

数据库管理系统 原理与实现

杜小勇 陈红 卢卫 主编
中国人民大学
信息学院

第二章 关系数据模型

章成源

湖南大学-信息科学与工程学院-计算机科学系

办公室：院楼403

Email: cyzhangcse@hnu.edu.cn

第二章 关系数据模型

- 2.1 关系模型的数据结构及形式化定义
- 2.2 关系代数
- 2.3 关系的完整性
- 2.4 小结

关系模型的数据结构及形式化定义

内容

关系、关系模式、关系数据库、关系模型的存储结构

目标

理解与掌握关系相关概念，如：域、笛卡尔积、关系、关系模式、候选、码、主码、主属性、非主属性；

理解和掌握关系模式相关概念；

了解关系数据库和关系模型的存储结构

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

- **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

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

关系

域是一组具有相同数据类型的值的集合。例：

- 自然数、
- 整数
- 实数
- 总字符数度小于25的变长字符串集合
- {男， 女}
- 0~100之间的正整数
- ...

学号	姓名	性别	课程	成绩
202101	张三	女	数据库	96
202102	李四	男	计算机	89
202103	王五	男	大数据	91

笛卡尔积

- 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:

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

- 所有域的所有取值的一个组合
- 不能重复

关系

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
- 例如: (1001, 张三, 北京), (1001, 张三, 上海) 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
- 例如: “1001”, “张三”, “北京”等都是分量

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：

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

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

关系

例如，给出3个域：

- $D_1 = \text{顾客ID (Customer ID)} = \{1001, 1002, 1003\}$
- $D_2 = \text{顾客姓名 (Customer Name)} = \{\text{张三, 李四, 王五}\}$
- $D_3 = \text{城市 (City)} = \{\text{北京, 上海, 广州}\}$
- D_1, D_2, D_3 的笛卡尔积为

关系

- **D1×D2×D3 = {**

(1001, 张三, 北京), (1001, 张三, 上海), (1001, 张三, 广州), (1001, 李四, 北京),
, (1001, 李四, 上海), (1001, 李四, 广州), (1001, 王五, 北京), (1001, 王五, 上海)
, (1001, 王五, 广州), (1002, 张三, 北京), (1002, 张三, 上海), (1002, 张三, 广
州), (1002, 李四, 北京), (1002, 李四, 上海), (1002, 李四, 广州), (1002, 王五,
北京), (1002, 王五, 上海), (1002, 王五, 广州), (1003, 张三, 北京), (1003, 张三
, 上海), (1003, 张三, 广州), (1003, 李四, 北京), (1003, 李四, 上海), (1003, 李
四, 广州), (1003, 王五, 北京), (1003, 王五, 上海), (1003, 王五, 广州) }。

- 其中, (1001, 张三, 北京), (1001, 张三, 上海) 等都是元组, “1001”, “张三”, “北京”等
都是元组的分量。
- 基数为 $3*3*3=27$

关系

笛卡儿积示例

顾客ID	顾客姓名	城市
1001	张三	北京
1001	张三	上海
1001	张三	广州
1001	李四	北京
1001	李四	上海
1001	李四	广州
1001	王五	北京
1001	王五	上海
1001	王五	广州
1002	张三	北京
.....

关系

D_1, D_2, \dots, D_n 的笛卡尔积的某个子集才有实际含义

例：上述的笛卡尔积没有实际意义，我们取出有实际意义的元组来构造关系如下：

下表展示了顾客编号与顾客姓名和所在城市的关系。将该关系取名为Customers

Customers关系

顾客ID (CID)	顾客姓名 (CName)	城市 (City)
1001	张三	北京
1002	李四	广州
1003	王五	上海

关系

关系

- 给定一组域 D_1, D_2, \dots, D_n , 允许其中某些域是相同的, $D_1 \times D_2 \times \dots \times D_n$ 的笛卡儿积 $D_1 \times D_2 \times \dots \times D_n$ 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)

