

数据库系统概论

北京理工大学

第三章 关系数据库

- ◆ 关系数据库使用关系代数的数学方法处理数据库中的数据
- ◆ 1970年Codd 提出并完善了关系数据库模型和相应的理论基础
- ◆ 70年代末
 - ◆ IBM 研制了 System R –SQL/DS
 - ◆ 加州大学研制了 INGRES 并产品化
- ◆ 目前关系型数据库产品很多
 - ◆ DB2 Oracle Ingres Sybase Informix SQL server...

关系模型概述

- ◆ 单一的数据结构—关系—二维表
- ◆ 关系操作
 - ◆ 关系操作是集合操作方式
 - ◆ 非过程化
- ◆ 完整性约束
 - ◆ 实体完整性
 - ◆ 参照完整性
 - ◆ 用户定义完整性

关系结构的形式化定义

- ◆单一结构 表一 元组集合
- ◆关系模型建立在集合代数基础上

关系

◆域(domain)

◆域是一组具有相同数据类型的值的集合

◆笛卡儿积

定义 2.2 给定一组域 D_1, D_2, \dots, D_n , 这些域中可以有相同的。 D_1, D_2, \dots, D_n 的笛卡尔积为:

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

◆元组，分量

◆基数

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

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

◆ 关系

定义 2.3 $D_1 \times D_2 \times \cdots \times D_n$ 的子集叫作在域 D_1, D_2, \cdots, D_n 上的关系, 表示为

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

这里 R 表示关系的名字, n 是关系的目或度 (Degree)。

◆ 关系中的每个元素是关系的元组 t

◆ 关系的列为属性

◆ N 是关系的目或者度

◆ $N=1$ 单元关系

◆ $N=2$ 二元关系

码

- ◆候选码：如果关系中的某一组属性值可以唯一的标识一个元组，则该属性组称为候选码
- ◆候选码中可以选择一个做为主码
- ◆主码的各个属性称为主属性
- ◆有些情况下，所有属性的集合是关系的候选码一称为全码

关系做为关系数据模型的扩充

- ◆关系数据模型中的关系是有限集合
- ◆取消关系元组的有序性

关系的特性

- ◆列是同质的
- ◆不同列可以来自相同的域，每个列是一个属性，赋予不同的名字
- ◆列是顺序无关的
- ◆没有两个元组是完全相同的
- ◆行的顺序无关
- ◆分量是原子值—分量是不可再分的

关系的规范化

- ◆最基本一条：关系的分量不可再分
- ◆规范化的关系称为范式（**normal form**）

关系模式

- ◆关系的描述称为关系模式（**relation Schema**） 一般形象表示为 $R(U,D,dom,f)$
- ◆其中
 - ◆R 关系名
 - ◆U 属性名的集合
 - ◆D 属性所来自的域
 - ◆Dom 属性向域的映像的集合
 - ◆F是属性间数据的依赖关系
- ◆简记为 $R(U)$ $R(A_1,A_2,...A_n)$

- ◆关系模式是静态的
- ◆关系是动态的
- ◆关系是关系模式在某一个时刻的状态或者内容

关系的完整性

- ◆关系模型中的完整性是对关系的某种约束条件
- ◆完整性约束分为三类
 - ◆实体完整性
 - ◆参照完整性
 - ◆用户定义的完整性

实体完整性

- ◆ 定义：若属性A是基本关系R的主属性，则属性A的取值不能为空
- ◆ 实体完整性指基本关系的所有主属性不能为空值
 - ◆ 实体完整性是针对基本关系而言的
 - ◆ 现实世界实体是可以区分的
 - ◆ 相应，关系模型以主码做为唯一性标识
 - ◆ 主属性不能取空值

