

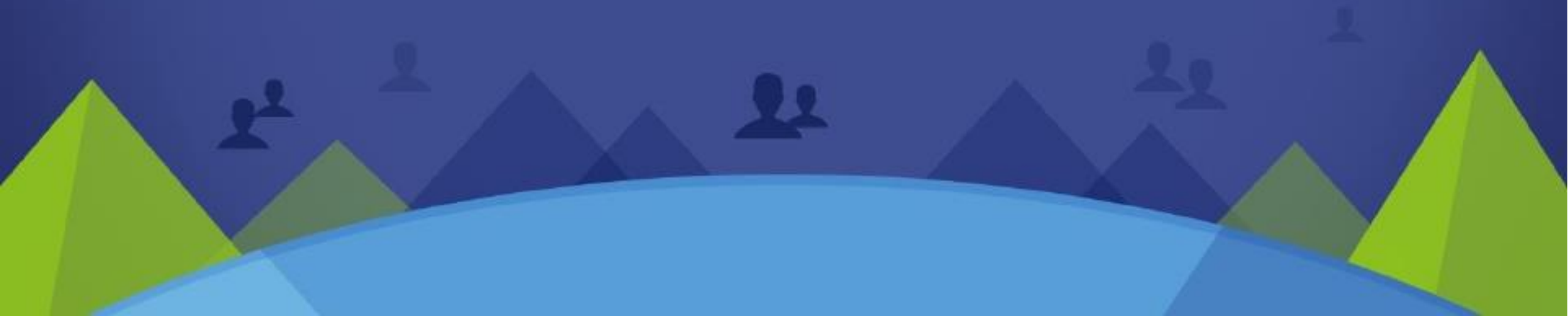
数据库原理及应用教程 (第4版)

“十二五”普通高等教育本科国家级规划教材
微课版，对重点和难点进行视频详解

中国工信出版集团

人民邮电出版社

第2章 关系数据库



2.1 关系模型的数据结构及其形式化定义

2.2 关系的码与关系的完整性

2.3 关系代数*

2.4 关系演算

2.5 小 结

本章主要按数据模型的三个要素讲述关系数据库的一些基本理论
(关系模型的数据结构、关系的定义和性质、关系的完整性、关系代数、
关系数据库的基本概念等)

掌握关系的定义及性质、关系键、外部键等基本概念
了解关系演算语言

重点掌握实体完整性和参照完整性的内容和意义、
常用的几种关系代数的基本运算等

2.1 关系模型的数据结构及其形式化定义

2.1.1 关系的形式化定义及其有关概念

域 (Domain)

整数、实数和字符串的集合都是域

域是一组具有相同数据类型的值的集合，又称为值域

域中所包含的值的个数称为域的基数（用 m 表示）。例如：

域名

例 $D1 = \{\text{李力, 王平, 刘伟}\}, m1 = 3;$

例 $D2 = \{\text{男, 女}\}; m2 = 2;$

例 $D3 = \{18, 19, 20\}; m3 = 3。$

笛卡尔积 (Cartesian Product)

给定一组域 D_1, D_2, \dots, D_n (它们可以包含相同的元素, 即可以完全不同, 也可以部分或全部相同)。 D_1, D_2, \dots, D_n 的笛卡尔积为

$$D_1 \times D_2 \times \dots \times D_n = \{ (d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n \}$$

每一个元素 (d_1, d_2, \dots, d_n) 中的每一个值 d_i 叫做一个分量 (Component), $d_i \in D_i$
每一个元素 (d_1, d_2, \dots, d_n) 叫做一个 n 元组 (n-Tuple), 简称元组 (Tuple)

笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M (即元素 (d_1, d_2, \dots, d_n) 的个数) 为所有域的基数的累乘之积, 即 $M = \prod_{i=1}^n m_i$ 。

例如, 上述表示教师关系中姓名、性别两个域的笛卡尔积为:

$D_1 \times D_2 = \{ (李力, 男), (李力, 女), (王平, 男), (王平, 女), (刘伟, 男), (刘伟, 女) \}$



分量: 李力、王平、刘伟、男、女

元组: $(李力, 男), (李力, 女)$, $M = m_1 \times m_2 = 3 \times 2 = 6$

2.1 关系模型的数据结构及其形式化定义

笛卡尔积可用二维表的形式表示：

姓 名	性 别
李力	男
李力	女
王平	男
王平	女
刘伟	男
刘伟	女

元组

同一域

笛卡尔积可以表示成为一个二维表

2.1 关系模型的数据结构及其形式化定义

关系 (Relation)

笛卡尔积 $D_1 \times D_2 \times \cdots \times D_n$ 的任一子集称为定义在域 D_1, D_2, \cdots, D_n 上的 n 元关系 (Relation)

关系的名字

$R(D_1, D_2, \dots, D_n)$

n 是关系的目或度

$D_1 \times D_2$ 笛卡尔积的子集 (关系 T_1)

