# 绪论

数据库技术是信息系统的核心和基础

## 数据库系统概述

1. 数据库4个基本概念：

数据(data)：描述事物的符号记录。描述事物的符号记录可以是文字、图形、图像、音频、视频等数据与其语义是不可分的，例如93可以代表成绩，也可代表体重

数据库(database DB)：长期储存在计算机内、有组织的、可共享的大量数据的集合

数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度（数据的重复存储）、较高的数据独立性和易扩展性，并可为各种用户共享。

数据库管理系统(DBMS)：位于用户与操作系统之间的一层数据管理软件（属于系统软件）

数据库管理系统的主要功能：

* 数据定义功能：提供数据定义语言(DDL)，例如定义表
* 数据组织、存储和管理：确定组织数据的文件结构和存取方式
* 数据操纵功能：数据操纵语言(DML)，查询、插入、删除和修改等
* 数据库的事务管理和运行管理：数据库在建立、运行和维护时由DBMS统一管理和控制
* 数据库的建立和维护功能
* 其他功能：DBMS与网络中的其他软件的通信功能、两个DBMS系统的数据转换、异构数据库之间的互访和互操作

数据库系统(DBS)：硬件平台及数据库 + 软件(DBMS) + 人员构成

2. 数据库管理技术的产生和发展：

人工管理阶段 -> 文件系统阶段 -> 数据库系统阶段

人工管理阶段：应用程序管理数据、数据不具有独立性、数据不保存、数据不共享（面向应用程序）

文件系统阶段：文件系统管理数据、数据独立性查、数据长期保存、数据不共享（面向应用程序）

3. 数据库系统的特点：

数据结构化：数据库系统实现整体数据的结构化（面向全组织，不面向某一个应用程序），这是数据库的主要特征之一，也是数据库系统与文件系统的本质区别

数据的共享性高、冗余度低且易扩充：由于面向全组织，因此数据共享，可以减少数据冗余，节约存储空间，避免数据之间的不相容性与不一致性，使系统易于扩充

数据独立性高：数据独立性是由数据库管理系统提供的二级映像功能来保证的

* 物理独立性：指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。数据的物理存储改变了，应用程序不用改变。
* 逻辑独立性：指用户的应用程序与数据库的逻辑结构是相互独立的。 数据的逻辑结构改变了，用户程序也可以不变

数据由数据库管理系统统一管理和控制：DBMS提供的数据控制功能

* 数据的安全性保护：保护数据，以防止不合法的使用造成的数据的泄密和破坏
* 数据的完整性检查：将数据控制在有效的范围内，或保证数据之间满足一定的关系
* 并发控制：对多用户的并发操作加以控制和协调，防止相互干扰而得到错误的结果
* 数据库恢复：将数据库从错误状态恢复到某一已知的正确状态

## 数据模型

数据模型是对现实世界数据特征的抽象，数据模型是数据库系统的核心和基础

1. 数据模型分类：

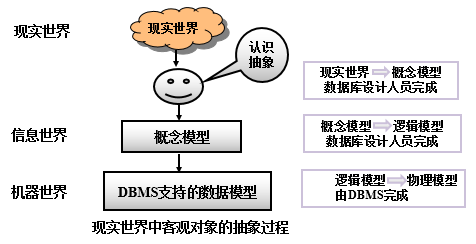
数据模型分为两类：第一类是概念模型，第二类是逻辑模型和物理模型

* 概念模型：按用户的观点来对数据和信息建模，主要用于数据库设计
* 逻辑模型：主要包括层次模型、网状模型、关系模型、面向对象数据模型、对象关系数据模型、半结构化模型，按照计算机系统的观点对数据建模，主要用于数据库管理系统的实现
* 物理模型：描述数据在系统内部的表示方式和存取方法，面向计算机系统的

