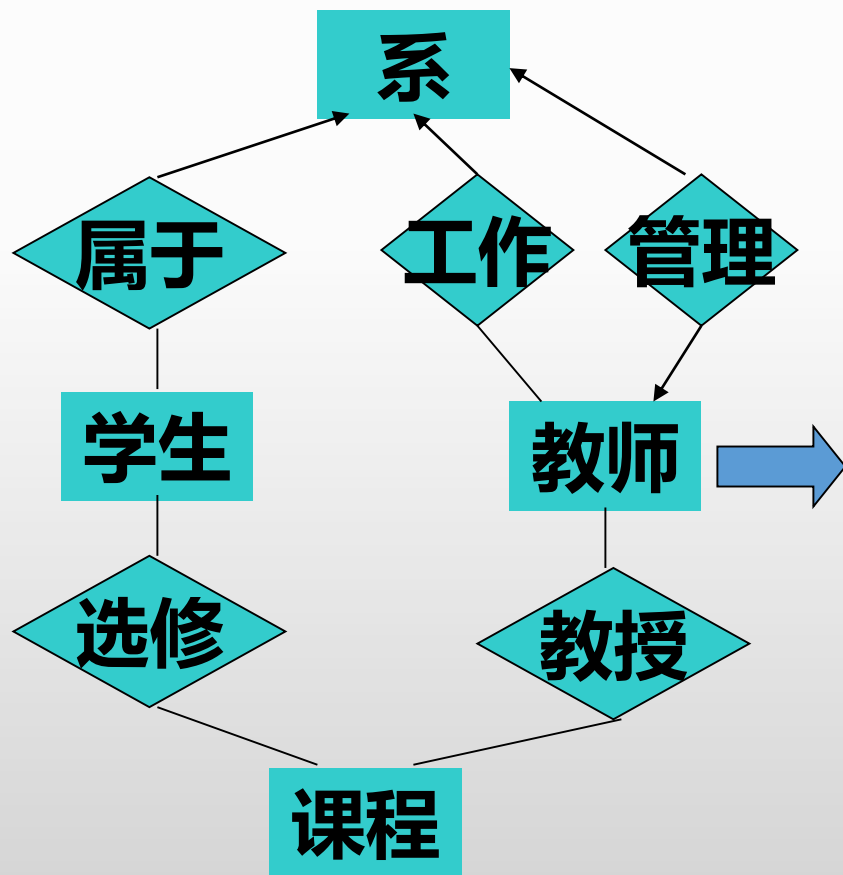


2.3 关系模型

- 2.3.1 数据结构
- 2.3.2 关系模式
- 2.3.3 关系的完整性约束
- 2.3.4 关系操作
- 2.3.5 关系语言

2.3 关系模型

• 2.3.1 数据结构



DEPT(D# , DN , DEAN)

S(S# , SN , SEX , AGE , D#)

C(C# , CN , PC# , CREDIT)

SC(S# , C# , SCORE)

EMPL(P# , PN, D# , SAL)

TEACH(P# , C#)



2.3 关系模型（续）

- 2.3.2 关系模式

- 1. 定义

关系的描述称作关系模式，包括关系名、关系中的属性名、属性向域的映象、属性间的数据依赖关系等。

其中，域名以及属性向域的映射常常直接反映属性的类型、长度。

形式化为： $R(U, D, dom, F)$ ，简记为 $R(A_1, A_2, \dots, A_n)$ 或 $R(U)$ 。

- 2. 说明



2.3 关系模型（续）

- 2.3.2 关系模式

关系模式	关系
型	值
静态、稳定	动态，内容 随时间变化

- 3. 关系数据库

- 关系模式的集合，数据库描述，数据库的内涵(Intension)
- 某一时刻关系的集合，数据库的外延(Extension)



2.3 关系模型（续）

- 2.3.3 关系的完整性约束（Integrity Constraint, IC）
- 1.作用
- 2.实体完整性
- 3.参照完整性
- 4.用户自定义完整性
- 5. 实现



2.3 关系模型（续）

- 2.3.3 关系的完整性约束 (Integrity Constraint, IC)

- 2. 实体完整性

- 空值：不知道、不存在或无意义
 - 关系的主属性中的属性值不能为空值
 - 意义



2.3 关系模型（续）

- 2.3.3 关系的完整性约束（Integrity Constraint, IC）
 - 3. 参照完整性
 - 如果关系 R_2 的外部码 F_k 与关系 R_1 的主码 P_k 相对应，则 R_2 中的每一个元组的 F_k 值或者等于 R_1 中某个元组的 P_k 值，或者为空值
 - 意义：
 - 示例：关系 S 在 $D\#$ 上的取值有两种可能
 - 空值,...
 - 若非空值,...
 - 约束方式：
 - 插入规则、删除规则、修改规则



2.3 关系模型（续）

- 2.3.3 关系的完整性约束（Integrity Constraint, IC）

- 3. 参照完整性

- 父表
- 子表
- 外码的取值
- 插入规则：在子表中插入记录时应遵循的规则
 - 限制，递归，忽略
- 删除规则：在父表中删除记录时应遵循的规则；
 - 级联，限制，置空值删除，忽略
- 更新规则：当父表的关键字被修改时应遵循的规则；
 - 级联，限制，忽略

举例:

- Class (clano, major, dept, tno)
- Students (sno, sname, sex, birthdate, clano)
- Sc (sno, cno, point1, point2)

本身不是
主属性的
外码

本身既是主
属性，又是
外部码



2.3 关系模型（续）

供应商关系S（主码是“供应商号”）

供应商号	供应商名	所在城市
B01	红星	北京
S10	宇宙	上海
T20	黎明	天津
Z01	立新	重庆

零件关系P（主码是“零件号”，外码是“供应商号”）

零件号	颜色	供应商号
010	红	B01
312	白	S10
201	蓝	T20

今要向关系P中插入新行，
新行的值分别列出如下。哪些
行能够插入？

- A. (null, ‘黄’ , ‘T20’)
- B. (‘201’, ‘红’ , ‘T20’)
- C. (‘105’, ‘蓝’ , ‘B01’)
- D. (‘101’, ‘黄’ , ‘T11’)
- E. (‘037’, ‘绿’ , null)



