



第一篇 基础篇

第二章 关系数据库(1)

福州大学数计学院

程 烨

chengye@fzu.edu.cn

2013年3月11日

关系数据库简介

关系数据库发展过程中的几件大事：

- 美国数据系统语言研究会 (CODASYL) 于1962年发表的“**信息代数**”一文；

- 68年David Child在IBM7090机上实现了**集合**数据结构

- E. F. Codd从**1970**年起发表了一系列的论文

- **70年代末**的在IBM 370 研制出**实验系统**System R

美国加州大学伯克利分校研制出**实验系统**Ingres

- 从**80年代**逐步走向成熟

关系数据库简介（续）

- 关系数据库系统——支持关系模型的数据库系统
- 关系数据库应用数学方法来处理数据库中的数据
- 80年代后，关系数据库系统成为最重要、最流行的数据库系统。
- 典型商用系统：
ORACLE, SYBASE, DB2, SQLSERVER等

关系数据库简介（续）

- 关系模型的组成
 - 关系数据结构——（数据的表示，用关系来表示信息世界中的实体及其相互联系）
 - 关系操作集合——（数据使用：增、删、改、查，使用系统提供的数据库语言实现，它基于关系运算。）
 - 关系完整性约束——（数据的语义：有效，合理。允许定义三类完整性约束）

第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数(重点与难点)

2.5 小结

2.1 关系数据结构及形式化定义

❖ 2.1.1 关系

❖ 2.1.2 关系模式

❖ 2.1.3 关系数据库

2.1.1 关系

❖ 单一的数据结构----关系

- 信息世界的实体以及实体间的各种联系均用关系来表示

❖ 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

❖ 建立在集合代数的基础上

关系(续)

- ❖ 1. 域 (**Domain**)
- ❖ 2. 笛卡尔积 (**Cartesian Product**)
- ❖ 3. 关系 (**Relation**)

1. 域 (Domain)

❖ 域是一组具有相同数据类型的值的集合。

➤ 介于某个取值范围的整数

➤ 指定长度的字符串集合

➤ {‘男’， ‘女’ }

➤ 介于某个取值范围的日期

2. 笛卡尔积 (Cartesian Product)

❖ 1) 笛卡尔积—定义在域上的集合运算

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

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

- 所有域的所有取值的组合
- 不能重复

笛卡尔积（续）

例 给出三个域：

$D_1 = \text{SUPERVISOR} = \{\text{张清玫}, \text{刘逸}\}$

$D_2 = \text{SPECIALITY} = \{\text{计算机专业}, \text{信息专业}\}$

$D_3 = \text{POSTGRADUATE} = \{\text{李勇}, \text{刘晨}, \text{王敏}\}$

则 D_1 , D_2 , D_3 的笛卡尔积为： $D_1 \times D_2 \times D_3 =$

$\{(\text{张清玫}, \text{计算机专业}, \text{李勇}), (\text{张清玫}, \text{计算机专业}, \text{刘晨}),$
 $(\text{张清玫}, \text{计算机专业}, \text{王敏}), (\text{张清玫}, \text{信息专业}, \text{李勇}),$
 $(\text{张清玫}, \text{信息专业}, \text{刘晨}), (\text{张清玫}, \text{信息专业}, \text{王敏}),$
 $(\text{刘逸}, \text{计算机专业}, \text{李勇}), (\text{刘逸}, \text{计算机专业}, \text{刘晨}),$
 $(\text{刘逸}, \text{计算机专业}, \text{王敏}), (\text{刘逸}, \text{信息专业}, \text{李勇}),$
 $(\text{刘逸}, \text{信息专业}, \text{刘晨}), (\text{刘逸}, \text{信息专业}, \text{王敏})\}$

笛卡尔积（续）

❖ 2) 元组 (Tuple)

- 笛卡尔积中每一个元素 (d_1, d_2, \dots, d_n) 叫作一个 **n 元组 (n-tuple)** 或简称**元组(Tuple)**
- (张清玫, 计算机专业, 李勇)、(张清玫, 计算机专业, 刘晨)等都是元组

❖ 3) 分量 (Component)

- 笛卡尔积元素 (d_1, d_2, \dots, d_n) 中的每一个值 d_i 叫作一个**分量**
- 张清玫、计算机专业、李勇、刘晨等都是分量

笛卡尔积（续）

❖ 4) 基数（Cardinal number）

- 若 D_i ($i=1, 2, \dots, n$) 为有限集，其基数为 m_i ($i=1, 2, \dots, n$)，则 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M 为：
$$M = \prod_{i=1}^n m_i$$
- 在上例中，基数： $2 \times 2 \times 3 = 12$ ，即 $D_1 \times D_2 \times D_3$ 共有 $2 \times 2 \times 3 = 12$ 个元组