参照完整性

◆反映现实生活中实体之间的某种联系



关系间的引用

- ◆ 在关系模型中实体及实体间的联系都是用关系来描述的，因此可能存在着关系与关系间的引用。

例1 学生实体、专业实体

学生（学号，姓名，性别，专业号，年龄）

主码

专业（专业号，专业名）

- ❖ 学生关系引用了专业关系的主码“专业号”。
- ❖ 学生关系中的“专业号”值必须是确实存在的专业的专业号，即专业

关系中有该专业的记录。

关系间的引用【续】

例2 学生、课程、学生与课程之间的多对多联系

学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

关系间的引用[续]

例3 学生实体及其内部的一对多联系

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
805	钱七	男	02	19	

❖ “学号” 是主码，“班长” 是外码，它引用了本关系的“学号”

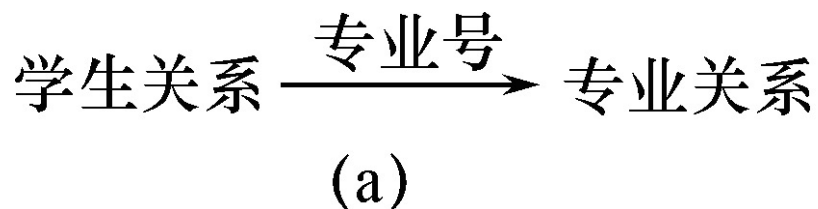
❖ “班长” 必须是确实存在的学生的学号

外码 (Foreign Key)

- ◆ 设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码。
如果 F 与基本关系 S 的主码 K_s 相对应，则称 F 是基本关系 R 的
外码
- ◆ 基本关系 R 称为**参照关系** (Referencing Relation)
- ◆ 基本关系 S 称为**被参照关系** (Referenced Relation)
或**目标关系** (Target Relation)

外码[续]

- ◆ [例1]：学生关系的“专业号”与专业关系的主码“专业号”相对应
 - “专业号”属性是学生关系的外码
 - 专业关系是被参照关系，学生关系为参照关系



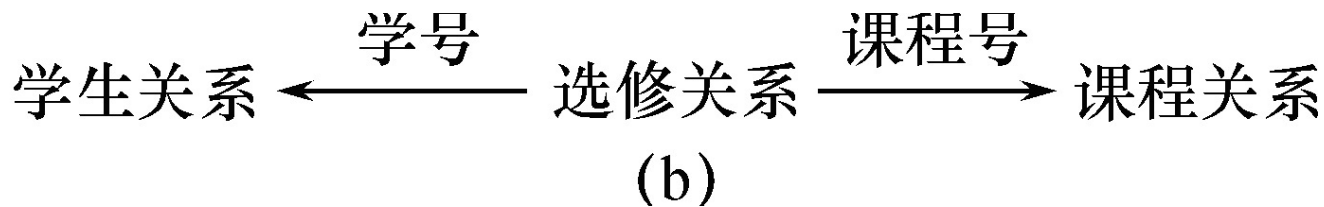
外码[续]

◆ [例2] :

选修关系的“学号”与学生关系的主码“学号”相对应

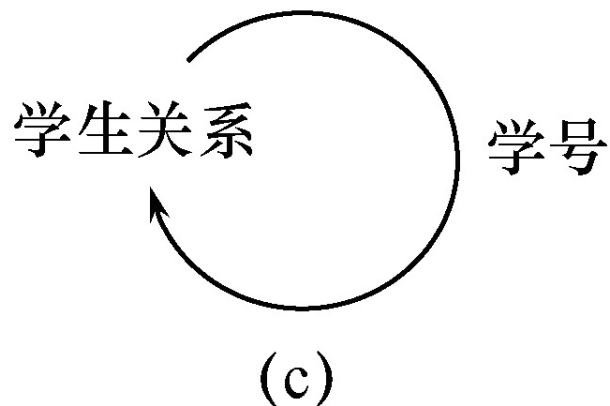
选修关系的“课程号”与课程关系的主码“课程号”相对应

- ◆ “学号”和“课程号”是选修关系的外码
- ◆ 学生关系和课程关系均为被参照关系
- ◆ 选修关系为参照关系



外码[续]

