

**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**



## 2.2.1基本关系操作

- 常用的关系操作
  - 查询：选择、投影、连接、除、并、交、差
  - 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



**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**



## 2.3 关系的完整性

### 2.3.1 关系的三类完整性约束

### 2.3.2 实体完整性

### 2.3.3 参照完整性

### 2.3.4 用户定义的完整性



## 2.3.1 关系的三类完整性约束

- 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 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



# 实体完整性

规则1：若属性A是基本关系R的主属性，则属性A不能为空值。

- 基本关系的所有主属性都不能取空值。



The diagram shows a table with three columns: 学号 (Student ID), 课程号 (Course ID), and 成绩 (Score). The first row contains the values S001, C001, and 80. The second row contains S001, an empty cell, and 90. The third row contains an empty cell, an empty cell, and 80. To the left of the table, there are two large 'X' marks. Two horizontal ovals are drawn across the table, one enclosing the second row and the other enclosing the third row, indicating that these rows violate the entity integrity rule because the primary key attribute (学号) is missing.

<u>学号</u>	<u>课程号</u>	成绩
S001	C001	80
S001		90
		80

# 实体完整性(续)

## 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，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

1. 关系间的引用
2. 外码
3. 参照完整性规则



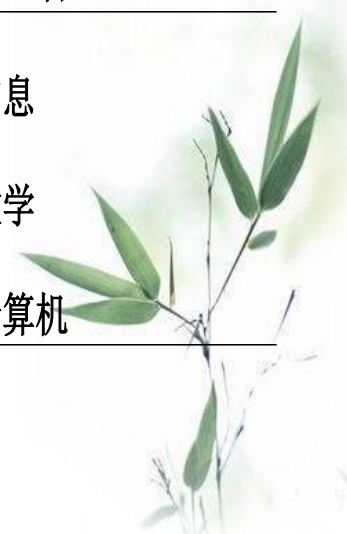
# 关系间的引用

**例1 学生实体、专业实体以及专业与学生间的一对多联系**  
**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。

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



# 关系间的引用

## 例2 学生、课程、学生与课程之间的多对多联系

学生 (学号, 姓名, 性别, 专业号, 年龄)

课程 (课程号, 课程名, 学分)

选修 (学号, 课程号, 成绩)

学号	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

学号	课程号	成绩
801	04	92
801	03	78
801	02	85
802	03	82
802	04	90
803	04	88

# 关系间的引用(续)

## 例3 学生实体及其内部的一对多联系

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



# 外码 (Foreign Key)

**Student**(sno,sname,sage,sdept)

**Course**(cno,cname)

- 外码

**Sc**(sno,cno,grade)

- 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Ks相对应，则称F是基本关系R的外码。

**F≠K<sub>r</sub>**

基本关系R(F , ...)

参照关系  
Referencing  
Relation

基本关系S (K<sub>s</sub>, ...)

被参照关系  
Referenced Relation  
或：目标关系  
Target Relation



# 外码说明

- 关系**R**和**S**不一定是不同的关系
- 目标关系**S**的主码**K<sub>s</sub>** 和参照关系的外码**F**必须定义在同一个（或一组）域上
- 关于取名
  - 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



候选码

候选码

外码

主码

学号	身份证号	姓名	性别	系别
0101	4201111985	张	男	CS
0102	36020211986	李	女	CS
0203	3671111985	赵	男	MA

系号	系名
CS	计算机
EN	英语
MA	数学系

主码

主码

学号	课号	成绩
0101	CS145	88
0101	CS148	90
0102	CS180	87
0203	CS145	78

课号	课名
CS145	数据库
CS148	操作系统
CS180	数据结构



# 参照完整性举例

例 1：学生实体与专业实体间的关系：

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

外码

参照关系

被参照关系

关系参照图

学生关系  $\xrightarrow{\text{专业号}}$  专业关系

例 2：学生，课程，学生与课程之间的多对多联系：

学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

被参照关系

被参照关系

参照关系

关系参照图

学生关系  $\xleftarrow{\text{学号}}$  选修关系  $\xrightarrow{\text{课程号}}$  课程关系

# 参照完整性举例

例 3：学生（学号，姓名，性别，专业号，年龄，班长）

参照关系

被参照关系

外码

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	



# 参照完整性规则

- 已知基本关系**R**的外码**F**（与基本关系**S**的主码**K<sub>s</sub>**相对应），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。

**$F \neq K_r$**

基本关系**R**(**F** , ...)

