



2024年春季学期

数据库系统概论

An Introduction to Database Systems

第二章 关系数据库

中国科学技术大学 大数据学院

黄振亚, huangzhy@ustc.edu.cn



关系数据库简介

2

- 1962年, CODASYL发表“信息代数”
- 1968年, IBM7090机实现集合论数据结构
- 提出关系模型的是美国**IBM**公司的**E.F.Codd**

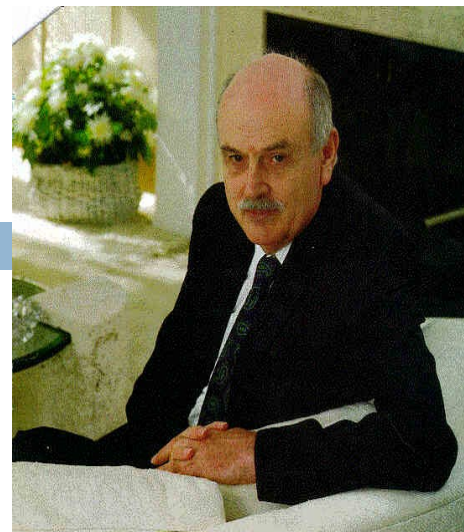
- 1970年, 提出关系数据模型

E.F.Codd, “A Relational Model of Data for Large

Shared Data Banks”, 《Communication of the ACM》, 1970

✓ 1958年以来的四分之一世纪具有里程碑意义的25篇研究论文之一

- 之后, 提出了关系代数和关系演算的概念
- 1972年, 提出了关系的第一、第二、第三范式
- 1974年, 提出了关系的**BC**范式





第二章 关系数据库

3

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 关系演算

2.6 小结



2.1 关系数据结构及形式化定义

4

- 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

5

□ 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

□ 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

□ 数学基础：建立在**集合代数**的基础上



关系(续)

6

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



1. 域 (Domain)

7

□ **域**是一组具有相同数据类型的值的集合。例：

➤ 整数

➤ 实数

➤ 介于某个取值范围的整数： $(0,100)$

➤ 指定长度的字符串集合： 长度为30

➤ { ‘男’ , ‘女’ }

➤



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 , 允许某些域是相同的

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 的一个组合

□ 不能重复

■ $D_1 =$ 导师集合 **SUPERVISOR** = {张清玫, 刘逸}

■ $D_2 =$ 专业集合 **SPECIALITY** = {计算机专业, 信息专业}

■ $D_3 =$ 研究生集合 **POSTGRADUATE** = {李勇, 刘晨, 王敏}



2. 笛卡尔积 (Cartesian Product)

9

□ 笛卡尔积 —— 一个例子

□ 给出3个域

■ D_1 = 导师集合SUPERVISOR = {张清玫, 刘逸}

■ D_2 = 专业集合SPECIALITY = {计算机专业, 信息专业}

■ D_3 = 研究生集合POSTGRADUATE = {李勇, 刘晨, 王敏}

□ D_1, D_2, D_3 的笛卡尔积 $D_1 \times D_2 \times D_3$ 为



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 |



笛卡尔积 (续)

11

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

□ 元组 (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)**
- 例如: (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

□ 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



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 |



笛卡尔积 (续)

13

□ 基数 (Cardinal number)

- 一个域允许的不同取值个数称为这个域的基数
- 若 $D_i (i=1, 2, \dots, n)$ 为有限集, 其基数为 $m_i (i=1, 2, \dots, n)$, 则笛卡尔积 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M 为:

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

- 例如, 上述例子中, 基数 $M=2*2*3=12$ 。
 - D_1 = 导师集合SUPERVISOR = {张清玫, 刘逸}
 - D_2 = 专业集合SPECIALITY = {计算机专业, 信息专业}
 - D_3 = 研究生集合POSTGRADUATE = {李勇, 刘晨, 王敏}



笛卡尔积 (续)

14

- 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（续）

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 |

分量

元组

M1 = 2
M2 = 2
M3 = 3
M = 12



3. 关系 (Relation)

16

□ 关系

$D_1 \times D_2 \times \dots \times D_n$ 的子集叫作在域 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)



3. 关系 (Relation)

17

□ 关系

$D_1 \times D_2 \times \dots \times D_n$ 的子集叫作在域 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)

□ 元组: 关系中的每个元素是关系中的元组, 通常用 t 表示

□ 单元关系与二元关系

■ 当 $n=1$ 时, 称该关系为单元关系 (Unary relation)
或一元关系

■ 当 $n=2$ 时, 称该关系为二元关系 (Binary relation)



关系 (Relation)

18

关系的表示

- ✓ 关系是笛卡尔积的有限子集
- ✓ 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 2.2 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 (Relation)

19

□ 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 (Relation)

20

□ 码

□ 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，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

□ 全码 (All-key)

最极端的情况：关系模式的所有属性组是这个关系模式的候选码，称为全码 (All-key)



关系 (Relation)

21

□ 码

□ 主码 系统建立index

- 若一个关系有多个候选码，则选定其中一个为主码 (Primary key)

□ 主属性

- 候选码的诸属性称为主属性 (Prime attribute)
- 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)



关系 (Relation)

22

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

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：SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

假设：导师与专业：1:1， 导师与研究生：1:n

主码：POSTGRADUATE（假设研究生不会重名）

表2.2 SAP关系

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) }

| SUPERVISOR | SPECIALITY | POSTGRADUATE |
|------------|------------|--------------|
| 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 |



关系 (Relation)

23

□ 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

□ 视图表 (View)

- ✓ 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
- ✓ 实际不存在



关系 (Relation)

| SUPERVISOR | SPECIALITY | POSTGRADUATE |
|------------|------------|--------------|
| 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 |

