第3章 关系数据库

北京理工大学　计算机学院
张文耀
zhwenyao@bit.edu.cn

主要内容

- 关系模型的基本概念
- 关系代数
- 关系演算
  - 元组关系演算
  - 域关系演算
- 关系数据语言
- 关系运算的安全性
- 关系运算的等价性
- 小结

1.关系模型的基本概念

- 关系模型的数据结构
  - 关系
  - 现实世界的实体以及实体间的各种联系均用关系来表示
- 关系模型的逻辑结构
  - 二维表
  - 从用户角度，关系模型中数据的逻辑结构是一张二维表
- 关系模型的数学基础
  - 集合代数

关系的定义

- 笛卡尔积（**Cartesian Product**）

  给定一组集合 D_1，D_2，…，D_n（可以有相同的），其笛卡尔积定义为：

  $$D_1 \times D_2 \times \ldots \times D_n =$$

  $$\{(d_1, d_2, \ldots, d_n) \mid d_i \in D_i,\ i = 1, 2, \ldots, n\}$$

例：$D_1=\{1, 2, 3\}$，$D_2=\{a, b\}$，

$$D_1 \times D_2 = \{(1, a),\ (1, b),\ (2, a),\ (2, b),\ (3, a),\ (3, b)\}.$$

- 元组（**Tuple**）
  - 笛卡尔积中每一个元素（d_1，d_2，…，d_n）叫作一个n元组（n-tuple）或简称元组。
- 分量（**Component**）
  - 笛卡尔积中的元素（d_1，d_2，…，d_n）中的第d_i值叫作第i个分量。
- 域（**Domain**）
  - 第i个分量d_i的取值范围$\boldsymbol{D_i}$，称为这个分量的（值）域。
  - 域是一组具有相同数据类型的值的集合。
  - 整数，实数，介于某个取值范围的整数。

- 基数（Cardinal number）
  - 若D_i（i=1，2，…，n）为有限集，其基数为m_i（i=1，2，…，n），则$D_1 \times D_2 \times \ldots \times D_n$的基数$M$为：

$$M = \prod_{i=1}^{n} m_i$$

- 笛卡尔积的表示方法
  - 笛卡尔积可表示为一个二维表；表中的每行对应一个元组，表中的每列对应一个域。

- 笛卡尔积示例
  - **D1=导师集合SUPERVISOR**

    =｛张清玫，刘逸｝
  - **D2=专业集合SPECIALITY**

    =｛计算机专业，信息专业｝
  - **D3=研究生集合POSTGRADUATE**

    =｛李勇，刘晨，王敏｝

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
张清玫	计算机专业	王敏
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
张清玫	信息专业	王敏
刘逸	计算机专业	李勇
刘逸	计算机专业	刘晨
刘逸	计算机专业	王敏
刘逸	信息专业	李勇
刘逸	信息专业	刘晨
刘逸	信息专业	王敏

- 关系（**Relation**）
  - $D_1 \times D_2 \times \ldots \times D_n$的子集称为$D_1$，$D_2$，…，$D_n$上的关系，表示为：

    R（D_1，D_2，…，D_n）

    R：关系名

    n：关系的目或度（Degree）

    集合D_1，D_2，…，D_n称为关系的域
  - 一元关系：$n=1$
  - 二元关系：$n=2$
  - n元关系

- 关系的表示
  - 关系也是一个二维表；
- 关系的元组和属性
  - 关系中的每一行对应一个元组，通常用 t 表示；
  - 关系中的每一列对应一个域；
  - 关系中的列称为属性，每一列用属性名表示；
  - $t[A_i]$ 表示元组 t 在属性 A_i 上的值；
  - 用符号 $\mathbf{dom}(A_i)$ 表示属性 A_i 的域。

- 关系的特殊性
  - 并非任何笛卡尔积的子集都有现实意义；
  - 关系元组的各个分量的次序是无关紧要的；
    （为关系的每个列附加一个属性名以取消其有序性）
  - 数学上的关系可以无限，但关系数据库中无限关系是无意义的，关系必须是有限元组的集合；
  - 关系是简单的二维表，每一列都是不可分的；
  - 严格的说，关系是一种规范化了的二维表行的集合。

$$(d_1,\ldots d_i,d_j,\ldots d_n)=(d_1,\ldots d_j,d_i,\ldots d_n),\ (i,j=1,2,\ldots,n)$$

姓名={张三，李四},性别={男，女};

姓名×性别＝{(张三，男), (张三，女),(李四，男),(李四，女)}

规范化的关系

满足以下性质的关系，称为规范化的关系

① 列是同质的

- 每一列中的值是同一类型的数据，来自同一个域。

② 不同的列可出自同一个域

- 每一列称为一个属性，不同的属性要给予不同的属性名。

③ 列的次序是无关紧要的

- 列的次序可以任意交换

④元组的每个分量都是原子的

- 每一个分量都必须是不可分的数据项。

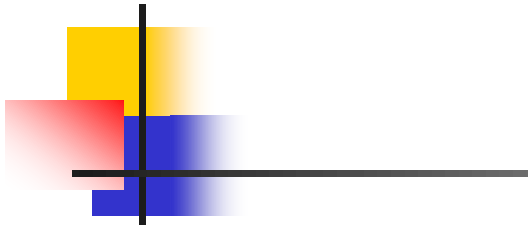
⑤ 元组的次序是无关紧要的

- 行的顺序无所谓，即行的次序可以任意交换

⑥各个元组都是不同的

- 关系中不允许出现重复元组

- 实际的商品化的数据库管理系统不一定都满足这些规范化要求
  - 有的允许重复元组
  - 有的区分元组的顺序和属性的次序



关系的定义

？

关系与关系模式

- 关系的型称为关系模式 (Relation Schema)
  - 是对关系的描述，该描述包括关系名、属性名、属性的类型和长度，以及属性间固有的数据关联关系。
  - 关系模式一般简记为关系名和属性名的集合

    $R(A_1, A_2, \ldots, A_n)$，或仅用关系名R表示。
  - 例：图书关系模式：图书(书号,书名,作者,单价,出版社)
    或简写为图书关系
- 关系的值是元组的集合，称为关系
  - 关系是对现实世界中事物在某一时刻状态的反映，关系的值是随时间在不断变化的。
  - 关系模式R上的一个关系通常写为r(R) 。

- 关系模式的形式化表示

  ***R*（*U*，*D*，*DOM*，*F*）**

  R 关系名

  U 组成该关系的属性名集合

  D 属性组***U***中属性所来自的域

  DOM 属性向域的映象集合

  F 属性间的数据依赖关系集合

关系数据库与关系数据库模式

- 关系模式的集合称为关系数据库模式
  - 是对数据库中所有数据逻辑结构的描述(关系数据库的型)
  - 表示为

    $R=\{R_1, R_2, \ldots, R_p\}$。
- 关系数据库
  - 关系数据库模式中每个关系模式上的关系的集合称为关系数据库。(关系数据库的值)
  - 表示为

