

The logo for SQL Server 2005 features a blue and green wave-like background. On the left, there are two spheres: a blue one and a green one. The text "SQL SERVER 2005" is displayed in a stylized, metallic-looking font across the top.

SQL SERVER 2005

数据库系统概论

第 2 章 关系数据库

第2章 关系数据库

- ❖ 第一节 关系数据结构及形式化定义
- ❖ 第二节 关系操作
- ❖ 第三节 关系的完整性
- ❖ 第四节 关系代数



本讲教学目标

❖ 掌握

- 关系的定义、特点；
- 关系的三类完整性约束；
- 传统集合运算和专门的关系运算

❖ 了解

- 关系的操作及其分类、关系演算

❖ 重点

- 关系数据结构、关系的三类完整性约束、关系代数

❖ 难点

- 专门的关系运算

第2章 关系数据库

● 第一节 关系数据结构及形式化定义

❖ 第二节 关系操作

❖ 第三节 关系的完整性

❖ 第四节 关系代数



关系数据结构及形式化定义

❖ 关系

- 域

- 笛卡尔积

- 关系

❖ 关系模式

❖ 关系数据库

域

❖ 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
- 指定长度的字符串集合
- {‘男’， ‘女’ }
- 介于某个取值范围的日期

笛卡尔积

❖ 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：

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

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

- 所有域的所有取值的一个组合
- 其中每一个**元素** (d_1, d_2, \dots, d_n) 叫作一个n元组 (n-tuple) 或简称**元组**(Tuple)
- 元素中的每一个值 d_i 叫做一个**分量**
- 一个域允许的不同取值个数称为这个域的**基数**

例：给出三个域：

$D_1 = \{ \text{张清玫, 刘逸} \}$ 导师集合

$D_2 = \{ \text{计算机专业, 信息专业} \}$ 专业集合

$D_3 = \{ \text{李勇, 刘晨, 王敏} \}$ 研究生集合

则 D_1, D_2, D_3 的笛卡尔积为：

$D_1 \times D_2 \times D_3 =$

{

(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

■ 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

$D1 = \{\text{张清玫、刘逸}\}$ $D2 = \{\text{计算机专业、信息专业}\}$

$D3 = \{\text{李勇、刘晨、王敏}\}$

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

有实际意义吗？

关系

❖ 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 : 关系名 $R(D_1, D_2, \dots, D_n)$
- n : 关系的目或度 (Degree)

例：在表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：专业与导师：1:n，导师与研究生：1:n

于是：SAP关系可以包含三个元组

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	计算机专业	王敏

❖ 关系的术语

- 表(table)、列(column)、行(row)
- 关系(relation)、元组(tuple)、属性(attribute)

关系或表

属性或列

元组或行

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	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

❖ 关系术语

- 候选码 (Candidate key) 全码 (All-key)
- 主码 主属性 非主属性

SC (Sno, Cno, Grade)

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

❖ 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

❖ 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

关系数据结构及形式化定义

❖ 关系

❖ 关系模式

- 什么是关系模式
- 定义关系模式
- 关系模式与关系

❖ 关系数据库

关系模式

关系

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	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

关系
模式

- ❖ 关系模式是对关系的描述，是静态的、稳定的
- ❖ 关系是关系模式在某一时刻的状态或内容，是动态的、随时间不断变化的
- ❖ 关系模式和关系往往统称为关系，通过上下文加以区别

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

定义关系模式

❖ 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

关系数据结构及形式化定义

❖ 关系

❖ 关系模式

❖ 关系数据库



关系数据库

❖ 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

- 关系数据库的型也称关系数据库模式，是对关系数据库的描述
- 关系数据库的值是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