24

基本关系的性质

- ① 列是同质的 (同一个类型数据)
- ② 不同的列可出自同一个域
 - 其中的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值不能相同 (增加id)
- 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（不可分）
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

关系的每一个分量必须是一个不可分的数据项



2.1 关系数据结构

26

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

27

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

28

关系数据库中

- 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
 - 完整性约束条件
 - 属性间数据的依赖关系集合
 - 如，取值范围，相互关联



2. 定义关系模式

29

定义2.4 关系模式可以形式化地表示为：

$R(U, D, DOM, F)$

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

第6章 关系数据理论



定义关系模式 (

| SUPERVISOR | SPECIALITY | POSTGRADUATE |
|------------|------------|--------------|
| 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 |

30

例: SAP模式:

SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

□ 定义域

□ **D1: PERSON. D2: SPECIALITY**

□ 导师和研究生出自同一个域 —— 人

□ 定义DOM

□ 取不同的属性名，并在模式中定义属性向域的映象，即说明它们分别出自哪个域：

DOM (SUPERVISOR)

= DOM (POSTGRADUATE)

= PERSON



定义关系模式 (续)

31

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象常常直接说明为属性的类型、长度

```
CREATE TABLE Student
  (Sno    CHAR(9) PRIMARY KEY,
   Sname  CHAR(20) UNIQUE,
   Ssex   CHAR(2)
   Sage   INT)
```

关系R

属性名

DOM



3. 关系模式与关系

32

□ 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

□ 注：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

33

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

34

- 关系数据库
 - 在一个给定的应用领域中，所有关系的集合构成一个关系数据库
- 关系数据库的型与值



2.1.3 关系数据库

35

- 关系数据库的型: 关系数据库模式
对关系数据库的描述。
- 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: 关系模式在某一时刻对应的关系的集合，简称为关系数据库



2.1.3 关系数据库（示例）

36

- 学生表: Student (Sno, Sname, Ssex, Sage, Sdept)
- 课程表: Course (Cno, Cname, Ccredit)
- 学生选课表: SC (Sno, Cno, Grade)

| Sno | Sname | Ssex | Sage | Sdept |
|-----|-------|------|------|-------|
| 001 | 李勇 | 男 | 20 | CS |
| 002 | 刘晨 | 女 | 19 | CS |

| Cno | Cname | Cpno | Ccredit |
|-----|-------|------|---------|
| 1 | 数据库 | 3 | 3.5 |
| 2 | 数学 | | 4 |
| 3 | 数据结构 | | 4 |

| Sno | Cno | Grade | Sno | Cno | Grade |
|-----|-----|-------|-----|-----|-------|
| 001 | 1 | 92 | 002 | 2 | 90 |
| 001 | 2 | 85 | 002 | 3 | 80 |
| 001 | 3 | 88 | | | |



2.1 关系数据结构

37

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 关系数据库的存储结构

38

- 关系数据库的逻辑结构表示：“表”
- 关系数据库的物理结构
 - 一个表对应一个操作系统文件，物理结构组织交给操作系统完成；
 - 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、索引等结构，并进行存储管理



第二章 关系数据库

39

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 关系演算

2.6 小结



2.2.1基本关系操作

40

- 常用的关系操作
 - 查询：选择、投影、连接、除、并、交、差
 - 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
 - 5种基本操作：选择、投影、并、差、笛卡尔积
-



2.2.2 关系数据库语言的分类

41

- 关系代数语言 $\pi_{Sno} (\sigma_{Cno='2'} (SC))$
 - 用对关系的运算来表达查询要求。代表：ISBL
- 关系演算语言：用谓词来表达查询要求
 - 元组关系演算语言
 - 谓词变元的基本对象是元组变量。代表：APLHA, QUEL
 - 域关系演算语言
 - 谓词变元的基本对象是域变量。代表：QBE
- 具有关系代数和关系演算双重特点的语言
 - 代表：SQL (Structured Query Language)
- 关系数据语言 (SQL)
 - 查询、数据定义语言、数据操纵语言、数据控制语言



2.2.2 关系数据库语言

42

□ 关系数据库语言的特点

□ 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

□ 语言具有完备的表达能力

□ 非过程化的操作语言，是描述性语言

- “要什么”，而非“怎么要”
- 不必请求建立特殊存取路径，存取路径的选择由DBMS完成
 - 例如，顺序查询，or 索引查询
 - 系统自动查询优化



第二章 关系数据库

43

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 关系演算

2.6 小结



2.3 关系的完整性

44

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

45

- 现实世界的要求
- 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（只需定义）
- 任何关系在任何时刻都满足这些语义约束



2.3 关系的完整性

46

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

47

规则2.1 实体完整性规则 (Entity Integrity)

- 若属性A 是基本关系R 的主属性，则属性A不能取空值NULL
 - 例：SAP (SUPERVISOR, SPECIALITY, POSTGRADUATE)
POSTGRADUATE: 主码 (假设研究生不会重名)
不能取空值
 - 例：选修 (Sno, Cno, Grade)
Sno和Cno: 两个属性均不能取空值
- 空值就是“不知道”或“不存在”或“无意义”的值。非0
 - 例如，Grade与空值的关系