笛卡尔积（续）

❖ 5) 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。

在上例中，**12**个元组可列成一张二维表

表 2.1 D_1, D_2, D_3 的笛卡尔积

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

3. 关系 (Relation)

1) 关系

$D_1 \times D_2 \times \dots \times D_n$ 的有限子集叫作在域 D_1, D_2, \dots, D_n 上的关系, 表示为

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

- R : 关系名
- n : 关系的目或度 (Degree)

关系（续）

2) 元组

关系中的每个元素是关系中的元组，通常用 t 表示。

3) 单元关系与二元关系

当 $n=1$ 时，称该关系为单元关系（Unary relation）

当 $n=2$ 时，称该关系为二元关系（Binary relation）

关系（续）

4) 关系的表示

关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域

表22 SAP关系

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

关系（续）

5) 属性

- 关系中不同列可以对应相同的域
- 为了加以区分，必须对每列起一个名字，称为属性（**Attribute**）
- n 目关系必有 n 个属性

关系（续）

6) 候选码 (Candidate key)

关系中能唯一地标识一个元组值的最小属性集，则称该属性集为候选码(码)。

如: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)

关系中可以有多个候选码

如:学生 (学号, 姓名, 性别, 专业号, 年龄)

如果学生不重名,姓名也可以是候选码

关系（续）

候选码(续)

主码:若一个关系有多个候选码，则选定其中一个为
主码（Primary key）如选学号为主码

包含在任何一个候选码中的属性称为主属性（
Prime attribute）如学号与姓名为主属性

不包含在任何候选码中的属性称为非码（主）属性
（Non-key attribute）

关系（续）

候选码(续)

全码 (All-key)

在最极端的情况下，关系的所有属性构成这个关系的候选码，称为全码 (All-key)

如:供应 (供应商号, 工程号, 零件号)

候选码是:供应商号+工程号+零件号

主属性为:供应商号,工程号,零件号

非主属性为空

关系（续）

❖ 一般来说， D_1, D_2, \dots, D_n 的笛卡尔积是没有实际语义的。只有它的某个子集才有实际含义

例：表2.1 的笛卡尔积中取出有实际意义的元组来构造关系

关系：SAP (SUPERVISOR, SPECIALITY, POSTGRADUATE)

假设：导师与专业：n:1，导师与研究生：1:n，

研究生与专业：n:1，研究生不会重名

于是：SAP关系可以包含三个元组

{ (张清玫, 计算机专业, 李勇),
(张清玫, 计算机专业, 刘晨),
(刘逸, 信息专业, 王敏) }

POSTGRADUATE可作为SAP关系的主码。

关系（续）

7) 三类关系

基本关系（基本表或基表）

实际存在的表，是实际存储数据的逻辑表示

查询表

查询结果对应的表（临时表）

视图表

由基本表或其他视图表导出的表，是虚表，
不对应实际存储的数据

关系（续）

- 关系是笛卡尔积的有限子集。无限关系在数据库系统中是无意义的。
- 由于笛卡尔积不满足交换律，即

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

但关系满足交换律，即

$$(d_1, d_2, \dots, d_i, d_j, \dots, d_n) = (d_1, d_2, \dots, d_j, d_i, \dots, d_n) \quad (i, j = 1, 2, \dots, n)$$

解决方法：为关系的每个列附加一个属性名以取消关系元组的有序性。

关系（续）

8)基本关系的六个性质

- ① 列是同质的（**Homogeneous**）
- ② 不同的列可出自同一个域
- ③ 列的顺序无所谓，列的次序可以任意交换
- ④ 任意两个元组的候选码不能相同
- ⑤ 行的顺序无所谓，行的次序可以任意交换

基本关系的性质(续)

⑥ 分量必须取原子值

- 每一个分量都必须是不可分的数据项
这是规范条件中最基本的一条

表2.3 非规范化关系

SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	信息专业	李勇	刘晨
刘逸	信息专业	王敏	

小表

2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

1. 什么是关系模式

- ❖ 关系模式（**Relation Schema**）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

2. 定义关系模式

关系模式可以形式化地表示为：

$R(U, D, \text{dom}, F)$

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 属性向域的映像集合

F 属性间的数据依赖关系集合

定义关系模式 (续)