第2章 关系数据库

❖ 第一节 关系数据结构及形式化定义

● 第二节 关系操作

❖ 第三节 关系的完整性

❖ 第四节 关系代数



关系操作

❖ 基本关系操作

❖ 关系数据库语言的分类

基本关系操作

❖ 常用的关系操作

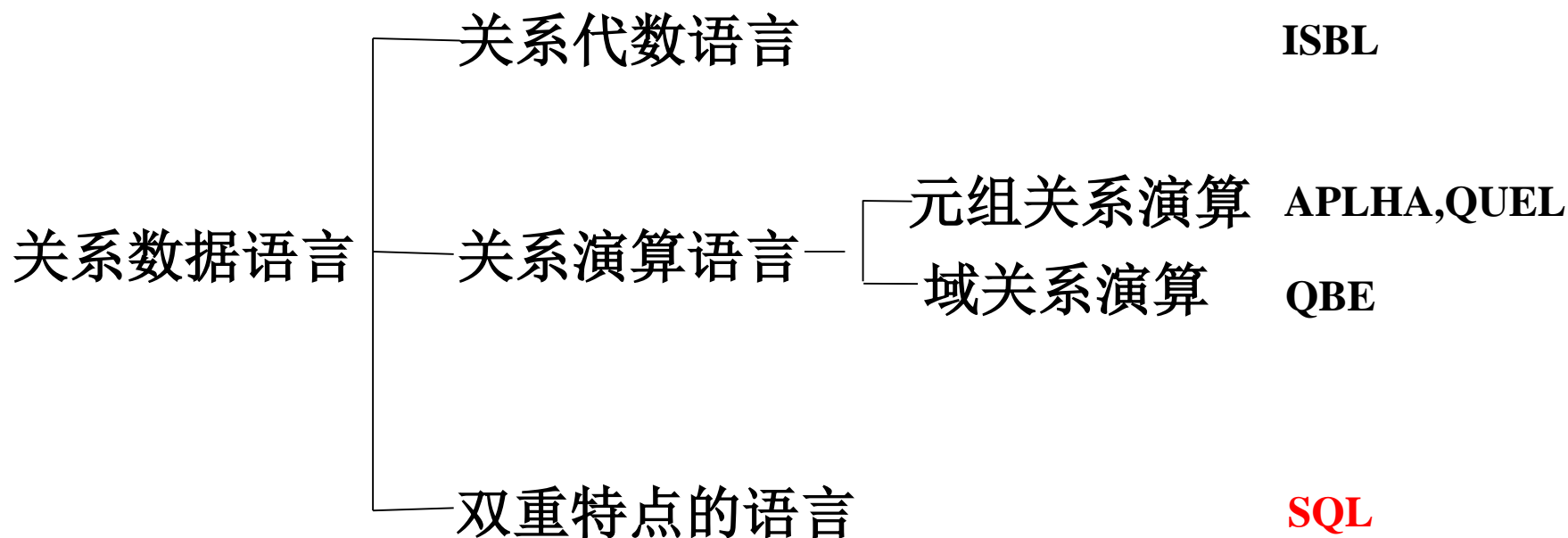
- 查询：选择、投影、连接、除、并、交、差
- 数据更新：插入、删除、修改
- 查询的表达能力是其中最主要的部分
- 选择、投影、并、差、笛卡尔积是5种基本操作

❖ 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

关系数据库语言的分类

■ 关系数据库语言的分类



第2章 关系数据库

- ❖ 第一节 关系数据结构及形式化定义
- ❖ 第二节 关系操作
- 第三节 关系的完整性
- ❖ 第四节 关系代数



第三节 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

用户定义的完整性

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

实体完整性

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

例：学生的选修（学号、课程号、成绩）中学号和课程号为主码，则学号、课程号都是主属性，都不能取空值

SC (Sno, Cno, Grade)

(201215121, 1, 92)