- ◆ [例3]：“班长”与本身的主码“学号”相对应
 - ◆ “班长”是外码
 - ◆ 学生关系既是参照关系也是被参照关系



外码【续】

- ◆ 关系 R 和 S 不一定是不同的关系
- ◆ 目标关系 S 的主码 K_s 和参照关系的外码 F 必须定义在同一个（或一组）域上
- ◆ 外码并不一定要与相应的主码同名

当外码与相应的主码属于不同关系时，往往取相同的名字，以便于识别

- ◆参照完整性规则：若属性（或者属性组）**F**是基本关系**R**的外码，它与基本关系**S**的主码**K_s**对应（基本关系 **R S**可以相同），则对于**R**中每个元组在**F**上的取值必须为：
 - ◆或者取空值（**F**的每个属性值均为空值）
 - ◆或者等于**S**中某个元组的主码值

其他约束

- ◆不同应用根据应用环境不同，需要指定特殊的约束条件
- ◆关系模型提供定义和检验这类完整性的机制，不需要由应用程序来完成。

用户定义的完整性【续】

例:

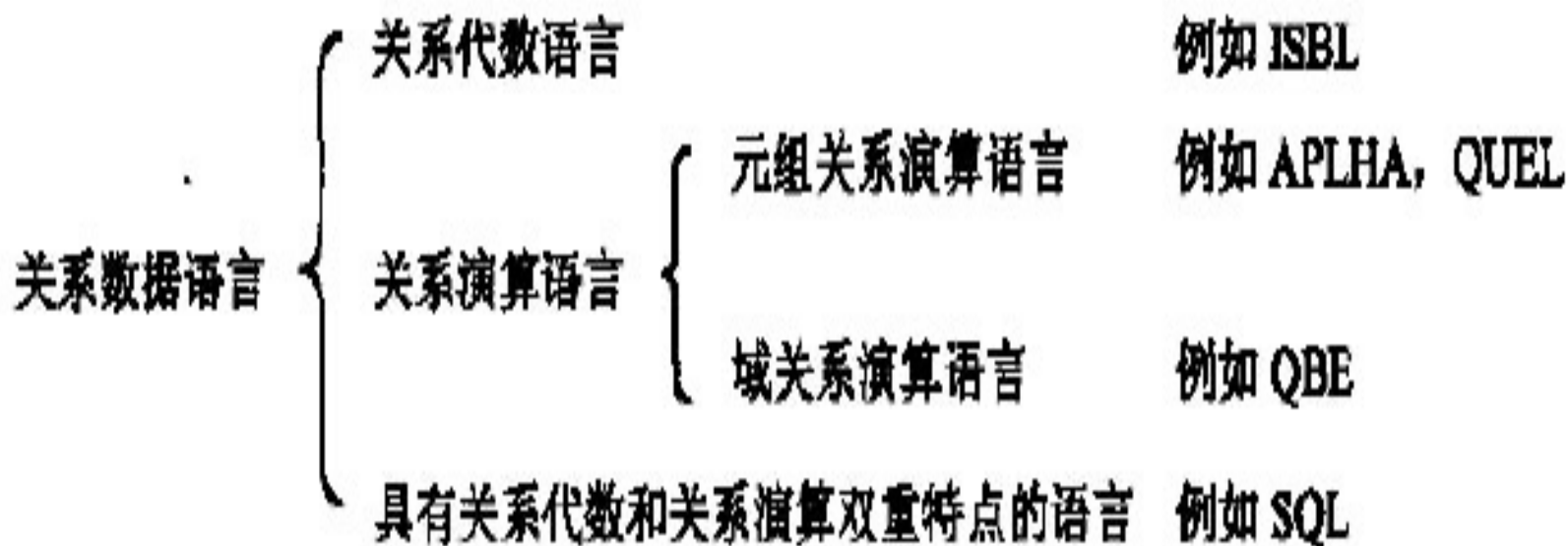
课程(课程号, 课程名, 学分)