实体完整性

48

实体完整性规则的说明

- (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，
这个规则称为**实体完整性**



2.3 关系的完整性

49

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



2.3.3 参照完整性

50

1. 关系间的引用（参照）
2. 外码（Foreign Key）
3. 参照完整性规则

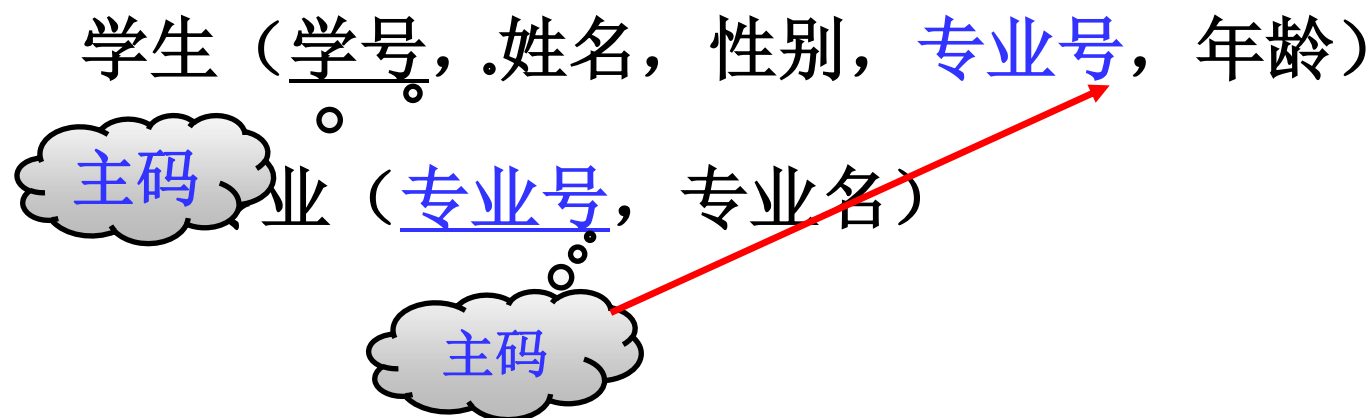


1. 关系间的引用

51

- 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2.1 学生实体、专业实体



- ✓ 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)

52

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，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)

53

| Sno | Sname | Ssex | Sage | Sdept |
|-----|-------|------|------|-------|
| 001 | 李勇 | 男 | 20 | CS |
| 002 | 刘晨 | 女 | 19 | CS |

| Cno | Cname | Cpno | Ccredit |
|-----|-------|------|---------|
| 1 | 数据库 | 3 | 3.5 |
| 2 | 数学 | | 4 |
| 3 | 数据结构 | | 4 |

| Sno | Cno | Grade |
|-----|-----|-------|
| 001 | 1 | 92 |
| 001 | 2 | 85 |
| 001 | 3 | 88 |
| 002 | 2 | 90 |
| 002 | 3 | 80 |



关系间的引用(续)

54

例2.3 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



2. 外码 (Foreign Key)

55

- 定义2.5 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 (Foreign Key)
- 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)

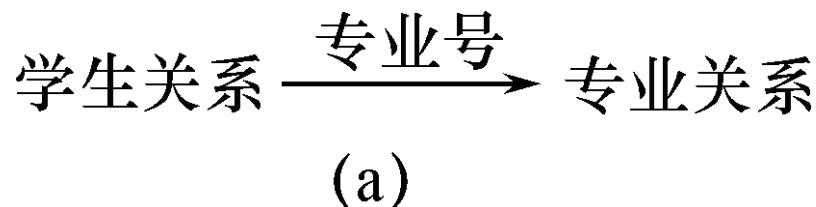




外码(续)

56

- [例2.1] : **学生** (学号, 姓名, 性别, **专业号**, 年龄)
专业 (专业号, 专业名)
- 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)

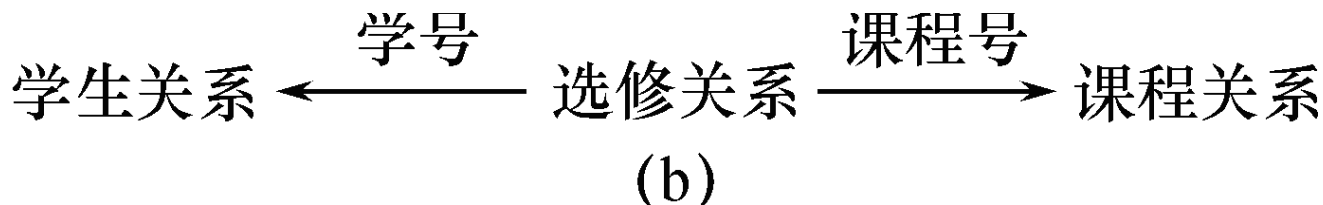
57

- [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 (学号, 课程号, 成绩)

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)

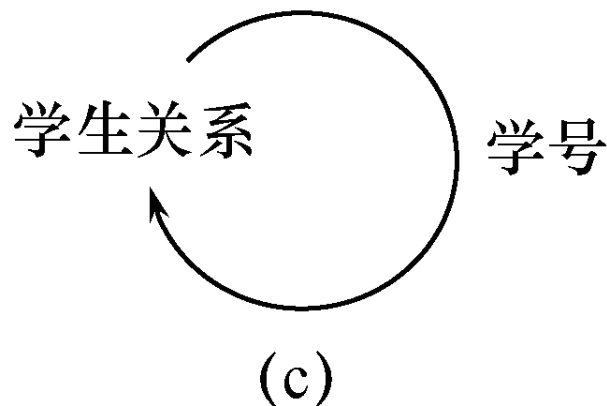
58

□ [例3] :

学生 (学号, 姓名, 性别, 专业号, 年龄, 班长)

“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(小结)

59

- 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
 - 可以改名，但含义相同，但
 - 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. 参照完整性规则

60

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



参照完整性规则(续)

61

[例2.1]: **学生** (学号, 姓名, 性别, **专业号**, 年龄)
专业 (专业号, 专业名)

