

第2章 关系模型与关系运算



第2章 关系模型与关系运算

2.1 关系模型与关系运算简述

- 关系模型的提出与作用
- 关系模型与关系运算概览
- 本章的目标

2.2 关系与关系模型

2.3 关系代数运算

2.4 关系元组演算

2.5 关系域演算

2.6 小结

2.1 关系模型简述

—关系模型的提出与作用

- 关系模型是从表(Table)及表的处理方式抽象出来的，是在对传统表及其操作严格定义基础上，引入集合理论与逻辑学理论提出的
- 是数据库的三大经典数据模型之一，也是现在大多数商品化数据库系统所仍然使用的数据模型
- 标准的数据库语言(SQL语言)是建立在关系模型基础之上的，数据库领域的众多理论也都是建立在关系模型基础之上的

2.1 关系模型简述

—关系模型与关系运算概览

- 形象地说，一个关系(relation)就是一个Table
- 关系模型就是处理Table的，它由三个部分组成：
(详细内容在后面讲述)
 - ▣ 描述DB各种数据的基本结构形式(采用Table描述)
 - ▣ 描述Table与Table之间所可能发生的各种操作(采用关系运算等描述)
 - ▣ 描述这些操作所应遵循的约束条件(被称为完整性条件)

2.1 关系模型简述

—关系模型与关系运算概览(续)

➤ 将要学习: Table如何描述, 有哪些操作、
结果是什么、有哪些约束等?

学生登记表

学号	姓名	班级	出生年月	家庭住址
11101	李德	2	1980.5	山东
11102	范燕美	2	1980.8	哈尔滨
11103	张靖	2	1981.3	北京
11104	许聪	2	1980.7	云南
11105	黄佩婷	2	1979.12	浙江

操作

有哪些?

学生成绩单

班级	姓名	语文	数学	英语
2	李德	75	86	71
2	范燕美	76	78	68
2	张靖	81	77	80
2	许聪	82	82	79
2	黄佩婷	80	79	82

结果

是什么?

2.1 关系模型简述

—关系模型与关系运算概览(续)

- 关系运算有**关系代数**和**关系演算**两种形式，关系演算又进一步区分：元组演算和域演算
- 关系代数示例：**基于集合的运算**

$$\pi_{\text{姓名,课程名}} \left(\sigma_{\text{课程号}=c_2} (R \bowtie S) \right)$$

- 元组演算示例：**基于逻辑的运算**

$$\{t | (\exists u)(R(t) \wedge W(u) \wedge t[3] < u[1])\}$$

- 域演算示例：**基于示例的运算**

$$\{t_1 t_2 t_3 | S(t_1 t_2 t_3) \wedge R(t_1 t_2 t_3) \wedge t_1 < 20 \wedge t_2 > 30\}$$

2.1 关系模型简述

——本章目标

- 理解关系(relation)，理解如何用Relation对Table进行抽象或说严格定义
- 理解关系/表(relation/table)所具有的各种特性，理解关系模型
- 熟练掌握关系代数、元组演算和域演算，用这些关系运算来表达各种复杂的检索需求，以便于后续SQL语言的学习