关系

- 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）

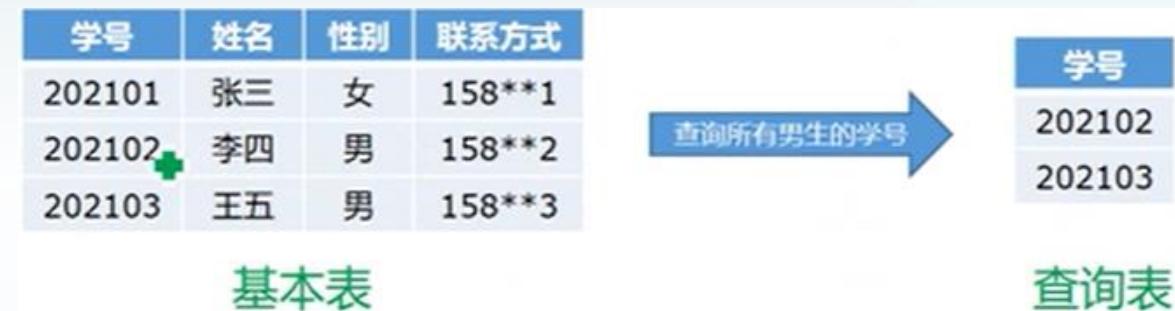
关系

- **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
- **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

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--- 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

当关系作为数据模型的数据结构时，需要对其进行必要的限定并增加属性名以便于记忆和使用

- ① 无限关系在数据库系统中是无意义的。因此，限定关系模型中的关系必须是有限集合。
- ② 通过为关系的每个列附加一个属性名的方法取消关系属性的有序性，使得列的次序可以任意交换：

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

关系

基本关系的性质

- ① 列是同质的 (Homogeneous)
- 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不能相同
- ⑤ 行的顺序无所谓，行的次序可以任意交换
- 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	

非规范化关系表

小表

关系模型的数据结构及形式化定义

- 2.1 关系
- 2.2 关系模式
- 2.3 关系数据库
- 2.4 关系模型的存储结构

关系模式

- **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

关系模式

- **关系模式**---对关系的描述
- 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 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

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象常常直接说明为
属性的类型、长度

- 关系模式和关系往往笼统称为关系

通过上下文加以区别

定义关系模式（续）

例：

导师和研究生出自同一个域——人，
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-PERSON)
= DOM (POSTGRADUATE-PERSON)
= PERSON

3. 关系模式与关系

关系模式可以形式化地表示为： $R (U, D, DOM, F)$

R 关系名; U 为组成该关系的属性名集合; D 为 U 中属性所来自的域; DOM 为属性向域的映象集合; F 为属性间数据的依赖关系的集合

学生成绩表				
学号	姓名	语文	数学	英语
202101	张三	96	95	62
202102	李四	90	94	90
202103	王五	75	82	84

R : 学生成绩表

U : {学号, 姓名, 语文, 数学, 英语}

D :

D_1 -学号集合

D_2 -姓名集合

D_3 -0~100的整数

DOM :

学号- D_1

姓名- D_2

语文、数学、英语- D_3

F : 学号->姓名, 语文, 数学, 英语

关系

- 码

候选码（Candidate key）

- 关系模式中的某一个属性或一组属性的值能唯一地标识一个元组，而它的真子集不能唯一地标识一个元组

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

- 码

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），或称主键。

主属性

候选码的诸属性称为主属性（Prime attribute）

不包含在任何候选码中的属性称为非主属性（Non-Prime attribute）或非码属性（Non-key attribute）

学号	姓名	性别	联系方式
202101	张三	女	158**1
202102	李四	男	158**2
202103	王五	男	158**3

两个候选码:学号,联系方式
主码:学号
两个主属性:学号,联系方式
两个非主属性:姓名,性别

课堂思考

成绩表：

表示学生
和课程之
间的联系

学号	课程号	成绩
2003002	A002	78
2003002	B025	88
2003002	A005	93
2003025	A002	90
2003025	B025	78
2003025	A005	77
2003025	B028	88
2004033	A002	87
2004033	B025	83
2005023	A005	79
2005023	B028	92
2005023	B025	89