(201215121, null, 92) ❌

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

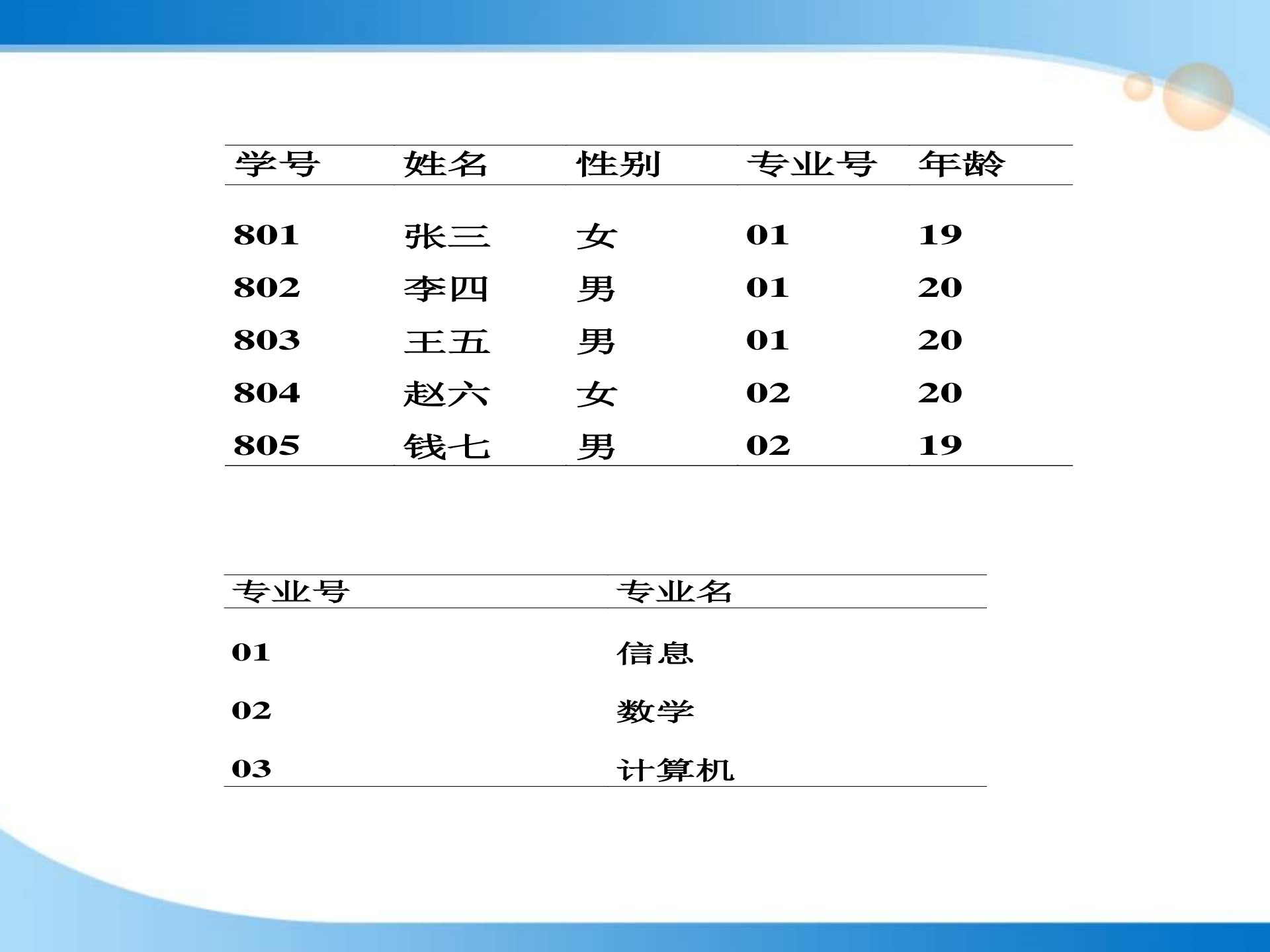
- 关系间的引用
- 外码
- 参照完整性规则

❖ 用户定义的完整性

关系间的引用

- ❖ 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





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

专业号	专业名
01	信息
02	数学
03	计算机

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

学生实体及其内部的一对多联系

[例1] 学生 (学号, 姓名, 性别, 专业号, 年龄, 班长)