基本关系**S** (**K<sub>s</sub>**, ...)



# 参照完整性规则举例

## 学生（学号，姓名，性别，专业号，年龄）

- “专业号”属性只取下面两类值：
  - 空值，表示尚未给该学生分配专业
  - 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到一个不存在的专业中

## 选修（学号，课程号，成绩）

- “学号”和“课程号”是选修关系中的主属性，按照实体完整性和参照完整性规则，它们只能取相应被参照关系中已经存在的主码值



# 参照完整性规则举例

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：
  - 空值，表示该学生所在班级尚未选出班长，或该学生本人即是班长；
  - 非空值，这时该值必须是本关系中某个元组的学号值



# 关系的完整性(续)

## 2.3.1 关系的三类完整性约束

## 2.3.2 实体完整性

## 2.3.3 参照完整性

## 2.3.4 用户定义的完整性



## 2.3.4 用户定义的完整性

- 针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求
- 关系模型应提供定义和检验这类完整性的机制，以便使用统一的系统的方法处理它们，而不要由应用程序承担这一功能



# 用户定义的完整性(续)

例:

课程(课程号, 课程名, 学分)

- “课程号” 属性必须取唯一值
- 非主属性 “课程名” 也不能取空值
- “学分” 属性只能取值{1, 2, 3, 4}





供应商关系S（主码是“供应商号”）

供应商号	供应商名	所在城市
B01	红星	北京
S10	宇宙	上海
T20	黎明	天津
Z01	立新	重庆

今要向关系P中插入新行，新行的值分别列出如下。哪些行能够插入？

- A、('037', '绿', null)
- B、(null, '黄', 'T20')
- C、('201', '红', 'T20')
- D、('105', '蓝', 'B01')
- E、('101', '黄', 'T11')

零件关系P（主码是“零件号”，外码是“供应商号”）

零件号	颜色	供应商号
010	红	B01
312	白	S10
201	蓝	T20



**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**



## 2.4 关系代数

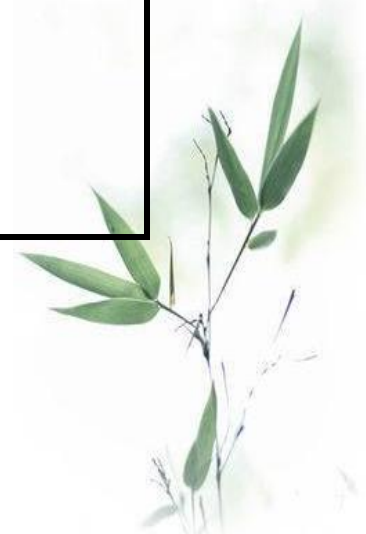
- 概述
- 传统的集合运算
- 专门的关系运算



# 概述

表2.4 关系代数运算符

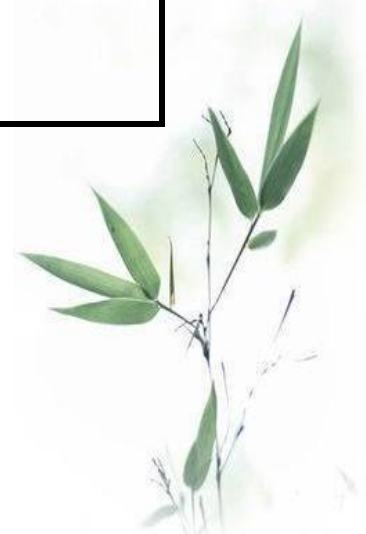
运算符		含义	运算符		含义
集合运算符	$\cup$	并	比较运算符	$>$	大于
	$-$	差		$\geq$	大于等于
	$\cap$	交		$<$	小于
	$\times$	笛卡尔积		$\leq$	小于等于
				$=$	等于
				$\neq$	不等于



## 概述(续)

表2.4 关系代数运算符（续）

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



## 2.4 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算



# 1. 并 (Union)

- $R$ 和 $S$

- 具有相同的目 $n$  (即两个关系都有 $n$ 个属性)
- 相应的属性取自同一个域

- $R \cup S$

- 仍为 $n$ 目关系, 由属于 $R$ 或属于 $S$ 的元组组成

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



# 并运算

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

S

A	B	C
3	4	5
7	2	3

RUS

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3
3	4	5