成绩表中是否有候选码？

候选码是否是全码？

候选码是什么？



关系模型的数据结构及形式化定义

- 2.1 关系
- 2.2 关系模式
- 2.3 关系数据库
- 2.4 关系模型的存储结构

关系数据库

- 关系数据库系统
 - 支持关系模型的数据库系统
 - 在关系模型中，实体以及实体间的联系都是用关系来表示的。
 - 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。

Customers关系

顾客ID (CID)	顾客姓名 (CName)	城市 (City)
1001	张三	北京
1002	李四	广州
1003	王五	上海

关系数据库

- **关系数据库的型与值**

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

关系模型的数据结构及形式化定义

- 2.1 关系
- 2.2 关系模式
- 2.3 关系数据库
- 2.4 关系模型的存储结构

关系模型的存储结构

- **关系数据库的物理组织**

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。

本书将在**第5章**介绍基于磁盘的数据库的组织与存储。

- **关系模型是关系数据的逻辑结构**

- 用关系数据定义语言描述。支持关系模型的关系数据库管理系统（RDBMS）将以一定的组织方式存储和管理数据，即设计和实现关系模型的存储结构，这是关系数据库管理系统的核功能之一。

第二章 关系数据模型

- 2.1 关系模型的数据结构及形式化定义
- 2.2 **关系代数**
- 2.3 关系的完整性
- 2.4 小结

关系代数

内容

传统的集合运算（并、差、交和笛卡尔积）、专门的关系运算（选择、投影、连接、除运算）

目标

理解并掌握传统的集合运算（并、差、交和笛卡尔积）、专门的关系运算（选择、投影、连接、除运算）

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
- 关系代数运算三要素
 - 运算对象
 - 运算结果
 - 运算符
 - 关系代数的运算符有两类：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

- 传统的集合运算是二目运算，包括并差交笛卡尔积4种运算。其将关系看成元组的集合，其运算是从关系的“水平”方向，即行的角度进行。
- 前提条件：R和S具有相同的目n（即两个关系都有n个属性），且相应的属性取自同一个域， n 是元祖变量， $t \in R$ ，表示t是R的一个元组

关系代数

关系代数运算符

运 算 符	含 义
集合 运算符	\cup
	-
	\cap
	\times
专门的 关系 运算符	σ
	π
	\bowtie
	\div

关系代数

- 2.2.1 传统的集合运算
- 2.2.2 专门的关系运算

2.4.1 传统的集合运算

- 传统的集合运算是二目运算，包括并差交笛卡尔积4种运算。其将关系看成元组的集合，其运算是从关系的“水平”方向，即行的角度进行。
- 前提条件：R和S具有相同的目n（即两个关系都有n个属性），且相应的属性取自同一个域， $t \in R$ ，表示t是R的一个元组

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

$$R \cup S = \{t | t_r \in R \vee t_s \in S\}$$

并 (Union)

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

差 (Difference)

- 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
- $R - S$
 - 仍为 n 目关系, 由属于 R 而不属于 S 的所有元组组成

$$R - S = \{t | t_r \in R \wedge t_s \notin S\}$$

差 (Difference)

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

交 (Intersection)

- 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
- $R \cap S$
 - 仍为 n 目关系, 由既属于 R 又属于 S 的元组组成

$$R \cap S = \{t | t_r \in R \wedge t_s \in S\}$$

交 (Intersection)

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

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

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

笛卡尔积 (Cartesian Product)