2.3 关系模型（续）

- 2.3.3 关系的完整性约束 (Integrity Constraint, IC)
 - 4. 用户定义的完整性
 - 用户针对具体的应用环境定义的完整性约束条件
 - 如S#要求是8位整数, SEX要求取值为“男”或“女”
 - 5. 系统支持
 - 实体完整性和参照完整性由系统自动支持
 - 系统应提供定义和检验用户定义的完整性的机制



2.3 关系模型（续）

- 2.3.4 关系操作
- 1. 说明
 - 关系操作，一次一集合（Set-at-a-time）的方式
 - 而非关系型的数据操作方式，一次一记录（Record-at-a-time）
 - 关系操作可以用关系代数和关系演算两种方式来表示
 - 关系代数是一种代数的符号
 - 关系演算是一种逻辑符号



2.3 关系模型（续）

- 2.3.4 关系操作
- 2.分类
 - 关系代数
 - 传统集合运算（并，交，差…）
 - 专门关系运算（选择，投影，连接）
 - 关系演算
 - 域关系演算
 - 元组（记录）关系演算



2.3 关系模型（续）

- 2.3.5 关系语言

- 1. 分类

- 关系代数
 - 用对关系的运算来表达查询，需要指明所用操作
- 关系演算
 - 用谓词来表达查询，只需描述所需信息的特性
- 元组关系演算
 - 谓词变元的基本对象是元组变量
- 域关系演算
 - 谓词变元的基本对象是域变量



2.3 关系模型 (续)

- 2.3.4 关系语言

- 2.实际应用

- SQL
- QUEL
- QBE



2.3 关系模型（续）

- 2.3.5 关系语言
- 3.关系数据语言的特点
 - 一体化
 - 非过程化
 - 面向集合的存取方式



关系模型的优缺点

- 优点

- * 数据结构简单（实体、联系、数据字典、索引）
- * 有扎实的理论基础（关系运算理论、关系模式设计理论）
- * 数据独立性强
- * 实现集合操作
- * 直接用关系表示M:N的联系

- 缺点

- * 查询效率低
- * 内存资源消耗大
- * 设计人员应熟悉关系理论



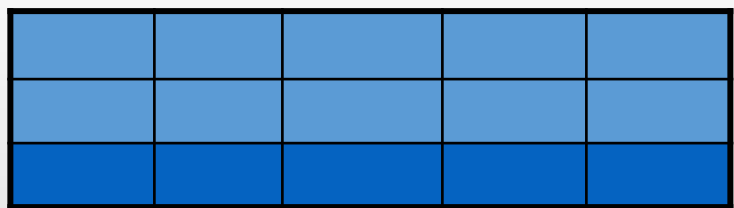
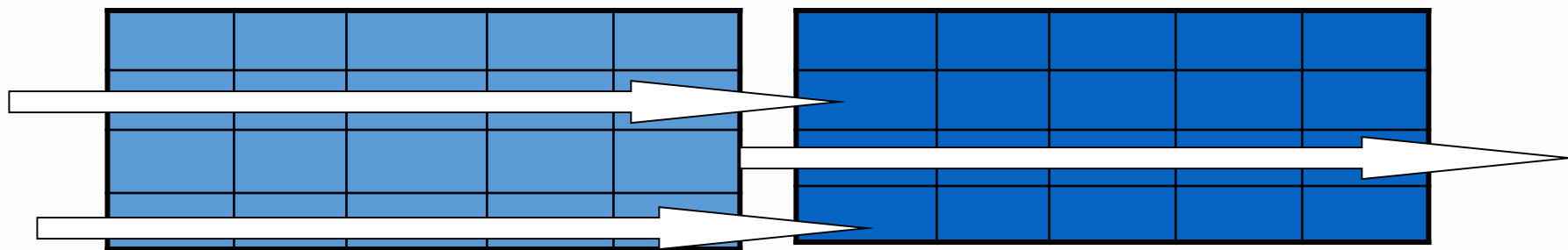
2.4 关系代数