    $d=\{r_1, r_2, \ldots, r_p\}$。

关系模式和关系往往统称为关系， 通过上下文加以区别

键（Key）

- 键（key，码）
- 关系数据库的重要概念
- 关系是元组的集合，为了区分不同元组，用其中一个或多个属性值来标识元组；能够惟一标识元组的属性或属性组称为关系的键。
- 若属性集K是关系模式R的键，对r (R)中任意二个不同的元组t_1、t_2应满足t_1 [K] ≠ t_2[K]，即任意二个不同的元组其K的值是不同的。

- 键的形式化定义

设关系模式**R(U)**，**K⊆U**，**r**是**R**上的任一关系，若对**r**中的任意二个不同的元组满足：

（1）$t_1[K] \neq t_2[K]$；

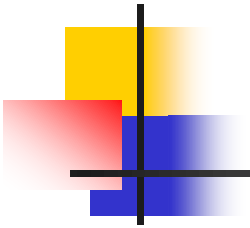
（2）不存在**K′⊂K**而使得$t_1[K'] \neq t_2[K']$成立；

则称**K**是**R**的键。

若仅条件(1)成立，则称**K**为**R**的超键（Superkey）

- 键除了能标识元组外，还应具有最小属性数。
- 超键不要求具有最小属性数。
- 超键包括了键。

- 候选键(Candidate Key)
  - 关系中能够起到标识作用的键（可能有多个）
- 主键(Primary Key)与候补键(Alternate Key)
  - 若一个关系有多个候选键，则选定其中一个作为主键；其余的键称为候补键。
- 联合键(Concatenated Key)
  - 由关系的多个属性组成的键
- 全键(All Key)
  - 由关系的所有属性组成的键
- 候选键中的属性称为主属性(Prime attribute)，不包含在任何候选键中的属性称为非主属性或非键属性(Non-key attribute)



- 外键(Foreign Key)
  - 当前关系的某个属性是另外一个关系的键，则称这个属性为当前关系的外键。
  - 给出了在不同关系间建立联系的一种方法。
  - 外键并不一定要与相应的主键同名；
  - 当外键与相应的主键属于不同关系时，往往取相同的名字，以便于识别。

- 示例：关系的键

学生关系S（学号，学生姓名，年龄，性别，班号）

班级关系C（班号，班长，学生人数，专业）

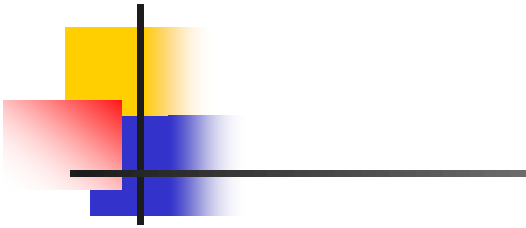
学生选课关系SC（学号，课程号，成绩）

用户关系U（身份证， 姓名， 性别，年龄）

供应关系Suppl（供应商，商品，商场）

关系的完整性约束

- 为了维护数据库中的数据与现实世界的一致性，关系模式上的所有关系必须满足一定的完整性约束条件。
  - 属性取值的约束
  - 属性间值的约束
  - 不同关系中属性值的约束
  - ……
- 完整性约束是语义上的概念，是现实世界对数据的限定。
  - 实体完整性约束
  - 参照完整性约束
  - 其他约束(用户定义的完整性约束)



- 实体完整性和参照完整性
  - 关系模型必须满足的完整性约束条件，称为关系的两个不变性
  - 由关系系统自动支持
- 用户定义的完整性
  - 应用领域需要遵循的约束条件
  - 体现了具体领域中的语义约束

- 实体完整性约束（**Entity Integrity Constraint**）
  - 对关系中主键取值的约束
  - 作为键的各个属性的值不能为空
  - 主键的值不能为空或部分为空，否则不能表示任何实体，因而无意义。
  - 空值：“不知道”、“不存在”或“无意义”的值。
- 参照完整性约束（**Reference Integrity Constraint**）
  - 对关系中外键取值的约束
  - 若关系R_1中属性A是另一个关系R_2中的主键，则对于关系R_1中的任一个元组在属性A上的值或者为空值，或者为另一个关系R_2中某个元组的主键的值。

- 参照完整性约束
  - 给出了关系间建立联系的约束规则：参照
  - 参照：某个关系与另一个关系通过定义在同一个域上的属性而建立的联系。
  - 参照关系
    例： $\mathbf{R_1}$ ， 包含外键
  - 被参照关系
    例： $\mathbf{R_2}$ ， 主键
  - 规定了外键的取值：
    (1)空值；
    (2)被参照关系中某个元组的主键的值。

- 参照完整性示例

学号	姓名	性别	专业号	年龄
801	张三	女	01	19
802	李四	男	01	20
803	王五	男	01	20
804	赵六	女	02	20
805	钱七	男		19

专业号	专业名
01	信息
02	数学
03	计算机

学号	姓名	性别	专业号	年龄
801	张三	女	01	19
802	李四	男	01	20
803	王五	男	01	20
804	赵六	女	02	20
805	钱七	男	02	19

课程号	课程名	学分
01	数据库	4
02	数据结构	4
03	编译	4
04	PASCAL	2

“学号”和“课程号”只能取相应被参照关系中已经存在的主键值。

学号	课程号	成绩
801	04	92
801	03	78
801	02	85
802	03	82
802	04	90
803	04	88

学生实体及其内部的领导联系(一对多)

学生（学号，姓名，性别，专业号，年龄，班长）

学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	
805	钱七	男	02	19	

“班长”属性值可以取空值或非空值

- 其他约束（用户定义的完整性）
  - 由用户根据实际应用定义的完整性约束；
  - 与具体的数据库应用系统有关；
  - 反映某一具体应用所涉及的数据必须满足的语义要求；
  - **DBMS**应提供定义和检查机制，不应由应用程序承担；

课程(课程号，课程名，学分)

1）“课程名”属性必须取唯一值；

2）非主属性“课程名”也不能取空值；

3）“学分”属性只能取值{1，2，3，4}。

- 完整性约束的实施

  –实体完整性
  –参照完整性
  } 关系系统自动支持

  –用户定义的完整性 —— 用户定义后由系统支持

关系与关系模式（小结）

- 关系模式是型
- 关系是值
- 关系模式是对关系的描述
  - 元组集合的结构
    - 属性构成
    - 属性来自的域
    - 属性与域之间的映象关系
  - 元组语义以及完整性约束条件
  - 属性间的数据依赖关系集合

- 关系模式
  - 对关系的描述
  - 静态的、稳定的
- 关系
  - 关系模式在某一时刻的状态或内容
  - 动态的、随时间不断变化的
- 关系模式和关系往往统称为关系

