

数据库系统概论

An Introduction to Database System

第二章 关系模型

第2章 关系数据库

2.1 关系模型的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 关系演算

本章小结

2.1 关系模型的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

2.1.4 关系模型的存储结构

2.1.1 关系

❖ 单一的数据结构----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（续）

1. 域（**domain**）
2. 笛卡儿积（**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。例：

- 整数的集合
- 实数的集合
- 介于某个取值范围的整数
- 指定长度的字符串集合
- {‘男’， ‘女’ }
-

集合中值（不同值）的个数称为域的基数

2.笛卡尔积 (Cartesian Product)

❖ 笛卡尔积：域上的一种集合运算。

给定一组域 D_1, D_2, \dots, D_n ，允许其中某些域是相同的。

D_1, D_2, \dots, D_n 的笛卡尔积为：

元组 $D_1 \times D_2 \times \dots \times D_n =$

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

■ 所有域取值的任意组合

■ 笛卡尔积可以看着是关系的“域”。

笛卡尔积（续）

例如，给出3个域：

- ❖ **D1=导师集合SUPERVISOR= { 张清玫,刘逸 } 基数2**
- ❖ **D2=专业集合SPECIALITY= { 计算机专业,信息专业 } 基数2**
- ❖ **D3=研究生集合POSTGRADUATE={李勇,刘晨,王敏} 基数3**
- ❖ **D1， D2， D3的笛卡尔积为**

笛卡尔积（续）

❖ $D1 \times D2 \times D3 = \{$

(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) }

❖ 该笛卡尔积的基数为 $2 \times 2 \times 3 = 12$

笛卡儿积（续）

❖ 元组（tuple）

- 笛卡儿积中每一个元素 (d_1, d_2, \dots, d_n) 叫作一个n元组（n-tuple）或简称元组
- (张清玫, 计算机科学与技术, 李勇)、
- (张清玫, 计算机科学与技术, 刘晨) 等都是元组

❖ 分量（Component）

- 笛卡儿积元素 (d_1, d_2, \dots, d_n) 中的每一个值 d_i 叫做一个分量
- 张清玫、计算机科学与技术、李勇、刘晨等都是分量

笛卡尔积（续）

❖ 基数（Cardinal number）

- 若 D_i ($i=1, 2, \dots, n$) 为有限集，其基数（该域允许的不同取值个数）为 m_i ($i=1, 2, \dots, n$)，则 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M 为：

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

❖ 笛卡尔积的表示方法

- 笛卡尔积可表示为一张二维表
- 表中的每行对应一个元组，表中的每列对应一个域

3. 关系 (relation)

- ❖ 关系模型中 D_1, D_2, \dots, D_n 的笛卡儿积一般没有实际语义，只有某个真子集才有实际含义
- ❖ 表2.1的笛卡儿积中许多元组是没有意义的
 - 在学校中一个专业方向有多个导师，而一个导师只在一个专业方向带研究生；
 - 一个导师可以带多名研究生，而一名研究生只有一个导师，学习某一个专业。
 - 表2.1中的一个子集才是有意义的，才可以表示导师与研究生的关系，把该关系取名为**SMP**。

关系（续）

❖ D_1, D_2, \dots, D_n 的笛卡尔积的某个子集才有实际含义

例：表2.1 的笛卡尔积没有实际意义

取出有实际意义的元组来构造关系

关系： **SMP(SUPERVISOR, MAJOR, POSTGRADUATE)**

表2.2 导师-研究生关系SMP

SUPERVISOR	MAJOR	POSTGRADUATE
张清玫	计算机科学与技术	李勇
张清玫	计算机科学与技术	刘晨
刘逸	信息管理与信息系统	王敏

关系（续）

（1）关系

$D_1 \times D_2 \times \dots \times D_n$ 的 **子集**（笛卡尔积中具有某一方面意义的那些元组）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（ d_1, d_2, \dots, d_n ）叫作一个 **n 元组（ n -tuple）**或简称元组，通常用 t 表示。

（3）属性

- 关系中不同列称为属性（**Attribute**），每个属性有一个名字
- n 目关系必有 **n** 个属性

当 **$n=1$** 时，称该关系为单元关系 或一元关系

当 **$n=2$** 时，称该关系为二元关系

关系（续）

（4）候选码（Candidate key）

若关系中的某一属性(组)的值能唯一地标识一个元组，若从该属性(组)中去掉任何一个属性，它就不具有这一性质了,则称该属性(组)为候选码。

简单的情况：候选码只包含一个属性

例：“选课(S#, C#, Sname, Cname, Grade)”，(S#,C#)联合起来是一个候选码；那“学生(S#, Sname, Sage, Sclass)”的（S#, Sname）??

■ 全码（All-key）

最极端的情况：关系模式的所有属性是这个关系模式的候选码，称为全码（All-key）比如关系“教师授课” (T#,C#)中的候选码(T#,C#)就是全码。

关系（续）

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**）

比如学生（学号，姓名，年龄，身份证号，籍贯），学号是候选码，身份证号也是候选码，**选择其中一个为主码。**

■ 主属性

候选码的诸属性称为主属性（Prime attribute） 学号、身份证号

不包含在任何侯选码中的属性称为非主属性（**Non-Prime attribute**）或
非码属性（**Non-key attribute**） **姓名，年龄，籍贯**

关系（续）

（5）关系可以有三种类型：

- ① 基本关系（又称为基本表或基表）：实际存在的表，是实际存储数据的逻辑表示；
- ② 查询结果：查询执行产生的结果对应的临时表；
- ③ 视图表：由基本表或其他视图表导出的表，是虚表，不存储实际数据。

关系（续）

（6）基本关系的性质

- ① 列是同质的（**Homogeneous**），即来自同一个域
- 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，区分列靠列名（属性名）
- ④ 任意两个元组的码不能相同
- ⑤ 行的顺序无所谓，行的次序可以任意交换，区分行靠码
- ⑥ 分量必须取原子值，即每一个分量都必须是不可分的数据项

⑥ 分量必须取原子值，即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

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

小表

2.1 关系模型的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

2.1.4 关系模型的存储结构

2.1.2 关系模式

1. 什么是关系模式
2. 定义关系模式
3. 关系模式与关系

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

- 描述关系的完整性约束

用户使用**DDL**描述其所要建立关系的模式结构

2. 定义关系模式

关系模式可以形式化地表示为：

$R(U, D, DOM, F)$ ——五元组

R 关系名

U 组成该关系的属性名集合

D U 中属性所来自的域

DOM 属性向域的映象集合

F 属性间数据的依赖关系的集合（第6章学习）

关系模式举例

❖ STUDENT (U, D, DOM, F)

■ $U\{\text{sno}, \text{name}, \text{age}\}$ // 属性名集合

■ $D\{\text{char}, \text{int}\}$ // U中属性所来自的域

■ $\text{DOM}\{\text{dom}(\text{sno})=\text{dom}(\text{name})=\text{char}, \text{dom}(\text{age})=\text{int}\}$
// 每个属性向域的映像的集合

■ $F\{\text{sno} \twoheadrightarrow \text{name}, \text{sno} \twoheadrightarrow \text{age}\}$ // 数据依赖关系的集合