关系模式通常可以简记为

$R(U)$ 或 $R(A_1, A_2, \dots, A_n)$

- **R** : 关系名
- **A_1, A_2, \dots, A_n** : 属性名

注：域名及属性向域的映像常常直接说明为
属性的类型、长度

3. 关系模式与关系

- ❖ 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



2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

1.关系数据库

❖ 关系数据库

在一个给定的应用领域中，所有实体及实体之间联系所对应的关系的集合构成一个关系数据库

2. 关系数据库的型与值

- ❖ 关系数据库也有型和值之分
- ❖ 关系数据库的型称为关系数据库模式是对关系数据库的描述，是关系模式的集合。关系数据库模式包括：
 - 若干域的定义
 - 在这些域上定义的若干关系模式
- ❖ 关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常简称为关系数据库

第二章 关系数据库

2.1 关系模型概述

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 小结

2.2.1 基本关系操作

❖ 常用的关系操作

- 查询

- 选择、投影、连接、除、并、交、差、笛卡尔积

- 数据更新

- 插入、删除、修改

- 查询的表达能力是其中最主要的部分

- 选择、投影、并、差、笛卡尔积是**5种基本操作**，其他操作可由基本操作来定义

基本关系操作（续）

❖ 关系操作的特点

- 集合操作方式，即操作的对象和结果都是集合，也成为一次一集合的方式。

❖ 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

关系数据库语言的分类（续）

❖ 关系演算语言：用谓词逻辑来表达查询要求

■ 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 (Structured Query Language)

关系数据库语言的分类（续）

- 关系数据语言的特点
 - 关系语言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、元组关系演算和域关系演算三种语言在表达能力上完全等价

第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 小结

2.3 关系的完整性

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.1 关系的三类完整性约束

- 在数据库中数据完整性是指**保证数据正确**的特性。它包括两方面的内容：
 - 与现实世界中应用需求的**数据**的相容性和正确性；
 - 数据库内**数据之间**的相容性和正确性。
- 在关系数据模型中一般将数据完整性分为**三类**
 - 实体完整性
 - 参照完整性
 - 用户定义完整性

2.3.1 关系的三类完整性约束

实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称作是关系的两个不变性，由关系系统自动支持。

用户定义的完整性是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。

2.3 关系的完整性

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.2 实体完整性

实体完整性规则(Entity Integrity)

如果A是关系模式R(U)的主码(键)，则A不能取空值。

注意：实体完整性规则规定关系的所有主码都不能取空值。

2.3 关系的完整性

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

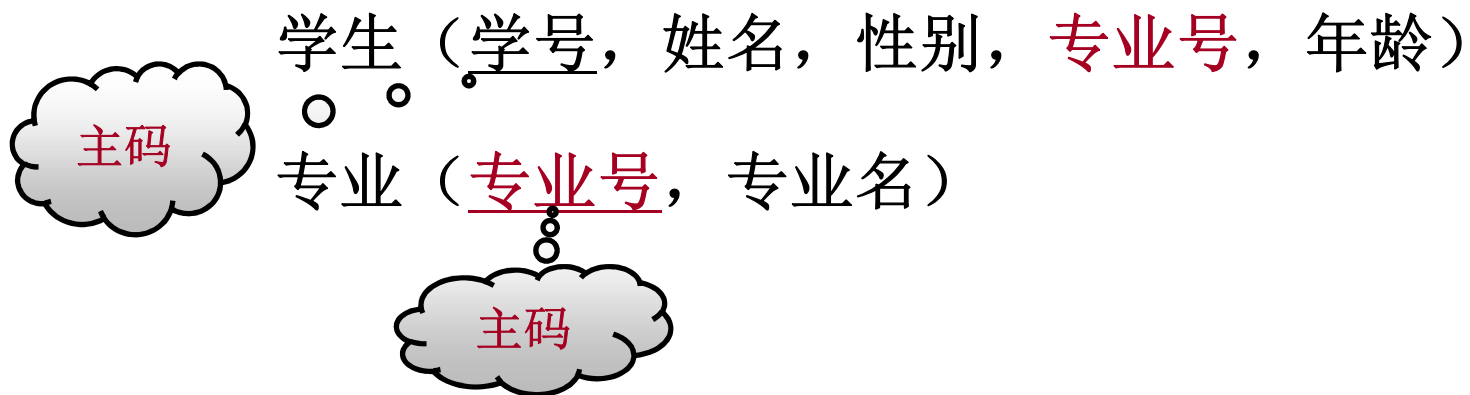
2.3.3 参照完整性

2.3.4 用户定义的完整性

1. 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、专业实体以及专业与学生间的一对多联系



关系间的引用(续)