如, 上例 $D_1 \times D_2$ 笛卡尔积的某个子集可以构成教师关系 T_1 , 如表2.2所示

姓 名	性 别
李力	男
王平	女
刘伟	男

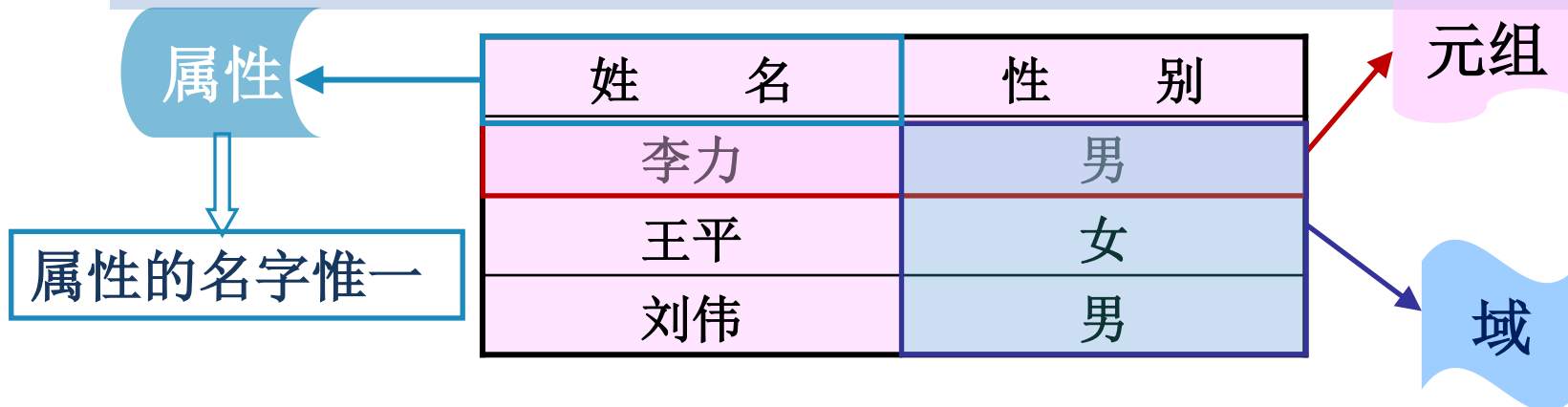
2.1 关系模型的数据结构及其形式化定义

在关系R中，当 $n=1$ 时，称为单元关系。
当 $n=2$ 时，称为二元关系，以此类推。

关系中的每个元素是关系中的元组，通常用t表示，
关系中元组个数是关系的基数

由于关系是笛卡尔积的子集，因此，
也可以把关系看成一个二维表。

具有相同关系框架的关系称为同类关系。



2.1 关系模型的数据结构及其形式化定义

姓 名	性 别
李力	男
李力	女



不符合实际意义的关系

在关系模型中，关系可进一步定义为：

不变

可变

关系头 (Heading) + 关系体 (Body)

由属性名的
集合组成

关系结构中的
内容或者数据

2.1 关系模型的数据结构及其形式化定义

2.1.2 关系的性质

关系是一种规范化了的二维表中行的集合

- 每一列中的分量必须来自同一个域，必须是同一类型的数据。
- 不同的属性可来自同一个域，不同的属性必须有不同的名字。
- 列的顺序可以任意交换。
- 关系中元组的顺序（即行序）可任意。
- 关系中不允许出现相同的元组。
- 关系中每一分量必须是不可分的数据项。

2.1 关系模型的数据结构及其形式化定义

在表2.8中，籍贯含有省、市 / 县两项，出现了“表中有表”的现象，则为非规范化关系，而应把籍贯分成省、市 / 县两列，将其规范化，如表2.9所示

姓名	籍	贯
	省	市 / 县
张强	吉林	长春
王丽	山西	大同

非规范化的关系 表2.8

姓	名	省	市 / 县
张强		吉林	长春
王丽		山西	大同

规范化的关系 表2.9

2.1 关系模型的数据结构及其形式化定义

2.1.3 关系模式

- 关系的描述称为**关系模式** (Relation Schema)