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

关系代数

- 2.2.1 传统的集合运算
- 2.2.2 专门的关系运算

专门的关系运算

先引入几个记号

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

设关系模式为 $R(A_1, A_2, \dots, A_n)$

它的一个关系设为 R

$t \in R$ 表示 t 是 R 的一个元组

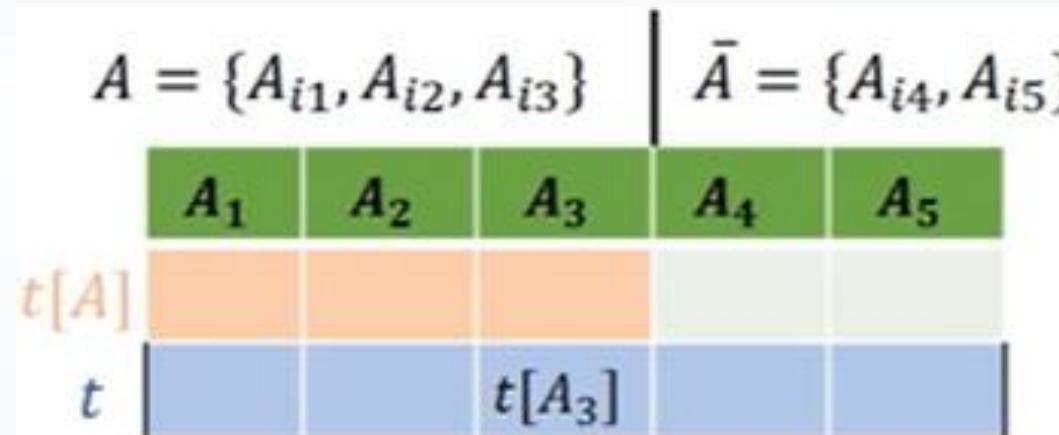
$t[A_i]$ 则表示元组 t 中相应于属性 A_i 的一个分量

$A = \{A_{i1}, A_{i2}, A_{i3}\}$			$\bar{A} = \{A_{i4}, A_{i5}\}$	
A_1	A_2	A_3	A_4	A_5
$t[A]$				
t			$t[A_3]$	

专门的关系运算

(2) A, t[A], \bar{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。
- \bar{A} 则表示 $\{A_1, A_2, \dots, A_n\}$ 中去掉 $\{A_{i1}, A_{i2}, \dots, A_{ik}\}$ 后剩余的属性组。



专门的关系运算

(3) $\widehat{t_r t_s}$

- R 为 n 目关系, S 为 m 目关系。
- $t_r \in R$, $t_s \in S$, $\widehat{t_r t_s}$ 称为元组的连接。
- $\widehat{t_r t_s}$ 是一个 $n + m$ 列的元组, 前 n 个分量为 R 中的一个 n 元组, 后 m 个分量为 S 中的一个 m 元组。

专门的关系运算

(4) 象集 Z_x

给定一个关系 $R(X, Z)$ ，
 X 和 Z 为属性组。

当 $t[X]=x$ 时， x 在 R 中的象集 (Images Set) 为：

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

它表示 R 中属性组 X 上值为 x 的诸元组在 Z 上分量的集合

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
 $Z_{x1} = \{Z_1, Z_2, Z_3\}$,
- x_2 在 R 中的象集
 $Z_{x2} = \{Z_2, Z_3\}$,
- x_3 在 R 中的象集
 $Z_{x3} = \{Z_1, Z_3\}$

专门的关系运算

当 $t[X]=x= (a, b, c)$ 时

$$Z_x = \{ (f, g), (l, k), (d, f) \}$$

当 $t[X]=x= (l, m, n)$ 时

$$Z_x = \{ (f, d), (s, n) \}$$

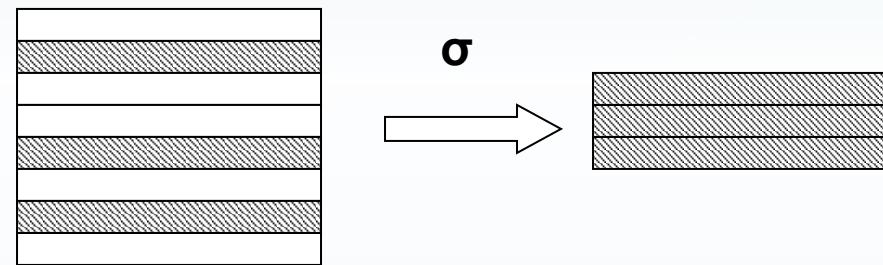
X			Z	
X ₁	X ₂	X ₃	Z ₁	Z ₂
a	b	c	f	g
a	b	c	l	k
a	b	c	d	f
l	n	m	f	d
l	n	m	s	n