学 号	姓 名	性 别	专 业 号	年 龄
8 0 1	张 三	女	0 1	1 9
8 0 2	李 四	男	0 1	2 0
8 0 3	王 五	男	0 1	2 0
8 0 4	赵 六	女	0 2	2 0
8 0 5	钱 七	男	0 2	1 9

专业号	专业名
01	信息
02	数学
03	计算机

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)

学生

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

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

学生选课

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

关系间的引用(续)

例3 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
805	钱七	男	02	19	

2. 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
- ❖ 基本关系 S 称为被参照关系或目标关系

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** 必须定义在同一组域
- ❖ 外码并不一定要与相应的主码同名

外码(续)

如 学生基础信息关系S1 (SNO , SNAME, 。 。)

学生详细信息关系S2 (SNO , SADD, 。 。)

则关系S1, S2中, SNO是主码也是类似外码特性的属性。

3. 参照完整性规则

规则2.2 参照完整性规则

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 K_s 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

关系的完整性(续)

2.3.1 关系的三类完整性约束

2.3.2 实体完整性

2.3.3 参照完整性

2.3.4 用户定义的完整性

2.3.4 用户定义的完整性


- ❖ 用户定义的完整性是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
- ❖ 关系模型应提供定义和检验这类完整性的机制，以便使用统一的系统的方法处理它们，而不要由应用程序承担这一功能

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}

- 
- 执行插入操作时检查完整性
 - 执行插入操作时需要分别检查实体完整性规则、参照完整性规则 and 用户定义完整性规则。
 - 执行删除操作时检查完整性
 - 执行删除操作时一般只需要检查参照完整性规则。
 - 执行更新操作时检查完整性
 - 执行更新操作可以看作是先删除旧的元组，然后再插入新的元组。所以执行更新操作时的完整性检查综合了上述两种情况。

第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 小结

2.4 关系代数

❖ 2.4.1 概述

❖ 2.4.2 传统的集合运算

交 (\cap)、并 (\cup)、差 ($-$)、笛卡尔积 (\times)

❖ 2.4.3 专门的关系运算

选择 (σ)、投影 (Π)、连接 (\Join)

和除 (或商 \div) (难点)

❖ 2.4.4 关系代数表达式(重点与难点)

2.4.1 概 述

(1) 关系代数

一种抽象的查询语言

用对关系的运算来表达查询

(2) 运算的三要素

- 运算对象
- 运算符
- 运算结果

概述(续)

(3) 关系代数运算的三个要素

运算对象：关系

运算结果：关系

运算符：集合运算符、专门的关系运算符、
算术比较符和逻辑运算符

概述(续)

表2.4 关系代数运算符

运算符		含义	运算符		含义
集合运算符	\cup	并	比较运算符	$>$	大于
	$-$	差		\geq	大于等于
	\cap	交		$<$	小于
	\times	笛卡尔积		\leq	小于等于
				$=$	等于
				\neq	不等于

概述(续)

表2.4 关系代数运算符（续）

运算符	含义		运算符	含义	
专门的关 系运算符	σ	选择	逻辑运算 符	\neg	非
	π	投影		\wedge	与
	\bowtie	连接		\vee	或
	\div	除			

概述(续)

(4) 关系代数运算的分类

- 集合运算
 - 将关系看成元组的集合
 - 运算是从关系的“水平”方向即行的角度来进行
- 专门的关系运算
 - 不仅涉及行而且涉及列
- 比较运算
 - 辅助专门的关系运算进行操作
- 逻辑运算
 - 辅助专门的关系运算进行操作

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

几个记号

例：关系模式表示为：S(学号，姓名，专业，年级)

关系 S 表示一个学生关系表（如含07级学生数据）

$t \in S$ 表示 t 是一个元组，是一个具体的学生数据。

$t[\text{姓名}]$ 则表示某一个学生的姓名，是元组 t 的一个分量。

几个记号

(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 称为属性集或属性组。

$t[A] = (t[A_{i1}], t[A_{i2}], \dots, t[A_{ik}])$ 表示元组 t 在属性集 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。

几个记号

例：关系模式表示为：S(学号，姓名，专业，年级)

U: {学号，姓名，专业，年级}

$A = \{\text{学号}, \text{姓名}\}$ 表示关系S属性集U的一个子集。

$t[A]$ 则表示某一个学生的学号与姓名，是元组t在A上的几个分量的集合。

$\bar{A} = \{\text{专业}, \text{年级}\} = U - A$ 表示关系S属性集U的一个子集

几个记号

(3) $t_r \frown t_s$ 连接

R 为 n 目关系, S 为 m 目关系。

$t_r \in R, t_s \in S$, $t_r \frown t_s$ 称为元组的**连接**。

$t_r \frown t_s$ 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。

几个记号

R

t_r

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

$R \times S$
 t_r t_s

S

t_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

几个记号

(4) 象集 Z_x

给定一个关系 $R(X, Z)$ ， X 和 Z 为属性集。

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 中属性集 X 上值为 x 的诸元组在 Z 上分量的集合