第2章关系模型与关系运算

2.1 关系模型与关系运算简述

2.2 关系与关系模型

--关系

--关系模型

2.3 关系代数运算

2.4 关系元组演算

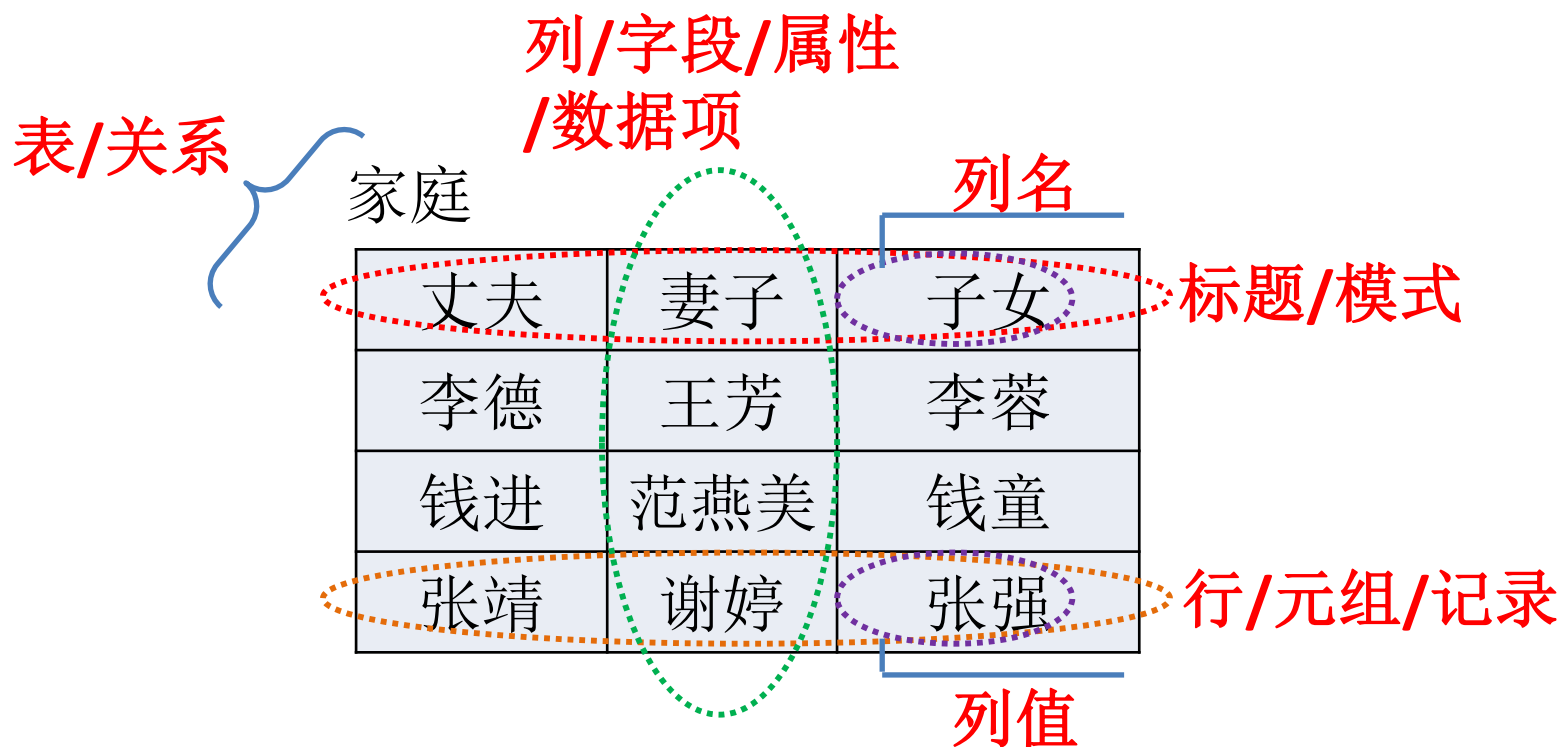
2.5 关系域演算

2.6 小结

2.2 关系与关系模型

—关系：Table的严格定义

➤ 如何严格地定义Table呢？



2.2 关系与关系模型

—关系：Table的严格定义(续)

➤ 首先定义“列”的取值范围“域(Domain)”

域(Domain)

- 一组值的集合，这组值具有相同的数据类型
 - ✓ 如整数的集合、字符串的集合、全体学生的集合
 - ✓ 再如，由8位数字组成的数字串的集合，由0到100组成的整数集合
- 集合中元素的个数称为域的基数(Cardinality)

2.2 关系与关系模型

--关系: Table的严格定义(续)

➤ 首先定义“列”的取值范围“域(Domain)”

域(Domain)

家庭

丈夫	妻子	子女
李德	王芳	李蓉
钱进	范燕美	钱童

D3=儿童集合(CHILD)=(李蓉, 钱童)
D2=女人集合(WOMAN)=(王芳, 范燕美)
D1=男人集合(MAN)=(李德, 钱进)