- ◆ “课程号” 属性必须取唯一值
- ◆ 非主属性 “课程名” 也不能取空值
- ◆ “学分” 属性只能取值{1, 2, 3, 4}

关系代数

- ◆关系代数是一种抽象的查询语言，是关系数据库操纵语言的一种传统表达方式。
- ◆运算的三大要素：运算对象 运算符 运算结果
- ◆关系代数的运算对象是关系 运算结果也是关系
- ◆关系是集合

关系操作的种类



关系代数运算符

运 算 符		含 义	运 算 符		含 义
集 合 运 算 符	\cup	并 差 交	比 较 运 算 符	$>$	大 于
	\cap			\geq	大 于 等 于
专 门 的 关 系 运 算 符	\times	广 义 笛 卡 尔 积 选 择 投 影 连 接 除	运 辑 运 算 符	$<$	小 于
	σ			\leq	小 于 等 于
	π			$=$	等 于
	\bowtie			\neq	不 等 于
	\div				
专 门 的 关 系 运 算 符				\neg	非
				\wedge	与
				\vee	或

传统的集合运算

◆传统的集合运算是两目运算

- ◆设关系R 和关系 S具有相同的目n（两个关系都有n个属性），且相应的属性来自同一个域，则传统的关系运算 并 差 交 定义如下：

◆并(Union)

关系R和S的并，其结果为R和S中的所有元组组成，记为：
 $R \cup S = \{t \mid t \in R \vee t \in S\}$

并[续]

R		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cup S$		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_2

◆差(Difference)

关系R和S的差，其结果由属于R但不属于S的所有元组组成，记为：

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

◆交(Intersection)

关系R和S的交结果为既属于R又属于S的所有元组组成，记为： $R \cap S = \{t \mid t \in R \wedge t \in S\}$

◆交运算不是必须的，交运算能用差运算来表示，即：

$$R \cap S = R - (R - S)。$$

差[续]

R		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R-S$		
A	B	C
a_1	b_1	c_1

交【续】

R		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cap S$		
A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

◆笛卡尔积(Cartesian Product)

◆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。若R和S的属性个数分别为k1和k2，则R和S的笛卡尔积的属性个数为k1+k2,。若R中有m个元组，S中有n个元组，则R和S的笛卡尔积的元组个数为m×n。记为：

$$◆ \quad R \times S = \{t \mid t = \overbrace{tr \quad ts}^{\text{concatenation}}, tr \in R \wedge ts \in S\}$$

$$r \begin{array}{|c|c|c|} \hline A & B & C \\ \hline \end{array}$$

$$a_1 \quad b_1 \quad c_1$$

$$a_1 \quad b_1 \quad c_2$$

$$a_2 \quad b_2 \quad c_1$$

$$s \begin{array}{|c|c|} \hline D & E \\ \hline \end{array}$$

$$d_1 \quad e_1$$

$$d_2 \quad e_2$$

$$r \times s \begin{array}{|c|c|c|c|c|} \hline A & B & C & D & E \\ \hline \end{array}$$

$$a_1 \quad b_1 \quad c_1 \quad d_1 \quad e_1$$

$$a_1 \quad b_1 \quad c_1 \quad d_2 \quad e_2$$

$$a_1 \quad b_1 \quad c_2 \quad d_1 \quad e_1$$

$$a_1 \quad b_1 \quad c_2 \quad d_2 \quad e_2$$

$$a_2 \quad b_2 \quad c_1 \quad d_1 \quad e_1$$

$$a_1 \quad b_2 \quad c_1 \quad d_2 \quad e_2$$

专门的关系运算

先引入几个记号

(1) R , $t \in R$, $t[A_i]$

设关系模式为 $R(A_1, A_2, \dots, A_n)$

它的一个关系设为 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

专门的关系运算【续】

(2) A , $t[A]$, \overline{A}

若 $A = \{A_{i1}, A_{i2}, \dots, A_{ik}\}$, 其中 $A_{i1}, A_{i2}, \dots, A_{ik}$ 是 A_1, A_2, \dots, A_n 中的一部分, 则 A 称为属性列或属性组。