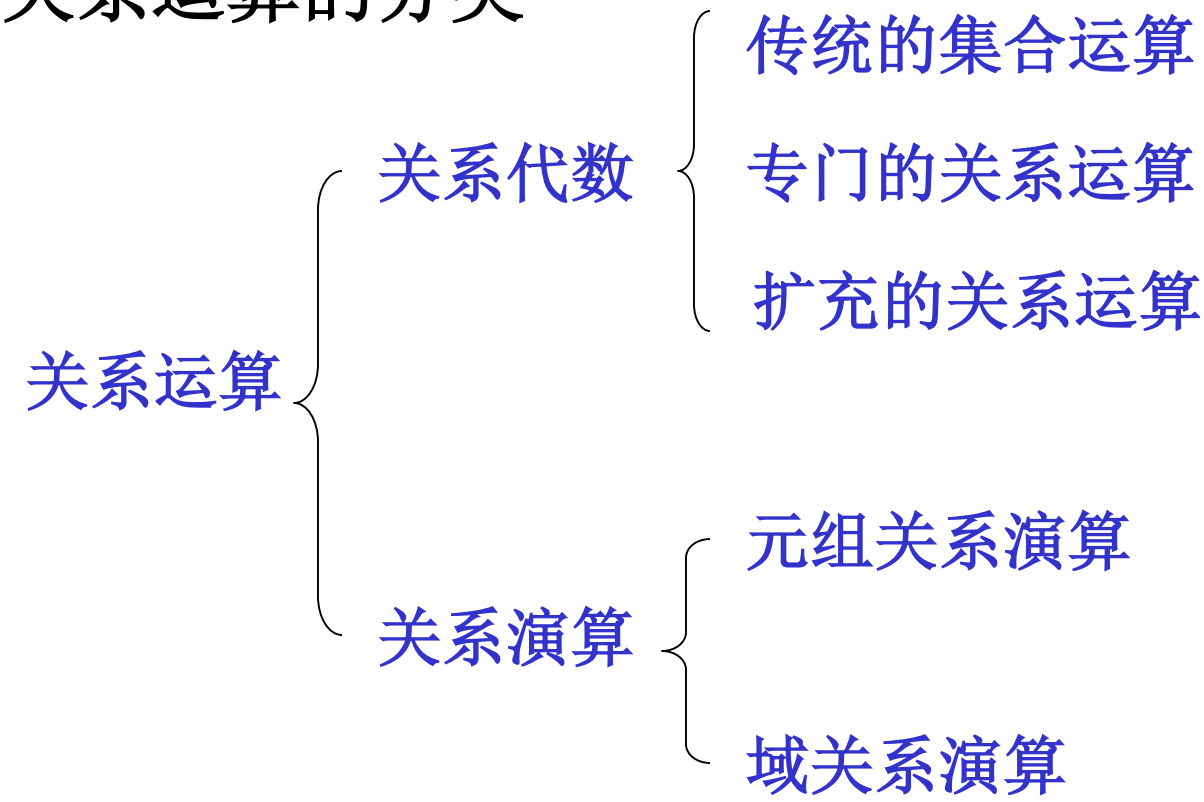
  通过上下文加以区别

2. 关系代数

- 关系运算概述
- 关系代数概述
- 传统的集合运算
- 专门的关系运算
- 扩充的关系运算
- 举例
- **ISBL** 语言

关系运算概述

- 关系模型对数据的各种操作都可以归结为对关系的运算；给出运算表达式就可得到所需要的结果。
- 关系运算的分类



关系代数概述

- 关系代数
  - 以集合代数为基础发展起来的，以关系为运算对象的一组运算集合；
  - 一种抽象的查询语言，用对关系的运算来表达查询；
  - 运算对象和运算结果都是关系；
- 关系代数运算的分类：
  - 传统的集合运算
    - 并、差、交、广义笛卡尔积
    - 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
  - 专门的关系运算
    - 选择、投影、连接、除
  - 扩充的关系运算（为数据库的应用而引进的特殊运算）

传统的集合运算—并

- *R*和*S*
  - 具有相同的目*n*（即两个关系都有*n*个属性）
  - 相应的属性取自同一个域
- $R \cup S$
  - 仍为*n*目关系，由属于*R*或属于*S*的元组组成

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

R

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

S

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

$R \cup S$

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

传统的集合运算—差

- *R*和*S*
  - 具有相同的目*n*
  - 相应的属性取自同一个域
- *R* - *S*
  - 仍为*n*目关系，由属于*R*而不属于*S*的所有元组组成

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

R

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

S

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

R-S

A	*B*	*C*
a1	*b1*	*c1*

传统的集合运算—交

- **R**和**S**
  - 具有相同的目**n**
  - 相应的属性取自同一个域
- **R**∩**S**
  - 仍为**n**目关系，由既属于**R**又属于**S**的元组组成

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

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

R

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

S

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

$R \cap S$

A	B	C
a1	b2	c2
a2	b2	c1

传统的集合运算—笛卡尔积

- 关系R和S的笛卡尔积为R中所有元组和S中所有元组的拼接
- R：n目关系，k_1个元组
- S：m目关系，k_2个元组
- $R \times S$
  - $R \times S = \{t_r\ t_s \mid t_r \in R \wedge t_s \in S\}$
  - 元组的前n列是关系R的一个元组，后m列是关系S的一个元组
  - $k_1 \times k_2$个$(n+m)$列的元组的集合

$R \times S$

R

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

S

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

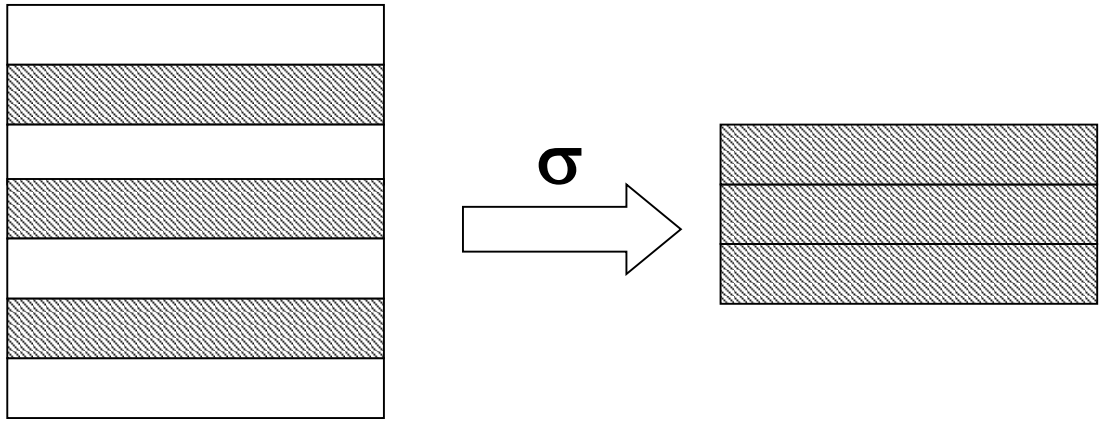
R.A	R.B	R.C	S.A	S.B	S.C
a1	b1	c1	a1	b2	c2
a1	b1	c1	a1	b3	c2
a1	b1	c1	a2	b2	c1
a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a1	b3	c2
a2	b2	c1	a2	b2	c1

专门的关系运算—选择（Selection）

- 选择运算是关系上的一元运算，是从关系中选择满足一定条件的元组子集

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

  - F 是限定条件的布尔表达式，由逻辑算符($\wedge$、$\vee$、$\neg$)连接比较表达式组成
  - 上式表示在关系 R 中选择使$F(t)$为真的所有元组
  - 选择运算是从行的角度进行的运算



- 选择操作示例

学生关系（Student）

学号 SNO	姓名 Sname	性别 Ssex	年龄 Sage	所在系 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