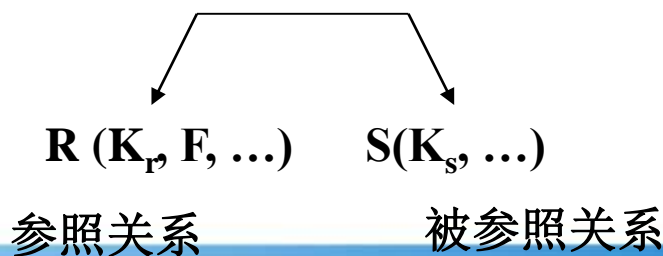
学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	

- “学号”是主码，“班长”引用了本关系的“学号”
- “班长”必须是确实存在的学生的学号

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， K_s 是基本关系 S 的主码。如果 F 与 K_s 相对应，则称 F 是基本关系 R 的**外码**

- 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



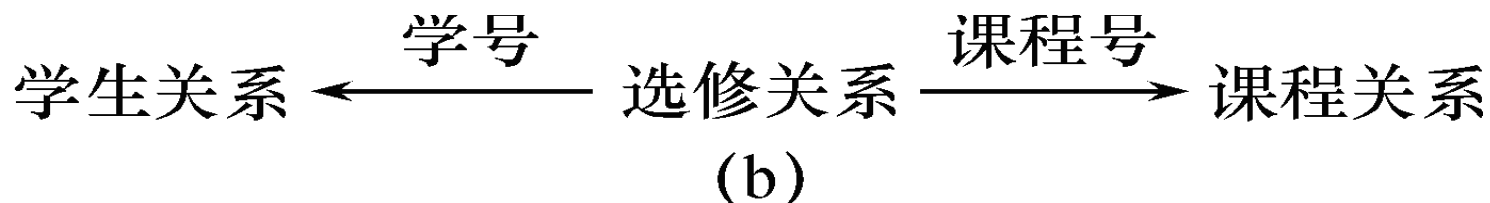
❖ [例2] :

学生 (学号、姓名、性别、专业号、年龄)

课程 (课程号、课程名、学分)

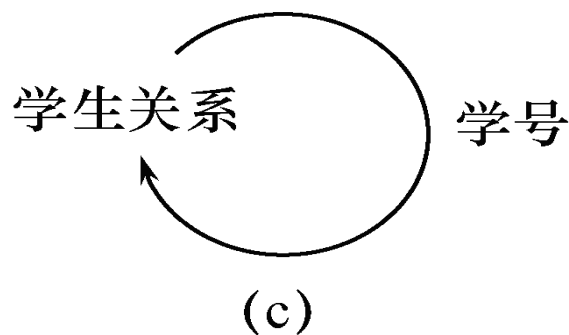
选修关系 (学号、课程号、成绩)

- “学号”和“课程号”是选修关系的外码
- 学生关系和课程关系均为被参照关系
- 选修关系为参照关系



❖ [例3] “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




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



[例4] 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

[例5] 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

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
- 用户定义的完整性

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，而不要由应用程序承担这一功能
- ❖ 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}

第2章 关系数据库

- ❖ 第一节 关系数据结构及形式化定义
- ❖ 第二节 关系操作
- ❖ 第三节 关系的完整性
- 第四节 关系代数



关系代数

❖ 传统集合运算

- 并(Union)
- 差(Except)
- 交(Intersection)
- 笛卡尔积(Cartesian Product)

❖ 专门的集合运算

概述

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

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			

传统的集合运算

❖ 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

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

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

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cup S$		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_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

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

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

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R-S$		
A	B	C
a_1	b_1	c_1

❖ 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

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

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

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

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cap S$		
A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

❖ 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
 - $R \times S = \{t_r t_s \mid t_r \in R \wedge t_s \in S\}$