- 2.4.1 关系代数运算介绍
- 2.4.2 传统的集合运算
- 2.4.3 专门的关系运算
- 2.4.4 其他操作



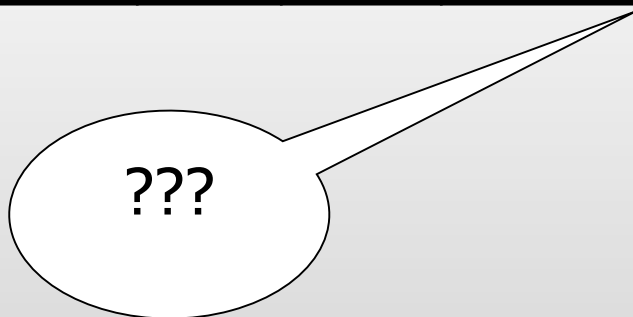
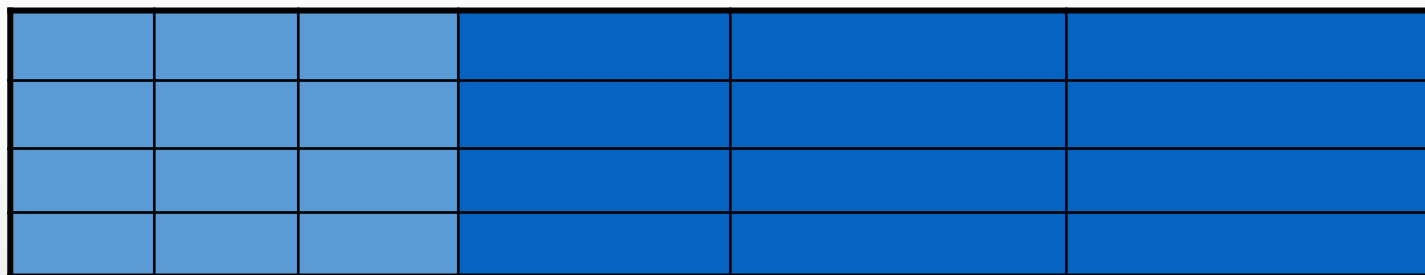
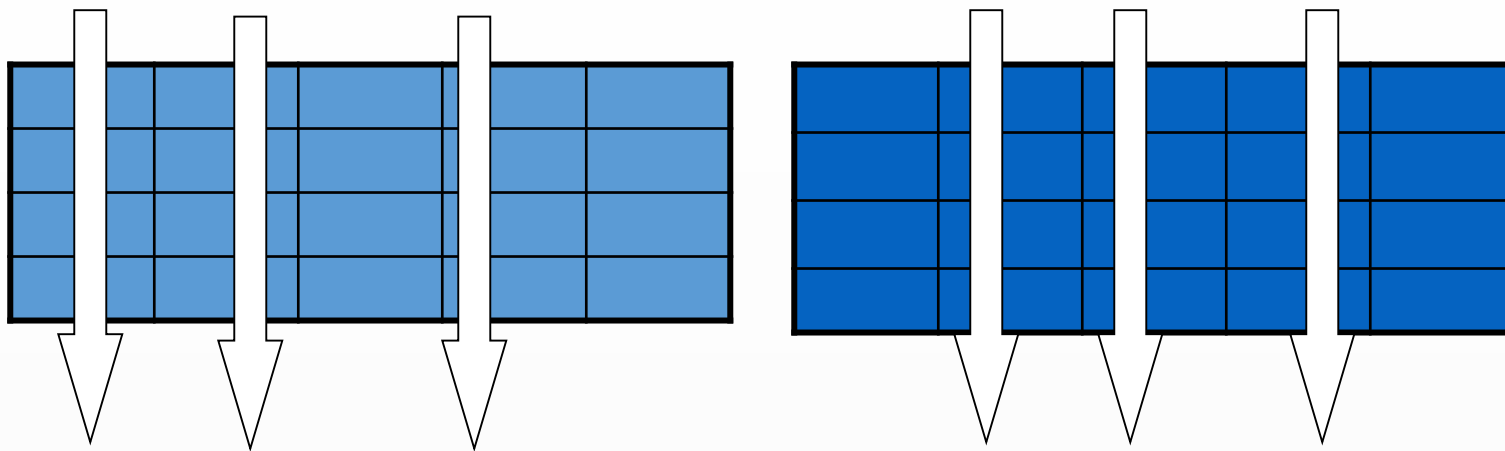
2.4 关系代数（续）

- 2.4.1 关系代数运算介绍
- 1. 运算汇总
 - 基本运算
 - 一元运算
 - 选择、投影
 - 多元运算
 - 笛卡儿积、并、交、集合差
 - 其它运算
 - 自然连接、除
 - 扩展运算
 - 内、外连接



???

传统集合运算
并、差、交、笛
卡尔积 \in



专门关系运算
投影、选择、联接



2.4 关系代数（续）

- 2.4.1 关系代数运算介绍

- 2. 记号说明

给定关系模式 $R(A_1, A_2, \dots, A_n)$ ，设 R 是它的一个具体的关系， $t \in R$ 是关系的一个元组

- 分量

设 $t \in R$ ，则 $t[A_i]$ 表示元组 t 中相应于属性 A_i 的一个分量

- 属性列

$A_i = \{A_{i1}, A_{i2}, \dots, A_{ik}\} \subseteq \{A_1, A_2, \dots, A_n\}$ ，称 A 为属性列

\bar{A} 表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 A_i 后剩余的属性组

$t[A_i] = (t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}])$



2.4 关系代数（续）

- 2.4.1 关系代数运算介绍
- 3. 关系代数表达式
 - 基本关系运算的有限次复合而成的式子。

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

$$\Pi_{DN}(\sigma_{S\# = 001}(S) \bowtie DEPT)$$

2.4 关系代数 (续)

• 2.4.3 传统的集合运算

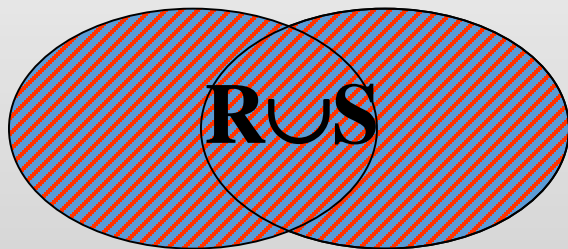
• 1. 并

• 定义

- 所有至少出现在两个关系中之一的元组集合

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

- 两个关系R和S若进行并运算，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 $\forall i$ ，R的第i个属性的域必须和S的第i个属性的域相同



并运算示例 I

R

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

S

A	B	C
b	g	a
b	a	f

T

D	E	F
g	h	i
j	k	l

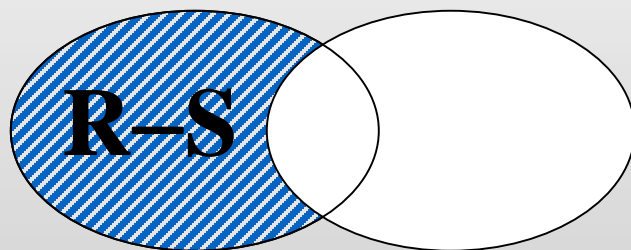
$R \cup S$

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



2.4 关系代数（续）

- 2.4.2 传统的集合运算
- 2. 差运算
 - 定义
 - 所有出现在一个关系而不在另一关系中的元组集合
$$R-S = \{ r \mid r \in R \wedge r \notin S \}$$
 - R和S必须是相容的



差运算示例 I

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

S

A	B	C
3	4	5
7	2	3

R—S ?