2.2 关系与关系模型

—关系：Table的严格定义(续)

➤ 再定义“元组”及所有可能组合成的元组：笛卡尔积
笛卡尔积(Cartesian Product)

□ 一组域 D_1, D_2, \dots, D_n 的笛卡尔积为：

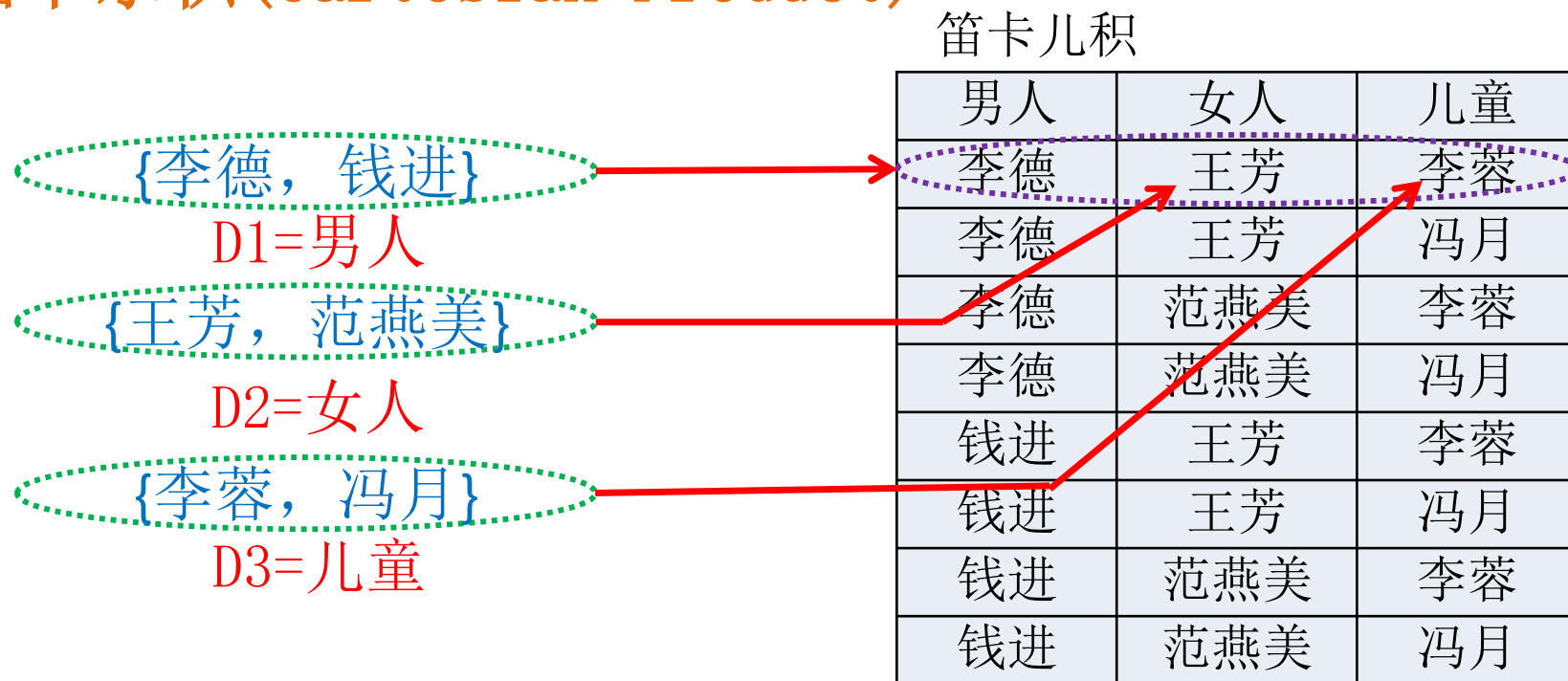
$$D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, \dots, n\}$$

□ 笛卡尔积的每个元素 (d_1, d_2, \dots, d_n) 称作一个
n元组 (n-tuple)