几个记号

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

关系 $R(X, Z)$,
设: $X=A, Z=U-X=\{B, C\}$

设: $t[X] = a_1$ 时,

a_1 在 R 中的象集为:

$$Z_{a_1} = \{ t[Z] \mid t \in R \wedge t[X] = a_1 \}$$
$$= \{ (b_1, c_2), (b_2, c_3), (b_2, c_1) \}$$

几个记号

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

象集举例

左图中

❖ 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

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

2.4 关系代数

❖ 2.4.1 概述

❖ 2.4.2 传统的集合运算

交 (\cap)、并 (\cup)、差 ($-$)、笛卡尔积 (\times)

❖ 2.4.3 专门的关系运算

选择 (δ)、投影 (Π)、连接 (\bowtie) 和除 (商 \div)

❖ 2.4.4 关系代数表达式

2.4.2 传统的集合运算

传统的集合运算——运算将关系看成元组的集合，运算是从关系的行的角度进行。

设R和S是n元关系，而且相应的属性取自同一个域，则交（ \cap ）、并（ \cup ）、差（ $-$ ）操作定义如下：

1. 并 (Union)

❖ 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

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

并(续)

如R为07(1)(2)班学生登记表,

S为07(3)班学生登记表。

则: $R \cup S$ 为07级全体学生登记表。

相应的 关系模式表示为:

Student-R (姓名, 学号, 年级, 专业, 系)

Student-S (姓名, 学号, 年级, 专业, 系)

Student- $R \cup S$ (姓名, 学号, 年级, 专业, 系)

并(续)

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
$a1$	$b1$	$c1$
$a1$	$b2$	$c2$
$a1$	$b3$	$c2$
$a2$	$b2$	$c1$

2. 差 (Difference)

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

如上例中 R 为07级各班学生登记表；

S 为07软件专业学生登记表

$R - S$ 为07级各班中非软件专业的学生登记表。

差(续)

R

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

R-S

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b1</i>	<i>c1</i>

S

<i>A</i>	<i>B</i>	<i>C</i>
<i>a1</i>	<i>b2</i>	<i>c2</i>
<i>a1</i>	<i>b3</i>	<i>c2</i>
<i>a2</i>	<i>b2</i>	<i>c1</i>

3. 交 (Intersection)

❖ R 和 S

- 具有相同的目 n
- 相应的属性取自同一个域

❖ $R \cap S$

- 仍为 n 目关系，由既属于 R 又属于 S 的元组组成

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

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

如 R 为07级学生登记表； S 为软件专业学生登记表

$R \cap S$ 为07级软件专业的学生登记表。

$R - S$ ：为07级非软件专业的学生登记表。

$R \cap S = R - (R - S)$ 为07级软件专业的学生登记表

交 (续)

R

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

$R \cap S$

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

S

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

4. 笛卡尔积 (Cartesian Product)

- ❖ 严格地讲应该是广义的笛卡尔积 (Extended Cartesian Product)
- ❖ R —— n 目关系, k_1 个元组
- ❖ S —— m 目关系, k_2 个元组
- ❖ $R \times S$
 - 列: ($n+m$) 列元组的集合
 - 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
 - 行: $k_1 \times k_2$ 个元组
 - $R \times S = \{\hat{t}_r t_s \mid t_r \in R \wedge t_s \in S\}$

广义笛卡尔积 (续)

R

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

S

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

$R \times S$

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

2.4 关系代数

❖ 2.4.1 概述

❖ 2.4.2 传统的集合运算

交 (\cap)、并 (\cup)、差 ($-$)、笛卡尔积 (\times)

❖ 2.4.3 专门的关系运算

选择 (δ)、投影 (Π)、连接 (\bowtie) 和除 (或商 \div)

❖ 2.4.4 关系代数表达式

1. 选择 (Selection)