A	B	C
3	6	7
2	5	7
4	4	3

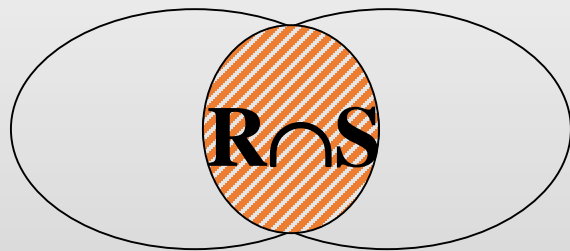
S—R ?

A	B	C
3	4	5



2.4 关系代数（续）

- 2.4.2 传统的集合运算
- 3. 交运算
 - 定义
 - 所有同时出现在两个关系中的元组集合 $R \cap S = \{ r \mid r \in R \wedge r \in S \}$
 - 交运算可以通过差运算来重写 $R \cap S = R - (R - S)$



交运算示例 I

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

S

A	B	C
3	4	5
7	2	3

$R \cap S$

A	B	C
7	2	3



交运算示例 II

- 示例

求同时选修了001号和002号课程的学生号

? ?

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

? ?

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



2.4 关系代数 (续)

- 2.4.2 传统的集合运算
- 4. 广义笛卡尔积
 - 定义1: 元组的连串 (Concatenation)
 - 若 $r = (r_1, \dots, r_n)$, $s = (s_1, \dots, s_m)$, 则定义 r 与 s 的连串为:

$$\widehat{rs} = (r_1, \dots, r_n, s_1, \dots, s_m)$$



2.4 关系代数 (续)

- 2.4.2 传统的集合运算
- 4. 广义笛卡尔积
 - 定义2
 - 两个关系R, S, 其度分别为n, m, 则它们的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×S的度为R与S的度之和, R×S的元组个数为R和S的元组个数的乘积

$$R \times S = \{ \overbrace{rs} \mid r \in R \wedge s \in S \}$$



广义笛卡尔积运算示例

r

<i>A</i>	<i>B</i>
α	1
β	2

s

<i>C</i>	<i>D</i>	<i>E</i>
α	10	<i>a</i>
β	10	<i>a</i>
β	20	<i>b</i>
γ	10	<i>b</i>

r X *s*

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
α	1	α	10	<i>a</i>
α	1	β	19	<i>a</i>
α	1	β	20	<i>b</i>
α	1	γ	10	<i>b</i>
β	2	α	10	<i>a</i>
β	2	β	10	<i>a</i>
β	2	β	20	<i>b</i>
β	2	γ	10	<i>b</i>

2.4 关系代数 (续)

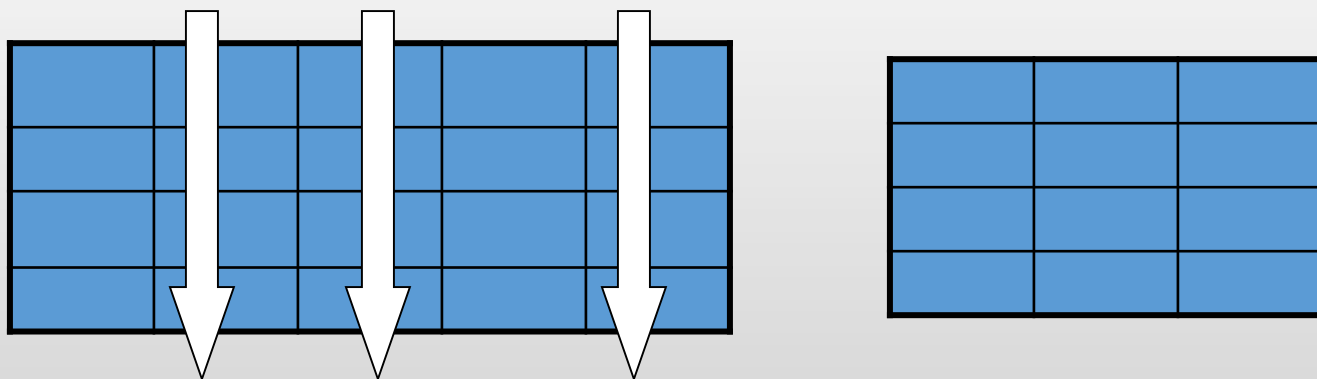
- 2.4.3 专门的关系运算

- 1. 投影

- 从关系R中取若干列组成新的关系 (从列的角度)

$$\Pi_A(R) = \{ t[A] \mid t \in R \}, A \subseteq R$$

- 投影的结果中要去掉相同的行*





投影运算示例

R

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

$\Pi_{B, C}(R)$

B	C
b	c
e	f

给出所有学生的姓名和年龄

$\Pi_{SN, AGE}(S)$

找001号学生所选修的课程号

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



投影

Student(S#,SN,Age)

Course(C#,CN)

SC(C#,S#,Score)

- 示例

给出所有学生的姓名和年龄

$$\Pi_{SN, AGE}(S)$$

找001号学生所选修的课程号

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



2.4 关系代数 (续)

- 2.4.3 专门的关系运算

- 2. 选择

- 在关系R中选择满足给定条件的元组 (从行的角度)

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

- F是选择的条件, $\forall t \in R$, $F(t)$ 要么为真, 要么为假
 - F的形式: 由 **逻辑运算符** 连接 **算术表达式** 而成