学生关系中每个元组的“**专业号**”属性只取两类值:

- (1) **空值**, 表示尚未给该学生分配专业
- (2) 非空值, 这时该值必须是**专业关系**中某个元组的“**专业号**”值, 表示该学生不可能分配一个不存在的专业



参照完整性规则(续)

62

〔例2.2〕：

- 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（学号，课程号，成绩）

“学号”和“课程号”可能的取值：

- (1) 选修关系中的主属性，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



参照完整性规则(续)

63

例2.3) :

学生 (学号, 姓名, 性别, 专业号, 年龄, 班长)

“班长”属性值可以取两类值:

- (1) 空值, 表示该学生所在班级尚未选出班长
- (2) 非空值, 该值必须是本学生关系中某个元组的学号值



完整性规则示例

64

建立一个“课程”表Course

```
CREATE TABLE Course
```

```
(Cno      CHAR(4) PRIMARY KEY,
```

```
  Cname   CHAR(40),
```

```
  Cpno,   CHAR(4),
```

```
  Ccredit INT,
```

```
  FOREIGN KEY Cpno REFERENCE Course(Cno)
```

```
)
```




关系的完整性(续)

65

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

66

- 用户定义完整性：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
 - 例如，课程关系中“课程名”可以要求“不能取空值”
- 关系模型应提供定义和检验这类完整性的机制，以使用统一的方法处理它们，而不要由应用程序承担这一功能
 - 例如，插入一条记录，需要检查其中的学号是否等于学生关系中的某个学号。若等于，则插入记录。否则，则拒绝



用户定义的完整性(续)

67

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}



练习

68

□ 一个关系只有一个（ ）。

A. 候选码 B. 外码 C. 超码 D. 主码

□ 现有如下关系：

患者（患者编号，患者姓名，性别，出生日期，所在单位）

医疗（患者编号，患者姓名，医生编号，医生姓名，诊断日期，诊断结果）

医生（医生编号，医生姓名，性别，出生日期，所在科室，职称）

➤ 其中，医疗关系中的外码是（ ）。

A. 患者编号 B. 患者姓名
C. 患者编号和患者姓名 D. 医生编号和患者编号

➤ 医疗关系是（ ），患者关系是（ ），医生关系是（ ）

A. 参照关系 B. 参考关系 C. 被参照关系 D. 被参考关系



练习

69

- 现有一个关系：借阅（书号，书名，库存数，读者号，借期，还期），假如同一本书允许一个读者多次借阅，但读者不能同时对一种书借多本，
则该关系模式的主码是（ ），为什么？
A. 书号 B. 读者号 C. 书号+读者号 D. 书号+读者号+借期
- 下面的选项不是关系数据库基本特征的是（ ）。
A. 不同的列应有不同的数据类型 B. 不同的列应有不同的列名
C. 与行的次序无关 D. 与列的次序无关
- 有一张职工表，里面含有“性别”属性，要求只能取“男”或“女”，这属于（ ），职工ID不能取空，这属于（ ）
A. 实体完整性 B. 参照完整性
C. 用户自定义的完整性 D. 关系不变性



第二章 关系数据库

73

2.1 关系模型概述

2.2 关系数据结构

2.3 关系的完整性

2.4 关系代数

2.5 关系演算

2.6 小结



2.4 关系代数

74

- 概述
- 传统的集合运算
- 专门的关系运算



概述

75

□ 关系代数概述

- 抽象的查询语言，用对关系的运算来表达查询
- 运算的三要素
 - 运算对象：关系
 - 运算结果：关系
 - （关系代数）运算符：集合运算符、专门的关系运算符
- 传统的集合运算
 - 将关系看成元组的集合，“水平”方向，行的角度
- 专门的关系运算
 - 涉及行，也涉及列
 - 还包含：比较运算符和逻辑运算符



概述

76

表2.4 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 |



表2.4 关系代数运算符（续）

| 运算符 | | 含义 | 运算符 | | 含义 |
|-------|--------|------|-------|----------|----|
| 比较运算符 | $>$ | 大于 | 逻辑运算符 | \neg | 非 |
| | \geq | 大于等于 | | \wedge | 与 |
| | $<$ | 小于 | | \vee | 或 |
| | \leq | 小于等于 | | | |
| | $=$ | 等于 | | | |
| | \neq | 不等于 | | | |



2.4 关系代数

78

- 概述
- 传统的集合运算
- 专门的关系运算



1. 并 (Union)

79

□ 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
 - 例如, 学生和课程Union

□ $R \cup S$

- 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)

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 |



2. 差 (Difference)

81

□ 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

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



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 |



3. 交 (Intersection)

83

- 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

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

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



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 |



4. 笛卡尔积 (Cartesian Product)

85

- 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: $(n+m)$ 目关系, $(n+m)$ 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
 - $R \times S = \{ \overbrace{t_r t_s} \mid t_r \in R \wedge t_s \in S \}$



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

87

- 概述
- 传统的集合运算
- 专门的关系运算



2.4.2 专门的关系运算

88

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

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



专门的关系运算(续)

89

(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。(R 关系中)



专门的关系运算(续)

90

(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(Concatenation)

或元组的串接。

$\widehat{t_r t_s}$ 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。

(注意: 存在相同的属性(来自相同的域))



专门的关系运算(续)

91

(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：

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

这里 $t[X]$ 表示元组 t 中相应于属性 X 的一个分量

一个理解：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)

92

| 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

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

□ x_1 在 R 中的象集

$$Z_{x1} = \{Z1, Z2, Z3\},$$

□ x_2 在 R 中的象集

$$Z_{x2} = \{Z2, Z3\},$$

□ x_3 在 R 中的象集

$$Z_{x3} = \{Z1, Z3\}$$



专门的关系运算(续)