$$R (U, D, DOM, F)$$

R--关系名

U--属性名集合

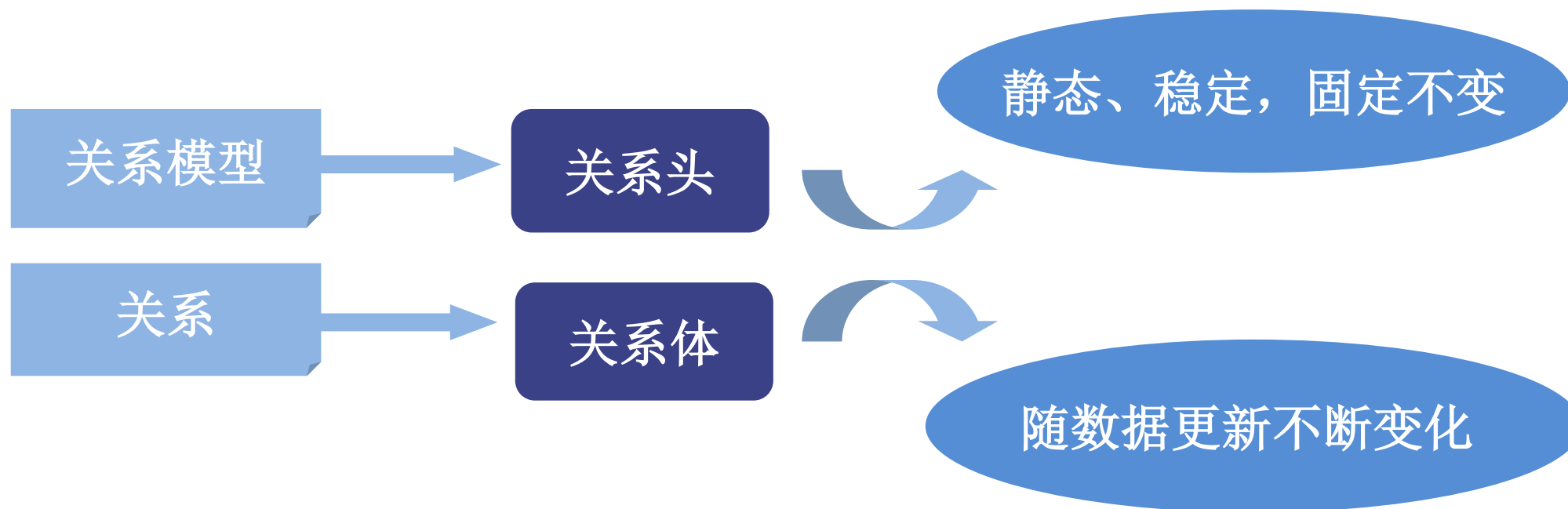
D--属性所来自的域

DOM--属性向域的映像集合

F--属性间数据的依赖关系集合

属性名

简记为: $R (U)$ 或 $R (A_1, A_2, \dots, A_n)$



2.1 关系模型的数据结构及其形式化定义

- 例如，在第1章的图1-22所示的教学数据库中，共有五个关系，其关系模式可分别表示为：

- 学生（学号，姓名，性别，年龄，系别）
- 教师（教师号，姓名，性别，年龄，职称，工资，岗位津贴，系别）
- 课程（课程号，课程名，课时）
- 选课（学号，课程号，成绩）
- 授课（教师号，课程号）

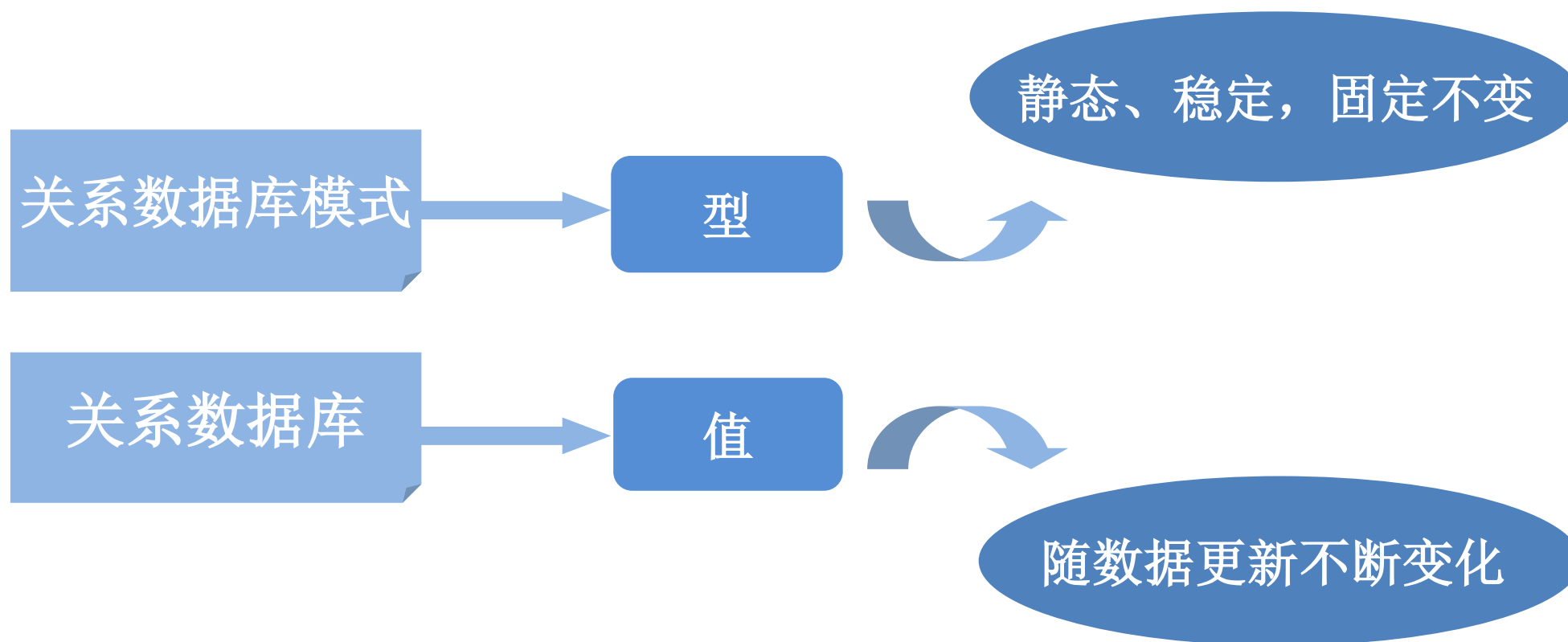
2.1 关系模型的数据结构及其形式化定义

- 与学生关系模式对应的数据库中的实例有如下6个元组，如图2-1所示。

S1	赵亦	女	17	计算机
S2	钱尔	男	18	信息
S3	孙珊	女	20	信息
S4	李思	男	21	自动化
S5	周武	男	19	计算机
S6	吴丽	女	20	自动化

图2-1 与学生关系模式对应的实例

2.1.4 关系数据库与关系数据库模式



2.2.1 候选码（键）与主码（键）

候选码 (Candidate Key)