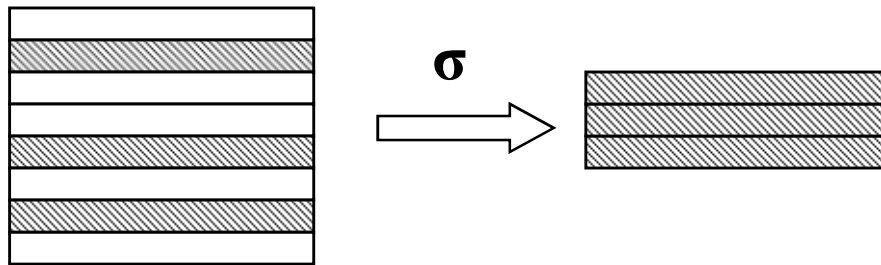
专门的关系运算——运算不仅涉及关系的行也涉及列，有时需要比较与逻辑运算来辅助进行操作。

1、选择操作 (Selection)：选择操作在关系R中选择满足给定条件的元组。

设R为一个n元关系，F是一个形如 $X_1 \theta Y_1$ 的逻辑表达式，F由逻辑运算符连接各算术表达式组成，R的选择操作定义为 $\delta_F(R) = \{ t \mid t \in R \wedge F(t) \}$ 。

选择（续）

- ❖ 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



- 举例：R为07级学生登记表

$\sigma_{\text{专业}='软件'}(R)$ 表示：

从07级学生登记表中选出07级软件专业学生。

2. 投影 (Projection)

2、投影操作 (Projection)：关系R上的投影操作是从关系R中选择出若干属性列，组成新的关系。

设R为一个n元关系，R的投影操作定义为：

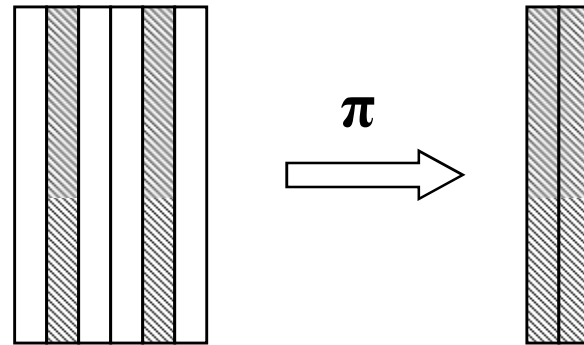
$$\Pi_A(R) = \{t[A] \mid t \in R\},$$

其中：A为R属性集的子集，

t[A]表示元组t中对应属性子集A的分量。

2. 投影 (Projection)

❖ 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、查询学生关系S中学生都来自哪些系

$\pi_{\text{系名}}(S)$:

结果:

系名
CS
IS
MA

选择和投影运算举例:

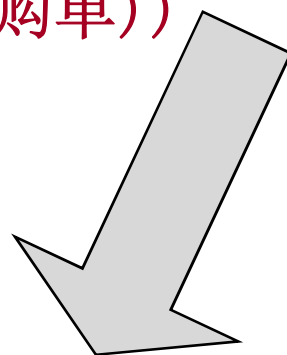
订购单关系

从订购单关系中，
选取出E3职工所经手
的订购单号和与之相
关的供应商号。

职工号	供应商号	订购单号	订购日期
E3	S7	OR67	2002/06/23
E1	S4	OR73	2002/07/28
E7	S4	OR76	2002/05/25
E6	S6	OR77	2002/06/19
E3	S4	OR79	2002/07/29
E1	S6	OR80	2002/06/22
E3	S6	OR90	2002/07/13
E3	S3	OR91	2002/10/27

π 供应商号, 订购单号 (σ 职工号='E3', (订购单))

供应商号	订购单号
S7	OR67
S4	OR79
S6	OR90
S3	OR91



3. 连接 (Join)

3、连接 (Join)：两个关系的连接操作是从两个关系的笛卡尔乘积中选取属性间满足一定条件的元组。

设： R 是 n 元关系， S 是 m 元关系，

A ， B 分别为 R 和 S 上可比的属性集

θ 是算术运算 $\{=, \neq, <, >, \leq, \geq\}$

$$\begin{aligned} R \bowtie_{A \theta B} S &= \{rs \mid r \in R \wedge s \in S \wedge (r[A] \theta s[B])\} \\ &= \delta_{A \theta B}(R \times S) \end{aligned}$$

如 **R**

A	B
1	2
4	5
7	8

R × **S** :

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

R ∞_{**B**<**C**} **S** :

S:

C	D
3	1
6	2

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

则属性**B**，**C**称为连接属性，**B**<**C**为连接条件

连接(续)

- 两类常用连接运算

- 等值连接

- 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：

$$\mathbf{R} \bowtie_{A=B} \mathbf{S} = \{\mathbf{rs} \mid \mathbf{r} \in \mathbf{R} \wedge \mathbf{s} \in \mathbf{S} \wedge (\mathbf{r}[A] = \mathbf{s}[B])\} = \boldsymbol{\delta}_{A=B}(\mathbf{R} \times \mathbf{S})$$