$t[A] = (t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}])$ 表示元组 t 在属性列 A 上诸分量的集合。

\overline{A} 则表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 $\{A_{i1}, A_{i2}, \dots, A_{ik}\}$ 后剩余的属性组。

专门的关系运算【续】

(3) $\widehat{t_r t_s}$

R 为 n 目关系， S 为 m 目关系。

$t_r \in R$, $t_s \in S$, $\widehat{t_r t_s}$ 称为元组的连接。

$\widehat{t_r t_s}$ 是一个 $n + m$ 列的元组，前 n 个分量为 R 中的一个 n 元组，后 m 个分量为 S 中的一个 m 元组。

专门的关系运算【续】

(4) 象集 Z_x

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：

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

它表示 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合

专门的关系运算【续】

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

$$Z_{x_1} = \{Z_1, Z_2, Z_3\},$$

◆ x_2 在 R 中的象集

$$Z_{x_2} = \{Z_2, Z_3\},$$

◆ x_3 在 R 中的象集

$$Z_{x_3} = \{Z_1, Z_3\}$$

象集举例

专门的关系运算【续】

- ◆ 选择
- ◆ 投影
- ◆ 连接
- ◆ 除

专门的关系运算【续】

4) 学生-课程数据库:

学生关系Student、课程关系Course和选修关系SC

Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
200215121	李勇	男	20	CS
200215122	刘晨	女	19	IS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

专门的关系运算【续】

Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 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

专门的关系运算【续】

SC

学号 Sno	课程号 Cno	成绩 Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

(c)

1. 选择 (Selection)

◆ 1) 选择又称为限制 (Restriction)

◆ 2) 选择运算符的含义

◆ 在关系 R 中选择满足给定条件的诸元组

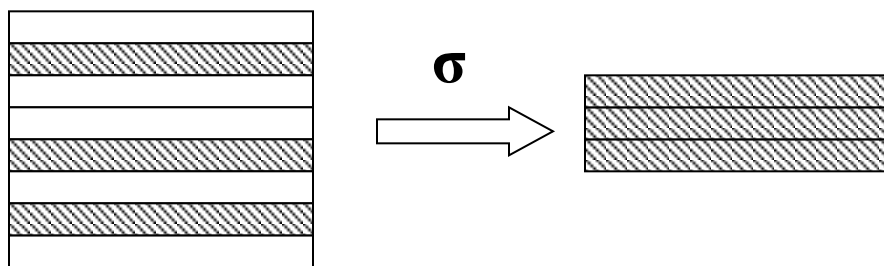
$$\sigma_F(R) = \{t | t \in R \wedge F(t) = \text{'真'}\}$$

◆ F : 选择条件, 是一个逻辑表达式, 基本形式为
:

$$X_1 \theta Y_1$$

选择（续）

- ◆3) 选择运算是从关系 R 中选取使逻辑表达式 F 为真的元组，是从行的角度进行的运算



选择（续）

[例1] 查询信息系（IS系）全体学生

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

结果：

Sno	Sname	Ssex	Sage	Sdept
200215122	刘晨	女	19	IS
200215125	张立	男	19	IS

选择（续）

[例2] 查询年龄小于20岁的学生

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

结果：

Sno	Sname	Ssex	Sage	Sdept
200215122	刘晨	女	19	IS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