候选码

候选码

“学生关系”中的**学号**能唯一标识每一个学生

“选课关系”中，只有属性的组合 **“学号+课程号”** 才能唯一地区分
每一条选课记录

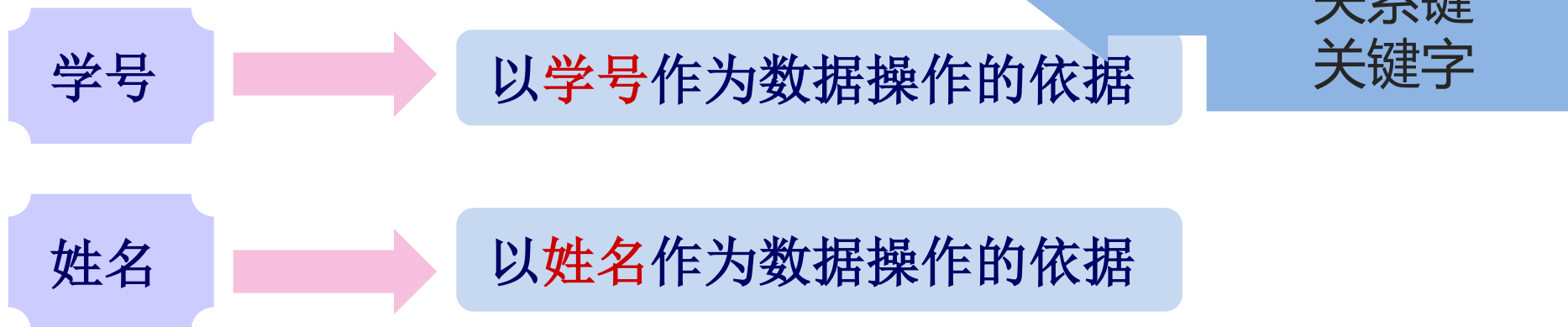
能**唯一标识**关系中元组的一个**属性或属性集**，称为**候选码(Candidate Key)**

唯一性

最小性

主码 (Primary Key)

- 从多个候选码中选择一个作为查询、插入或删除元组的操作变量，被选用的候选码称为 **主码**。



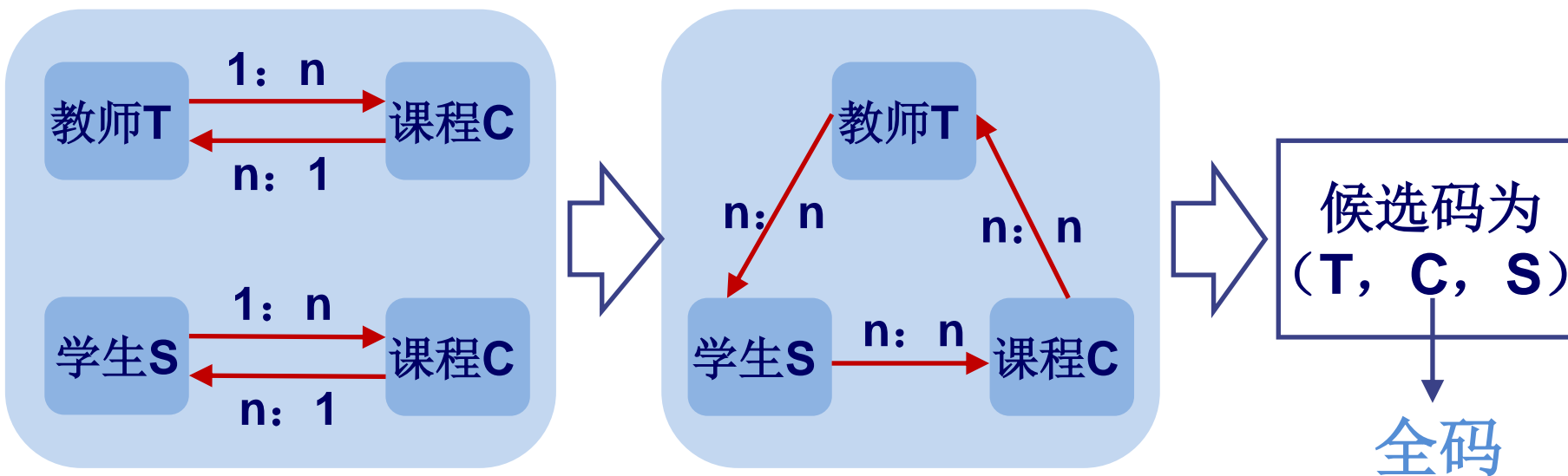
每个关系必须选择一个主码，且不能随意改变

主属性 (Prime Attribute) 与非码属性 (Non-Prime Attribute)