定义关系模式（续）

关系模式通常可以简记为——二元组

$R(U)$ 或 $R(A_1, A_2, \dots, A_n)$

- R : 关系名

- A_1, A_2, \dots, A_n : 属性名

SC(Sno,Cno,Grade,Semester,Teachingclass)

3. 关系模式与关系

❖ “型”与“值”的关系

- 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（**实例**）

❖ 关系模式和关系往往笼统称为关系

通过上下文加以区别

2.1 关系模型的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

2.1.4 关系模型的存储结构

2.1.3 关系数据库

❖ 关系数据库

- 支持关系模型的数据库系统
- 关系模型中，实体以及实体间的联系都用关系表示例如学生实体、课程实体、学生与课程之间选修课程的多对多联系
- 在一个关系数据库中，某一时刻所有关系模式对应的关系的集合构成一个关系数据库

关系数据库（续）

- ❖ “学生选课数据库” **模式**：学生、课程和学生选课3个关系模式：

学生表Student(Sno,Sname,Ssex,Sbirthdate,Smajor)

课程表Course(Cno,Cname,Ccredit,Cpno)

学生选课表SC(Sno,Cno,Grade,Semester,Teachingclass)

- ❖ **2019年的学生选课数据库值（实例）**：

2019年学校中所有学生的记录

2019年学校开设的所有课程的记录

2019年所有学生选课的记录

- ❖ **2020年的学生选课数据库值（实例）**：

2020年学校中所有学生的记录

2020年学校开设的所有课程的记录

2020年所有学生选课的记录

- ❖ **2020年度和2019年度“学生选课数据库”模式对应的2个实例是不同的**

关系数据库（续）

■ 学生关系模式：Student(Sno, Sname, Ssex, Sbirthdate, Smajor)

包括学号、姓名、性别、出生日期和主修专业等属性

学号 Sno	姓名 Sname	性别 Ssex	出生日期 Sbirthdate	主修专业 Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	刘晨	女	1999-9-1	计算机科学与技术
20180003	王敏	女	2001-8-1	计算机科学与技术
20180004	张立	男	2000-1-8	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180006	赵明	男	2000-6-12	数据科学与大数据技术
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

关系数据库（续）

■ 课程关系模式：Course(Cno,Cname,Ccredit,Cpno)

包括课程号、课程名、学分、先修课（直接先修课）等

课程号 Cno	课程名 Cname	学分 Ccredit	先修课 Cpno
81001	程序设计基础与C语言	4	
81002	数据结构	4	81001
81003	数据库系统概论	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

关系数据库（续）

■ 学生选课关系模式：SC(Sno,Cno, Grade,Semester,Teachingclass)

包括学号、课程号、成绩、选课学期、教学班等

学号 Sno	课程号 Cno	成绩 Grade	选课学期 Semester	教学班 Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

2.1 关系模型的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

2.1.4 关系模型的存储结构

2.1.4 关系模型的存储结构

❖ 关系数据库的物理组织—**DBMS**负责完成

- 有的关系数据库管理系统中一个表对应一个操作系统文件，将物理数据组织交给操作系统完成
- 有的关系数据库管理系统从操作系统那里申请若干个大的文件，自己划分文件空间，组织表、索引等存储结构，并进行存储管理

第2章 关系数据库

2.1 关系模型的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 关系演算

本章小结

2.2 关系操作

2.2.1 基本的关系操作

2.2.2 关系数据语言的分类

2.2.1 基本的关系操作

❖ 常用的关系操作

- 查询操作：表达能力很强，包括：**选择、投影、连接、除、并、差、交、笛卡尔积**

- 选择、投影、并、差、笛卡尔积是**5种基本操作**，其他操作可以用基本操作来定义和导出

- 数据更新：插入、删除、修改

❖ 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

- 关系操作的所有输入和输出均是关系，包括关系操作的中间结果也是关系

第2章 关系数据库

2.1 关系模型的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 关系演算

本章小结

关系的三类完整性约束

❖ 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**，应该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

2.3 关系的完整性

2.3.1 实体完整性

2.3.2 参照完整性

2.3.3 用户定义的完整性

2.3.1 实体完整性

❖ 规则2.1 实体完整性规则 (entity integrity)

- 若属性（指一个或一组属性）**A**是基本关系**R**的主属性，则**A**不能取空值
- 空值就是“不知道”或“不存在”或“无意义”的值

例：

学生选课（学号，课程号，成绩，选课学期，教学班）

（学号、课程号）为主码

“学号”和“课程号”两个属性都不能取空值

实体完整性（续）

❖ 说明

（1）实体完整性规则是针对基本关系而言的。

一个基本表通常对应现实世界的一个实体集。

（2）现实世界中的实体是可区分的，即它们具有某种唯一性标识。

（3）关系模型中以主码作为唯一性标识。

（4）主码中的属性即主属性不能取空值。

主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与第（2）点相矛盾，因此这个规则称为实体完整性

实体完整性（续）

■ 实体完整性规则 (entity integrity rule)

• 例如

Student(Sno, Sname, Ssex, Sbirthdate, Smajor)

字段含义：学号、姓名、性别、出生日期和主修专业

```
CREATE TABLE Student  
(Sno CHAR(8) PRIMARY KEY,  
Sname VARCHAR(20) UNIQUE,  
Ssex CHAR(6),  
Sbirthdate Date,  
Smajor VARCHAR(40));
```

Sno	Sname	Ssex	Sbirthdate	Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	李晨	女	1999-9-1	计算机科学与技术
20180002	王敏	女	2001-8-1	计算机科学与技术
NULL	张立	男	2000-1-8	计算机科学与技术

2.3 关系的完整性

2.3.1 实体完整性

2.3.2 参照完整性

关系与关系之间
存在着互相引用、
互相制约的情况

2.3.3 用户定义的完整性

2.3.2 参照完整性

1. 关系间的引用
2. 外码
3. 参照完整性约束

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，自然存在着关系与关系间的引用。

[例2.1] “学生”实体、“专业”实体

主码

学生（学号，姓名，性别，出生日期，主修专业）

专业（专业名，专业编号）

主码

- ❖ 两个关系存在着属性的引用
- ❖ 学生关系引用了专业关系的主码“专业名”
- ❖ 学生关系中的“主修专业”值必须是确实存在的专业的专业名

关系间的引用（续）

例[2.2] 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，出生日期，主修专业）

课程（课程号，课程名，学分，先修课）

学生选课（学号，课程号，成绩，选课学期，教学班）

- 学生选课关系引用学生关系的主码“学号”和课程关系的主码“课程号”
- 学生选课关系中的“学号”值必须是确实存在的学生的学号
- 学生选课关系中的“课程号”值也必须是确实存在的课程的课程号

关系间的引用（续）

❖ 同一关系内部属性间也可能存在引用关系

例[2.3]在课程（[课程号](#)，课程名，学分，先修课）中

- “课程号”属性是主码
- “先修课”属性表示选修该门课程之前需要完成先修课程的课程号，它引用了本关系“课程号”属性，即“课程号”必须是确实存在的课程的课程号

课程号 Cno	课程名 Cname	学分 Ccredit	先修课 Cpno
81001	程序设计基础与C语言	4	
81002	数据结构	4	81001
81003	数据库系统概论	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