SNO	Sname	Ssex	Sage	Sdept
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

选取年龄小于20岁的学生

$\sigma_{Sage < 20}(Student)$

选取信息系（IS系）全体学生

$\sigma_{Sdept = 'IS'}(Student)$

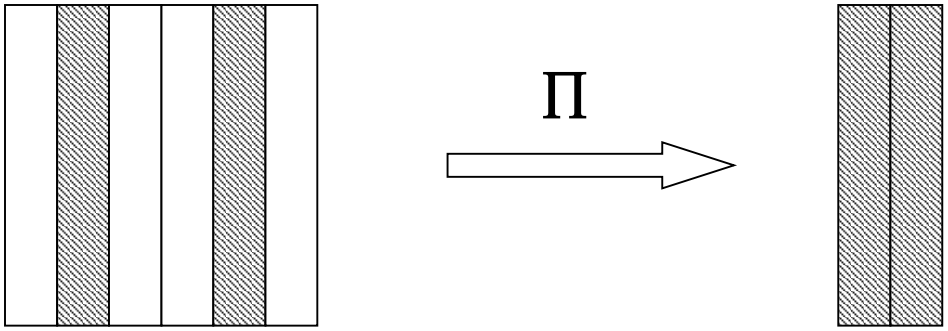
SNO	Sname	Ssex	Sage	Sdept
95002	刘晨	女	19	IS
95004	张立	男	19	IS

专门的关系运算—投影（Projection）

- 在模式R上的投影运算表示为

$$\Pi_X(R)=\{t[X] \mid t \in R\}$$

  - Π是投影算符，X是模式 R 属性的子集，$t[X]$表示R中元组在属性集 X 上的值，或为元组t在 X 上的投影
  - 结果：从R中选择出若干属性列组成新的关系
  - 投影操作主要是从列的角度进行运算



- 投影操作示例
  查询学生的姓名和所在系，即求**Student**关系上学生姓名和所在系两个属性上的投影

$\Pi_{Sname,\ Sdept}(Student)$　　$\Pi_{Sdept}(Student)$

学号 SNO	姓名 Sname	性别 Ssex	年龄 Sage	所在系 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

姓名 Sname	所在系 Sdept
李勇	CS
刘晨	IS
王敏	MA
张立	IS

所在系 Sdept
CS
IS
MA

专门的关系运算—连接（Join）

- 连接运算是把二个关系中的元组按条件连接起来，形成一个新关系
  - 条件连接
  - 自然连接
- 条件连接也称θ连接，是将二个关系中满足θ条件的元组拼接起来形成新元组的集合。
  - 设属性A和B分别是关系R和S上的属性，且定义在同一个域上，R和S的连接记为：

$$R \underset{A\,\theta\,B}{\bowtie} S = \{ t \mid t = t_r\, t_s,\ t_r \in R \wedge t_s \in S \wedge t_r[A]\ \theta\ t_s[B] \}$$

⋈ 是连接符，$A\theta B$为连接条件，θ是比较符。

- 条件连接示例

R

A	B	C
a1	b1	5
a1	b2	6
a2	b3	8
a2	b4	12

S

B	E
b1	3
b2	7
b3	10
b3	2
b5	2

$R \underset{C<E}{\bowtie} S$

A	R.B	C	S.B	E
a_1	b_1	5	b_2	7
a_1	b_1	5	b_3	10
a_1	b_2	6	b_2	7
a_1	b_2	6	b_3	10
a_2	b_3	8	b_3	10

- 条件连接的转换
  - 从R和S的笛卡尔积$R\times S$中选取R关系在A属性组上的值与S关系在B属性组上的值满足比较条件的元组

$$R \underset{A\,\theta\,B}{\bowtie} S = \sigma_{A\,\theta\,B}(R\times S)$$

- 等值连接，最常用的连接
  - 连接条件是二个属性值的相等比较；
  - θ为“=”的连接运算称为等值连接

- 等值连接示例

R

A	B	C
a1	b1	5
a1	b2	6
a2	b3	8
a2	b4	12

S

B	E
b1	3
b2	7
b3	10
b3	2
b5	2

$R \bowtie S$

R.B=S.B

A	R.B	C	S.B	E
a_1	b_1	5	b_1	3
a_1	b_2	6	b_2	7
a_2	b_3	8	b_3	10
a_2	b_3	8	b_3	2

- 自然连接（Natural join）
  - 自然连接是一种特殊的等值连接；它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列去掉