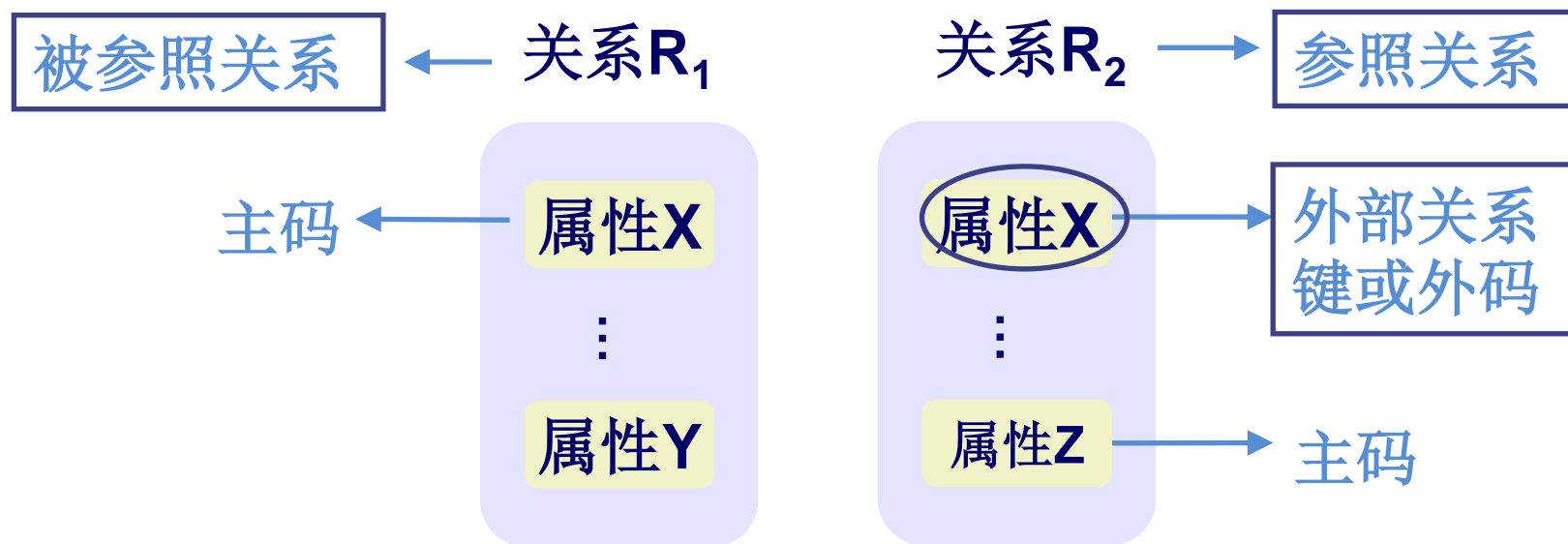
主属性： 包含在主码中的各个属性称为主属性

非码属性： 不包含在任何候选码中的属性称为非主属性（或非码属性）

全码： 所有属性的组合是关系的候选码

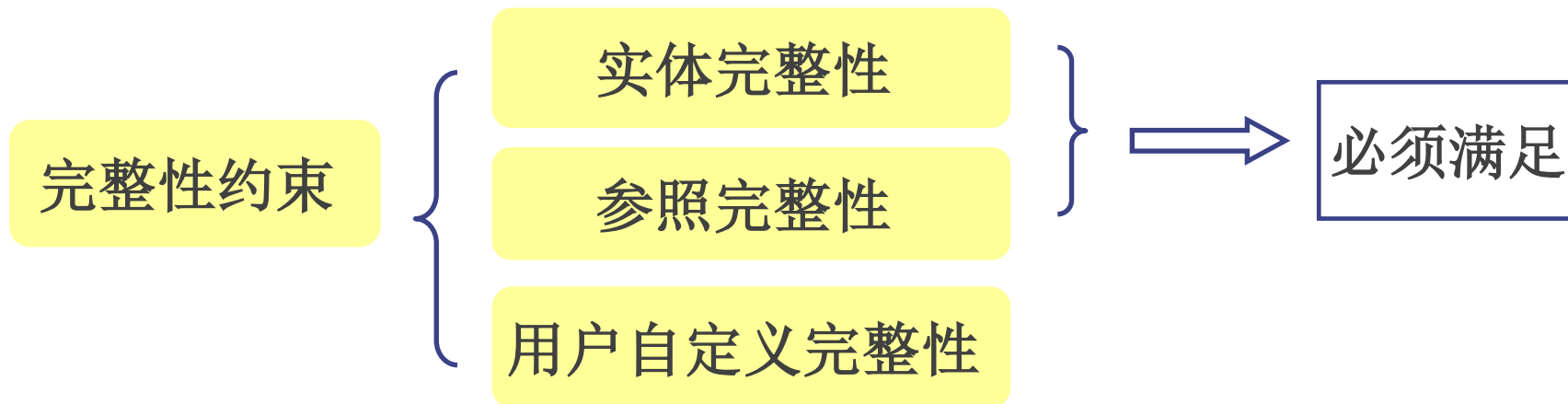


2.2.2 外码



被参照关系的主码和参照关系的外码必须定义在同一个域上

2.2.3 关系的完整性



体现具体领域中的语义约束

实体完整性 (Entity Integrity)

主码的值不能为空或部分为空

学生关系中的主码“学号”不能为空
选课关系中的主码“学号+课程号”不能部分为空，
即“学号”和“课程号”两个属性都不能为空

参照完整性 (Referential integrity)

如果关系**R2**的外码**X**与关系**R1**的主码相符，则**X**的每个值或者等于**R1**中主码的某一个值，或者取空值

S (学生关系)

SNo 学号	SN 姓名	Sex 性别	Age 年龄	Dept 系别
S1	赵亦	女	17	计算机
S2	钱尔	男	18	信息
...				
S11	王威	男	19	

D (系别关系)