93

- 选择 (select)
- 投影 (project)
- 连接 (join)
- 除 (division)



专门的关系运算(续)

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 |
|-----------|-------------|------------|------------|--------------|
| 201215121 | 李勇 | 男 | 20 | CS |
| 201215122 | 刘晨 | 女 | 19 | IS |
| 201215123 | 王敏 | 女 | 18 | MA |
| 201215125 | 张立 | 男 | 19 | IS |

SC

| 学号 Sno | 课程号 Cno | 成绩 Grade |
|-----------|------------|-------------|
| 201215121 | 1 | 92 |
| 201215121 | 2 | 85 |
| 201215121 | 3 | 88 |
| 201215122 | 2 | 90 |
| 201215122 | 3 | 80 |

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 | C语言 | 6 | 4 |



1. 选择 (Selection)

95

- 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

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

- F : 选择条件, 是一个逻辑表达式,

- 基本形式为: $X_1 \theta Y_1$

- $X_1 \theta Y_1$

- θ 表示 比较运算符
 - X_1, Y_1 为属性名、或常量
 - 注意: 运算符的优先级

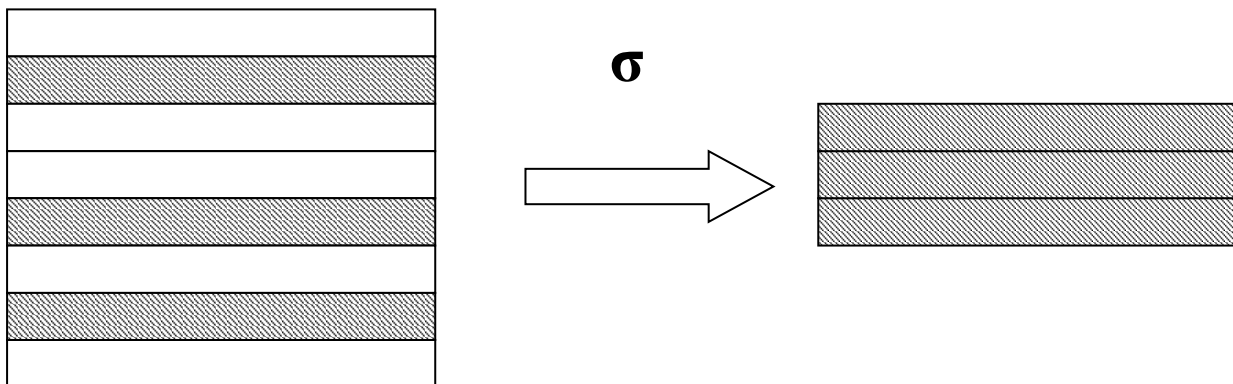
| 运算符 | | 含义 | 运算符 | | 含义 |
|-------|--------|------|-------|----------|----|
| 比较运算符 | > | 大于 | 逻辑运算符 | \neg | 非 |
| | \geq | 大于等于 | | \wedge | 与 |
| | < | 小于 | | \vee | 或 |
| | \leq | 小于等于 | | | |
| | = | 等于 | | | |
| | < | 不等于 | | | |
| | > | | | | |



选择（续）

96

- 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）

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 |
|-----------|-------------|------------|------------|--------------|
| 201215121 | 李勇 | 男 | 20 | CS |
| 201215122 | 刘晨 | 女 | 19 | IS |
| 201215123 | 王敏 | 女 | 18 | MA |
| 201215125 | 张立 | 男 | 19 | IS |



选择（续）

98

[例2.4] 查询信息系（IS系）全体学生

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

结果：

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



选择（续）

| Sno | Sname | Ssex | Sage | Sdept |
|-----------|-------|------|------|-------|
| 201215121 | 李勇 | 男 | 20 | CS |
| 201215122 | 刘晨 | 女 | 19 | IS |
| 201215123 | 王敏 | 女 | 18 | MA |
| 201215125 | 张立 | 男 | 19 | IS |

99

[例2.5] 查询年龄小于20岁的学生

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

结果:

| Sno | Sname | Ssex | Sage | Sdept |
|-----------|-------|------|------|-------|
| 201215122 | 刘晨 | 女 | 19 | IS |
| 201215123 | 王敏 | 女 | 18 | MA |
| 201215125 | 张立 | 男 | 19 | IS |



2. 投影 (Projection)

100

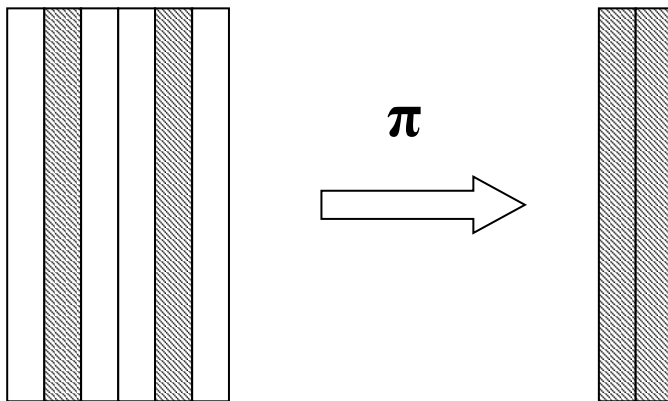
□ 1) 投影运算符的含义

□ 从 R 中选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

□ 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,元组数可能减少）



专门的关系运算(续)

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 |
|-----------|-------------|------------|------------|--------------|
| 201215121 | 李勇 | 男 | 20 | CS |
| 201215122 | 刘晨 | 女 | 19 | IS |
| 201215123 | 王敏 | 女 | 18 | MA |
| 201215125 | 张立 | 男 | 19 | IS |