客观对象的抽象过程：

（1）现实世界中的客观对象抽象为概念模型

（2）把概念模型转换为某一DBMS支持的数据模型



2. 数据模型的组成要素：

数据结构：描述数据库的组成对象，以及对象之间的联系

数据操作：对数据库中各种对象(型)的实例(值)允许执行的操作及有关的操作规则

数据的完整性约束条件：一组完整性规则

3. 关系模型：

* 关系模型中数据的逻辑结构是一张二维表，它由行和列组成
* 关系模型的数据操作主要包括查询、插入、更新、删除操作
* 关系模式的完整性约束包括实体完整性、参照完整性、用户自定义完整性

一些基本的关系术语：

|  |  |
| --- | --- |
| 关系术语 | 含义 |
| 关系 | 表 |
| 元组 | 表的一行 |
| 属性 | 表的一列 |
| 码 | 可以唯一确定一个元祖的属性组 |
| 域 | 一组具有相同数据类型的值的集合（取值范围） |
| 分量 | 元祖中的一个属性值 |
| 关系模式 | 关系名（属性1，属性2，……，属性n）  学生（学号，姓名，年龄，性别，系，年级） |

关系必须是规范化的：关系的每一个分量必须是一个不可分的数据项，表中不能还有表

## 数据库系统结构

1. 数据库系统模式的概念：

* 型(Type)：对某一类数据的结构和属性的说明
* 值(Value)：是型的一个具体赋值

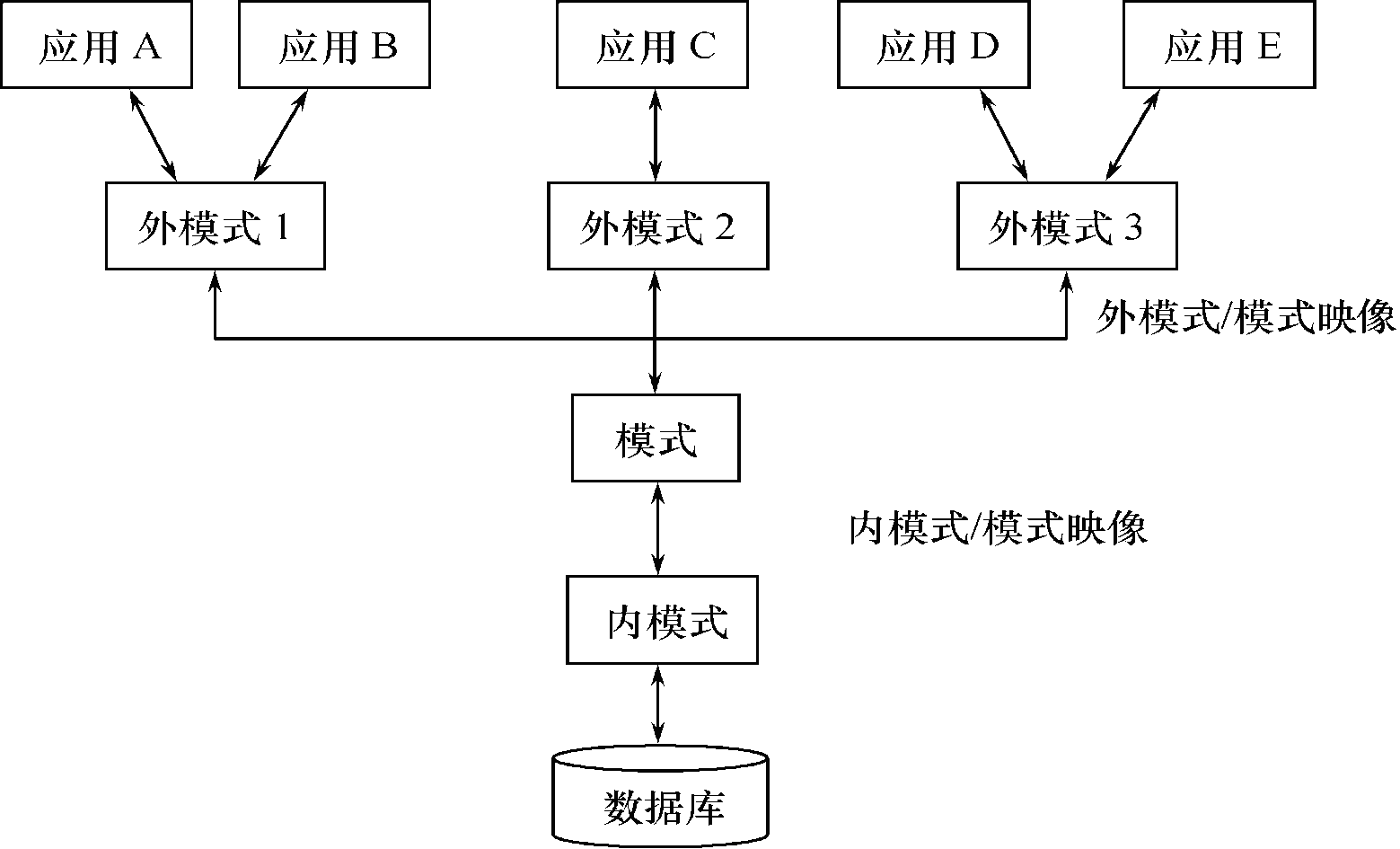
例如：

学生记录型：（学号，姓名，性别，系别，年龄，籍贯）

一个记录值：（201315130，李明，男，计算机，19，江苏）

2. 三级模式结构和二级映像功能：

数据库系统的三级模式结构是指数据库系统是由外模式、模式、内模式三级构成



（1）模式：（逻辑模式）数据库中全体数据的逻辑结构和特征的描述，一个数据库只有一个模式

（2）外模式：（子模式或用户模式）数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述，模式与外模式的关系：一对多，外模式与应用的关系：一对多