连接(续)

- 自然连接
 - 什么是自然连接
 - 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

$$R \bowtie S = \prod_{\text{Att}(R) \cup (\text{Att}(S) - \{B\})} \delta_{R.B=S.B} (R \times S)$$

连接(续)

❖ 举例

<i>A</i>	<i>B</i>	<i>C</i>
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

R

<i>B</i>	<i>E</i>
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

S

连接(续)

一般连接 $R \bowtie_{C <_E} S$ 的结果如下:

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

连接(续)

等值连接 $R \bowtie S$ 的结果如下: 自然连接 $R \bowtie S$ 的结果如下:
 $R.B=S.B$

<i>A</i>	<i>R.B</i>	<i>C</i>	<i>S.B</i>	<i>E</i>
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

<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

连接(续)

自然连接 $R \bowtie S$ 的结果如下:

<i>A</i>	<i>B</i>	<i>C</i>
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

R

<i>B</i>	<i>E</i>
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

S

<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2

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 JOIN**)。

连接(续)

下图是关系 R 和关系 S 的外连接

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

R

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

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
a_2	b_4	12	NULL
NULL	b_5	NULL	2

(a) 外连接

连接(续)

图(b)是关系 R 和关系 S 的左外连接,图(c)是右外连接

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

(b) 左外连接

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
NULL	b_5	NULL	2

(c) 右外连接

象集的例子

职工号	供应商号	订购单号	订购日期
E3	S7	OR67	2002/06/23
E1	S4	OR73	2002/07/28
E7	S4	OR76	2002/05/25
E6	S6	OR77	2002/06/19
E3	S4	OR79	2002/07/29
E1	S6	OR80	2002/06/22
E3	S6	OR90	2002/07/13
E3	S3	OR91	2002/10/27

- 设有如右上表所示的订购单关系，把它命名为 R ，并且进一步设 X 为属性职工号， Y 为属性集 {供应商号, 订购单号, 订购日期}，
则当 X 取值为 E3 时

$$Y_x = \{ (S7, OR67, 2002/06/23), (S4, OR79, 2002/07/29), (S6, OR90, 2002/07/13), (S3, OR91, 2002/10/27) \}$$

$$= \Pi_{\text{供应商号, 订购单号, 订购日期}} (\sigma_{\text{职工号}='E3'}(R))$$

4. 除 (Division)

给定关系 $R(X, Y)$ 和 $S(Y, Z)$ ，其中 X, Y, Z 为属性集。

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：

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

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

4. 除 (Division)

仓库号	供应商号
WH1	S1
WH1	S2
WH1	S3
WH2	S3
WH3	S1
WH3	S2
WH5	S1
WH5	S2
WH5	S4
WH6	S2

被除关系

÷

仓库号
WH1
WH3
WH5

除关系

=

供应商号
S1
S2

商关系

- 它的含义是：至少向WH1、WH3、WH5供货的供应商号。

4. 除 (Division)

[例6] 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**}

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**a₁**的象集包含了**S**在(**B**, **C**)属性组上的投影

所以 $R \div S = \{a_1\}$

4. 除 (Division)

以学生-课程数据库为例 (P60)

[例7] 查询至少选修1号课程和3号课程的学生号

首先建立一个临时关系 K :

Cno
1
3

然后求: $\pi_{\text{Sno}, \text{Cno}}(\text{SC}) \div K$

4. 除 (Division)

❖ 例 7续 $\pi_{Sno,Cno}(SC)$

95001象集{1, 2, 3}

95002象集{2, 3}

$K=\{1, 3\}$

于是: $\pi_{Sno,Cno}(SC) \div K=\{95001\}$

Sno	Cno
95001	1
95001	2
95001	3
95002	2
95002	3

2.4.4 关系代数表达式

前述八种关系代数运算，其中并、差、笛卡尔乘积、选择、投影五种为基本运算，另外三种运算：交、连接和除可用五种基本运算来表达。

在关系代数中，关系运算经有限次复合而成的式子称关系代数表达式。

在关系代数中，采用关系代数表达式描述一个关系查询。ISBL是典型的关系代数查询语言。

2.4.4 关系代数表达式

例：设S为学生关系S(S#, SN, ...)

SC为学生选课关系SC(S#, C#, ...)

C为课程关系 C(C#, ...)