SC

| 学号 Sno | 课程号 Cno | 成绩 Grade |
|-----------|------------|-------------|
| 201215121 | 1 | 92 |
| 201215121 | 2 | 85 |
| 201215121 | 3 | 88 |
| 201215122 | 2 | 90 |
| 201215122 | 3 | 80 |

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 | C语言 | 6 | 4 |



投影（续）

102

- [例2.6] 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

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

结果:

| Sname | Sdept |
|-------|-------|
| 李勇 | CS |
| 刘晨 | IS |
| 王敏 | MA |
| 张立 | IS |



投影（续）

103

[例2.7] 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)

104

□ 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

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

➤ A 和 B : 分别为 R 和 S 上**列数相等且可比**的属性组

➤ θ : 比较运算符

□ 一种理解: 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 的元组作为结果元组



连接(续)

105

□ 3) 两类常用连接运算

□ 1. 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)

106

□ 2. 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 , Z 是 R 和 S 的全体属性集合

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

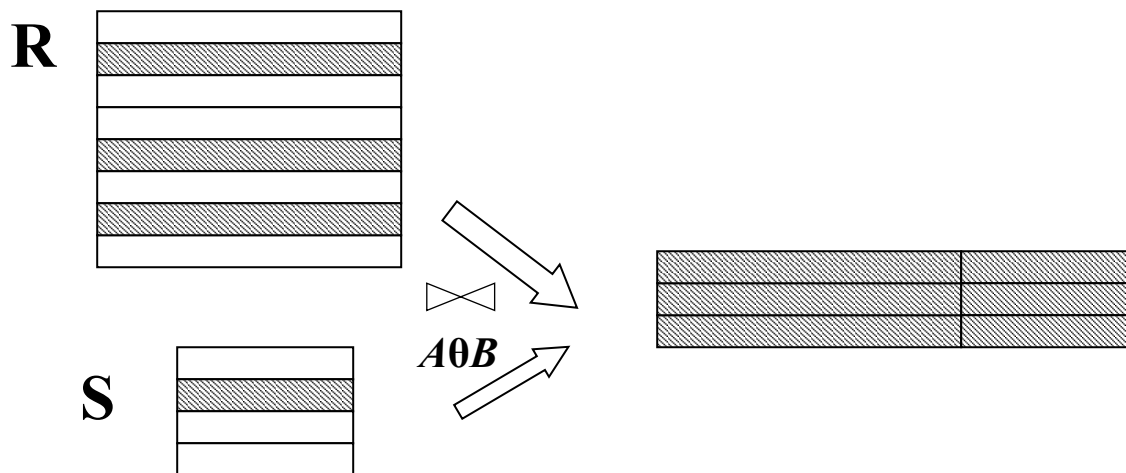
注意：自然连接与等值连接的区别



连接(续)

107

- 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**，所以是同时**从行和列**的角度进行运算。



连接(续)

108

□ [例5]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 | | | S | |
|----------------|----------------|----|----------------|----|
| A | B | C | 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 ₄ | 12 | b ₃ | 2 |
| | | | b ₅ | 2 |

| $R \bowtie_{C < E} S$ | | | | |
|-----------------------|----------------|---|----------------|----|
| 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)

110

等值连接 $R \bowtie_{R.B=S.B} S$ 的结果如下:

| 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 |

| 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)

111

自然连接 $R \bowtie S$ 的结果如下:

| <i>R</i> | | | <i>S</i> | |
|----------|----------|----------|----------|----------|
| <i>A</i> | <i>B</i> | <i>C</i> | <i>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_4 | 12 | b_3 | 2 |
| | | | b_5 | 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 |

悬浮元组 (dangling tuple)



连接(续)

112

□ 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。



连接(续)

113

□ 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)。记作：P55页

□ 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)

114

下图是例5中关系*R*和关系*S*的外连接

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

| <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 |
| NULL | <i>b</i> ₅ | NULL | 2 |

(a) 外连接



连接(续)

115

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

| R | | | S | |
|----------------|----------------|----|----------------|----|
| A | B | C | 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 |
| a ₂ | b ₄ | 12 | b ₅ | 2 |

| 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



4. 除 (Division)

116

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 在 R 中的象集, $x = t_r[X]$

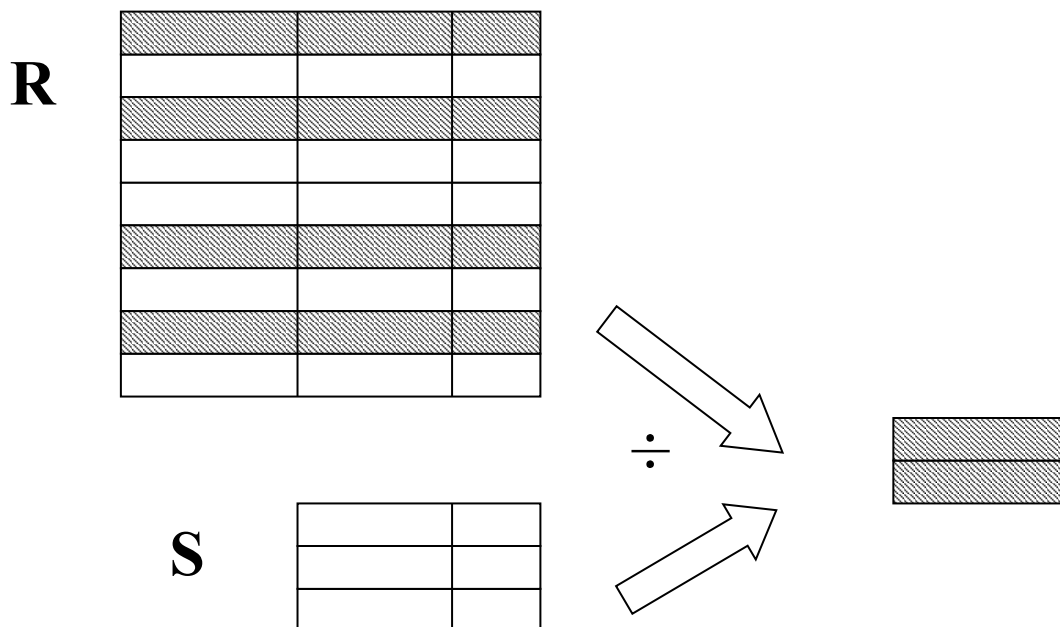
一种理解: $P(X)$ 包含所有在 R 但不在 S 中的属性 X 及其值的元组,
而且 $P(X)$ 的元组与 S 的元组的所有组合都在 R 中