逻辑运算符: \wedge, \vee, \neg

算术表达式: $X \theta Y$

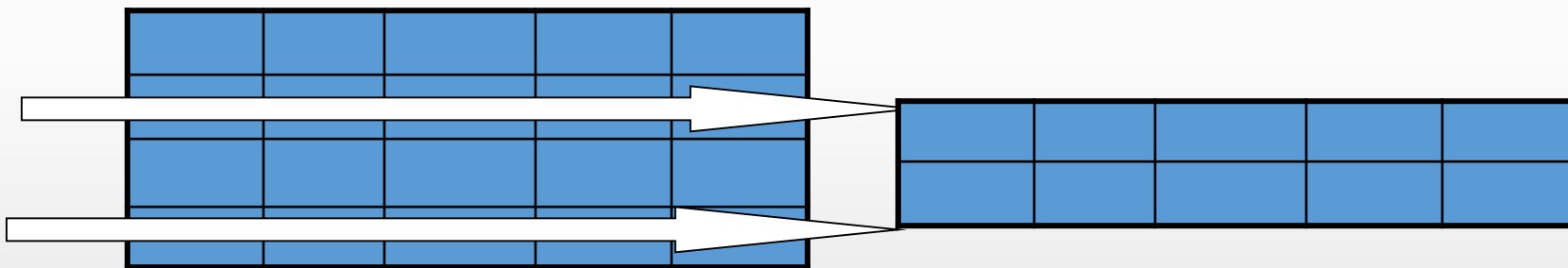
X, Y是属性名、常量、或简单函数

θ 是比较算符, $\theta \in \{>, \geq, <, \leq, =, \neq\}$

2.4 关系代数 (续)

- 2.4.3 专门的关系运算

- 2. 选择





选择运算示例

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

$\sigma_{A<5}(R)$

A	B	C
3	6	7
2	5	7
4	4	3

$\sigma_{A<5 \wedge C=7}(R)$

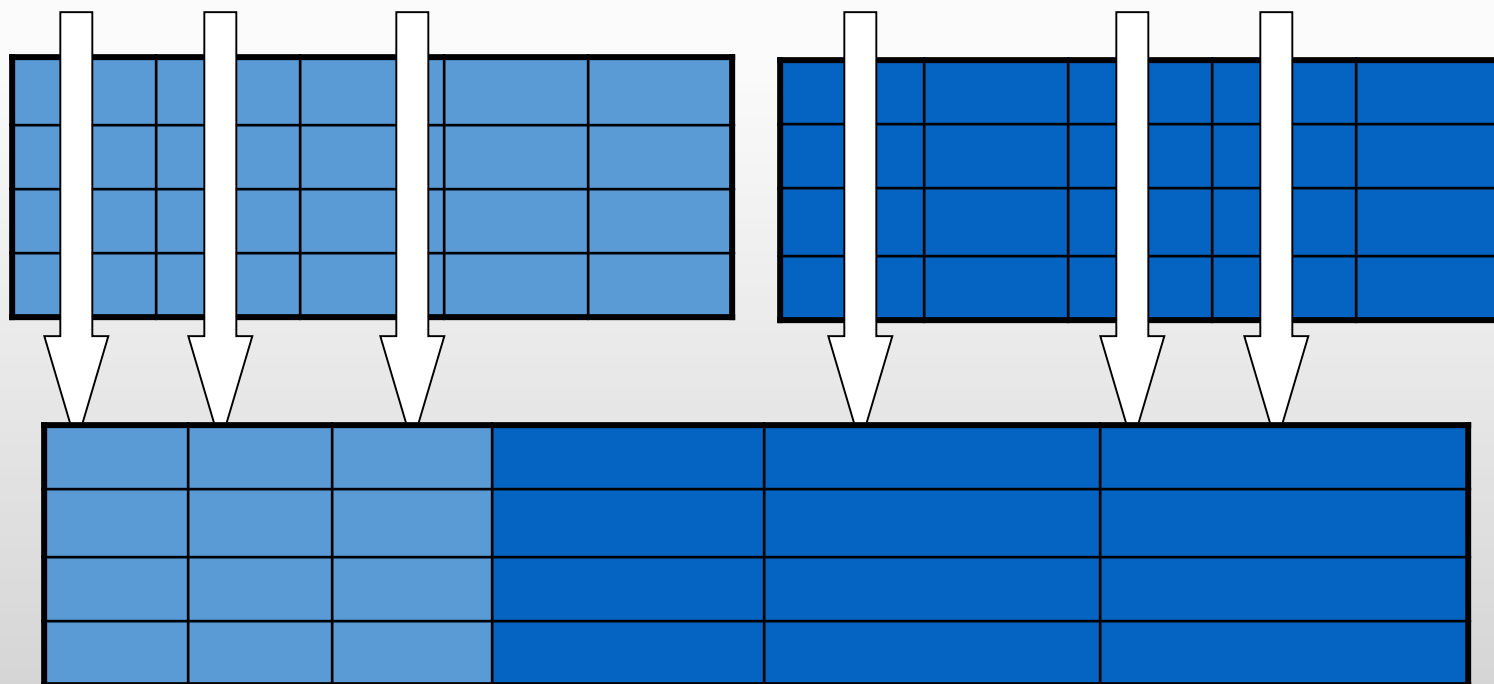
A	B	C
3	6	7
2	5	7

2.4 关系代数 (续)

- 2.4.3 专门的关系运算

- 3. 连接

- (1) θ 连接
- (2) 自然连接





2.4 关系代数（续）

- 2.4.3 专门的关系运算

- 3. 连接

- θ 连接

- 从两个关系的广义笛卡儿积中选取给定属性间满足一定条件的元组

$$R \bowtie_{A \theta B} S = \{ \overbrace{rs} \mid r \in R \wedge s \in S \wedge r[A] \theta s[B] \}$$

- A, B 为 R 和 S 上度数相等且可比的属性列
 - θ 为算术比较符，为等号时称为等值连接

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

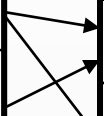
θ连接示例

R

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

S

D	E
3	1
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

θ连接示例 II

• $R \bowtie S$

A	B	C	D	E
α	1	α	10	a
α	1	β	19	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

• $\sigma_{A=C}(R \bowtie S)$

• $R \mid_{X/S}$

$A=C$

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



2.4 关系代数（续）

- 2.4.3 专门的关系运算

- 3. 连接

- 自然连接

- 从两个关系的广义笛卡儿积中选取在相同属性列B上取值相等的元组，去掉重复的列。

$$R \bowtie S = \{ \overbrace{rs}^{\text{弧}}[\bar{B}] \mid r \in R \wedge s \in S \wedge r[B] = s[B] \}$$