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

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \times S$					
$R.A$	$R.B$	$R.C$	$S.A$	$S.B$	$S.C$
a_1	b_1	c_1	a_1	b_2	c_2
a_1	b_1	c_1	a_1	b_3	c_2
a_1	b_1	c_1	a_2	b_2	c_1
a_1	b_2	c_2	a_1	b_2	c_2
a_1	b_2	c_2	a_1	b_3	c_2
a_1	b_2	c_2	a_2	b_2	c_1
a_2	b_2	c_1	a_1	b_2	c_2
a_2	b_2	c_1	a_1	b_3	c_2
a_2	b_2	c_1	a_2	b_2	c_1

关系代数

❖ 传统集合运算

❖ 专门的集合运算

- 选择
- 投影
- 连接
- 除

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

(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。

(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) 给定一个关系R (X, Z) X和Z为属性组。

当 $t[X]=x$ 时, x 在R中的象集定义为

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

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。

学生-课程数据库

❖ 学生关系Student、课程关系Course和选修关系SC

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

Student

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

Course

❖ 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

选择 (Selection)

❖ 选择又称为限制 (Restriction)

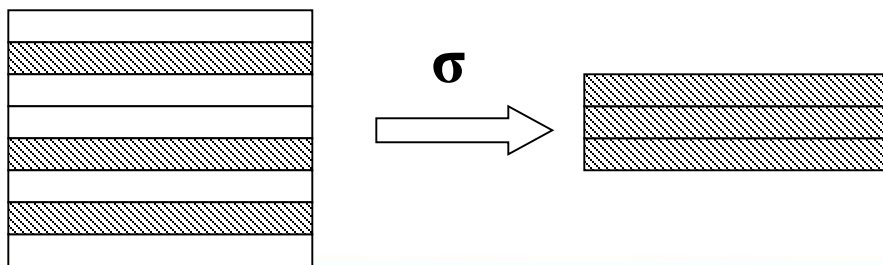
❖ 选择运算符的含义

- 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, 基本形式为: $X_1 \theta Y_1$

❖ 选择运算是从关系 R 中选取使逻辑表达式 F 为真的元组, 是从行的角度进行的运算



[例1] 查询信息系（IS系）全体学生

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

Sno	Sname	Ssex	Sage	Sdept
201215122	刘晨	女	19	IS
201215125	张立	男	19	IS

思考：查询年龄小于20岁且性别为男的学生

投影 (Projection)

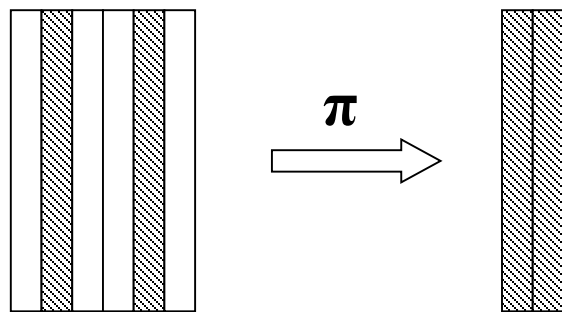
❖ 1) 投影运算符的含义

- 从 R 中选择出若干属性列组成新的关系

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

A : R 中的属性列

❖ 2) 投影操作主要是从列的角度进行运算



- 但投影之后不仅取消了原关系中的某些列，而且还可能取消某些元组（避免重复行）

❖ [例2] 查询学生的姓名和所在系

即求Student关系上学生姓名和所在系两个属性上的投影

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

Sname	Sdept
李勇	CS
刘晨	IS
王敏	MA
张立	IS

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

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

- 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 属性组上值满足比较关系 θ 的元组

❖ 两类常用连接运算

■ 等值连接 (equijoin)