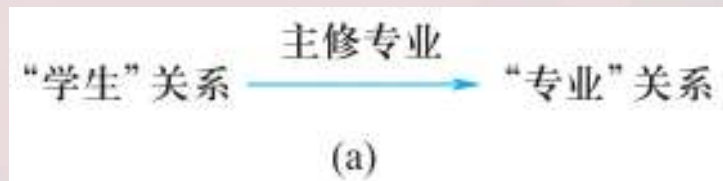
2.外码 (Foreign Key)

- ❖ 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(引用关系 S 的主码 K_s 的值)，则称 F 是 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)

R 和 S 可以是不同关系，也可以是同一个关系，**关系数据模型中关系间（表间）的联系是用外码隐含地表示的。**

外码（续）

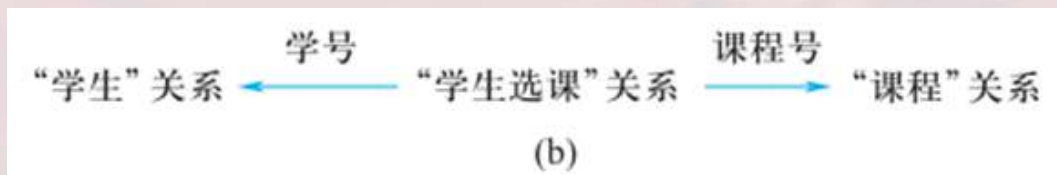
- ❖ [例2.1]中学生关系的“主修专业”与专业关系的主码“专业名”相对应
 - “主修专业”属性是学生关系的外码
 - “专业”关系是被参照关系，“学生”关系为参照关系



外码（续）

❖ [例2.2]中

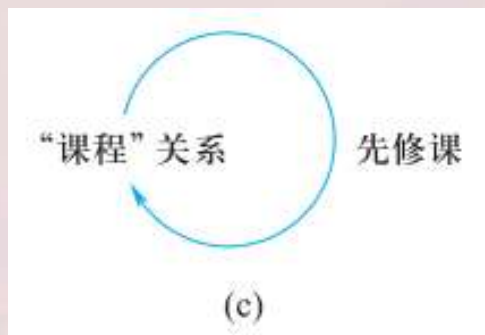
- “学生选课”关系的“学号”与“学生”关系的主码“学号”相对应
- “学生选课”关系的“课程号”与“课程”关系的主码“课程号”相对应
- “学号”和“课程号”是“学生选课”关系的外码
- “学生”关系和“课程”关系均为被参照关系
- “学生选课”关系为参照关系



外码（续）

❖ [例2.3] “课程”关系中“先修课”属性与本身的主码“课程号”属性相对应

- “先修课”是外码
- “课程”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

3.参照完整性约束

❖ 参照完整性约束

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

参照完整性约束（续）

[例2.1]中

“学生”关系中每个元组的“主修专业”属性只取两类值：

- （1）空值，表示该学生尚未选择主修专业
- （2）非空值，这时该值必须是专业关系中某个元组的“专业名”值，表示该学生不可能选一个不存在的专业

参照完整性约束（续）

[例2.2] 中

学生选课（学号，课程号，成绩，选课学期，教学班）

“学号”和“课程号”可能的取值：

- （1）学生选课关系中的主属性，不能取空值
- （2）只能取相应被参照关系中已经存在的主码值

参照完整性约束（续）

[例2.3] 中

课程（课程号，课程名，学分，先修课）

“先修课”属性值可以取两类值：

- （1）空值，表示该门课程不存在先修课
- （2）非空值，该值必须是本关系中某个元组的课程号

2.3 关系的完整性

2.3.1 实体完整性

2.3.2 参照完整性

2.3.3 用户定义的完整性

2.3.3 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，而不需由应用程序承担这一功能

用户定义的完整性（续）

例2.1 “学生”关系中

- 若要求学生不能没有姓名，则可以定义学生“姓名”不能取空值
- “学生选课”关系中“成绩”的取值范围可以定义在0~100之间

用户定义的完整性(续)

❖ **DBMS**系统通常提供了如下机制:

❖ **(1)** 使用户可以自行定义有关的完整性约束条件

❖ **(2)** 当有更新操作发生时, **DBMS**将自动按照完整性约束条件检验更新操作的正确性, 即是否符合用户自定义的完整性

第2章 关系数据库

2.1 关系模型的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 关系演算

本章小结

2.4 关系代数

❖ 关系代数是一种抽象的查询语言，它用对关系的运算来表达查询

❖ 关系代数

■ 运算对象是关系

■ 运算结果亦为关系

$\Pi_{S\#, SNAME}(\sigma_{C\# = 'c2'}(S \bowtie SC))$

■ 关系代数的运算符有两类：传统的集合运算符和专门的关系运算符

❖ 传统的集合运算是从关系的“水平”方向即行的角度进行

❖ 专门的关系运算不仅涉及行而且涉及列

关系代数运算符

运 算 符		含 义
集合 运算符	\cup	并
	$-$	差
	\cap	交
	\times	笛卡尔积
专门的 关系 运算符	σ	选择
	π	投影
	\bowtie	连接
	\div	除

2.4 关系代数

2.4.1 传统的集合运算

2.4.2 专门的关系运算

1 并 (Union)

❖ $R \cup S$

- 具有相同的目 n (即两个关系都有 n 个属性)，结果仍为 n 目关系，由属于 R 或属于 S 的元组组成

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

并运算是将两个关系的元组合并成一个关系，**在合并时去掉重复的元组。**

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

RUS

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

并操作的语义

❖ 若 R 为计算机学院的学生， S 为材料学院的学生

则： $R \cup S$ 为

❖ 若 R 为学过数据库课程的学生， S 为学过软件工程课程的学生

则： $R \cup S$ 为

汉语中的“或者...或者...”通常意义是并运算的要求。

首先要准确理解汉语的查询要求，然后再找到正确的操作

2 差 (Difference)

❖ $R - S$

- 仍为 n 目关系，由属于 R 但不属于 S 的所有元组组成

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

❖ $R - S$ 与 $S - R$ 是不同的

差 (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-S

A	B	C
a1	b1	c1

差操作的语义

- ❖ 若 R 为计算机学院的学生， S 为四年级的学生，则：
 - ❖ $R - S$ 为 [redacted]
 - ❖ $S - R$ 为 [redacted]
- ❖ 若 R 为学过数据库课程的学生， S 为学过软件工程课程的学生，则：
 - ❖ $R - S$ 为 [redacted]
- ❖ 汉语中的“是...但不含...”通常意义是差运算的要求。

3 交 (Intersection)