选择

- 在关系 R 中选择满足给定条件的诸元组，记作

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

- 其中 F 表示选择条件，它是一个逻辑表达式，取逻辑值“True”或“False”。选择运算实际上是从关系 R 中选取使逻辑表达式 F 为“True”的元组，是从行角度进行的运算。



选择

- 【例】查询种类（Category）为‘数码产品’的所有商品。

$$\sigma_{\text{Category}=\text{'数码产品'}}(\text{Products})$$

- 结果如下表所示。

商品ID (PID)	商品名称 (PName)	价格 (Price)	类别 (Category)	供应商ID (SID)
P0001	智能手机	1999	数码产品	S001
P0002	老人专用手机	899	数码产品	S001
P0003	平板电脑	1688	数码产品	S001

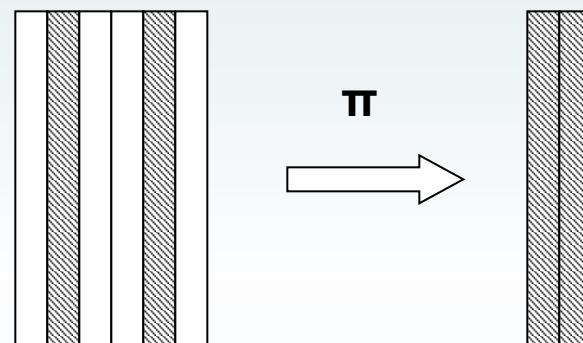
投影

- 关系R上的投影是从R中选择若干属性列组成新的关系。记作

$$\prod_{\mathbf{A}}(R) = \{t[\mathbf{A}] | t \in R\}$$

- 其中A是R的一组属性集合。

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，而且可能取消某些元组（避免重复行）

投影

- [例]查询所有的顾客姓名和所在城市。

$$\prod_{CName, City} (\text{Customers})$$

- 查询结果如下表所示。

查询所有的顾客姓名和所在城市

顾客姓名 (CName)	城市 (City)
张三	北京
李四	广州
王五	上海

连接

- 连接操作是从两个关系的笛卡儿积中选取其属性间满足一定条件的元组。记作

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

- 其中，A和B分别为关系R和S上列数相等且可比的属性列，是比较运算符。连接运算是从笛卡儿积 $R \times S$ 中选取关系R在属性列A上的值与关系S在属性列B上的值满足比较关系 θ 的元组。

连接

- 自然连接是一种特殊的等值连接。它要求两个关系中进行比较的分量必须是同名的属性列，并且在结果中把重复的属性列去掉。即若R和S中具有相同的属性列B，U为R和S的全体属性集合，则自然连接可记作：

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

连接

- 对于顾客关系（Customers,下面简称C）和订单关系（Orders，下面简称O）

Customers关系

顾客ID (CID)	顾客姓名 (CName)	城市 (City)
1001	张三	北京
1002	李四	广州
1003	王五	上海

Orders关系

订单编号 (OID)	顾客ID (CID)	创建时间 (CreateTime)
O001	1001	2023-1-1 18:00:10
O002	1002	2023-1-1 18:20:01
O003	1002	2023-1-5 09:10:20
O004	1003	2023-2-5 19:20:10

连接

C.CID=O.CID

C.CID	CName	City	OID	O.CID	CreateTime
1001	张三	北京	O001	1001	2023-1-1 18:00:10
1002	李四	广州	O002	1002	2023-1-1 18:20:01
1002	李四	广州	O003	1002	2023-1-5 09:10:20
1003	王五	上海	O004	1003	2023-2-5 19:20:10

连接

C.CID>O.CID

C.CID	CName	City	OID	O.CID	CreateTime
1002	李四	广州	O001	1001	2023-1-1 18:00:10
1003	王五	上海	O002	1002	2023-1-1 18:20:01
1003	王五	上海	O003	1002	2023-1-5 09:10:20