➤ 什么是等值连接

θ 为 “=” 的连接运算称为等值连接

➤ 等值连接的含义

从关系 R 与 S 的广义笛卡尔积中选取 A 、 B 属性值相等的那些元组，即等值连接为：

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

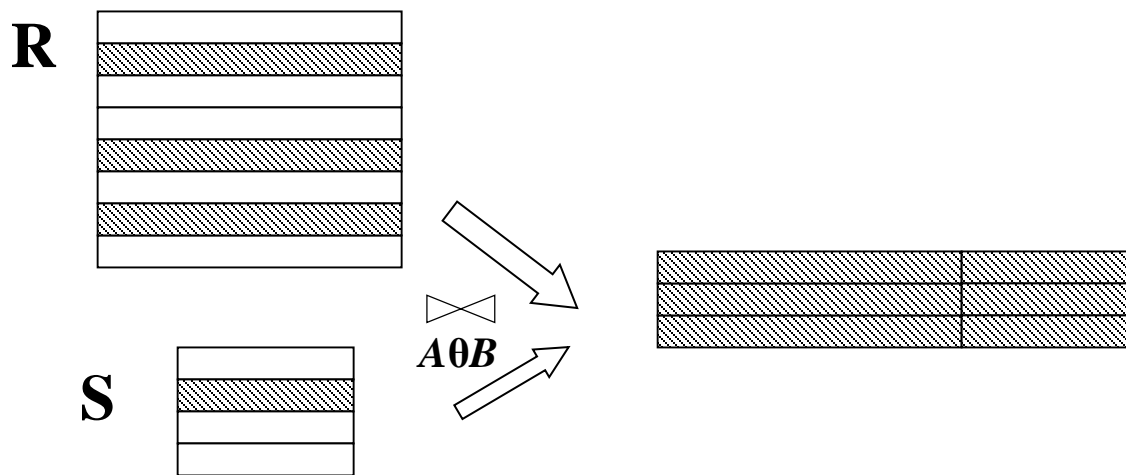
■ 自然连接 (Natural join)

- 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

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

❖ 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。

❖ [例3] 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

❖ 一般连接 $R \bowtie_{C < E} S$ 的结果如下

$R \bowtie_{C < E} S$				
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

❖ 等值连接 $R \bowtie_{R.B=S.B} S$ 的结果

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

❖ 自然连接 $R \bowtie S$ 的结果

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

❖ 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)

$$R$$

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

❖ 下图是例5中关系 R 和 S 的外连接、左连接、右连接

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
NULL	b_5	NULL	2

(a) 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

除 (Division)

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

R 中的 Y 与 S 中的 Y 可以有不同的属性名, 但必须出自相同的

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。

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

Y_x : x 在 R 中的象集, $x = t_r[X]$

除(续)

R

<i>A</i>	<i>B</i>	<i>C</i>
<i>a</i> ₁	<i>b</i> ₁	<i>c</i> ₂
<i>a</i> ₂	<i>b</i> ₃	<i>c</i> ₇
<i>a</i> ₃	<i>b</i> ₄	<i>c</i> ₆
<i>a</i> ₁	<i>b</i> ₂	<i>c</i> ₃
<i>a</i> ₄	<i>b</i> ₆	<i>c</i> ₆
<i>a</i> ₂	<i>b</i> ₂	<i>c</i> ₃
<i>a</i> ₁	<i>b</i> ₂	<i>c</i> ₁

S

<i>B</i>	<i>C</i>	<i>D</i>
<i>b</i> ₁	<i>c</i> ₂	<i>d</i> ₁
<i>b</i> ₂	<i>c</i> ₁	<i>d</i> ₁
<i>b</i> ₂	<i>c</i> ₃	<i>d</i> ₂

R* ÷ *S

<i>A</i>
<i>a</i> ₁

分析:

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

综合举例

以学生-课程数据库为例 (P.60)

[例4] 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: $\pi_{\text{Sno. Cno}}(\text{SC}) \div K$

综合举例(续)

❖ [例 4] 续 $\pi_{\text{Sno.Cno}}(\text{SC})$

201215121象集{1, 2, 3}

201215122象集{2, 3}

$$\pi_{\text{Cno}}(K) = \{1, 3\}$$

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

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



❖ 关系结构

❖ 关系操作

❖ 关系的完整性

❖ 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

休息…



子曰：
『君子不器。』