❖ $R \cap S$

- 仍为 n 目关系，由既属于 R 又属于 S 的元组组成

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

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

交 (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

$R \cap S$

A	B	C
a1	b2	c2
a2	b2	c1

交操作的语义

❖ 若R为年龄小于20岁的学生，S为计算机学院的学生，则：

$R \cap S$ 为

❖ 若R为学过数据库课程的学生，S为学过软件工程课程的学生，则：

$R \cap S$ 为

❖ 汉语中的“既...又...”，“..., 并且...”通常意义是交运算的要求

4 笛卡尔积 (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

● $R \times S = \{\widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S\}$

$R \times S = S \times R$

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 传统的集合运算

2.4.2 专门的关系运算

学生-课程数据库

学生关系**Student**、课程关系**Course**和学生选课关系**SC**

Student

学号 Sno	姓名 Sname	性别 Ssex	出生日期 Sbirthdate	主修专业 Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	刘晨	女	1999-9-1	计算机科学与技术
20180003	王敏	女	2001-8-1	计算机科学与技术
20180004	张立	男	2000-1-8	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180006	赵明	男	2000-6-12	数据科学与大数据技术
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

(a)

学生-课程数据库

Course

课程号 Cno	课程名 Cname	学分 Ccredit	先修课 Cpno
81001	程序设计基础与C语言	4	
81002	数据结构	4	81001
81003	数据库系统概论	4	81002
81004	信息系统概论	4	81003
81005	操作系统	4	81001
81006	Python语言	3	81002
81007	离散数学	4	
81008	大数据技术概论	4	81003

(b)

学生-课程数据库

SC

学号 Sno	课程号 Cno	成绩 Grade	选课学期 Semester	教学班 Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

(c)

一些记号

(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

一些记号（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。

全部
属性

一些记号（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。

(4) 象集 Z_x

比较抽象，先举个例子

表A

姓名	书号
喵喵	1
喵喵	2
汪汪	1
汪汪	3
萌萌	2
萌萌	3

喵喵在表A中的象集为{1, 2}:

1
2

汪汪在表A中的象集为{1, 3}:

1
3

萌萌在表A中的象集为{2, 3}:

2
3

象集：本质上是一次选择行的运算和一次选择列的运算。

求 x_1 在表A中的象集，就是先选出所有 x 属性中 $x=x_1$ 的那些行，然后选择出不包含 x_1 的那些列。

一些记号（续）

（4）象集 Z_x

正式的定义

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 上分量的集合

一些记号

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

❖ 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

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

2.4.2 专门的关系运算

1. 选择
2. 投影
3. 连接
4. 除运算

1. 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

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

■ F : 选择条件, 是一个逻辑表达式, 取值为“真”或“假”

● 基本形式为: $X_1 \theta Y_1$

● X_1 , Y_1 是属性名, 或为常量, 或为简单函数; 也可以用它的序号来代替

● θ 表示比较运算符, 它可以是 $>$, \geq , $<$, \leq , $=$ 或 \neq

● 可通过逻辑运算符进一步进行与、或、非运算

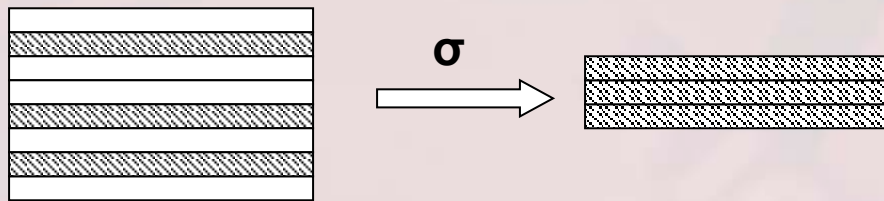
\neg 命题的“非”运算

\wedge 命题的“合取” (“与”) 运算

\vee 命题的“析取” (“或”, “可兼或”) 运算

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



选择（续）

[例2.4] 查询信息安全专业全体学生。



结果：

Sno	Sname	Ssex	Sbirthdate	Smajor
20180001	李勇	男	2000-3-8	信息安全

选择（续）

[例2.5] 查询2001年之后（包括2001年）出生的学生。

结果：

Sno	Sname	Ssex	Sbirthdate	Smajor
20180003	王敏	女	2001-8-1	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

进一步：

查询信息安全专业且**2001**年之后（包括**2001**年）出生的学生？

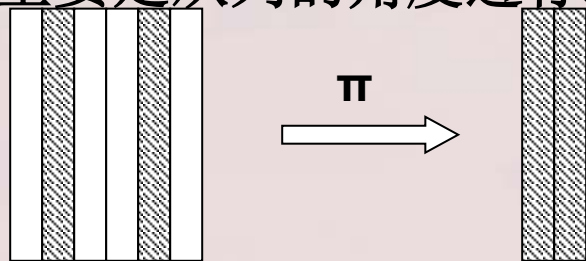
查询信息安全专业或**2001**年之后（包括**2001**年）出生的学生？

2. 投影 (Projection)

- 从 R 中选择出若干属性列组成新的关系

$$\pi_A(R) = \{ t[A] \mid t \in R \}, \quad A: R \text{中的属性列}$$

- 投影操作主要是从列的角度进行运算



- 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2.6] 查询学生的姓名和主修专业。

即求**Student**关系上学生姓名和主修专业两个属性上的投影



结果:

学号Sno	专业 Smajor
20180001	信息安全
20180002	计算机科学与技术
20180003	计算机科学与技术
20180004	计算机科学与技术
20180005	信息管理与信息系统
20180006	数据科学与大数据技术
20180007	数据科学与大数据技术

投影（续）

[例2.7] 查询学生关系**Student**中都主修了哪些专业。



结果：去掉了重复的值

专业Smajor
信息安全
计算机科学与技术
信息管理与信息系统
数据科学与大数据技术

3. 连接 (Join)

❖ 连接也称为 θ 连接

❖ 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

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

- 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 属性组上的值满足比较关系 θ 的元组

■ $R \underset{i\theta j}{\bowtie} S = \sigma_{i\theta(r+j)}(R \times S)$ 选择和笛卡尔积的组合