除(续)

119

- 2) 除操作是同时从行和列角度进行运算





除(续)

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

Y_x : x 在 R 中的象集, $x = t_r[X]$

120

[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 |
| 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)

❖ 在关系 R 中, A 可以取四个值

$\{a_1, a_2, a_3, a_4\}$

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



5. 综合举例

121

以学生-课程数据库为例 (P56)

[例2.10] 查询至少选修1号课程和3号课程的学生号码

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 |

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 |

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 | C语言 | 6 | 4 |

3/16/2024



5. 综合举例

122

以学生-课程数据库为例 (P56)

[例2.10] 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)

K=

Cno

1

3

□ [例2.10] $\pi_{Sno,Cno}(SC)$

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)

124

[例 2.11] 查询选修了2号课程的学生们的学号。

$$\pi_{\text{Sno}} (\sigma_{\text{Cno}='2'} (\text{SC}))$$
$$= \{ 200215121, 200215122 \}$$

请写出其他表达式？

| 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 |



综合举例(续)

125

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名
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 |

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 |

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 | C语言 | 6 | 4 |



综合举例(续)

126

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno, Sname}(Student))$

Step1: $\sigma_{Cpno='5'}(Course)$

| 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit |
|------------|--------------|-------------|---------------|
| 1 | 数据库 | 5 | 4 |

| 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 | C语言 | 6 | 4 |



综合举例(续)

127

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno, Sname}(Student))$

Step2: $\sigma_{Cpno='5'}(Course) \bowtie SC$

| 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit |
|------------|--------------|-------------|---------------|
| 1 | 数据库 | 5 | 4 |



| 学号 Sno | 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit | 成绩 Grade |
|-----------|------------|--------------|-------------|---------------|-------------|
| 200215121 | 1 | 数据库 | 5 | 4 | 92 |

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 |



综合举例(续)

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno, Sname}(Student))$

Step3: $\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno, Sname}(Student)$

$\pi_{Sno, Sname}(Student)$



| 学号 Sno | 姓名 Sname |
|-----------|-------------|
| 200215121 | 李勇 |
| 200215122 | 刘晨 |
| 200215123 | 王敏 |
| 200215125 | 张立 |

| 学号 Sno | 姓名 Sname | 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit | 成绩 Grade |
|-----------|-------------|------------|--------------|-------------|---------------|-------------|
| 200215121 | 李勇 | 1 | 数据库 | 5 | 4 | 92 |



综合举例(续)

129

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno, Sname}(Student))$

Step4: $\pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \pi_{Sno, Sname}(Student))$

| 学号 Sno | 姓名 Sname | 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit | 成绩 Grade |
|---------------|-------------|------------|--------------|-------------|---------------|-------------|
| 200215 121 | 李勇 | 1 | 数据库 | 5 | 4 | 92 |

| |
|-------------------|
| 姓名 Sname 李勇 |
|-------------------|



综合举例(续)

130

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

方式1: $\pi_{\text{Sname}}(\sigma_{\text{Cpno}='5'}(\text{Course}) \bowtie \text{SC} \bowtie \pi_{\text{Sno}, \text{Sname}}(\text{Student}))$

方式2: $\pi_{\text{Sname}}(\pi_{\text{Sno}}(\sigma_{\text{Cpno}='5'}(\text{Course}) \bowtie \text{SC}) \bowtie \pi_{\text{Sno}, \text{Sname}}(\text{Student}))$

方式3: $\pi_{\text{Sname}}(\sigma_{\text{Cpno}='5'}(\text{Course} \bowtie \text{SC} \bowtie \text{Student}))$



综合举例(续)

131

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

方法3: $\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(Course) \bowtie SC) \bowtie \pi_{Sno, Sname}(Student))$

Step1: $\sigma_{Cpno='5'}(Course)$

| 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit |
|------------|--------------|-------------|---------------|
| 1 | 数据库 | 5 | 4 |

| 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 | C语言 | 6 | 4 |



综合举例(续)

132

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname} (\pi_{Sno} (\sigma_{Cpno='5'} (Course) \bowtie SC) \bowtie \pi_{Sno, Sname} (Student))$

Step2: $\sigma_{Cpno='5'} (Course) \bowtie SC$

| 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit |
|------------|--------------|-------------|---------------|
| 1 | 数据库 | 5 | 4 |



| 学号 Sno | 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit | 成绩 Grade |
|-----------|------------|--------------|-------------|---------------|-------------|
| 200215121 | 1 | 数据库 | 5 | 4 | 92 |

| 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 |



综合举例(续)