2. 投影（Projection）

◆1) 投影运算符的含义

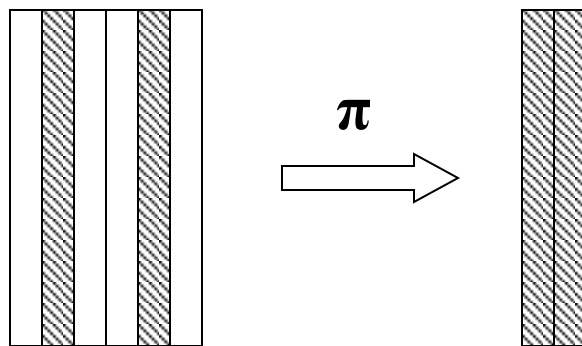
◆从 R 中选择出若干属性列组成新的关系

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

A : R 中的属性列

2. 投影 (Projection)

◆2) 投影操作主要是从列的角度进行运算



◆但投影之后不仅取消了原关系中的某些列，而且还可能取消某些元组（避免重复行）

投影（续）

◆[例3] 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

$\pi_{\text{Sname}, \text{Sdept}}(\text{Student})$

结果：

投影（续）

Sname	Sdept
李勇	CS
刘晨	IS
王敏	MA
张立	IS

投影（续）

查询学生关系**Student**中都有哪些系

$\pi_{\text{Sdept}}(\text{Student})$

结果：

Sdept
CS
IS
MA

3. 连接 (Join)

◆ 1) 连接也称为 θ 连接

◆ 2) 连接运算的含义

从两个关系的笛卡尔积中选取属性间满足一定条件的元组

$$R \bowtie_{A\theta B} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B] \}$$

➤ A 和 B : 分别为 R 和 S 上度数相等且可比的属性组

➤ θ : 比较运算符

◆ 连接运算从 R 和 S 的广义笛卡尔积 $R \times S$ 中选取 (R 关系) 在 A 属性组上的值与 (S 关系) 在 B 属性组上值满足比较关系 θ 的元组

连接【续】

◆3) 两类常用连接运算

◆等值连接 (equijoin)

➤什么是等值连接

θ 为“=”的连接运算称为等值连接

➤等值连接的含义

从关系 R 与 S 的广义笛卡尔积中选取 A 、 B 属性值相等的那些元组，即等值连接为：

$$R \bowtie_{A=B} S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B] \}$$

连接【续】

◆自然连接（Natural join）

◆自然连接是一种特殊的等值连接

- 两个关系中进行比较的分量必须是相同的属性组
- 在结果中把重复的属性列去掉

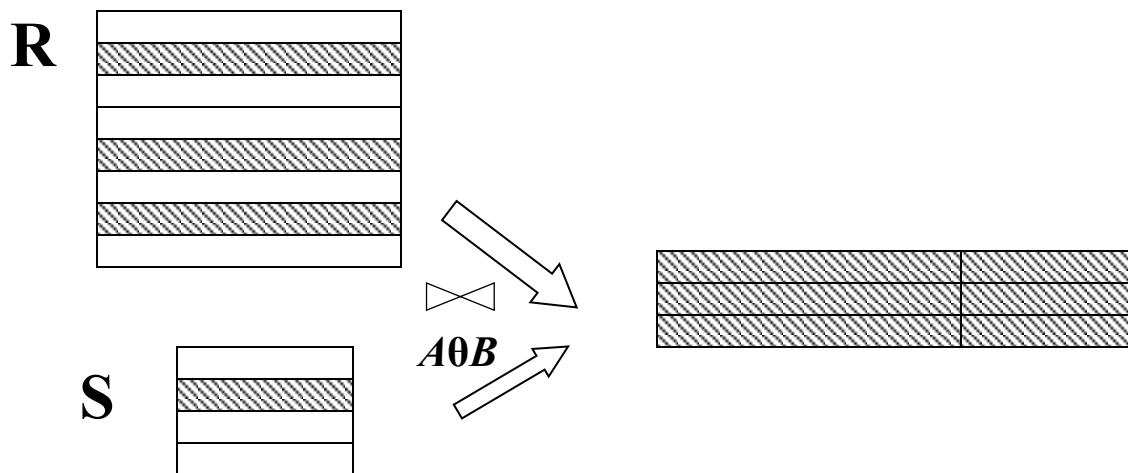
◆自然连接的含义

R 和 S 具有相同的属性组 B

$$R \bowtie S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[B] = t_s[B] \}$$