连接（续）

❖ 两类常用连接运算

■ 等值连接（equijoin）

- θ 为 “=” 的连接运算称为等值连接
- 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

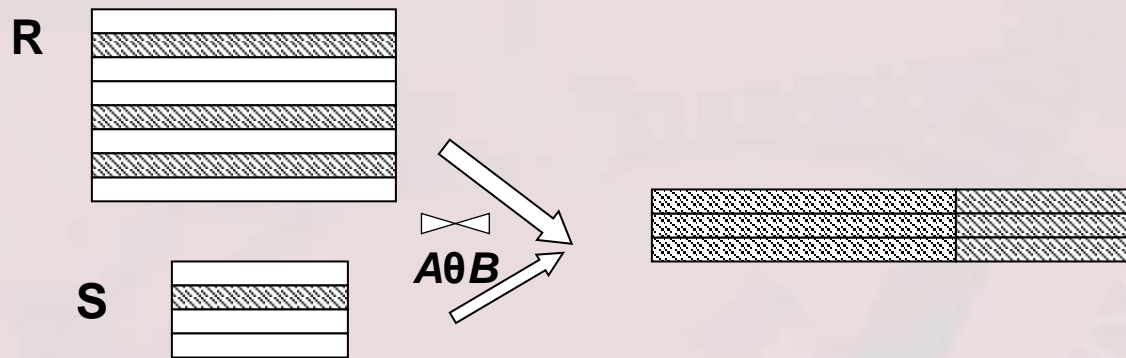
R 和 S 具有相同的属性组 B

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



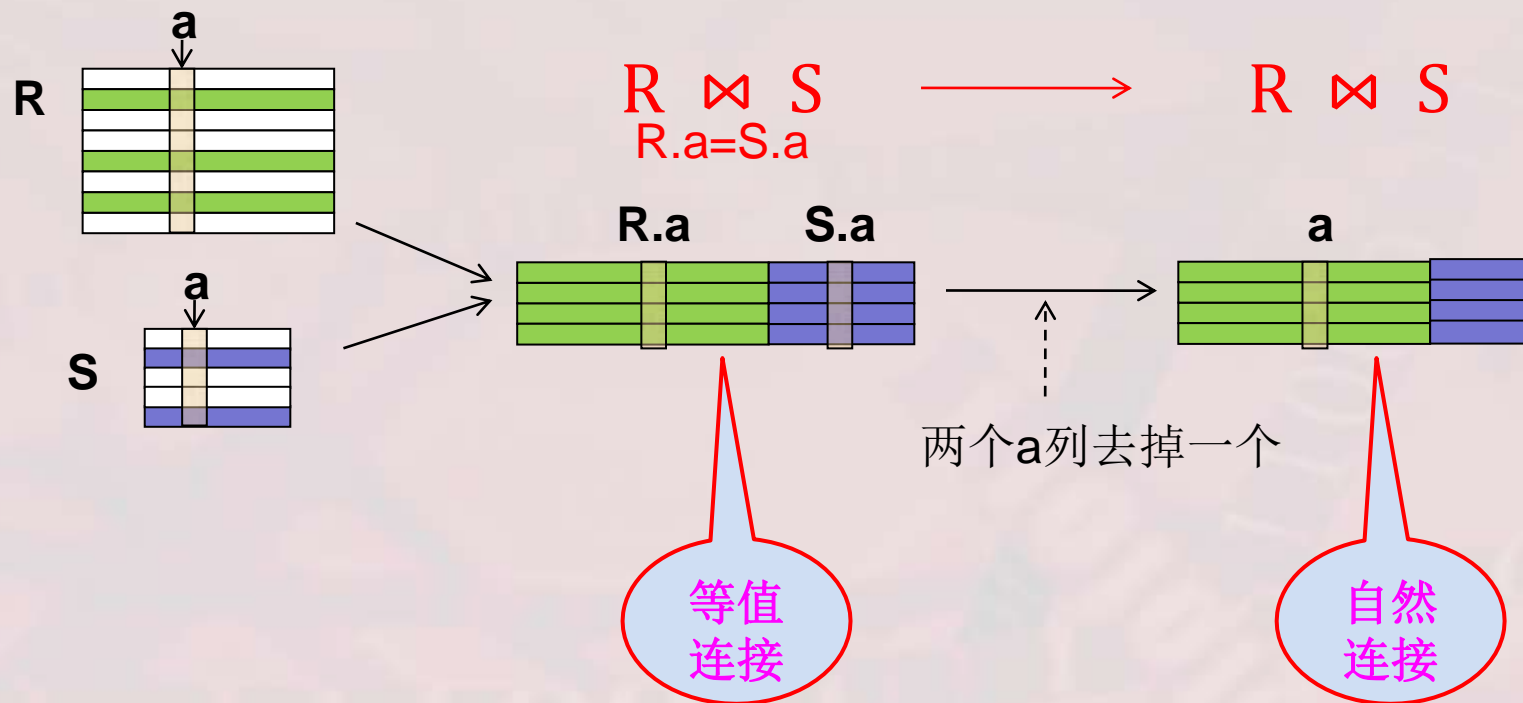
连接（续）

❖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。

连接（续）



- $R \bowtie S$ 的计算过程:

(1) 计算 $R \times S$

(2) 设 R 和 S 的公共属性是 B_1, \dots, B_k ,

挑选 $R \times S$ 中满足 $R.B_1 = S.B_1, \dots, R.B_k = S.B_k$ 的元组

(3) 去掉冗余属性 $S.B_1, \dots, S.B_k$

两个关系没有公共属性，自然连接就转化为笛卡儿积操作

连接（续）

❖ [例2.8]关系 R 和关系 S 如下所示：

R

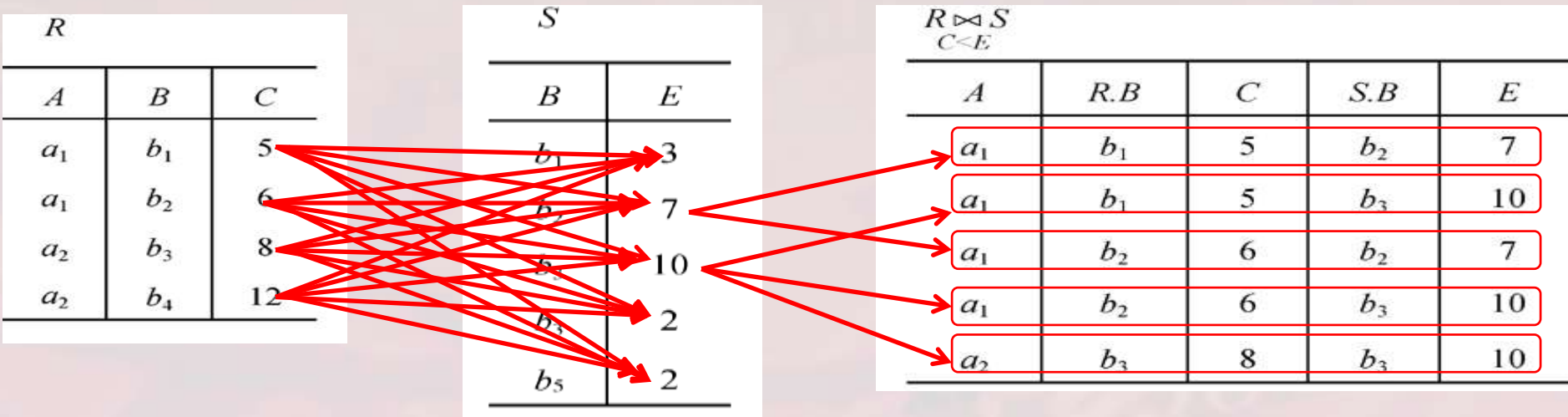
A	B	C
a1	b1	5
a1	b2	6
a2	b3	8
a2	b4	12

S

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

连接(续)

一般连接 $R \bowtie_{C < E} S$ 的结果如下:



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
a1	b1	5	b1	3
a1	b2	6	b2	7
a1	b2	6	b2	2
a2	b3	8	b3	10
a2	b3	8	b3	2

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
b2	2

连接（续）

自然连接 $R \bowtie S$ 的结果如下：

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a1	b2	6	2
a2	b3	8	10
a2	b3	8	2

思考题

对学生-课程数据库中的关系进行查询，得到以下相应的结果，请写出对应的关系表达式。

Sname	Cname	Grade
刘晨	程序设计基础与C语言	80
刘晨	数据结构	98
刘晨	数据库系统概论	71
张立	程序设计基础与C语言	56
张立	数据库系统概论	97
李勇	程序设计基础与C语言	85
李勇	数据结构	96
李勇	数据库系统概论	87
王敏	程序设计基础与C语言	81
王敏	数据结构	76
陈新奇	数据库系统概论	68