Dept 系别	Addr 地址
计算机	1 号楼
信息	1 号楼
自动化	2 号楼

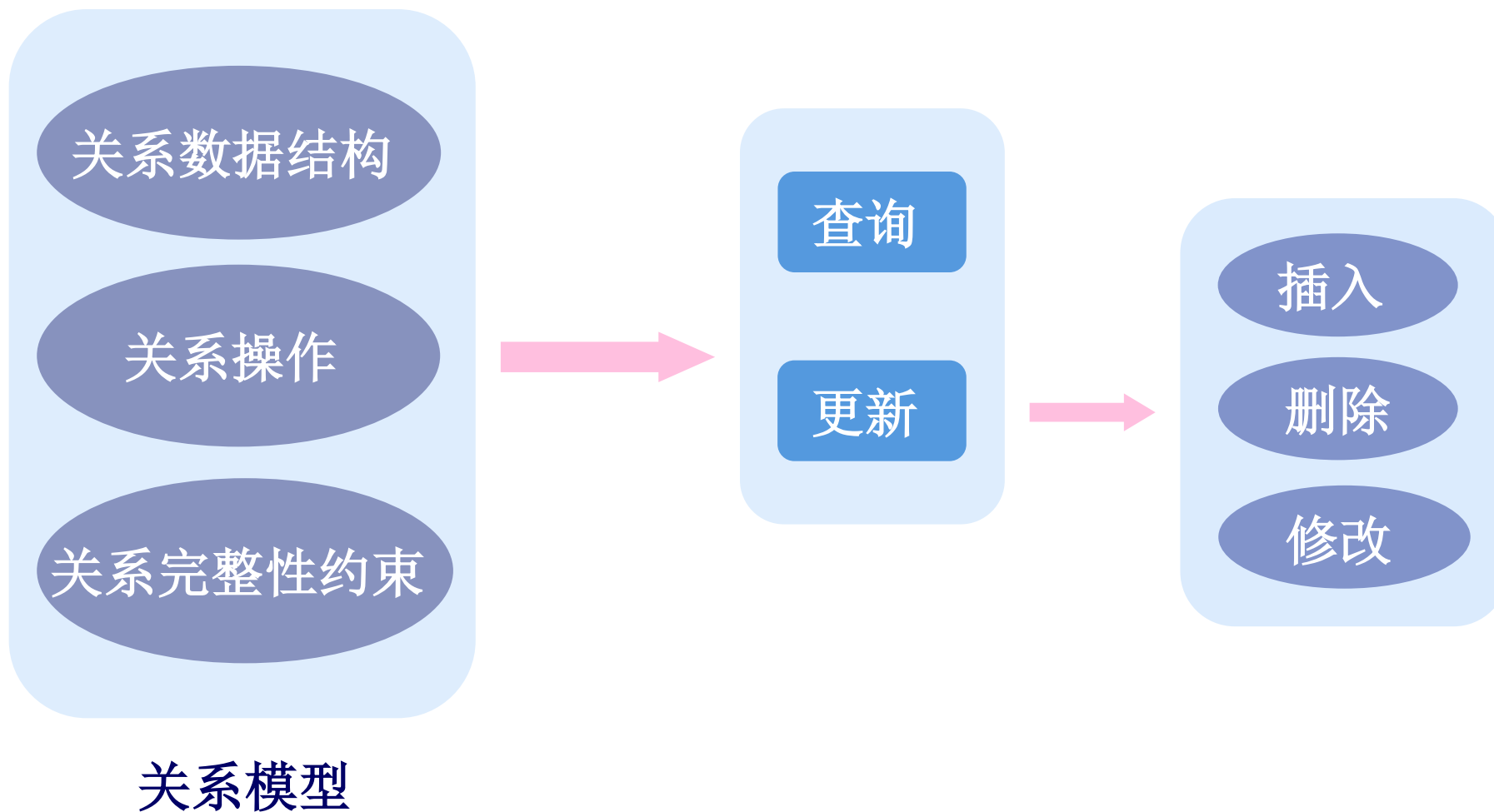
未分配系别

用户定义完整性 (User-defined Integrity)

针对某一具体关系数据库的约束条件

反映某一具体应用所涉及的数据必须满足的语义要求

如：成绩属性的取值范围在**0-100**之间



2.3.1 关系代数的分类及其运算符

- 关系代数是一种抽象的查询语言
- 关系代数的运算对象与运算结果都是关系
- 关系代数运算符

集合运算符

\cup
 \cap
 \times

关系运算符

σ
 π
 \bowtie
 $\bowtie x \theta y$
 \div

比较运算符

$>$, \geq
 $<$, \leq
 $=$, \neq

逻辑运算符

\wedge
 \vee
 \neg



关系代数的运算按**运算符**的不同主要分为两类：

- 传统的集合运算：

- 把关系看成元组的集合，以元组作为集合中元素来进行运算，其运算是从关系的“水平”方向即行的角度进行的。
- 包括并、差、交和笛卡尔积等运算。

- 专门的关系运算：

- 不仅涉及行运算，也涉及列运算，这种运算是为数据库的应用而引进的特殊运算。
- 包括选取、投影、连接和除法等运算。

2.3.2 传统的集合运算

设给定两个关系 R 、 S ，若满足：

- (1) 具有相同的列数（或称度数） n ;
- (2) R 中第 i 个属性和 S 中第 i 个属性必须来自同一个域。

则说关系 R 、 S 是**相容的**。

- 除笛卡尔积外，其他的集合运算要求参加运算的关系必须满足上述的相容性定义。

并 (Union) : $R \cup S = \{t \mid t \in R \vee t \in S\}$

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

交 (Intersection) : $R \cap S = \{t \mid t \in R \wedge t \in S\}$

广义笛卡尔积 (Extended Cartesian Product) :

$R \times S = \{tr \cap ts \mid tr \in R \wedge ts \in S\}$

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

【例2-4】 如图2-3(a)、(b)所示的两个关系 R 与 S 为相容关系，(c)为 R 与 S 的并，(d)为 R 与 S 的差，(e)为 R 与 S 的交，(f)为 R 与 S 的广义笛卡尔积。

R				S		
A	B	C		A	B	C
a1	b1	c1	—	a1	b1	c1
a1	b1	c2		a2	b2	c1
a2	b2	c1	↘	a2	b3	c2

(a) (b)

$R \cup S$

A	B	C
a1	b1	c1
a1	b1	c2
a2	b2	c1
a2	b3	c2

(c)

 $R \cap S$

A	B	C
a1	b1	c1
a2	b2	c1

(e)

 $R - S$

A	B	C
a1	b1	c2

(d)

$R \times S$

A	B	C	A	B	C
a1	b1	c1	a1	b1	c1
a1	b1	c1	a2	b2	c1
a1	b1	c1	a2	b3	c2
a1	b1	c2	a1	b1	c1
a1	b1	c2	a2	b2	c1
a1	b1	c2	a2	b3	c2
a2	b2	c1	a1	b1	c1
a2	b2	c1	a2	b2	c1
a2	b2	c1	a2	b3	c2

(f)

图2-3 传统的集合运算

2.3.3 专门的关系运算

- 由于传统的集合运算，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。
- 在讲专门的关系运算之前，为叙述上的方便先引入几个概念。
 - （1）设关系模式为 $R(A_1, A_2, \dots, A_n)$ ，它的一个关系为 R ， $t \in R$ 表示 t 是 R 的一个元组， $t[A_i]$ 则表示元组 t 中相应于属性 A_i 的一个分量。