（3）内模式：（存储模式）数据物理结构和存储方式的描述，一个数据库只能由一个内模式

数据库的二级映像包括：外模式/模式映像、模式/内模式映像

* 外模式/模式映像：保证数据的逻辑独立性，当模式改变时，数据库管理员修改有关的外模式／模式映象，使外模式保持不变
* 模式/内模式映像：保证数据的物理独立性，当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式／内模式映象，使模式保持不变

# 关系数据库

## 关系数据结构及形式化定义

1. 关系：

域：是一组具有相同数据类型的值的集合--取值范围构成的集合

例如：整数、实数、介于某个取值范围的整数、{‘男’，‘女’}等

笛卡儿积：所有域的所有取值的一个组合

给定一组域D1,D2,...,Dn,允许其中某些域是相同的。D1,D2,…,Dn的笛卡尔积为：

D1×D2×…×Dn ＝｛（d1，d2，…，dn）｜di ∈ Di，i＝1，2，…，n｝

其中每一个元素(d1，d2，…，dn)叫作一个元组，元素中的每一个值di叫做一个分量，一个域允许的不同取值个数称为这个域的基数

-----------------------------------------------------------------------

例：给出三个域：

*D*1={ 张清玫，刘逸 } 导师集合 基数为2

*D*2={计算机专业，信息专业} 专业集合 基数为2

*D*3={李勇，刘晨，王敏} 研究生集合 基数为3

则*D*1，*D*2，*D*3的笛卡尔积为：（该笛卡儿积的基数为2x2x3 = 12）

*D*1×*D*2×*D*3 ＝

分量

{

(张清玫，计算机专业，李勇)，(张清玫，计算机专业，刘晨)，

元祖

(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：D1×D2×…×Dn的子集叫作在域D1，D2，…，Dn上的关系，表示为：