R

A	*B*	*C*
a1	*b1*	5
a1	*b2*	6
a2	*b3*	8
a2	*b4*	12

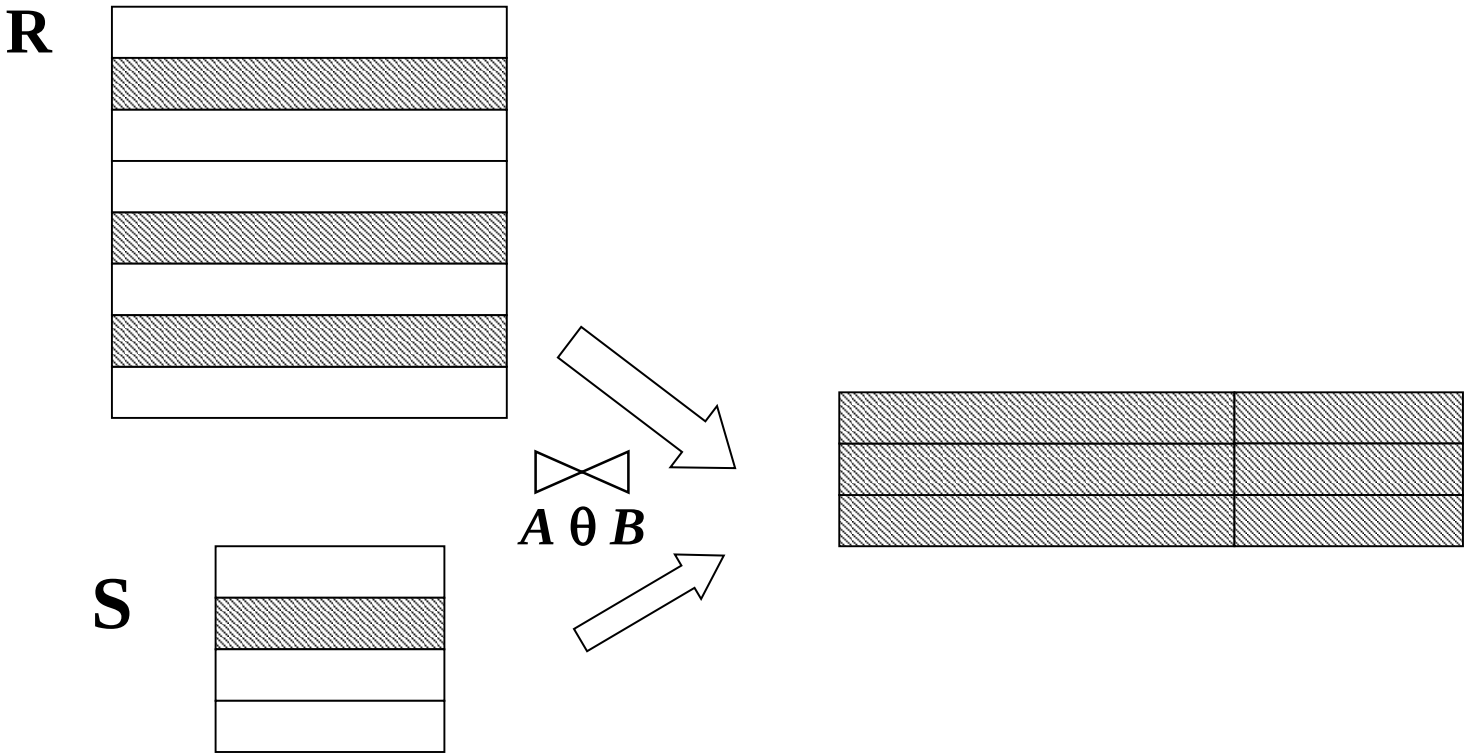
S

B	*E*
b1	3
b2	7
b3	10
b3	2
b5	2

自然连接 $R \bowtie S$

A	*B*	*C*	*E*
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

- 一般的连接操作是从行的角度进行运算。



- 自然连接还需要取消重复列，所以是同时从行和列的角度进行运算。

专门的关系运算—除法(Division)

- 象集

  给定一个关系R（X，Z），X和Z为属性组。当$t[X]=x$时，x在R中的象集（Images Set）为：

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

它表示R中属性组X上值为x的诸元组在Z分量上的取值的集合。

- 象集示例

R

x_1	Z_1
x_1	Z_2
x_1	Z_3
x_2	Z_2
x_2	Z_3
x_3	Z_1
x_3	Z_3

- x_1在R中的象集

  $Z_{x1}=\{Z_1, Z_2, Z_3\}$,
- x_2在R中的象集

  $Z_{x2}=\{Z_2, Z_3\}$,
- x_3在R中的象集

  $Z_{x3}=\{Z_1, Z_3\}$

- 除法

给定关系$R(X, Y)$和$S(Y, Z)$，其中X，Y，Z为属性组，R中的Y与S中的Y可以有不同的属性名，但必须出自相同的域集。R与S的除运算得到一个新的关系$P(X)$，P是R中*满足下列条件的元组*在X属性列上的投影：

元组在X分量上的值x的象集Y_x包含S在Y上投影的集合。

即：

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

Y_x是x在R中的象集，$x = t_r[X]$

- 除法示例

R

A	B	C
a_1	b_1	c_2
a_2	b_3	c_7
a_3	b_4	c_6
a_1	b_2	c_3
a_4	b_6	c_6
a_2	b_2	c_3
a_1	b_2	c_1

S

B	C	D
b_1	c_2	d_1
b_2	c_1	d_1
b_2	c_3	d_2

$R \div S$

A
a_1

在关系R中，A可以取四个值{a1，a2，a3，a4}

a_1的象集为 $\{(b_1, c_2), (b_2, c_3), (b_2, c_1)\}$

a_2的象集为 $\{(b_3, c_7), (b_2, c_3)\}$

a_3的象集为 $\{(b_4, c_6)\}$

a_4的象集为 $\{(b_6, c_6)\}$

S在(B, C)上的投影为

$\{(b_1, c_2), (b_2, c_1), (b_2, c_3)\}$

只有a_1的象集包含了S在(B, C)属性组上的投影，所以

$R \div S = \{a_1\}$

- 除法运算的结果属性是*R*的属性中去掉与*S*相同的属性后剩下的其它属性。

SC'

Sno	Cno
200701	C1
200701	C2
200701	C3
200702	C1
200702	C2
200703	C2
200703	C3

C'

Cno
C1

SC'÷C'

Sno
200701
200702

选修了C1课的所有学生

C''

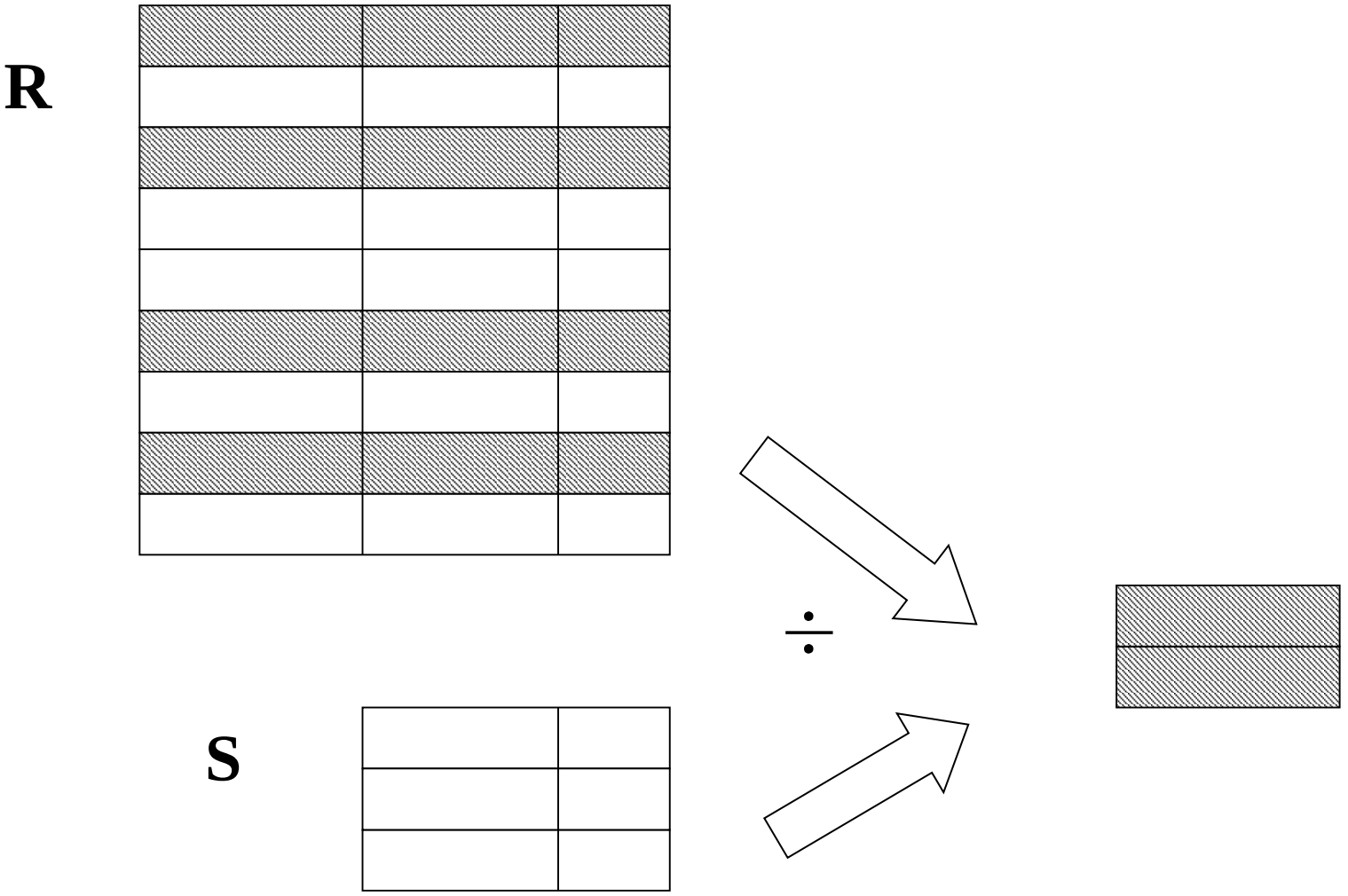
Cno
C2
C3

SC'÷C''