课堂练习

请对学生-课程数据库中的学生关系**Student**、学生选课关系**SC**进行自然连接，给出相应的结果。

Student ⋈ **SC**

学号 Sno	姓名 Sname	性别 Ssex	出生日期 Sbirthdate	主修专业 Smajor
20180001	李勇	男	2000-3-8	信息安全
20180002	刘晨	女	1999-9-1	计算机科学与技术
20180003	王敏	女	2001-8-1	计算机科学与技术
20180004	张立	男	2000-1-8	计算机科学与技术
20180005	陈新奇	男	2001-11-1	信息管理与信息系统
20180006	赵明	男	2000-6-12	数据科学与大数据技术
20180007	王佳佳	女	2001-12-7	数据科学与大数据技术

学号 Sno	课程号 Cno	成绩 Grade	选课学期 Semester	教学班 Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

自然连接

执行结果：

Sno	Sname	Ssex	Sbirthdate	Smajor	Cno	Grade	Semester	Teachingclass
20180001	李勇	男	2000-03-08	信息安全	81001	85	20192	81001-01
20180001	李勇	男	2000-03-08	信息安全	81002	96	20201	81002-01
20180001	李勇	男	2000-03-08	信息安全	81003	87	20202	81003-01
20180002	刘晨	女	1999-09-01	计算机科学与技术	81001	80	20192	81001-02
20180002	刘晨	女	1999-09-01	计算机科学与技术	81002	98	20201	81002-01
20180002	刘晨	女	1999-09-01	计算机科学与技术	81003	71	20202	81003-02
20180003	王敏	女	2001-08-01	计算机科学与技术	81001	81	20192	81001-01
20180003	王敏	女	2001-08-01	计算机科学与技术	81002	76	20201	81002-02
20180004	张立	男	2000-01-08	计算机科学与技术	81001	56	20192	81001-02
20180004	张立	男	2000-01-08	计算机科学与技术	81003	97	20201	81002-02
20180005	陈新奇	男	2001-11-01	信息管理与信息...	81003	68	20202	81003-01

赵明和王佳佳同学呢？--被舍弃的元组称为**悬浮元组**

连接（续）

❖ 悬浮元组（Dangling tuple）

- 两个关系 R 和 S 在做自然连接时，关系 R 中某些元组有可能在 S 中不存在公共属性上值相等的元组，从而造成 R 中这些元组在操作时被舍弃了，这些被舍弃的元组称为悬浮元组。（同样情况， S 中也可能存在悬浮元组）

悬浮元组

R			S	
A	B	C	C	D
1	1	1	2	1
2	2	2	3	1

悬浮元组

悬浮元组

$R \bowtie S$

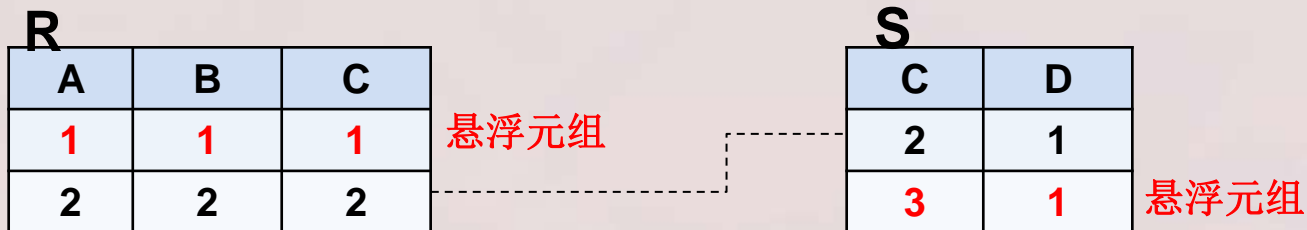
A	B	C	D
2	2	2	1

连接（续）

❖ 外连接（Outer Join）

- 如果把悬浮元组也保存在结果关系中，而在其他属性上填空值(Null)，就叫做外连接
- 左外连接(LEFT OUTER JOIN或LEFT JOIN)
 - 只保留左边关系 R 中的悬浮元组
- 右外连接(RIGHT OUTER JOIN或RIGHT JOIN)
 - 只保留右边关系 S 中的悬浮元组

外连接 \bowtie



外连接

$R \bowtie S$

A	B	C	D
2	2	2	1
1	1	1	null
null	null	3	1

左外连接

$R \bowtie S$

A	B	C	D
1	1	1	null
2	2	2	1

右外连接

$R \bowtie S$

A	B	C	D
2	2	2	1
null	null	3	1

课堂练习

请对学生-课程数据库中的学生关系**Student**、选修关系**SC**进行左外连接，给出相应的结果。

左外连接

执行结果:

Student ⋈ SC

Sno	Sname	Ssex	Sbirthdate	Smajor	Cno	Grade	Semester	Teachingclass
20180001	李勇	男	2000-03-08	信息安全	81001	85	20192	81001-01
20180001	李勇	男	2000-03-08	信息安全	81002	96	20201	81002-01
20180001	李勇	男	2000-03-08	信息安全	81003	87	20202	81003-01
20180002	刘晨	女	1999-09-01	计算机科学与技术	81001	80	20192	81001-02
20180002	刘晨	女	1999-09-01	计算机科学与技术	81002	98	20201	81002-01
20180002	刘晨	女	1999-09-01	计算机科学与技术	81003	71	20202	81003-02
20180003	王敏	女	2001-08-01	计算机科学与技术	81001	81	20192	81001-01
20180003	王敏	女	2001-08-01	计算机科学与技术	81002	76	20201	81002-02
20180004	张立	男	2000-01-08	计算机科学与技术	81001	56	20192	81001-02
20180004	张立	男	2000-01-08	计算机科学与技术	81003	97	20201	81002-02
20180005	陈新奇	男	2001-11-01	信息管理与信息...	81003	68	20202	81003-01
20180006	赵明	男	2006-06-12	数据科学与大数...	NULL	NULL	NULL	NULL
20180007	王佳佳	女	2001-12-07	数据科学与大数...	NULL	NULL	NULL	NULL

4. 除运算 (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上全部投影的集合, 记作:

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

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

除运算（续）

[例2.9] 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
a1	b1	c2
a1	b2	C1
a1	b2	C3
a2	b3	C7
a2	b2	C3
a3	b4	C6
a4	b6	c6