2.2 关系与关系模型

—关系：Table的严格定义(续)

➤ 再定义“元组”及所有可能组合成的元组：笛卡尔积
笛卡尔积(Cartesian Product)



2.2 关系与关系模型

—关系：Table的严格定义(续)

➤ 再定义“元组”及所有可能组合成的元组：笛卡尔积
笛卡尔积(Cartesian Product)

- 元组 (d_1, d_2, \dots, d_n) 的每一个值 d_i 叫做一个分量(component)
- 元组 (d_1, d_2, \dots, d_n) 是从每一个域任取一个值所形成的一种组合，笛卡尔积是所有这种可能组合的集合，即：笛卡尔积是由 n 个域形成的所有可能的 n 元组的集合
- 若 d_i 的基数为 m_i ，则笛卡尔积的基数，即元组个数为 $m_1 \times m_2 \times \dots \times m_n$

2.2 关系与关系模型

—关系：Table的严格定义(续)

➤ 再定义“元组”及所有可能组合成的元组：笛卡尔积
笛卡尔积(Cartesian Product)

笛卡儿积

男人	女人	儿童	→ 域名
李德	王芳	李蓉	
钱进	范燕美	钱童	
钱进	谢婷	张强	→ 域值
张靖	谢婷	冯月	

2.2 关系与关系模型

—关系：Table的严格定义(续)

➤ 由于笛卡尔积中的所有元组并不都是有意义的，因此…

关系(Relation)

- 一组域 D_1, D_2, \dots, D_n 的笛卡尔积的子集：
- 笛卡尔积中具有某一方面意义的那些元组被称作一个关系(Relation)
- 由于关系的不同列可能来自同一个域，为区分，需要为每一列起一个名字，该名字即为属性名。不同列名的列值可以来自相同域。

2.2 关系与关系模型

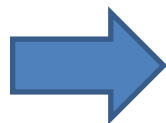
--关系: Table的严格定义(续)

- 由于笛卡尔积中的所有元组并不都是有意义的, 因此...

关系(Relation)

笛卡儿积

男人	女人	儿童
李德	王芳	李蓉
李德	王芳	冯月
李德	范燕美	李蓉
李德	范燕美	冯月
钱进	王芳	李蓉
钱进	王芳	冯月
钱进	范燕美	李蓉
钱进	范燕美	冯月



家庭

丈夫	妻子	子女
李德	王芳	李蓉
钱进	范燕美	钱童

列名(属性名)

列值: 来自域

2.2 关系与关系模型

--关系: Table的严格定义(续)

➤ 由于笛卡尔积中的所有元组并不都是有意义的, 因此...

关系(Relation)

- ❑ 关系可用 $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ 表示, 可简记为 $R(A_1, A_2, \dots, A_n)$, 这种描述又被称为关系模式(Schema)或表标题(head)
- ❑ R 是关系的名字, A_i 是属性, D_i 是属性所对应的域, n 是关系的度或目(degree), 关系中元组的数目称为关系的基数(Cardinality)
- ❑ 关系模式中 $A_i (i=1, \dots, n)$ 必须是不同的, 而 $D_i (i=1, \dots, n)$ 是可以相同的

2.2 关系与关系模型

--关系: Table的严格定义(续)

➤ 由于笛卡尔积中的所有元组并不都是有意义的，因此…

关系(Relation)

▣ 例如下图的关系为一3目关系，描述为
家庭(丈夫:男人, 妻子:女人, 子女:儿童)
或家庭(丈夫, 妻子, 子女)

家庭

丈夫	妻子	子女
李德	王芳	李蓉
钱进	范燕美	钱童

2.2 关系与关系模型

--关系: Table的严格定义(续)

- 由于笛卡尔积中的所有元组并不都是有意义的, 因此...