Sno
200701
200703

同时选修了C2和C3课的所有学生

- 除操作是同时从行和列角度进行运算



扩充的关系运算—属性重命名

- 设r是模式R上的一个关系，A是R中的一个属性，B为属性名，B不是R中的属性，B和A具有相同的域。设R'=(R-A)∪B，则属性A被重命名为B后，得到的关系r'记为：

$$r'(R') = \delta_{A \to B}(r)$$

- 重命名后的关系r'可表示如下：

$$r'(R') = \{t' \mid t' \in r' \wedge t \in r \wedge t'[R-A] = t[R-A] \wedge t'[B] = t[A]\}$$

- 重命名运算可以作用于一组属性

- 通过属性重命名运算，可以
  - 做同一个关系的笛卡儿积
  - 在同一个关系上做自然连接运算
  - 将两个关系的等值连接方便地表示为自然连接（？）

- 例：把学生关系中的学号和姓名**Sno**和**Sname**重命名为**Sno'**和**Sname'** 。

  $\delta_{Sno,Sname \rightarrow Sno',Sname'}(Student)$

扩充的关系运算—外连接

- 连接运算是把二个关系中的元组按条件连接起来，结果为满足条件的元组集合，这样的连接称为内连接（**inner join**），还有一种连接称为外连接。
- 外连接（**outer join**）是对自然连接运算的扩展。外连接结果中除了满足连接条件的元组外还包含没有被连接的元组。
- 外连接分
  - 左外连接
  - 右外连接
  - 完全外连接

- 左外连接
  - $R \overset{L}{\bowtie} S$　　$R \supset\!\bowtie S$
  - 连接结果中包含了关系R (左边关系)中不满足连接条件的元组，在这些元组对应关系S属性上的值为空值。
  - 对R中任意元组，若S中找不到匹配的元组，则S用空元组与之对应；R的信息在左外连接的结果中都得到保留。

EMP

ENAME	CITY
张丽	北京
王小	大连
李宏	长春
邓平	上海

SAL

ENAME	SALARY
张丽	6000
王小	5000
赵刚	3000

$EMP \overset{L}{\bowtie} SAL$

ENAME	CITY	SALARY
张丽	北京	6000
王小	大连	5000
李宏	长春	NULL
邓平	上海	NULL

- 右外连接
  - $R \bowtie_R S$　　$R \rtimes S$
  - 连接结果中包含了关系S (右边关系) 中不满足连接条件的元组，在这些元组对应关系R属性上的值为空值。
  - 对S中任意元组，若R中找不到匹配的元组，则R用空元组与之对应。S的信息在右外连接的结果中都得到保留。

EMP

ENAME	CITY
张丽	北京
王小	大连
李宏	长春
邓平	上海

SAL

ENAME	SALARY
张丽	6000
王小	5000
赵刚	3000

EMP $\bowtie_R$ SAL

ENAME	CITY	SALARY
张丽	北京	6000
王小	大连	5000
赵刚	NULL	3000

- 完全外连接
  - $R \overset{\bowtie}{} {}_F S$　　$R \overset{\supset\bowtie\subset}{} S$
  - 连接结果中包含了关系R中不满足连接条件的元组，同时也包含了关系S中不满足连接条件的元组；即连接结果是左外连接和右外连接结果的并。

EMP

ENAME	CITY
张丽	北京
王小	大连
李宏	长春
邓平	上海

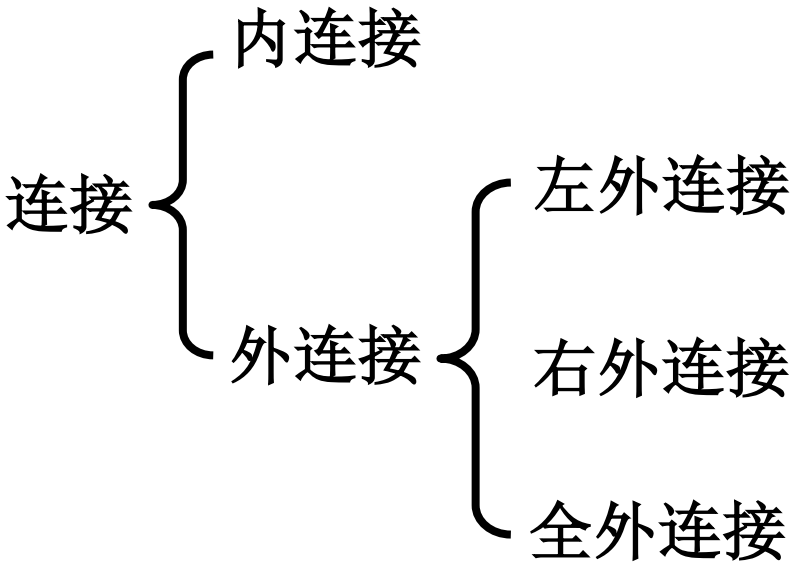
SAL

ENAME	SALARY
张丽	6000
王小	5000
赵刚	3000

$EMP \overset{\bowtie}{} {}_F SAL$

ENAME	CITY	SALARY
张丽	北京	6000
王小	大连	5000
李宏	长春	NULL
邓平	上海	NULL
赵刚	NULL	3000

- 关系代数的连接运算



关系代数表达式

- 关系代数运算
  - 关系代数运算
    并、差、交、笛卡尔积、投影、选择、连接、除
  - 基本运算
    并、差、笛卡尔积、投影、选择
  - 交、连接、除法可以用5种基本运算来表达

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

$$R \underset{A\,\theta\,B}{\bowtie} S = \sigma_{A\,\theta\,B}(R \times S)$$

$$R \div S = \prod_X(R) - \prod_X((\prod_X(R) \times \prod_Y(S)) - R)$$

X为R中除去与S相同属性Y之后的其余属性

- 关系代数表达式
  - 关系代数运算经有限次复合而形成的表达式，称为关系代数表达式。
  - 用关系代数表达式可以表示对数据库的各种操作
    - 检索（查询）
    - 插入
    - 删除
    - 修改（先删除，再插入）
- 以关系代数运算为基础的数据子语言可以实现对数据库的所有查询和更新操作。关系代数的这种处理能力称为关系的完备性。
- 针对特定操作的关系代数表达式并不唯一。

关系代数对数据库操作示例

- 检索计算机系学生的学号和姓名。

Sno	Sname	Sage	Ssex	Sdept
200701	刘明亮	18	男	计算机
200702	李和平	17	男	外语
200703	王　茵	21	女	计算机
200704	张小芳	20	女	数学

$$\Pi_{Sno,\ Sname}(\sigma_{Sdept=\ '计算机'}(S))$$

- 检索选修了**C1**课的学生信息。

Sno	Cno	Grade
200701	C1	85
200701	C2	70
200701	C3	78
200702	C1	81
200702	C2	84
200703	C2	75
200703	C3	90