查选修了全部课程的学生号和**学生名**

- 1、 $\Pi_{C\#}(C)$ 或 $\Pi_{C\#}(SC)$ — 从**课程或选课**关系中取所有课程号集合
- 2、 $\Pi_{S\#, C\#}(SC)$ — 从关系SC取属性S#、C#形成新关系。
- 3、 $\Pi_{S\#, C\#}(SC) \div \Pi_{C\#}(C)$ — 列出选修了全部课程的学号
- 4、 $(\Pi_{S\#, C\#}(SC) \div \Pi_{C\#}(C)) \bowtie \Pi_{S\#, SN}(S)$

2.4.4 关系代数表达式

例： 设S为学生关系S(S#, SN, ...)

SC为学生选课关系SC(S#, C#, ...)

C为课程关系 C(C#, ...)

查所学课程包含学生S3所学所有课程的学生学号。

- 1、 $\Pi_{C\#} (\sigma_{S\#='S3'} (SC))$ ——学生S3所学课程
- 2、 $\Pi_{S\#, C\#} (SC)$ — 从关系SC取属性S#、C#形成新关系。
- 3、 $\Pi_{S\#, C\#} (SC) \div \Pi_{C\#} (\sigma_{S\#='S3'} (SC))$

<u>仓库号</u>	城 市	面 积
------------	-----	-----

仓库

仓库号	<u>职工号</u>	工 资
-----	------------	-----

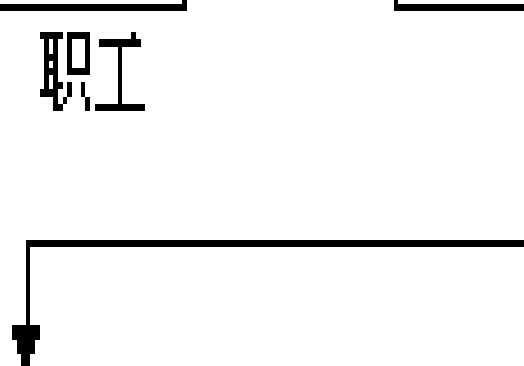
职工

<u>供应商号</u>	供应商名	地 址
-------------	------	-----

供应商

职工号	供应商号	<u>订购单号</u>	订购日
-----	------	-------------	-----

订购单



关系代数表达式

例：以上图数据库为例，用关系代数完成以下检索：
检索和北京的所有供应商**都有**业务联系的职工的工资。

1. $\pi_{\text{供应商号}}(\sigma_{\text{地址}='北京'}(\text{供应商}))$

2. $\pi_{\text{职工号}, \text{供应商号}}(\text{订购单}) \div \pi_{\text{供应商号}}(\sigma_{\text{地址}='北京'}(\text{供应商}))$

3. $\pi_{\text{职工号}, \text{工资}}(\text{职工} \bowtie (\pi_{\text{职工号}, \text{供应商号}}(\text{订购单}) \div \pi_{\text{供应商号}}(\sigma_{\text{地址}='北京'}(\text{供应商}))))$

关系代数表达式

P74第5题: 设有以下关系模式:

供应商S (SNO, SNAME, STATUS, CITY)

零件P (PNO, PNAME, COLOR, WEIGHT)

工程J (JNO, JNAME, CITY)

供应情况SPJ (SNO, PNO, JNO, QTY)

关系代数表达式

(4) 查询**没有使用**天津供应商生产的红色零件的工程号

1. 所有工程号: $\Pi_{JNO}(\text{SPJ})$

2. 使用天津供应商生产的红色零件的工程号:

$$A = \Pi_{JNO}(\Pi_{SNO}(\delta_{CITY='天津'}(S)) \bowtie \Pi_{SNO,PNO,JNO}(\text{SPJ}) \bowtie \Pi_{PNO}(\delta_{COLOR='红'}(P)))$$

3. $\Pi_{JNO}(\text{SPJ}) - A$

关系代数表达式

(5) 查询至少用了供应商S1所提供的全部零件的工程号

1. 被除关系: $\Pi_{JNO, PNO}(SPJ)$

2. 除关系:

$\Pi_{PNO}(\delta_{SNO='S1'}(SPJ))$

3. $\Pi_{JNO, PNO}(SPJ) \div \Pi_{PNO}(\delta_{SNO='S1'}(SPJ))$

第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系的完整性

2.4 关系代数

2.5 小结

小结

- ❖ 关系数据库系统与非关系数据库系统的区别是，关系系统只有“表”这一种数据结构；而非关系数据库系统还有其他的数据结构，以及对这些数据结构的操作。
- ❖ 代数方式表达的关系语言：关系代数