关系(Relation)

- 关系模式 $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ 中属性向域的映象在很多DBMS中一般直接说明为属性的类型、长度等

例如:

Student(S# char(8), Sname char(10), Ssex char(2), Sage integer,
D# char(2), Sclass char(6))

再如:

Course (C# char(3), Cname char(12), Chours integer,
Credit float(1), T# char(3))
SC(S# char(8), C# char(3), Grade float(1))

2.2 关系与关系模型

--关系: Table的严格定义(续)

➤ 关系模式与关系

- 同一关系模式下，可有很多的关系
- 关系模式是关系的结构，关系是关系模式在某一时刻的数据
- 关系模式是稳定的；而关系是某一时刻的值，是随时间可能变化的

Student(S# char(7), Sname char(10), Ssex char(2))

Student

S#	Sname	Ssex
1110101	张三	女
1110102	李四	男
1110201	王五	男

Student

S#	Sname	Ssex
1110101	张三	女
1110102	李四	男
1110201	王五	男
1110202	刘六	女

2.2 关系与关系模型

--关系(relation/table)的特性

- 列是同质：即每一列中的分量来自同一域，是同一类型的数据

Student

S#	Sname	Ssex	Sage	D#	Sclass
1110101	张三	女	21	03	11101
1110102	李四	男	20	03	11101
1110103	王五	男	19	软件	11101
1110201	刘六	大学	21	软件	11102
1110202	钱七	男	20	02	11102
1110203	赵八	高中	21	01	11102

2.2 关系与关系模型

—关系(relation/table)的特性

- 不同的列可来自同一个域，称其中的每一列为一个属性，不同的属性要给予不同的属性名。

例，假如：我们定义一个域为Person = 所有男人、女人和儿童的集合={李德，钱进，王芳，范燕美，李蓉，钱童，张峰}，则下述“家庭”关系的三个列将来自同一个域Person，因此需要不同的属性名“丈夫”“妻子”“子女”以示区分。

家庭

丈夫	妻子	子女
李德	王芳	李蓉
钱进	范燕美	钱童

2.2 关系与关系模型

--关系(relation/table)的特性(续)

- 列位置互换性：区分哪一列是靠列名
- 行位置互换性：区分哪一行是靠某一或某几列的值(关键字/键字/码字)
- 关系是以内容(名字或值)来区分的，而不是属性在关系的位置来区分
- 如下面两个关系是完全相同的关系

家庭

丈夫	妻子	子女
李德	王芳	李蓉
钱进	范燕美	钱童

家庭

妻子	子女	丈夫
王芳	李蓉	李德
范燕美	钱童	钱进

2.2 关系与关系模型

--关系(relation/table)的特性(续)

- 任意两个元组不能完全相同。(集合的要求：集合内不能有相同的两个元素)
- 元组相同是指两个元组的每个分量都相同

Student

S#	Sname	Ssex	Sage	D#	Sclass
1110101	张三	女	21	03	11101
1110102	李四	男	20	03	11101
1110103	王五	男	19	02	11101
1110201	刘六	男	21	02	11102
1110102	李四	男	20	03	11101
1110203	赵八	女	21	01	11102

在同一个关系(表)中存在相同的元组,去掉其中的重复元组

2.2 关系与关系模型

--关系(relation/table)的特性(续)

➤ 属性不可再分特性: 又被称为关系第一范式

Student

S#	Sname	Ssex
1110101	张三	女
1110102	李四	男

符合第一范式(Table)

Student

S#	姓名		Ssex
	姓	名字	
1110101	李	四	男
1110102	王	五	男
1110103	刘	六	男

不符合第一范式
(Not Table)

2.2 关系与关系模型

--关系(relation/table)的特性(续)

➤ 属性不可再分特性: 又被称为关系第一范式(续)

家庭

丈夫	妻子	子女
李德	王芳	李蓉
钱进	范燕美	钱童

符合第一范式(Table)

家庭

丈夫	妻子	子女	
		第一	第二
李德	王芳	李刚	李蓉
钱进	范燕美	钱瑞	钱童

不符合第一范式
(Not Table)

2.2 关系与关系模型

--关系(relation/table)的特性(续)

➤ 属性不可再分特性: 又被称为关系第一范式(续)

下面的关系符不符合第一范式(Table)?