连接

自然连接

CID	CName	City	OID	CreateTime
1001	张三	北京	O001	2023-1-1 18:00:10
1002	李四	广州	O002	2023-1-1 18:20:01
1002	李四	广州	O003	2023-1-5 09:10:20
1003	王五	上海	O004	2023-2-5 19:20:10

连接（续）

- 内连接（Dangling tuple）
 - 两个关系做自然连接时，连接的结果是满足条件的元组保留下来，不满足条件的元组被舍弃了。
- 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)

- 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

连接 (续)

❖ 给定关系 **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
b5	2

连接（续）

下图是例2.8中关系R和关系S的外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL
NULL	b5	NULL	2

连接 (续)

图(b)是例2.8中关系R和关系S的左外连接,图(c)是右外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL

图(b)

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
NULL	b5	NULL	2

图(c)

连接（续）

下图是例2.8中关系R和关系S的外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL
NULL	b5	NULL	2

连接 (续)

图(b)是例2.8中关系R和关系S的左外连接,图(c)是右外连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
a2	b4	12	NULL

图(b)

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2
NULL	b5	NULL	2

图(c)

课堂思考

学生-课程数据库

学生关系-Student

Sno	Sname	Ssex	Sage	Sdept
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

课程关系-Course

Cno	Cname	Cpno	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

选修关系-SC

Sno	Cno	Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80



例1：查询所有学生的学号、姓名、课程号及成绩。

$\Pi_{Sno,Sname,Cno,Grade}((Student \bowtie SC))$

课堂思考

学生-课程数据库

学生关系-Student

Sno	Sname	Ssex	Sage	Sdept
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

课程关系-Course

Cno	Cname	Cpno	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

选修关系-SC

Sno	Cno	Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

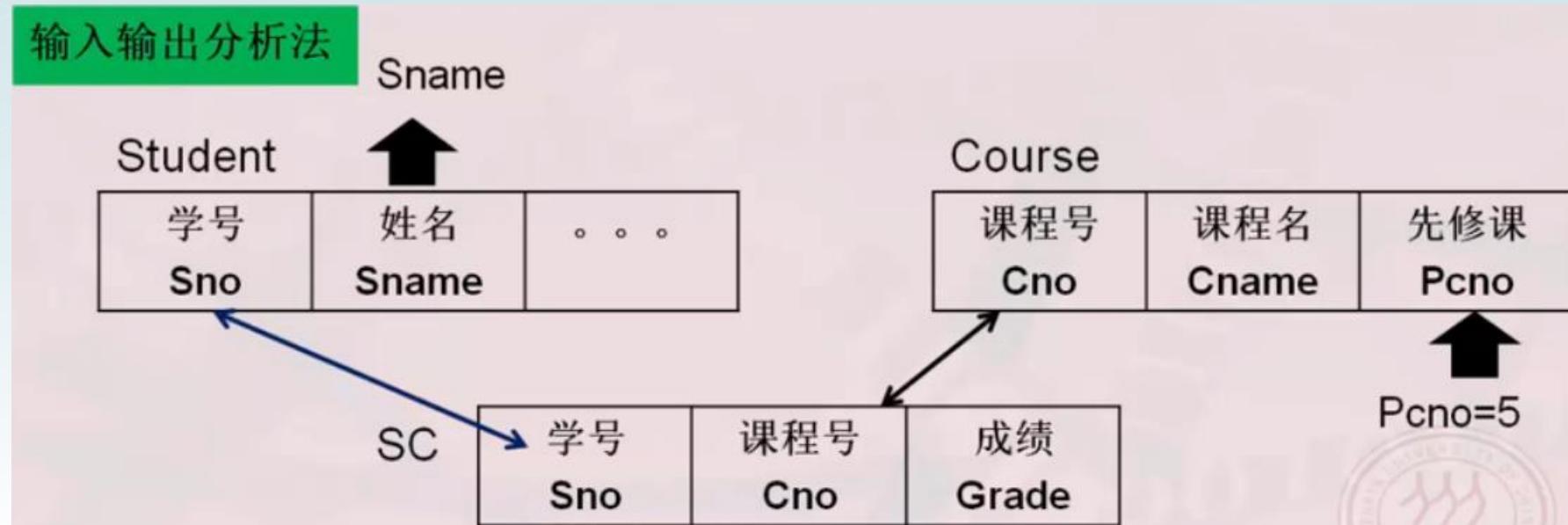
例2：查询CS系的学生的学号、课程号及成绩。

$\Pi_{Sno,Cno,Grade}(\sigma_{Sdept='CS'}(Student \bowtie SC))$

$\Pi_{Sno,Cno,Grade}(\sigma_{Sdept='CS'}(Student) \bowtie SC)$



课堂思考：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？



或

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



4. 除运算 (Division)