*R*：关系名

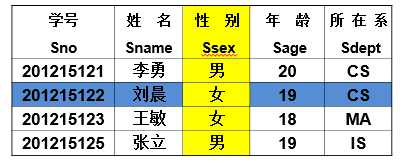
*n*：关系的目或度（Degree）--属性

关系实质上就是将广义笛卡尔积上无用的砍掉，有用的留下，我们把这个构成的集合叫做狭义笛卡尔积

**关系的术语：**

候选码：某一个属性组（有一个或多个）的值能够唯一地标识一个元组，而其子集不能，则称该属性组为候选码。（一个关系中可以有多个候选码）

主码：若一个关系有多个候选码，则选定其中一个为主码



假设上述该表的候选码可以为学号，可以为姓名，那么选择学号为主码。

主属性：只要改属性是候选码，就是主属性

非主属性：只要改属性不是候选码，就是非主属性

全码：如果一个关系中的所有属性都是候选码，那么我们称为全码

基本关系的性质

* 列是同质的（Homogeneous），就是列来自于同一个域
* 不同的列可出自同一个域，其中每一列称为一个属性，不同的属性要给予不同的属性名
* 列的顺序无所谓, 列的次序可以任意交换
* 任意两个元组的候选码不能相同
* 行的顺序无所谓，行的次序可以任意交换
* 分量必须取原子值（表中不能有表）

2. 关系模式：



* 关系模式是对关系的描述，是静态的、稳定的
* 关系是关系模式在某一时刻的状态或内容，是动态的、随时间不断变化的

关系模式可以形式化地表示为：R（U，D，DOM，F）

## 关系操作

常用的关系操作：选择、投影、并、差、笛卡尔积是5种基本操作

关系操作的特点：集合操作方式：操作的对象和结果都是集合，一次一集合的方式

## 关系完整性

关系的三类完整性约束：实体完整性、参照完整性、用户自定义完整性（其中实体完整性和参照完整性被称作关系的两个不变性）

实体完整性：若属性*A*是基本关系*R*的主属性，则属性*A*不能取空值—主属性不能为空

参照完整性：若属性（或属性组）*F*是基本关系*R*的外码，它与基本关系*S*的主码Ks相对应（基本关系*R*和*S*不一定是不同的关系），则对于*R*中每个元组在*F*上的值必须为：

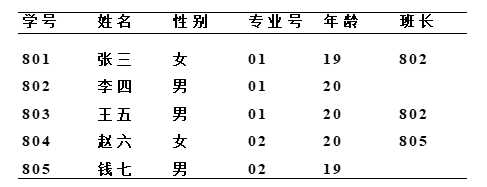
* 或者取**空值**（*F*的每个属性值均为空值）
* 或者等于*S*中某个元组的**主码值**

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，Ks是基本关系S的主码。如果F与Ks相对应，则称F是基本关系*R*的外码

-----------------------------------------------------------------------

[例 1] 学生（学号，姓名，性别，专业号，年龄，班长）



* “学号”是主码，“班长” 引用了本关系的“学号”
* “班长”必须是确实存在的学生的学号，班长是外码
* 学生关系既是参照关系也是被参照关系

[例2]

学生（学号、姓名、性别、专业号、年龄）

课程（课程号、课程名、学分）

选修关系(学号、课程号、成绩)

* “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

-----------------------------------------------------------------------

用户定义的完整性: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

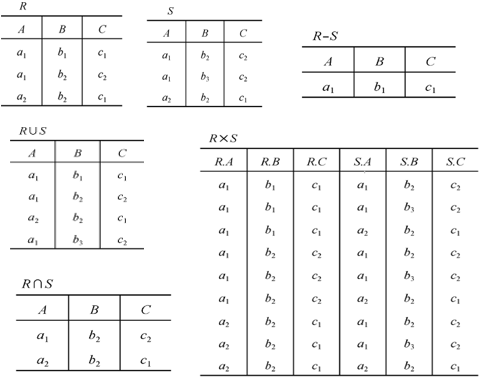
## 关系代数

1. 传统的集合运算

传统的集合运算包括：并、交、差、笛卡儿积

对于并、差、交三种运算来说的前提条件：

*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



## 专门的集合运算

先引入几个记号

（1） R，t∈R，t[Ai]