家庭

丈夫	妻子	子女	家庭住址
李德	王芳	李蓉	中国山东省威海市文化西路二号
钱进	范燕美	钱童	中国北京市王府井大街十号云丽弄三门

2.2 关系与关系模型

--关系(relation/table)上的一些重要概念

➤ 候选码(Candidate Key)/候选键

- 关系中的一个属性组，其值能唯一标识一个元组，若从该属性组中去掉任何一个属性，它就不具有这一性质了，这样的属性组称作候选码。
- 例如关系“学生(S#, Sname, Sage, Sclass)”中**S#**就是一个候选码，在学生关系中，任何两个元组的S#是一定不同的，而这两个元组的Sname, Sage, Sclass都可能相同(同名、同龄、同班)，所以S#是候选码。
- 再如关系“选课(S#, C#, Sname, Cname, Grade)”中**(S#, C#)**联合起来是一个候选码

2.2 关系与关系模型

--关系(relation/table)上的一些重要概念

➤ 候选码(Candidate Key)/候选键

□ 有时，关系中有很多组候选码，

•例如：

学生(S#, Sname, Sage, Sclass, Saddress)

其中属性S#是候选码，属性组(Sname, Saddress)也是候选码(同名同地址的两个同学是不存在的)

•再如

Employee(EmpID, EmpName, Mobile)

每一雇员有唯一的EmpID，没有两个雇员有相同的手机号Mobile，则EmpID是候选码，Mobile也是候选码

2.2 关系与关系模型

--关系(relation/table)上的一些重要概念

➤ 主码(Primary Key)/主键

- ▣ 当有多个候选码时，可以选定一个作为主码。
- ▣ DBMS以主码为主要线索管理关系中的各个元组。
 - 例如可选定属性S#作为“学生”表的主码，也可以选定属性组(Sname, Saddress)作为“学生”表的主码。
 - 选定EmpID为Employee的主码。

2.2 关系与关系模型

--关系(relation/table)上的一些重要概念

➤ 主属性与非主属性

- ▣ 包含在任何一个候选码中的属性被称作主属性，如“选课”中的S#，D#。
- ▣ 而其他属性被称作非主属性，如“选课”中的Sname, Cname, Grade。

➤ 在最简单的情况下，候选码只包含一个属性

2.2 关系与关系模型

--关系(relation/table)上的一些重要概念

- 主属性与非主属性(续)
- 在最简单的情况下，候选码只包含一个属性
- 在最极端的情况下，关系的所有属性组是这个关系的候选码，称为全码(All-Key)。比如关系“教师授课”(T#, C#)中的候选码(T#, C#)就是全码。

教师授课表

T#	C#
几何	张三
物理	李四
代数	王五



2.2 关系与关系模型

—关系(relation/table)上的一些重要概念

➤ 外码(Foreign Key)/外键

- 关系R中的一个属性组，它不是R的候选码，但它与另一个关系S的候选码相对应，则称这个属性组为R的外码或外键。

• 例如“合同”关系中的**客户号**不是候选码，但却是外码。因它与“客户”关系中的候选码“**客户号**”相对应。

合同			外码	客户		
主码	合同号	合同名称	客户号	主码	客户号	客户名称
	HIT001	电脑采购	ZQ		ZQ	志强电脑
	HIT002	外墙修补	JA		JA	吉安涂装
	HIT003	文件打印	ZQ		YY	用友印刷
						联系人
						张三
						李四
						王五

2.2 关系与关系模型

--关系(relation/table) 的概念小结

- 域(Domain)
- 元组(Tuple)
- 笛卡尔积
- 关系(Relation)
- 关系模式(Relational Schema)
- 属性(Attribute)
- 关系的度(Degree)与关系的基数(Cardinality)
- 候选码、主码、主属性与非主属性、外码