连接[续]

- ◆4) 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

连接[续]

◆ [例]关系 R 和关系 S 如下所示:

R		
A	B	C
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

S	
B	E
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

连接【续】

一般连接 $R \bowtie_{C < E} S$ 的结果如下:

$R \bowtie_{C < E} S$				
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

连接【续】

等值连接 $R \bowtie_{R.B=S.B} S$ 的结果如下：

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

连接【续】

自然连接 $R \bowtie S$ 的结果如下：

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

扩充的关系运算

◆属性重命名

- ◆有时候为了实现关系自身和自身之间的运算，可以将某些属性的名称进行换名操作，以便可以在同一个关系上进行自然链接等运算

◆外连接

- ◆对自然连接的扩展，包括符合链接条件的元组，也包括部分不符合条件的元组

连接[续]

◆外连接

- ◆如果把舍弃的元组也保存在结果关系中，而在其他属性上填空值(Null)，这种连接就叫做外连接（**OUTER JOIN**）。

◆左外连接

- ◆如果只把左边关系 R 中要舍弃的元组保留就叫做左外连接(**LEFT OUTER JOIN**或**LEFT JOIN**)

◆右外连接

- ◆如果只把右边关系 S 中要舍弃的元组保留就叫做右外连接(**RIGHT OUTER JOIN**或**RIGHT**

连接[续]

下图是例5中关系 R 和关系 S 的外连接

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
a_2	b_4	12	NULL
NULL	b_5	NULL	2

(a) 外连接

连接[续]

图(b)是例5中关系R和关系S的左外连接,图(c)是右外连接

<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>
<i>a</i> ₁	<i>b</i> ₁	5	3
<i>a</i> ₁	<i>b</i> ₂	6	7
<i>a</i> ₂	<i>b</i> ₃	8	10
<i>a</i> ₂	<i>b</i> ₃	8	2
<i>a</i> ₂	<i>b</i> ₄	12	NULL

(b) 左外连接

<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>
<i>a</i> ₁	<i>b</i> ₁	5	3
<i>a</i> ₁	<i>b</i> ₂	6	7
<i>a</i> ₂	<i>b</i> ₃	8	10
<i>a</i> ₂	<i>b</i> ₃	8	2
NULL	<i>b</i> ₅	NULL	2

(c) 右外连接

4. 除 (Division)

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:

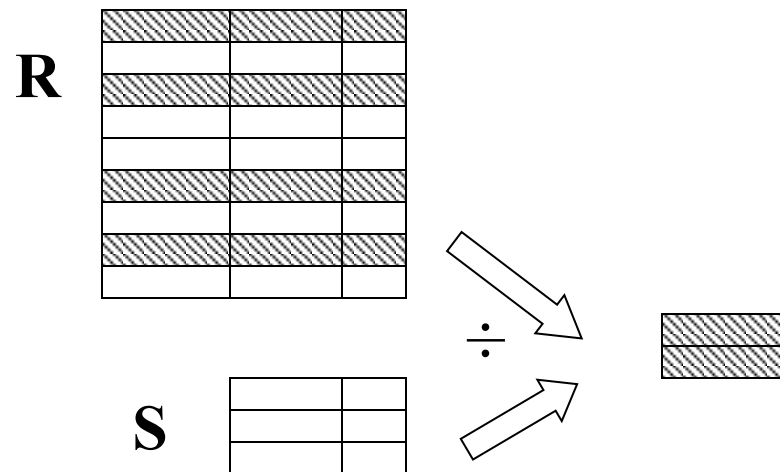
元组在 X 上分量值 x 的象集 Y_x 包含 S 在 Y 上投影的集合, 记作:

