



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](mailto:huangzhy@ustc.edu.cn)



# 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

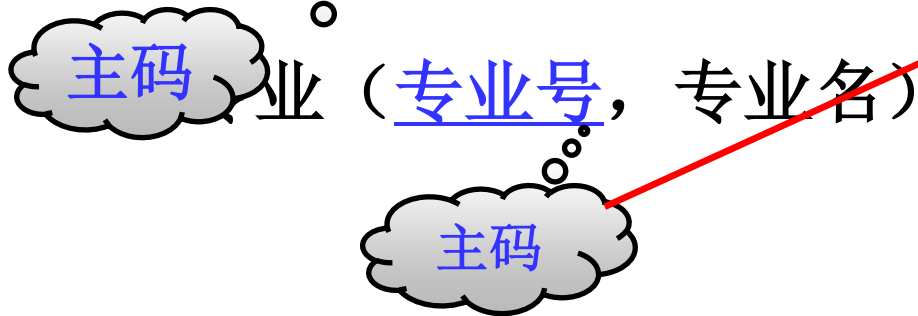


# 1. 关系间的引用

- 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 (学号, 姓名, 性别, **专业号**, 年龄)



- ✓ 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)

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)

## 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)

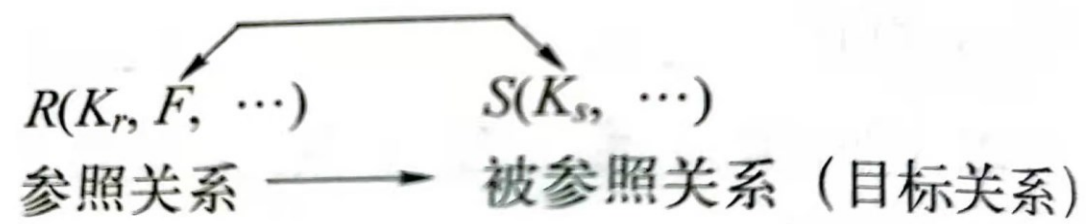
学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	

- ✓ “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)

- 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)

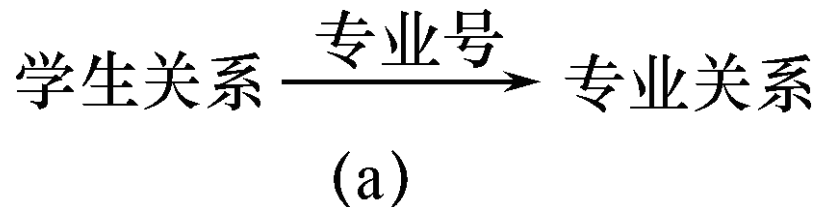




# 外码(续)

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



(b)





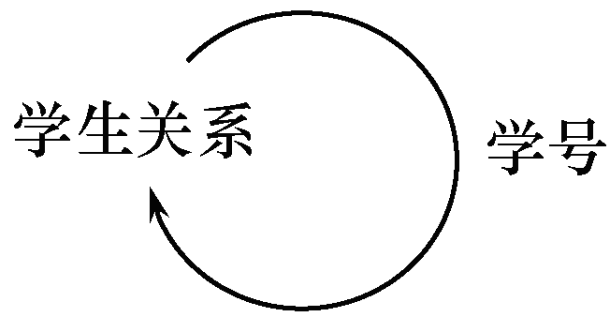
# 外码(续)

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



(c)



# 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

- 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

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	$\sigma$	选择
	$\pi$	投影
	$\bowtie$	连接
	$\div$	除





# 概述

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 与 或
	$\geq$	大于等于			
	$<$	小于			
	$\leq$	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**

<b>A</b>	<b>B</b>	<b>C</b>
a1	b1	c1
a1	b2	c2
a2	b2	c1

**S**

<b>A</b>	<b>B</b>	<b>C</b>
a1	b2	c2
a1	b3	c2
a2	b2	c1

**R-S**

<b>A</b>	<b>B</b>	<b>C</b>
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**

<b>A</b>	<b>B</b>	<b>C</b>
a1	b1	c1
a1	b2	c2
a2	b2	c1

**S**

<b>A</b>	<b>B</b>	<b>C</b>
a1	b2	c2
a1	b3	c2
a2	b2	c1

**$R \cap S$**

<b>A</b>	<b>B</b>	<b>C</b>
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 × 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

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

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



## 专门的关系运算(续)

89

### (2) $A$ , $t[A]$ , $\bar{A}$

若  $A = \{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$ , 其中  $A_{i_1}, A_{i_2}, \dots, A_{i_k}$  是  $A_1, A_2, \dots, A_n$  中的一部分, 则  $A$  称为属性列或属性组。

$t[A] = (t[A_{i_1}], t[A_{i_2}], \dots, t[A_{i_k}])$  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  $\{A_{i_1}, A_{i_2}, \dots, A_{i_k}\}$  后剩余的属性组。(R关系中)



# 专门的关系运算(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(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)

$R$

$x_1$	$Z_1$
$x_1$	$Z_2$
$x_1$	$Z_3$
$x_2$	$Z_2$
$x_2$	$Z_3$
$x_3$	$Z_1$
$x_3$	$Z_3$

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

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