图a

S

B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

图b

$R \div S$

A
a1

图c

除运算（续）

❖ 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 $R \div S = \{a_1\}$

R		
A	B	C
a1	b1	c2
a1	b2	c1
a1	b2	c3
a2	b3	c7
a2	b2	c3
a3	b4	c6
a4	b6	c6

图a

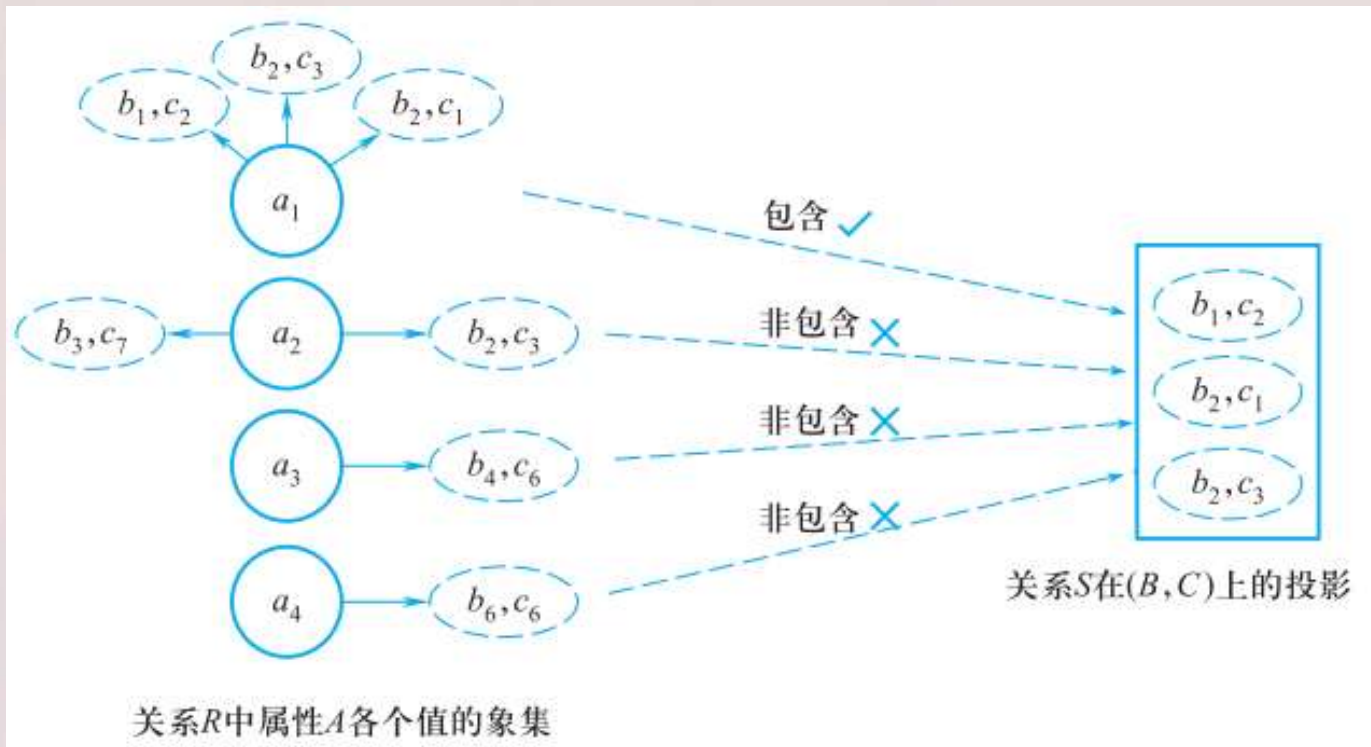
S		
B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

$R \div S$

A
a1

图c

除运算（续）



除运算的结果列

$$R(A,B,C,D) \div S(C,D) = T(A,B)$$

$$R(A,B,C,D,E) \div S(C,D) = T(A,B,E)$$

$$R(A,B,C,D) \div S(B,C,D) = T(A)$$

除运算（示例）

选课

Sno	Cno	GRADE
S1	C1	80
S3	C1	90
S1	C2	70
S3	C2	85
S1	C3	95
S4	C4	70

课程

Cno	CNAME	TEACHER
C1	Maths	Ma
C2	Physics	Shi
C3	Chemistry	Zhou

检索：学习全部课程的学生学号

全体学生(学号,课号) ÷ 所有课程的(课号)=(学号)

↓
 $\pi_{Sno, cno}$ (选课)

÷

↓
 π_{Cno} (课程)

Sno

s1

关系代数

(综合训练)

以学生-课程数据库为例

综合举例

[例2.10]查询至少选修81001号课程和81003号课程的学生学号。

首先建立一个临时关系***K***:

Cno
81001
81003

然后求: $\pi_{Sno,Cno}(SC) \div K = \{20180001, 20180002\}$

综合举例（续）

❖ [例2.10]续

$\pi_{Sno,Cno}(SC)$

20180001象集{81001, 81002, 81003}

20180002象集{81001, 81002, 81003}

20180003象集{81001, 81002}

20180004象集{81001, 81003}

20180005象集{81003}

$K=\{81001, 81003\}$

于是： $\pi_{Sno,Cno}(SC) \div K = \{20180001, 20180002\}$

sno	cno
20180001	81001
20180002	81001
20180003	81001
20180004	81001
20180001	81002
20180002	81002
20180003	81002
20180001	81003
20180002	81003
20180004	81003
20180005	81003

综合举例（续）

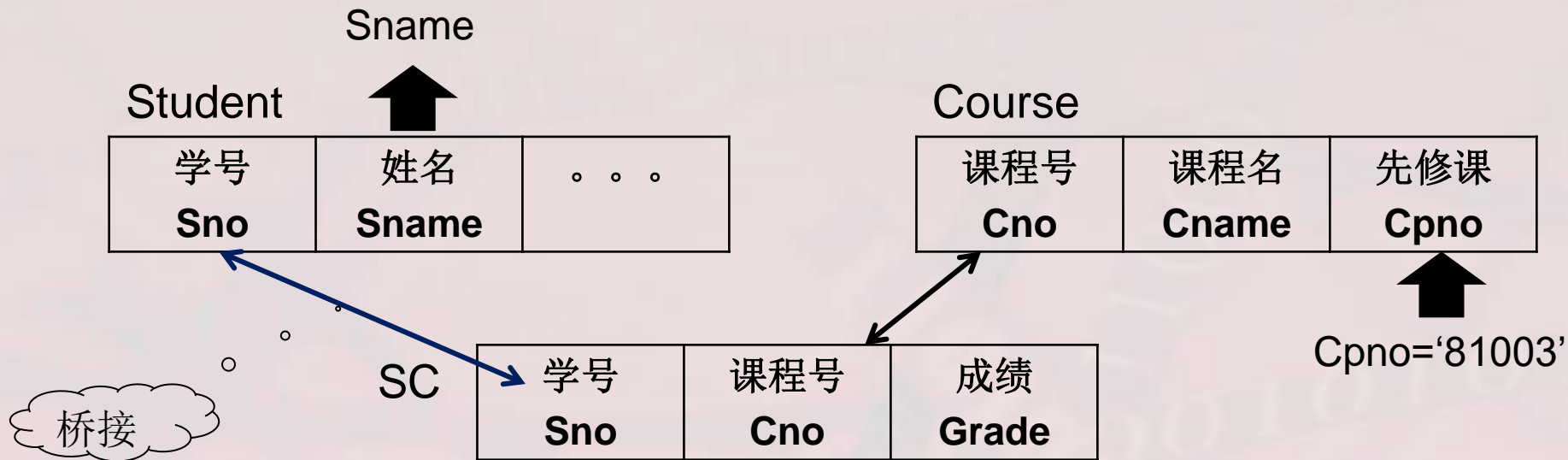
[例2.11] 查询选修了81002号课程的学生的学号。

$\pi_{Sno}(\sigma_{Cno='81002'}(SC)) = \{20180001, 20180002, 20180003\}$



综合举例（续）

[例2.12] 查询至少选修了一门其直接先行课为**81003**号课程的学生姓名



综合举例（续）

[例2.12]查询至少选修了一门其直接先行课为81003号课程的学生姓名

$$\pi_{\text{Sname}}(\sigma_{\text{Cpno}='81003'}(\text{Course} \bowtie \text{SC} \bowtie \text{Student}))$$
$$\pi_{\text{Sname}}(\pi_{\text{Sno}}(\sigma_{\text{Cpno}='81003'}(\text{Course}) \bowtie \text{SC}) \bowtie \pi_{\text{Sno}, \text{Sname}}(\text{Student}))$$