设关系模式为*R(A1，A2，…，An)*它的一个关系设为*R*

*t*∈*R*表示*t*是*R*的一个元组

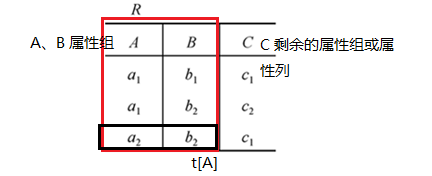
*t*[*Ai*]则表示元组*t*中相应于属性*Ai*的一个分量

（2） A，t[A]， A

若*A*={*Ai*1，*Ai*2，…，*Aik*}，其中*Ai*1，*Ai*2，…，*Aik*是*A*1，*A*2，…，*An*中的一部分，则*A*称为**属性列**或属性组。

*t[A]*=(*t*[*Ai*1]，*t*[*Ai*2]，…，*t*[*Aik*])表示元组*t*在属性列*A*上诸分量的集合。

*A*则表示{*A*1，*A*2，…，*An*}中去掉{*Ai*1，*Ai*2，…，*Aik*}后剩余的属性组。



（3） *t*r *t*s

*R*为*n*目关系，*S*为*m*目关系。 *t*r ∈*R*，*t*s∈*S*， *t*r *t*s称为元组的连接。

*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。

（4）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**定义为

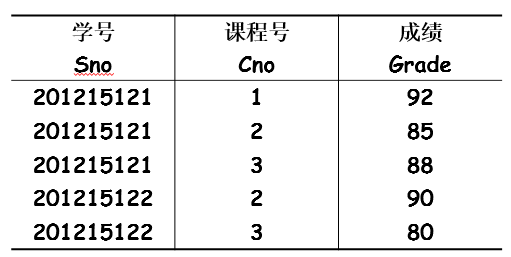


它表示R中属性组X上值为**x的诸元组在Z上分量的集合**。

**问题描述：**

学生关系Student、课程关系Course和选修关系SC





**选择（Selection）----**选择又称为限制（Restriction）

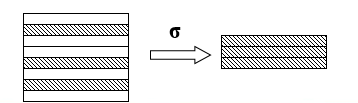
选择运算符的含义

在关系*R*中选择满足给定条件的诸元组

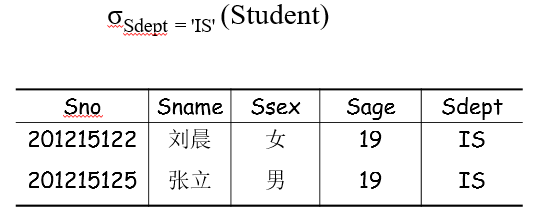
**σF(*R*) = {*t*|*t*∈*R*∧*F*(*t*)= '真'}**