133

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname} (\pi_{Sno} (\sigma_{Cpno='5'} (Course) \bowtie SC) \bowtie \pi_{Sno, Sname} (Student))$

Step3: $\pi_{Sno} (\sigma_{Cpno='5'} (Course) \bowtie SC)$

| 学号 Sno |
|-----------|
| 200215121 |



| 学号 Sno | 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit | 成绩 Grade |
|-----------|------------|--------------|-------------|---------------|-------------|
| 200215121 | 1 | 数据库 | 5 | 4 | 92 |



综合举例(续)

134

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

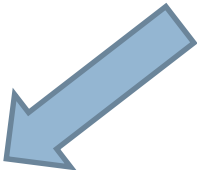
$\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(Course) \bowtie SC) \bowtie \pi_{Sno, Sname}(Student))$

Step4: $\pi_{Sno}(\sigma_{Cpno='5'}(Course) \bowtie SC) \bowtie \pi_{Sno, Sname}(Student)$

| 学号 Sno |
|-----------|
| 200215121 |



| 学号 Sno | 姓名 Sname |
|-----------|-------------|
| 200215121 | 李勇 |



$\pi_{Sno, Sname}(Student)$

| 学号 Sno | 姓名 Sname |
|-----------|-------------|
| 200215121 | 李勇 |
| 200215122 | 刘晨 |
| 200215123 | 王敏 |
| 200215125 | 张立 |



综合举例(续)

135

[例2.12] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(Course) \bowtie SC) \bowtie \pi_{Sno, Sname}(Student))$

Step5: $\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(Course) \bowtie SC) \bowtie \pi_{Sno, Sname}(Student))$

| 学号 Sno | 姓名 Sname |
|-----------|-------------|
| 200215121 | 李勇 |



| 姓名 Sname |
|-------------|
| 李勇 |



综合举例(续)

136

[例2.13] 查询选修了全部课程的学生号码和姓名。

- 问题思考：
1. 找到全部课程（Course表）
 2. 找到选修了全部课程的学生学号（SC表）；
 3. 找到相应的学生姓名（Student表）



专门的关系运算(续)

[例2.13] 查询选修了全部课程的学生号码和姓名。

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 |

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 |

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 | C语言 | 6 | 4 |



综合举例(续)

138

[例2.13] 查询选修了全部课程的学生号码和姓名。

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



综合举例(续)

[例2.13] 查询选修了全部课程的学生号码和姓名。

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

Step1: $\pi_{Cno} (Course)$

Course

| 课程号 Cno | 课程名 Cname | 先行课 Cpno | 学分 Ccredit | 课程号 Cno |
|------------|--------------|-------------|---------------|------------|
| 1 | 数据库 | 5 | 4 | 1 |
| 2 | 数学 | | 2 | 2 |
| 3 | 信息系统 | 1 | 4 | 3 |
| 4 | 操作系统 | 6 | 3 | 4 |
| 5 | 数据结构 | 7 | 4 | 5 |
| 6 | 数据处理 | | 2 | 6 |
| 7 | C语言 | 6 | 4 | 7 |



综合举例(续)

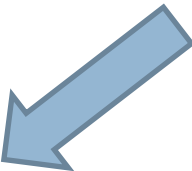
[例2.13] 查询选修了全部课程的学生号码和姓名。

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

修改后的SC

Step2: $\pi_{Sno, Cno} (SC)$

| 学号 Sno | 课程号 Cno |
|-----------|------------|
| 200215121 | 1 |
| 200215121 | 2 |
| 200215121 | 3 |
| 200215121 | 4 |
| 200215121 | 5 |
| 200215121 | 6 |
| 200215121 | 7 |
| 200215122 | 3 |



| 学号 Sno | 课程号 Cno | 成绩 Grade |
|-----------|------------|-------------|
| 200215121 | 1 | 92 |
| 200215121 | 2 | 85 |
| 200215121 | 3 | 88 |
| 200215121 | 4 | 92 |
| 200215121 | 5 | 85 |
| 200215121 | 6 | 88 |
| 200215121 | 7 | 90 |
| 200215122 | 3 | 80 |



综合举例(续)

[例2.13] 查询选修了全部课程的学生号码和姓名。

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

Step3: $\pi_{Sno, Cno} (SC) \div \pi_{Cno} (Course)$

| $\pi_{Sno, Cno} (SC)$ | | $\pi_{Cno} (Course)$ | | | | |
|-----------------------|------------|----------------------|--------------|--------------|-----------|--|
| 学号 Sno | 课程号 Cno | 课程号 Cno | | | 学号 Sno | |
| 200215121 | 1 | 1 | <div>÷</div> | <div>→</div> | 200215121 | |
| 200215121 | 2 | 2 | | | | |
| 200215121 | 3 | 3 | | | | |
| 200215121 | 4 | 4 | | | | |
| 200215121 | 5 | 5 | | | | |
| 200215121 | 6 | 6 | | | | |
| 200215121 | 7 | 7 | | | | |
| 200215122 | 3 | | | | | |



综合举例(续)

142

[例2.13] 查询选修了全部课程的学生号码和姓名。

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

Step4: $\pi_{Sno, Cno} (SC) \div \pi_{Cno} (Course) \bowtie \pi_{Sno, Sname} (Student)$

$\pi_{Sno, Sname} (Student)$

| 学号 Sno |
|-----------|
| 200215121 |



| 学号 Sno | 姓名 Sname |
|-----------|-------------|
| 200215121 | 李勇 |
| 200215122 | 刘晨 |
| 200215123 | 王敏 |
| 200215125 | 张立 |



| 学号 Sno | 姓名 Sname |
|-----------|-------------|
| 200215121 | 李勇 |