设关系R除以关系S的结果为关系T，则T包含所有在R但不在S中的属性及其值，且T的元组与S的元组的所有组合都在R中。

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：

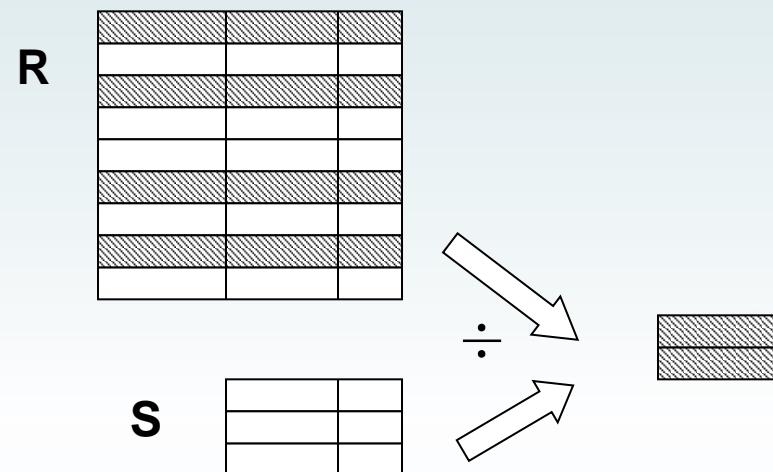
元组在X上分量值x的象集 Y_x 包含S在Y上投影的集合，记作：

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

Y_x : x在R中的象集， $x = t_r[X]$

除运算（续）

- 除操作是同时从行和列角度进行运算



除运算 (续)

[例2.9]设关系R、S分别为下图的(a)和(b), RS的结果为图(c)

R

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

S

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

$R \div S$

A
a1

- 在关系R中, A可以取四个值{a1, a2, a3, a4}
 - a_1 的象集为 $\{(b_1, c_2), (b_2, c_3), (b_2, c_1)\}$
 - a_2 的象集为 $\{(b_3, c_7), (b_2, c_3)\}$
 - a_3 的象集为 $\{(b_4, c_6)\}$
 - a_4 的象集为 $\{(b_6, c_6)\}$
- S在(B, C)上的投影为
 $\{(b1, c2), (b2, c1), (b2, c3)\}$
- 只有 a_1 的象集包含了S在(B, C)属性组上的投影
所以 $R \div S = \{a_1\}$

综合举例

以学生-课程数据库为例

[例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)

- [例2.10]续

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

201215121象集{1, 2, 3}

201215122象集{2, 3}

K={1, 3}

于是： $\pi_{Sno,Cno}(SC) \div K = \{201215121\}$

Sno	Cno
201215121	1
201215121	2
201215121	3
201215122	2
201215122	3

除运算 (续)

什么情况下用到除运算?

解决“查询...全部/所有的....”的问题，比如：

查询选修了全部课程的学生的学号



注意与“查询全部/所有...”区分：

查询所有信息系的学生

课堂思考 查询选修了全部课程的学生号码和姓名

第一步：求出选修了全部课程的学生的学生号码：

$$P(sno, cno) = \pi_{Sno, Cno}(SC) \div \pi_{Cno}(Course)$$

第二步：通过学号求学生姓名

$$P \bowtie Student$$

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



关系代数小结

- 关系代数运算
 - 关系代数运算
 - 并、差、交、笛卡尔积、投影、选择、连接、除
 - 基本运算
 - 并、差、笛卡尔积、投影、选择
 - 交、连接、除
 - 可以用5种基本运算来表达
 - 引进它们并不增加语言的能力，但可以简化表达
- 关系代数表达式
 - 关系代数运算经有限次复合后形成的式子