Sno	Sname	Sage	Ssex	Sdept
200701	刘明亮	18	男	计算机
200702	李和平	17	男	外语
200073	王　茵	21	女	计算机
200704	张小芳	20	女	数学

$$\Pi_{Sno}(\sigma_{Cno='C1'}(SC)) \bowtie S$$

- 检索不选**C1**课的学生信息。

$$S-(\Pi_{Sno}(\sigma_{Cno='C1'}(SC)) \bowtie S)$$

- 检索选修了全部课程的学生的学号

Sno	Cno	Grade
200701	C1	85
200701	C2	70
200701	C3	78
200702	C1	81
200702	C2	84
200703	C2	75
200703	C3	90

Cno	Cname	Cdept
C1	C语言	计算机
C2	英语	外语
C3	数据库	计算机
C4	数学	数学

$$\Pi_{Sno,Cno}(SC) \div \Pi_{Cno}(C)$$

- 插入学号为200701的学生选修了C4课、成绩为88分的选课记录。

$$SC \cup \{‘200701’,‘C4’, 88\}$$

- 删除学生刘明亮选修的英语课。

$$SC - (\Pi_{Sno}(\sigma_{Sname=‘刘明亮’}(S)) \bowtie SC \bowtie \Pi_{Cno}(\sigma_{Cname=‘英语’}(C)))$$

- 检索与李勇在同一个系的学生的学号和姓名

Sno	Sname	Sage	Ssex	Sdept

Sno’	Sname’	Sage’	Ssex’	Sdept

$$\Pi_{Sno',\ Sname'}(\sigma_{Sname=‘李勇’}(S \bowtie \delta_{Sno,Sname,Sage,Ssex \to Sno',Sname',Sage',Ssex'}(S)))$$

$$\Pi_{Sno,\ Sname,\ Sdept}(S) \div \Pi_{Sdept}(\sigma_{Sname=‘李勇’}(S))$$

P46例3-15有误

- 查询至少选修C1课程和C3课程的学生的学号

首先建立一个临时关系K：

Cno
C1
C3

然后求

$$\Pi_{Sno,Cno}(SC) \div K$$

结果

$$\Pi_{Sno,Cno}(SC) \div K = \{2007001\}$$

- 查询至少选修了一门其直接先行课号**Cpno**为**5**的课程的学生姓名。

S	Sno	Sname	Sage	Ssex	Sdept

SC	Sno	Cno	Grade

C	Cno	Cname	Cpno	Ccredit

$$\Pi_{Sname}(\sigma_{Cpno='5'}(C \bowtie SC \bowtie S))$$

$$\Pi_{Sname}(\sigma_{Cpno='5'}(C) \bowtie SC \bowtie \Pi_{Sno,\ Sname}(S))$$

$$\Pi_{Sname}(\Pi_{Sno}(\sigma_{Cpno='5'}(C) \bowtie SC) \bowtie \Pi_{Sno,\ Sname}(S))$$

- 查询选修了全部课程的学生的学号和姓名。

$$\Pi_{Sno,Cno}(SC) \div \Pi_{Cno}(C)$$

$$\Pi_{Sno,\ Sname}(S)$$

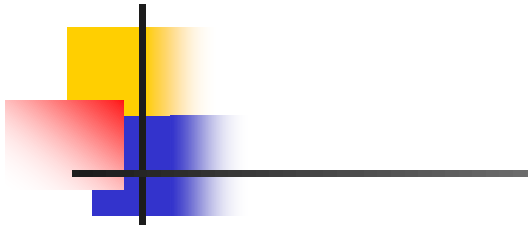
$$\Pi_{Sno,Cno}(SC) \div \Pi_{Cno}(C) \bowtie \Pi_{Sno,\ Sname}(S)$$

典型关系代数语言

- **ISBL（Information System Base Language）**
  - 由**IBM United Kingdom**研究中心研制
  - 用于**PRTV（Peterlee Relational Test Vehicle）**实验系统
  - 与关系代数十分相近
  - 允许将关系表达式的值赋给一个关系变量
  - 允许属性的重命名
  - 可以完成关系代数的五种基本运算，是关系完备的
  - 参阅表**3-4**

表 3-4 ISBL 与关系代数操作符的对应

ISBL	关系代数
R+S	$R \cup S$
R−S	$R - S$
R·S	$R \cap S$
R：F	$\sigma_F(R)$
$R\%_{A1,A2,\ldots,AK}$	$\prod_{A1,A2,\ldots,AK}(R)$
R*S	$R \bowtie S$



【例 3-16】检索计算机系学生的学号和姓名。

LIST Student：Sdept=’计算机’%Sno，Sname

【例 3-17】检索选修了 C1 课的学生姓名和学生所在系。

R = Student *SC

LIST R：Cno=’C1’% Sname，Sdept

3. 关系演算

- 关系演算：使用谓词演算表达对关系数据的查询
  - 用谓词的形式给出查询结果应该满足的条件；
  - 是以谓词演算为基础的关系数据查询语言；
  - 如何实现查询由系统自己解决；
  - 高度非过程化的。
- 关系演算
  - 元组关系演算
    谓词变量是元组变量
  - 域关系演算
    谓词变量是域变量

元组关系演算

- 用元组演算表达式表示查询
- 元组演算表达式的形式

  {t: φ(t)}
  - t 是元组变量，表示某一关系中的元组；
  - φ(t)是由原子公式和运算符组成的公式。
- “检索计算机系的学生姓名”的元组演算表达式为：

  {t[2]: S(t)∧t[5]='计算机'}

  检索选修了C1课的学生信息

  {t: S(t)∧∃u(SC(u)∧u[2]='C1'∧u[1]=t[1])}

- 元组关系演算语言ALPHA
  - E.F.Codd提出来的（P50-52）
  - 语言的基本格式为：

    (操作符) <工作空间名> (表达式) [:<条件>]
    - 操作符为命令语句完成的功能，包括**Get**、**Put**、**Update**、**Hold**、**Delete**等
    - 工作空间是用户与数据库间的数据通信区，用字母**w**表示
    - 表达式用于指定语句的操作对象，它可以是关系名或(和)属性名，一条语句可以同时操作多个关系或多个属性。
    - 条件是一个含逻辑运算符的谓词公式（逻辑表达式），用于将操作结果限定在满足条件的元组中；操作条件可以为空。

- Alpha查询示例
  - 查询所有被选修的课程号码
    GET W (SC.Cno)
  - 查询信息系(IS)中年龄小于20的学生学号和年龄
    GET W (Student.Sno，Student.Sage):
    Student.Sdept='IS'∧ Student.Sage<20
  - 查询计算机科学系(CS)学生的学号、年龄，结果按年龄降序排序。
    GET W (Student.Sno，Student.Sage):
    Student.Sdept='CS' DOWN Student.Sage