综合举例（续）

[例2.13] 查询选修了全部课程的学生学号和姓名。

第一步：求出选修了全部课程的学生学号

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

第二步：通过学号求学生姓名

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

综合举例（续）

用关系代数表达式表达以下每个查询：

➤ 第一步：分析操作的数据库对象

单表、多表（ \times 或 \bowtie 类型）

➤ 第二步：分析元组条件

做选择运算（ σ ）保留所需的行/元组

➤ 第三步：分析需要的属性

做投影运算（ Π ）保留所需的列/属性

(1) 检索选修了81002号课程的学生的学号与成绩

$\Pi_{\text{Sno}, \text{Grade}} (\sigma_{\text{Cno} = '81002', (\text{SC}))$ 或 $\Pi_{1,3} (\sigma_{2='81002', (\text{SC}))$

(2) 检索选修了c2号课程的学生的学号与姓名

$\Pi_{\text{Sno}, \text{Sname}} (\sigma_{\text{Cno} = '81002', (\text{Student} \bowtie \text{SC}))$

综合举例（续）

(3)检索选修了课程数据结构的学生的学号与姓名

$\Pi_{Sno, Sname}(\sigma_{Cname = '数据结构'}(Student \bowtie SC \bowtie Course))$

(4)检索选修了81001或81003号课程的学生的学号

$\Pi_{Sno}(\sigma_{Cno = '81001' \vee Cno = '81003'}(SC))$

(5)检索至少选修了81001和81003号课程的学生的学号

$\Pi_{Sno}(\sigma_{Cno = '81001' \wedge Cno = '81003'}(SC))$??



综合举例（续）

SC

×

SC

Sno	Cno	Grade	Semester	Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

Sno	Cno	Grade	Semester	Teachingclass
20180001	81001	85	20192	81001-01
20180001	81002	96	20201	81002-01
20180001	81003	87	20202	81003-01
20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01
20180002	81003	71	20202	81003-02
20180003	81001	81	20192	81001-01
20180003	81002	76	20201	81002-02
20180004	81001	56	20192	81001-02
20180004	81003	97	20201	81002-02
20180005	81003	68	20202	81003-01

综合举例（续）

SC×SC

从中选择：

- 1、第一列和第六列相等的记录（同一名学生选修了两门课）
- 2、第二列是81001并且第七列是81003（选修了81001和81003这两门课）

Sno	Cno	Grade	Semester	Teachingclass	Sno	Cno	Grade	Semester	Teachingclass
20180001	81003	87	20202	81003-01	20180001	81003	87	20202	81003-01
20180001	81002	96	20201	81002-01	20180001	81003	87	20202	81003-01
20180001	81001	85	20192	81001-01	20180001	81003	87	20202	81003-01
20180005	81003	68	20202	81003-01	20180002	81001	80	20192	81001-02
20180004	81003	97	20201	81002-02	20180002	81001	80	20192	81001-02
20180004	81001	56	20192	81001-02	20180002	81001	80	20192	81001-02
20180003	81002	76	20201	81002-02	20180002	81001	80	20192	81001-02
20180003	81001	81	20192	81001-01	20180002	81001	80	20192	81001-02
20180002	81003	71	20202	81003-02	20180002	81001	80	20192	81001-02
20180002	81002	98	20201	81002-01	20180002	81001	80	20192	81001-02
20180002	81001	80	20192	81001-02	20180002	81001	80	20192	81001-02
20180001	81003	87	20202	81003-01	20180002	81001	80	20192	81001-02
20180001	81002	96	20201	81002-01	20180002	81001	80	20192	81001-02
20180001	81001	85	20192	81001-01	20180002	81001	80	20192	81001-02

综合举例（续）

(5)检索至少选修了81001和81003号课程的学生的学号

$$\Pi_1(\sigma_{1=6 \wedge 2='81001' \wedge 7='81003'}(SC \times SC))$$

Sno
20180001
20180002
20180004

$$\Pi_{Sno, Cno}(SC) \div \Pi_{Cno}(\sigma_{Cno='81001' \vee C\#='81003'}(Course)) \quad \checkmark$$

交操作呢？大家思考一下

综合举例（续）

(6)检索**不选**81002号课程的学生姓名——**差操作注意并兼容性**

$\Pi_{Sname}(Student) - \Pi_{Sname}(\sigma_{Cno = '81002'}(Student \bowtie SC))$

(7)检索学习**全部**课程的学生姓名

学生选课情况表示为:



全部课程的课程号为:



学了全部课程的**学号**可用除法操作表示为:



从S#求学生**姓名**为:



综合举例（续）

(8)检索所学课程包含学生20180003所学课程的学生学号

学生选课情况表示为：



学生20180003所学课程表示为：



所学课程包含学生20180003所学课程的学生学号为：



总结:

◆书写关系代数表达式的基本思路

- ✓ 检索是否涉及多个表，如不涉及，则可直接采用并、差、交、选择与投影，只要注意条件书写正确与否即可，当不同运算符在一起时，要注意运算符的优先次序，**优先次序自高至低为{ 括弧； θ ； \neg ； \wedge ； \vee }**
- ✓ 如涉及多个表，则检查
 1. 能否使用自然连接，将多个表连接起来(多数情况是这样的)
 2. 如不能，能否使用等值或不等值连接(θ -连接)
 3. 还不能，则使用广义笛卡尔积，注意相关条件的书写
- ✓ 连接完后，可以继续使用选择、投影等运算

$\Pi_{Sno, Sname}(\sigma_{Cno = '81001'}(Student \bowtie SC))$

第2章 关系数据库

2.1 关系模型的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 关系演算（略）

本章小结

本章小结（续）

❖ 关系数据结构

■ 关系

- 域
- 笛卡尔积
- 关系
 - 关系，属性，元组
 - 候选码，主码，主属性
 - 基本关系的性质

■ 关系模式

■ 关系数据库

■ 关系模型的存储结构

本章小结（续）

❖ 关系操作

■ 查询

- 选择、投影、连接、除、并、交、差、笛卡儿积等

■ 数据更新

- 插入、删除、修改

本章小结（续）

❖ 关系的完整性

- 实体完整性
- 参照完整性
 - 外码
- 用户定义的完整性

本章小结（续）

❖ 关系代数运算

- 五个基本运算： $\{\sigma, \Pi, \cup, -, \times\}$,
- 理解各种代数运算的含义，难点是连接、除运算，
- 理解所构成的表达式的含义。