- (2) 若 $A = \{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$, 其中 $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ 是 A_1, A_2, \dots, A_n 中的一部分, 则 A 称为属性列或域列, \tilde{A} 则表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 $\{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$ 后剩余的属性组。
 $t[A] = \{t[A_{i_1}], t[A_{i_2}], \dots, t[A_{i_k}]\}$ 表示元组 t 在属性列 A 上诸分量的集合。
- (3) R 为 n 元关系, S 为 m 元关系, $t_r \in R, t_s \in S, t_r \cap t_s$ 称为元组的连接(Concatenation), 它是一个 $n+m$ 列的元组, 前 n 个分量为 R 的一个 n 元组, 后 m 个分量为 S 中的一个 m 元组。
- (4) 给定一个关系 $R(X, Z)$, X 和 Z 为属性组, 定义当 $t[X] = x$ 时, x 在 R 中的像集(Image set), 为 $Z_x = \{t[Z] | t \in R, t[X] = x\}$, 它表示 R 中的属性组 X 上值为 x 的各元组在 Z 上分量的集合。

R

x1	Z1
x1	Z2
x1	Z3
x2	Z2
x2	Z3
x3	Z1
x3	Z3

- x1 在R中的像集
 - $Z_{x1} = \{Z1, Z2, Z3\}$
- x2 在R中的像集
 - $Z_{x2} = \{Z2, Z3\}$
- x3 在R中的像集
 - $Z_{x3} = \{Z1, Z3\}$

选取 (Selection)

 F 为选取的条件

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

[例2-5] 查询计算机系的全体学生。

$\sigma_{\text{Dept}=\text{'计算机'}}(S)$ 或 $\sigma_5=\text{'计算机'}}(S)$
(其中5为属性Dept的序号) 运算结果如图

SNo	SN	Sex	Age	Dept
S1	赵亦	女	17	计算机
S5	周武	男	19	计算机

从行的角度进行的运算


[例2-6] 查询工资高于1000元的男教师。

$\sigma_{(\text{Sal} > 1000) \wedge (\text{Sex} = \text{'男'})}(\text{T})$


运算结果如图

TNo	TN	Sex	Age	Prof	Sal	Comm	Dept
T1	李力	男	47	教授	1500	3000	计算机

投影 (Projection)



A 为 R 中的属性列


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

[例2-7] 查询教师的姓名、教师号及其职称。

$\Pi_{TN, TNo, Prof}(T)$ 或

$\Pi_{2,1,5}(T)$ (其中2, 1, 5分别为属性TN、TNo和Prof的序号)

运算结果如图

从列的角度
进行的运算

TN	TNo	Prof
李力	T1	教授
王平	T2	讲师
刘伟	T3	讲师
张雪	T4	教授
张兰	T5	副教授

- **[例2-8]** 查询教师关系中有哪些系。

$\Pi_{\text{Dept}}(T)$

运算结果如图

Dept
计算机
信息
自动化

- **[例2-9]** 查询讲授C5课程的教师号。

$\Pi_{TNo}(\sigma_{CNo='C5'}(TC))$

运算结果如图

TNo
T2
T3
T5

θ 连接 (θ Join)

$$R \bowtie_{x \theta y} S = \{t_r \cap t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[X] \theta t_s[Y] \text{ 为真} \}$$

θ 为算术比较运算符

=	等值连接
<	小于连接
>	大于连接

自然连接：在**等值连接**的情况下，当连接属性X与Y具有相同属性组时，把在连接结果中重复的属性列去掉，记为： **$R \bowtie S$**

[例2-10] 设有如图2-9 (a)、(b) 所示的两个关系 R 与 S , (c) 为 R 和 S 的大于连接 ($C \succ D$), (d) 为 R 和 S 的等值连接 ($C = D$), (e) 为 R 和 S 的等值连接 ($R.B = S.B$), (f) 为 R 和 S 的自然连接。

R		
A	B	C
a1	b1	2
a1	b2	4
a2	b3	6
a2	b4	8

(a)

S	
B	D
b1	5
b2	6
b3	7
b3	8

(b)

大于连接 ($C > D$)

(c)

<i>A</i>	<i>R.B</i>	<i>C</i>	<i>S.B</i>	<i>D</i>
a2	b3	6	b1	5
a2	b4	8	b1	5
a2	b4	8	b2	6
a2	b4	8	b3	7

等值连接 ($C = D$)

(d)

<i>A</i>	<i>R.B</i>	<i>C</i>	<i>S.B</i>	<i>D</i>
a2	b3	6	b2	6
a2	b4	8	b3	8

等值连接 ($R.B = S.B$)

(e)

<i>A</i>	<i>R.B</i>	<i>C</i>	<i>S.B</i>	<i>D</i>
a1	b1	2	b1	5
a1	b2	4	b2	6
a2	b3	6	b3	7
a2	b3	6	b3	8

自然连接

(f)

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
a1	b1	2	5
a1	b2	4	6
a2	b3	6	7
a2	b3	6	8

等值连接与自然连接的区别

自然连接要求相等属性值的属性名相同，而等值连接不要求
自然连接是去掉重复列的等值连接

[例2-11] 查询讲授“数据库”课程的教师姓名。

$\Pi_{TN}(\sigma_{CN='数据库'}(C) \bowtie TC \bowtie \Pi_{TNo, TN}(T))$ 或

$\Pi_{TN}(\Pi_{TNo}(\sigma_{CN='数据库'}(C) \bowtie TC) \bowtie \Pi_{TNo, TN}(T))$

运算结果如图

TN
王平
刘伟
张兰

除法(Division)

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

Y_x 为 x 在 R 中的像集, $x = t_r[X]$

- 除法运算同时从行和列的角度进行运算, 适合于包含“全部”之类的短语的查询。



【例2-12】 已知关系R和S，如图2-11 (a)， (b) 所示，则 $R \div S$ 如图 (c) 所示。

R

A	B	C	D
a1	b2	c3	d5
a1	b2	c4	d6
a2	b4	c1	d3
a3	b5	c2	d8

(a)