- 自然连接与等值连接的不同

自然连接运算示例 I

$$r$$

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

$$s$$

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

$$r \bowtie s$$

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

自然连接运算示例 II

R

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

S

C	D
3	1
6	2

$R \bowtie S$

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



2.4 关系代数（续）

- 2.4.3 专门的关系运算
- 4. 除
 - 1) 特殊用途
 - 查询选修了全部课程的学生姓名
 - 查询选修了所有开课院系为管理学院的课程的学生
 - All, Every
 - 和整数的除运算的比较

姓名	课程
张蕊	物理
王红	数学
张蕊	数学

课程
数学
物理

张蕊同学所
选修的全部
课程



2.4 关系代数（续）

- 2.4.3 专门的关系运算

- 4. 除

- 2) 象集(Image Set)

- 关系 $R(X, Z)$, X, Z 是属性组, x 是 X 上的取值, 定义 x 在 R 中的象集为

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

- 从 R 中选出在 X 上取值为 x 的元组, 去掉 X 上的分量, 只留 Z 上的分量



2.4 关系代数（续）

- 2.4.3 专门的关系运算

- 4. 除

R	X		Z	
	姓名	课程	Z	
	张军	物理	课程	
	王红	数学	数学	
S	张军	数学	物理	



2.4 关系代数（续）

如何得到选修了全部课程的学生？

做法：逐个考虑**选课关系** SC 中的元组 r ，求 r 在**姓名** SN 上的分量 x ，再求 x 在**选课关系**中的象集**课程** C_x ，若 C_x 包含了所有的课程 C ，则 x 是满足条件的一个元组

选修全部课程的学生

x 同学所选修的全部课程

全部课程

$$\{ x \mid x=r[SN] \wedge r \in SC \wedge C_x \supseteq C \}$$

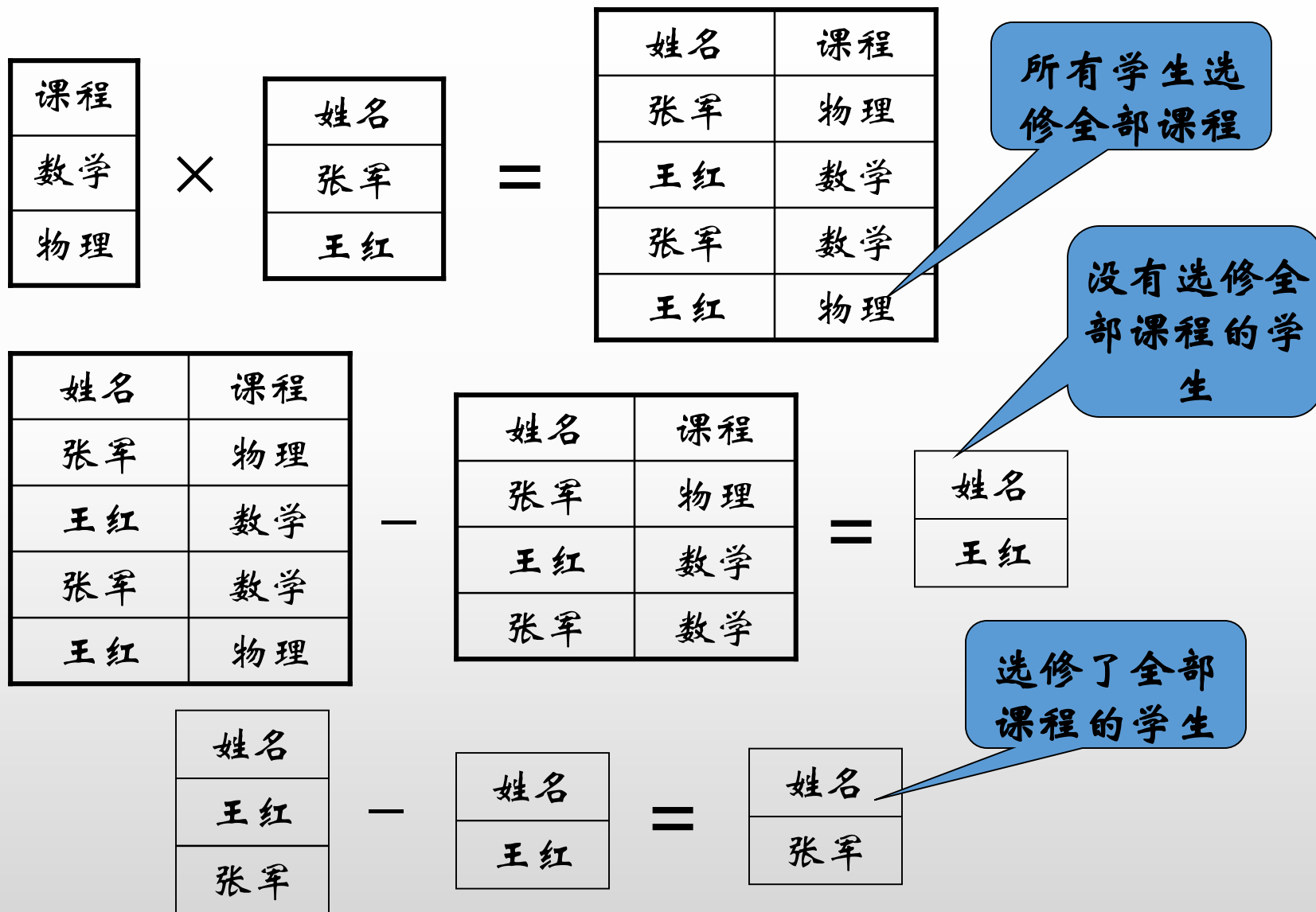


2.4 关系代数（续）

- 2.4.3 专门的关系运算
- 4. 除
 - 3) 定义
 - $R(X, Y) \div S(Y) = \{x \mid x = r[x] \wedge r \in R \wedge Y_x \supseteq \Pi_Y(S)\}$
 - $R \div S = \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R)$