域关系演算

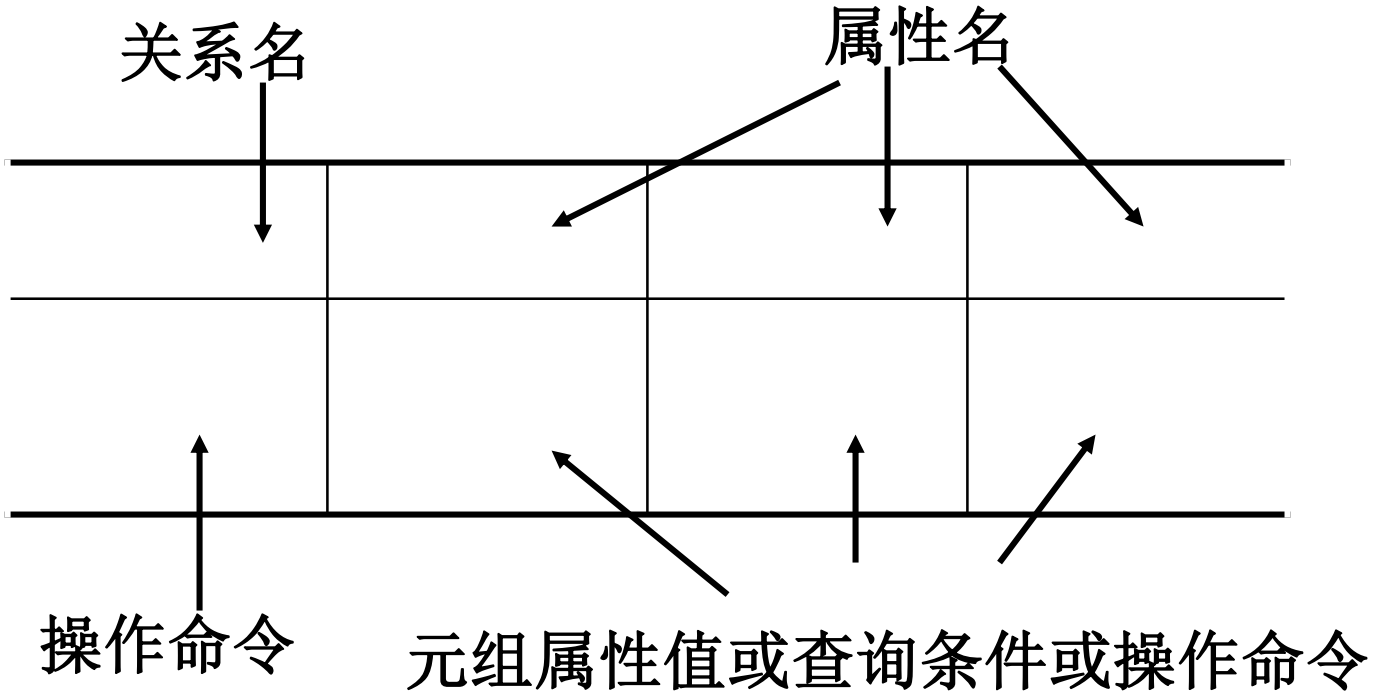
- 关系运算表达式中的变量若是域变量，则为域关系演算。
- 与元组变量不同，域变量的变化范围是指定属性的域而不是整个关系。
- 域关系演算表达式的形式

  $\{t_1, t_2, \ldots, t_k: \varphi(t_1, t_2, \ldots, t_k)\}$
  - $t_1, t_2, \ldots, t_k$是域变量
  - $\varphi(t_1, t_2, \ldots, t_k)$是由原子公式和运算符组成的公式
- “检索计算机系的学生姓名”

  $\{t_1: S(Sname:t_1, Sdept:计算机)\}$

- 域关系演算语言QBE－Query By Example
  - 由M.M.Zloof提出
  - 1978年在IBM370上得以实现
  - 基于屏幕表格的查询语言
  - 以填写表格的方式构造查询
  - 用示例元素(域变量)来表示查询结果可能的情况
  - 查询结果以表格形式显示
  - 是一种非专业用户使用的高级语言

- **QBE的主要操作命令有**
  P.（显示）、I.（插入）D.（删除）、U.（修改）
- **QBE操作框架**



QBE检索操作示例

Student	Sno	Sname	Sage	Ssex	Sdept
P.					

Student	Sno	Sname	Sage	Ssex	Sdept
		P. SN	>18		计算机

Student	Sno	Sname	Sage	Ssex	Sdept
		P. SN1 P. SN2	>18		计算机

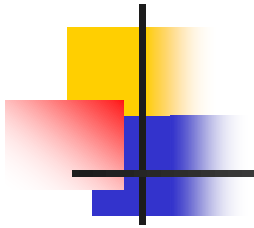
QBE－Access示例

4. 关系数据语言

关系数据语言
- 关系代数语言
- 关系演算语言
  - 元组关系演算语言
  - 域关系演算语言
- 具有关系代数和关系演算双重特点的语言（SQL）

- 关系数据语言的特点
  - 是一种高度非过程化的语言
    - 存取路径的选择由DBMS的优化机制来完成
    - 用户不必用循环结构就可以完成数据操作
  - 能够嵌入高级语言中使用

- 关系代数语言
  - 用对关系的运算来表达查询要求
  - 典型代表：**ISBL**
- 关系演算语言：用谓词来表达查询要求。
  - 按谓词变元的基本对象的不同分：
    - 元组关系演算语言
      - 谓词变元的基本对象是元组变量
      - 典型代表：**APLHA, QUEL**
    - 域关系演算语言
      - 谓词变元的基本对象是域变量
      - 典型代表：**QBE**



- 具有关系代数和关系演算双重特点的语言
  - 典型代表：**SQL(Structurel Query Language)**
  - 它是集**Query**、**DDL**、**DML**、**DCL**于一体的关系数据语言，充分体现了关系数据语言的特点和优点，是关系数据库的标准语言。

5. 关系运算的安全限制

- 关系是集合论中的概念，在集合论中，关系可以是无限的。
- 在关系数据库中，关系被限定为有限的。
- 不产生无限关系的关系表达式称为安全运算表达式，所采取的措施称安全限制
  - 关系代数运算是安全的，只要参加运算的关系是有限的；关系运算的结果关系也是有限的；关系运算的有限次复合不会出现无限关系。
  - 关系演算不一定是安全的，需要加以限制
    通常方法是定义一个有限的符号集

6. 关系运算的等价性

- 在加以安全限制后，三种关系运算（关系代数、元组关系演算、域关系演算）在表达关系的功能上是等价的。
- 如果**E**是一个关系代数表达式，则必定存在一个与**E**等价的安全的元组关系演算表达式；对于每个安全的元组关系演算表达式，必定存在一个等价的安全的域关系演算表达式；对于每个安全的域关系演算表达式，也必定存在一个等价的关系代数表达式。

本章小结

- 关系数据结构
  - 关系，关系模式，关系数据库
- 关系的完整性约束
  - 实体完整性、参照完整性、其他完整性
- 关系操作
  - 检索、插入、删除、修改等
  - 关系代数
  - 关系演算
    - 元组关系演算
    - 域关系演算
- 关系数据语言
- 关系运算的安全性和等价性

关系模型的三要素