**F：选择条件，是一个逻辑表达式，基本形式为： *X*1θ*Y*1**

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**



**[例1] 查询信息系（IS系）全体学生**



**投影（Projection）**

1）投影运算符的含义

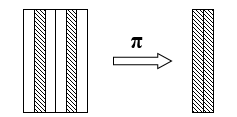
* 从*R*中选择出若干**属性列**组成新的关系

π*A*(*R*) = { *t*[*A*] | *t* ∈*R* }

*A：R*中的属性列

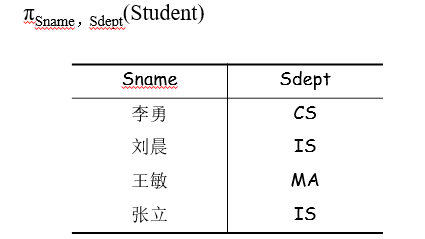
2）投影操作主要是从列的角度进行运算

* 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）

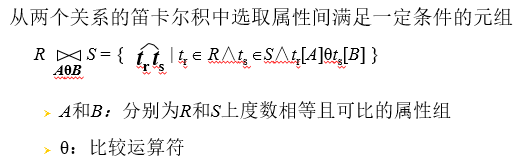


**[例2] 查询学生的姓名和所在系**

**即求Student关系上学生姓名和所在系两个属性上的投影**

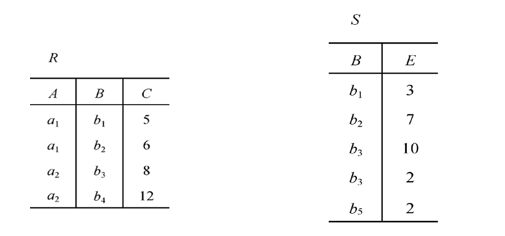


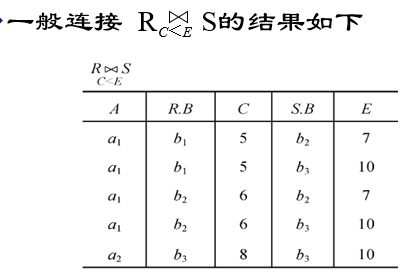
**连接（Join）----也称为θ连接**



连接运算从*R*和*S*的广义笛卡尔积*R*×*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

**[例3] 关系*R*和关系*S* 如下所示：**





**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：



**自然连接（Natural join）**

自然连接是一种特殊的**等值连接**

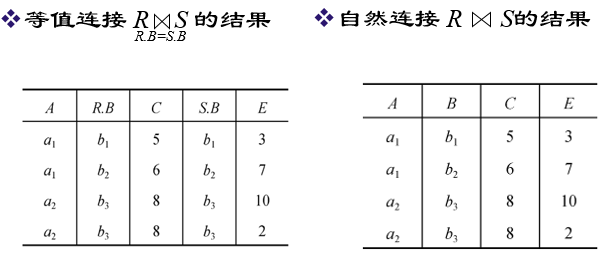
（1）两个关系中进行比较的分量必须是**相同的属性组**

（2）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*





**外连接**

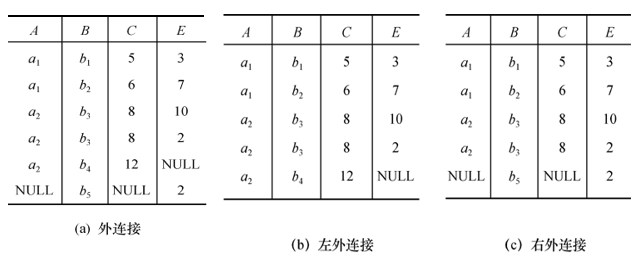
如果把舍弃的元组也保存在结果关系中，而在其他属性上填空值(Null)，这种连接就叫做外连接（OUTER JOIN）

**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 JOIN)

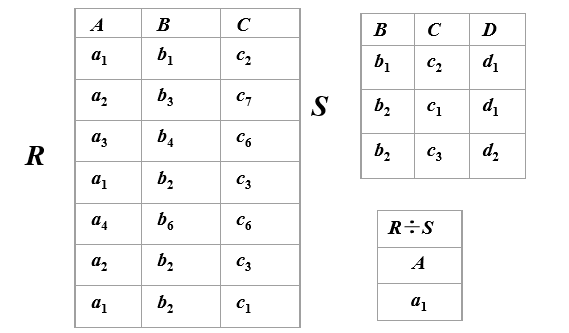


**除（Division）**

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*Yx*包含*S*在*Y*上投影的集合。

*R*÷*S* = {*t*r [*X*] | *t*r ∈ *R*∧πY (*S*) ⊆ *Yx* }

*Yx*：*x*在*R*中的象集，*x* = *t*r[*X*]



在关系R中，A可以取四个值{a1，a2，a3，a4}

a1的象集为 {(b1，c2)，(b2，c3)，(b2，c1)}

　a2的象集为 {(b3，c7)，(b2，c3)}

　a3的象集为 {(b4，c6)}

　a4的象集为 {(b6，c6)}

*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÷S ={a1}**