设关系R和S的元数分别为r和s（设 $r > s > 0$ ），那么 $R \div S$ 是一个 $(r-s)$ 元的元组集合。 $(R \div S)$ 是满足下列条件的最大关系：其中每个元组t与S中每个元组u组成的新元组 $\langle t, u \rangle$ 必在关系R中。为方便起见，我们假设S的属性为R中后s个属性。

除运算示例 I



除运算示例 II-1

R

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

S

C	D
c	d
e	f

$\Pi_{AB}(R)$

A	B
a	b
b	c
e	d

除运算示例 II-2

$\Pi_{AB}(R) \times \Pi_{CD}(S)$

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

$\Pi_{AB}(R) \times \Pi_{CD}(S) - R$

A	B	C	D
b	c	c	d

$$R \div S = \begin{array}{|c|c|} \hline A & B \\ \hline a & b \\ \hline b & c \\ \hline e & d \\ \hline \end{array} - \begin{array}{|c|c|} \hline A & B \\ \hline b & c \\ \hline \end{array}$$

$$= \begin{array}{|c|c|} \hline A & B \\ \hline a & b \\ \hline e & d \\ \hline \end{array}$$



R

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

S

A	D	E
3	6	4
1	2	3
2	4	1
7	2	2

T

B	C	F
7	3	4
8	9	5

$$\sigma_{R.B > R.C} (R \times S)_{R.C < S.E}$$

$$\Pi_{(R.A, R.B, S.E)} (R \times S)_{R.C = S.D}$$



并运算示例 II

- SC(C#,S#,Score)

- 求选修了001号或002号课程的学生号

方案1:

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

方案2:

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



交运算示例 II

- 示例

求同时选修了001号和002号课程的学生号

? ?

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

? ?

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



差运算示例 II

- 示例

求选修了001号而没有选002号课程的学生号

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



查询示例

Sailor(Sid,Sname,Rating,Age)

Boat(Bid, Bname, color)

Reserve(Sid, Bid, Day)

水手编号 Sid	水手姓名 Sname	等 级 Rating	年龄Age
22	Dustin	7	45
29	Brutus	1	33
32	Andy	8	25
64	Horatio	7	35
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63



水手编号 Sid	水手姓名 Sname	等级 Rating	年龄Age
22	Dustin	7	45
29	Brutus	1	33
32	Andy	8	25
64	Horatio	7	35
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63



Reserve(Sid, Bid, Day)

Sid	Bid	Day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
32	102	11/10/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Boat(Bid,Bname,Color)

Bid	Bname	Color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	red



查询

Sailor(Sid,Sname,Rating,Age)

Boat(Bid, Bname, color)

Reserve(Sid, Bid, Day)

- (1) 查询预定了编号为103号船的水手
- (2) 查询预定了红颜色船只的水手
- (3) 查询被名字为Lubber预定了的船只
- (4) 查询至少预定了一只船的水手姓名
- (5) 查询预定了红色或绿色船只的水手的姓名
- (6) 查询预定了红色和绿色船只的水手的姓名
- (7) 查询至少预定了两只船的水手的姓名
- (8) 查询年龄超过20但是没有预定船只的水手的姓名
- (9) 查询预定了所有船只的水手的姓名
- (10) 查询预定了所有名为“Interlake”的船只的水手的姓名



2.5 关系演算

- 2.5.1 概述
- 2.5.2 元组关系演算
 1. 定义
 2. 表达式的安全性
 3. 元组关系演算与关系代数的等价性
- 2.5.3 域关系演算



2.5 关系演算

• 2.5.2 元组关系演算

- 1. 定义
 - 用元组作为谓词变量的一种谓词演算方法

$$\{ t / P(t) \}$$

表示所有使谓词 P 为真的元组集合

- t 为元组变量
 - 如果元组变量前有“全称” (\forall) 或“存在” (\exists) 量词, 则称其为约束变量, 否则称为自由变量
- P 是公式
 - 由原子公式和运算符组成



2.5 关系演算

• 2.5.2 元组关系演算

- 原子公式

- $R(t)$

- t 是关系 R 中的一个元组

- $t[x] \theta u[y]$

- $t[x]$ 与 $u[y]$ 为元组分量，他们之间满足比较关系 θ

- $t[x] \theta c$

- 分量 $t[x]$ 与常量 c 之间满足比较关系 θ



2.5 关系演算

• 2.5.2 元组关系演算

- 公式的递归定义
 - 原子公式是公式
 - 如果 P 是公式, 那么 $\neg P$ 也是公式
 - 如果 P_1, P_2 是公式, 则 $P_1 \wedge P_2, P_1 \vee P_2$ 也是公式
 - 如果 $P(t)$ 是公式, R 是关系, 则 $\exists t \in R (P(t))$ 和 $\forall t \in R (P(t))$ 也是公式