第二章 关系数据模型

- 2.1 关系模型的数据结构及形式化定义
- 2.2 关系代数
- 2.3 关系的完整性
- 2.4 小结

2.3 关系的完整性

内容

实体完整性、参照完整性、用户定义的完整性

目标

能准确识别输入的SQL语句是否满足实体完整性、参照完整性（外码）、用户定义的完整性原则，如果不满足能指出其存在的问题。

2.3 关系的完整性

- 关系的三类完整性约束：实体完整性、参照完整性和用户定义的完整性
- 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

- 实体完整性目的是保证关系数据库中每个元组是可区分的，并且是唯一的
- 实体完整性约束
 - 若属性A是基本关系R的主属性，则属性A不能取空值
 - 空值就是“不知道”或“不存在”或“无意义”的值
- 例：
 - Customers关系的主码为CID，则CID不可为空。
 - 注意，若关系主码包含多个属性，则每个属性均不能为空。例如，OrderItems关系的主码为(OID, PID)，则两个都不能取空值。

2.3.1 实体完整性

- 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。例如，“Customers”关系对应顾客的集合。

(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。例如，每个顾客都是独立的个体，是不一样的。

(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

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

2.3.2 参照完整性

1. 关系间的引用
2. 外码
3. 参照完整性规则

2.3.2 参照完整性

- 现实世界中的实体之间往往存在某种联系，在关系模型中实体及实体间的联系都是用关系来描述的，这样就自然存在着关系与关系间的引用。
- 例：“供应商”和“商品”可以用下面的关系模式来表示。其中下横线表示该属性为主码。
 - 供应商（供应商ID, 供应商名称, 城市）
 - 商品（商品ID, 商品名称, 价格, 类别, 供应商ID）
 - 商品关系引用了供应商关系的主码“供应商ID”。在商品关系中出现的供应商ID必须是真实存在的供应商的ID，也就是说，商品关系中“供应商ID”的取值要参照供应商关系中“供应商ID”的取值。

2.3.2 参照完整性

- 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 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）
- 在商品关系的例子中，商品关系中的“供应商ID”是外码，参照了供应商关系的主码“供应商ID”。商品关系是参照关系，供应商关系是被参照关系。

2.3.2 参照完整性

- 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
 - 例如，在商品关系的例子中，商品关系中的“供应商ID”或者为空值，或者等于供应商关系中某个元组的“供应商ID”。

参照完整性规则（续）

[例2.1] 学生实体、专业实体

学生 (学号, 姓名, 性别, 专业号, 年龄)

专业 (专业号, 专业名)

[例2.1]中

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）

[例2.2] 中

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）

[例2.3] 中

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，该值必须是本关系中某个元组的学号值

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
- 【例如】
 - 要求供应商名称不能重复
 - 要求商品价格必须大于0
 - 要求订单项关系中的折扣必须是0~1之间的数

2.3 关系的完整性小结

实体完整性

关系的所有主属性都不能为空

参照完整性

相关联的两个表之间的约束

用户定义的完整性

用户根据实际情况定义的约束，例如学生成绩取值在**0-100**之间

课堂思考

- 在选修表中学号、课程号或学分能否为空？为什么？

学生-课程数据库

学生关系-Student

Sno	Sname	Ssex	Sage	Sdept
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

课程关系-Course

Cno	Cname	Cpno	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

选修关系-SC

Sno	Cno	Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80



第二章 关系数据模型

- 2.1 关系模型的数据结构及形式化定义
- 2.2 关系代数
- 2.3 关系的完整性
- 2.4 小结

2.4 小结

- 主要介绍
 - 关系模型的数据结构和形式化定义
 - 关系代数
 - 关系的完整性
- 难点：关系代数，特别是里面的连接运算

本章作业

- 王珊老师《数据库系统概论》第6版
- P70
 - 第4道题
 - 第6道题