2.2 关系与关系模型

——关系模型 注意区别：关系模式

➤ 关系模型 同一关系模式下，可有很多的关系

▣ DB数据的基本结构：Relation/Table

▣ DB数据的基本操作有：

\cup (并, UNION)、 \cap (交, INTERSECTION)、 \bowtie (连接, JOIN)、

$-$ (差, DIFFERENCE)、 \times (广义积, PRODUCT)、

σ (选择, SELECTION)、 π (投影, PROJECTION)、

\div (除, DIVISION) 等运算

▣ DB数据的结构与操作受三个完整性的约束：

实体完整性、参照完整性和用户自定义完整性

2.2 关系与关系模型

—关系模型中的完整性

- Relation/Table已在前面介绍过
- DB数据的基本操作对象是Relation/Table，操作的结果仍然是Relation/Table；
- 关系操作是集合操作，操作的对象及结果都是集合，是一次一集合的方式。而非关系型的数据操作方式是一次一记录，具体操作将在关系代数中介绍

下面将介绍：关系的完整性

2.2 关系与关系模型

—关系的实体完整性

➤ 实体完整性

- ❑ 关系的主码中的属性值不能为空值；
- ❑ 空值：不知道或无意义的值；
- ❑ 意义：关系中的元组对应到现实世界相互之间可区分的一个个个体，这些个体是通过主码来唯一标识的；若主码为空，则出现不可标识的个体，这是不容许的。

Student

主 码	S#	Sname	Ssex
	1110101	张三	女
		李四	男
	1110102	王五	男

2.2 关系与关系模型

—关系的实体完整性(续)

➤ 空值的含义

- 空值：不知道、不存在或无意义的值；
- 在进行关系操作时，有时关系中的某属性值在当前是填不上的，
 - 比如：档案中有“生日不详”、“下落不明”、“日程尚待公布”等，这时就需要空值来代表这种情况。关系模型中用‘?’表征



2.2 关系与关系模型

—关系的实体完整性(续)

➤ 空值的含义(续)

- ❑ 数据库中有了空值，会影响许多方面，如影响聚集函数运算的正确性，不能参与算术、比较或逻辑运算等
- 例如：“ $3 + ?$ ”结果是多少呢？ “ $3 * ?$ ”结果是多少呢？ “ $? \text{ and } (A=A)$ ”结果又是多少呢？
- 再例如，一个班有30名同学，如所有同学都有成绩，则可求出平均成绩；如果有一个同学没有成绩，怎样参与平均成绩的计算呢，是当作0，还是当作100呢？还是不考虑他呢？
- 因此有空值的时候是需要特殊处理的，要特别注意。

2.2 关系与关系模型

—关系的参照完整性

➤ 参照完整性

- 如果关系R1的外码Fk与关系R2的主码Pk相对应，则R1中的每一个元组的Fk值或者等于R2 中某个元组的Pk值，或者为空值
- 意义：如果关系R1的某个元组t1参照了关系R2的某个元组t2，则t2必须存在
- 例如关系Student在“系别”上的取值有两种可能：
 - ✓ 空值，表示该学生尚未分到任何系中
 - ✓ 若非空值，则必须是Dept关系中某个元组的“系别”值，表示该学生不可能分到一个不存在的系中

R1--Student
外码Fk—系别

Student

学号	姓名	系别
101	张三	01
102	李四	06
103	王五	✓

R2--Dept
主码Pk—系别

Dept

系别	系名	人数
01	机械	900
02	船舶	850
03	软件	870

2.2 关系与关系模型

—关系的用户自定义完整性

➤ 用户自定义完整性

- ▣ 用户针对具体的应用环境定义的完整性约束条件

要求名字在5个汉字字符之内

Student

S#	Sname	Ssex	Sage
1110101	张三	女	21
1110102	李四	男	20
1110103	理查德·克莱德曼	男	19
1110201	刘六	0	21
1110202	李四	男	200

性别只能是“男”或“女”

年龄在[12, 35]之间

颠覆完整性的示例



2.2 关系与关系模型

--DBMS对关系完整性的支持

- 实体完整性和参照完整性由DBMS系统自动支持
- DBMS系统通常提供了如下机制：它使用户可以自行定义有关的完整性约束条件，当有更新操作时，自动检验更新操作的正确性，即是否符合用户自定义的完整性