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

$$Y_x: x \text{ 在 } R \text{ 中的象集, } Y_x = \{ t_r[Y] \mid t_r[X] = x \}$$

除[续]

◆2) 除操作是同时从行和列角度进行运算



除[续]

设关系 R 、 S 分别为下图的(a)和(b)， $R \div S$ 的结果为图(c)

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

(a)

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

(b)

$R \div S$
A
a_1

(c)

分析

- ◆ 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

$\{(b1, c2), (b2, c1), (b2, c3)\}$

- ◆ 只有 a_1 的象集包含了S在(B, C)属性组上的投影

所以 $R \div S = \{a_1\}$

综合举例

以学生-课程数据库为例

[例] 查询至少选修1号课程和3号课程的学生号码

首先建立一个临时关系 K :

Cno
1
3

然后求: $\pi_{Sno, Cno}(SC) \div K$

综合举例【续】

◆ 续 $\pi_{Sno,Cno}(SC)$

200215121象集{1, 2, 3}

200215122象集{2, 3}

$K=\{1, 3\}$

于是:

$\pi_{Sno,Cno}(SC) \div K = \{200215121\}$

Sno	Cno
200215121	1
200215121	2
200215121	3
200215122	2
200215122	3

综合举例【续】

[例] 查询选修了全部课程的学生号码和姓名。

$\pi_{Sno, Cno} (SC) \div \pi_{Cno} (Course) \bowtie \pi_{Sno, Sname} (Student)$

学生关系 Student

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

课程关系 Course

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

选课关系 SC

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

【例 3-9】 检索计算机系学生的学号和姓名。

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

【例 3-10】 检索选修了 C1 课的学生信息。

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

【例 3-11】 检索不选 C1 课的学生信息。

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

【例 3-12】 检索选修了全部课的学生的学号。

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

【例 3-13】 插入学号为 200504 的学生选修了 C4 课、成绩为 88 分的选课记录。

$$SC \cup \{ '200504', 'C4', 88 \}$$

【例 3-14】 删除学生刘明亮选修的英语课。

$$SC - (\Pi_{Sno} (\sigma_{Sname='刘明亮'} (Student)) \bowtie SC \bowtie \Pi_{Cno} (\sigma_{Cname='英语'} (Course)))$$

【例 3-15】 检索在同一个系的学生的学号和姓名。

$$\Pi_{Sno, Sname} (Student \bowtie \delta_{Sno, Sname, Sage, Ssex \rightarrow Sno', Sname', Sage', Ssex'} (Student))$$

关系代数的综合应用

- ◆关系代数比较典型的实际例子是**ISBL**
(**information sysytem base language**) ,由**IBM**开发

关系演算

- ◆关系演算来源于数理逻辑的谓词演算
- ◆按照变元不同
 - ◆元组关系演算
 - ◆域关系演算
- ◆一般的用法是使用谓词给出查询结果需要满足的条件，系统完成查询操作，高度非过程化（关系代数需要指明运算顺序）

元组关系演算语言**ALPHA**

- ◆元组关系演算谓词表达式的对象是元组
- ◆通过谓词给出元组需要满足的条件
- ◆常用的谓词包括
 - ◆比较 等于 不等于 大于 小于 大于等于 小于等于
 - ◆逻辑 逻辑与 或非
 - ◆量词
 - ◆存在量词，全称量词
- ◆Alpha是一种概念的关系演算语言

三种关系运算是等价的

- ◆关系代数，元组关系演算，域关系演算在功能上是等价的
- ◆域关系演算和元组关系演算只是表达式的变量不同
- ◆关系代数的五种基本运算 **并 差 笛卡尔积 选择 投影** 是可以元组演算表达式表达

本章小节

- ◆ 关系模型
- ◆ 以表做为单一的数据结构
- ◆ 理论基础扎实
 - ◆ 关系代数
 - ◆ 关系演算
 - ◆ 元组关系演算
 - ◆ 域关系演算