2.5 关系演算

• 2.5.2 元组关系演算

- 运算符及其优先顺序
 - 算术比较符最高
 - 存在量词和全称量词次之
 - 逻辑运算符最低
 - 括号中的运算优先级最高



元组演算表达式举例：

$\{ t \mid S(t) \wedge t[A] > 2 \}$ S 中 A 属性大于2的元组的集合

$\{ t \mid R(t) \wedge \neg S(t) \}$ 在 R 中不在 S 中出现的元组的集合

$\{ t \mid (\exists u) (S(t) \wedge R(u) \wedge t[C] < u[B]) \}$

S 中满足下述条件的元组的集合： C 属性小于 R 中某一个元组的 B 属性的值。

$\{ t \mid (\forall u) (R(t) \wedge S(u) \wedge t[C] > u[A]) \}$

R 中满足下述条件的元组的集合： C 属性大于 S 中每个元组的 A 属性的值。



关系演算示例 I

R

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

$$\{ t \mid t \in S \wedge t[A] > 2 \}$$

N₁

A	B	C
3	4	6
5	6	9

S

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

$$\{ t \mid t \in R \wedge \neg t \in S \}$$

N₂

A	B	C
4	5	6
7	8	9



关系演算示例 II

R

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

S

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

N_3

A	B	C
1	2	3
3	4	6

$$\{ t \mid t \in S \wedge \exists u \in R (t[C] < u[B]) \}$$



关系演算示例 III

R

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

S

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

N_4

A	B	D
4	5	6
7	8	9

$$\{ t \mid t \in R \wedge \forall u \in S (t[D] > u[A]) \}$$



关系演算示例 IV

R

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

S

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

$$\{ t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[A] > v[B] \\ \wedge t[A] = u[B] \wedge t[B] = v[C] \wedge t[C] = u[A]) \}$$

R.B	S.C	R.A
5	3	4
8	3	7
8	6	7
8	9	7



关系演算示例 V

PROF(P#, PNAME, SAL,DNO)

- 找出工资在800元以上的老师

$$\{t \mid t \in \text{PROF} \wedge t[\text{SAL}] > 800\}$$

Get W (Prof):Prof.Sal>800

- 找出工资在800元以上的老师的姓名

$$\{t \mid \exists s \in \text{PROF} (t[\text{PNAME}] = s[\text{PNAME}] \wedge s[\text{SAL}] > 800)\}$$

- 给出计算机系老师的姓名

$$\{t \mid \exists u \in \text{DEPT} (u[\text{DNAME}] = \text{“计算机系”} \wedge \exists s \in \text{PROF} (s[\text{DNO}] = u[\text{DNO}] \wedge t[\text{PNAME}] = s[\text{PNAME}]))\}$$



关系演算示例 VI

求选修了全部课程的学生号

SC(CNO, SNO, SCORE)

C(CNO, CNAME)

$$\{t \mid \forall u \in C (\exists s \in SC (s[CNO] = u[CNO] \wedge t[SNO] = s[SNO])) \}$$



2.5 关系演算

• 2.5.2 元组关系演算

• 2. 表达式的安全性

• 问题

• 域

- 引入公式 P 的域概念，用 $dom(P)$ 表示

$dom(P)$ = 显式出现在 P 中的值 + 在 P 中出现的关系的元组中出现的值
(不必是最小集)

如 $dom(t \mid \neg (t \in R))$ 是 R 中出现的所有值的集合

- 如果出现在表达式 $\{t \mid P(t)\}$ 结果中的所有值均来自 $dom(P)$ ，则称 $\{t \mid P(t)\}$ 是安全的



R

A	B
A1	B1
A1	B2
A2	B3

$\{ t \mid \neg (t \in R) \}$

A	B
A1	B3
A2	B1
A2	B2

$$\text{dom}(\neg R(t)) = \{\{A1, A2\}, \{B1, B2, B3\}\}$$



2.5 关系演算

- 2.5.2 元组关系演算

- 3. 元组关系演算与关系代数的等价性

- 并: $R \cup S \equiv \{t \mid R(t) \vee S(t)\}$

- 差: $R - S = \{t \mid t \in R \wedge \neg t \in S\}$



2.5 关系演算

• 2.5.2 元组关系演算

• 3. 元组关系演算与关系代数的等价性

- 笛卡儿积: $R \times S = \{t^{(r+s)} \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge t[1]=u[1] \wedge t[2]=u[2] \wedge \cdots \wedge t[r+1]=v[1] \wedge \cdots \wedge t[r+s]=v[s])\}$
- 投影: $\pi_{i_1, \dots, i_m}(R) = \{t^{(m)} \mid \exists (u) R(u) \wedge t[1]=u[i_1] \wedge t[2]=u[i_2] \wedge \cdots \wedge t[m]=u[i_m]\}$
- 选择: $\sigma_{F(A)}(R) = \{t \mid t \in R \wedge F(t[A])\}$



2.5 关系演算

• 2.5.3 域关系演算

• 形式化定义

$$\{ x_1 x_2 \cdots x_n / P (x_1, x_2, \cdots, x_n) \}$$

x_i 代表域变量, P 为由原子构成的公式

• 原子公式

- $(x_1, x_2, \cdots, x_n) \in R, \text{记} R(x_1, x_2, \cdots, x_n)$

- x_i 是域变量或域常量

- $x \theta y$

- 域变量 x 与 y 之间满足比较关系 θ

- $x \theta c$

- 域变量 x 与常量 c 之间满足比较关系 θ



R

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

S

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

W

D	E
7	5
4	8

$$R1 = \{ x \ y \ z \mid R(x, y, z) \wedge x < 5 \wedge y > 3 \}$$

A	B	C
4	5	6



R

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

W

D	E
7	5
4	8

$$R2 = \{ x \ y \ z \mid (\exists u) (\exists v) (R(z, x, u) \wedge W(y, v) \wedge u > v) \}$$

B	D	A
5	7	4
8	7	7
8	4	7

R

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

S

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

W

D	E
7	5
4	8

$$R2 = \{xyz \mid R(x, y, z) \vee (S(x, y, z) \wedge y=4)\}$$

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



A1	A2	A3
d	ce	5
d	bd	3
g	ef	7
d	cd	9

A1	A2	A3
d	ce	5
d	bd	3

$$R_1 = \{XYZ \mid R(XYZ) \wedge Z < 8 \wedge X = d\}$$

等价于关系代数表达式:

$$R_1 = \sigma_{A1='d' \wedge A3 < 8}(R)$$

等价于元组关系演算为:

$$R_1 = \{t \mid R(t) \wedge t[1] = d \wedge t[3] < 8\}$$



小结

- 关系模型
- 关系数据结构及定义
- 完整性约束
- 关系代数:5个基本运算
- 关系演算:元组、域