S

C	D	F
c3	d5	f3
c4	d6	f4

(b)

$R \div S$

A	B
a1	b2

(c)

- 与除法的定义相对应，本题中

$$X=\{A,B\}=\{(a1,b2),(a2,b4),(a3,b5)\}, \quad Y=\{C,D\}=\{(c3,d5),(c4,d6)\}$$

$$Z=\{F\}=\{f3,f4\}$$

其中，元组在X上各个分量值的像集分别为：

- $(a1,b2)$ 的象集为 $\{(c3,d5),(c4,d6)\}$
 - $(a2,b4)$ 的象集为 $\{(c1,d3)\}$
 - $(a3,b5)$ 的象集为 $\{(c2,d8)\}$
 - S在Y上的投影为 $\{(c3,d5),(c4,d6)\}$
- 显然只有 $(a1,b2)$ 的像集包含S在Y上的投影，所以 $R \div S = \{(a1,b2)\}$



– [例2-13] 查询选修了全部课程的学生学号和姓名。

$$\Pi_{SNo, CNo}(SC) \div \Pi_{CNo}(C) \bowtie \Pi_{SNo, SN}(S)$$

– [例2-14] 查询至少选修了C1课程和C3课程的学生学号。

$$\Pi_{SNo, CNo}(SC) \div \Pi_{CNo}(\sigma_{CNo='C1' \vee CNo='C3'}(C))$$

只有S4同学的像集至少包含了C1课程和C3课程，因此，查询结果为S4。

关系代数



过程化语言，怎样做

关系演算



非过程化语言，做什么

关系演算



元组关系演算

→ ALPHA,
QUEL

域关系演算

→ QBE

2.4.1 元组关系演算语言

ALPHA语言

QUEL语言

2.4.1 元组关系演算语言

ALPHA语言

- 基本格式:

<操作符> <工作空间名> (<目标表>)[:<操作条件>]

GET
PUT
HOLD
UPDATE
DELETE
DROP

内存空间 (W)

- 数据查询

- ① 简单查询
- ② 条件查询
- ③ 排序查询
- ④ 定额查询
- ⑤ 带元组变量的查询
- ⑥ 带存在量词的查询
- ⑦ 库函数查询

- 数据更新

- ① 修改
- ② 插入 → PUT
- ③ 删除 → DELETE

① 简单查询

[例2-16] 查询所有被选修的课程号码。GET W (SC. CNo)

② 条件查询

[例2-17] 查询计算机系工资高于1000元（不包括1000）的教师的姓名和工资。
GET W (T. TN, T. Sal) : T. Dept= '计算机' ^ T. Sal > 1000

③ 排序查询

[例2-18] 查询S3同学所选课程号及成绩，并按成绩降序排列。
GET W (SC. CNo, SC. Score) : SC. SNo= 'S3' DOWN SC. Score

④ 定额查询

[例2-20] 查询一名男教师的教师号和姓名，并使他的年龄最小。
GET W (1) (T. TNo, T. TN) : T. Sex= '男' UP T. Age

⑤ 带元组变量的查询

[例2-21] 查询S3同学所选课程号。

RANGE SC X

GET W (X. CNo) : X. SNo = 'S3'

使用RANGE来说明元组变量，
X为关系SC上的元组变量

⑥ 带存在量词的查询

[例2-23] 查询至少选修一门其课时数为80的课程的学生的姓名。

RANGE C CX

SC SCX

GET W (S. SN) : SCX (SCX. SNo = S. SNo \wedge CX (CX. CNo = SCX. CNo \wedge CX. CT = 80))

⑦ 库函数查询

ALPHA常用的库函数及其功能

函数名称	功能
AVG	按列计算平均值
TOTAL	按列计算值的总和
MAX	求一列中的最大值
MIN	求一列中的最小值
COUNT	按列值计算元组个数

[例2-25] 求学号为S1学生的平均分。

GET W (AVG(SC. Score):S. SNo= 'S1')

[例2-26] 求学校共有多少个系。

GET W (COUNT(S. Dept))

(2) 数据更新

① 修改

读数据

修改

[例2-27] 把刘伟教师转到信息系。
HOLD W(T. Dept):T. TN= '刘伟'
MOVE '信息' TO W. Dept
UPDATE W

送回



② 插入

建立新元组

[例2-28] 在SC表中插入一条选课记录 (S6, C1, 85) 。

- **MOVE 'S6' TO W. SNo**
- MOVE 'C1' TO W. CNo**
- MOVE 85 TO W. Score**
- PUT W(SC)**

写数据



③ 删除

读数据

- **[例2-29]** 删除学号为S6的学生的信息。

HOLD W(S) : S. SNo= 'S6'

DELETE W

删除

QUEL语言

数据更新

数据定义	CREATE <关系名> (<属性名=数据类型及长度>[,<属性名=数据类型及长度>…])
数据查询	RANGE OF t1 IS R1 …… RANGE OF tk IS Rk RETRIEVE (目标表) WHERE <条件>
修改	REPLACE (<属性名=数据类型及长度>[,<属性名=数据类型及长度>…])
插入	APPEND TO <关系名> (<属性名=数据类型及长度>[,<属性名=数据类型及长度>…])
删除	DELETE <关系名>

2.4.2 域关系演算语言QBE

QBE操作框架表

关系名	属性1	...	属性n
操作命令	属性值或查询条件	...	属性值或查询条件

↓

P. (打印或显示)
U. (修改)
I. (插入)
D. (删除)

注：红色字体为用户操作
蓝色字体为系统操作

