数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

第二章 关系数据库

Principles of Database Systems

关系数据库简介

- 提出关系模型的是美国IBM公司的E.F.Codd
  - 1970年首次提出关系数据模型
    - E.F.Codd, “A Relational Model of Data for Large Shared Data Banks”, Communication of the ACM, 1970
  - 之后，提出了关系代数和关系演算的概念
  - 1972年提出了关系的第一、第二、第三范式
  - 1974年提出了关系的BC范式

关系模型的特征：

数据结构简单

表达能力强大

数据独立性好

关系模型的组成

- 数据结构：
  - 二维表（关系），数据库中全部数据及数据间联系都以关系来表示。
- 数据操作：

  读：查询(query)；写：增加(insert)、删除(delete)、修改(update)
  - 例如：选择(select)、投影(project)、连接(join)、除(divide)、并(union)、交(intersection)、差(difference)……
  - 理论基础：关系代数、元组关系演算、域关系演算
  - 具体实现：关系数据库语言——ALPHA、QBE、SQL……
- 数据的约束条件:

  三类完整性约束：实体完整性、参照完整性、用户自定义完整性

第二章 关系数据库

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

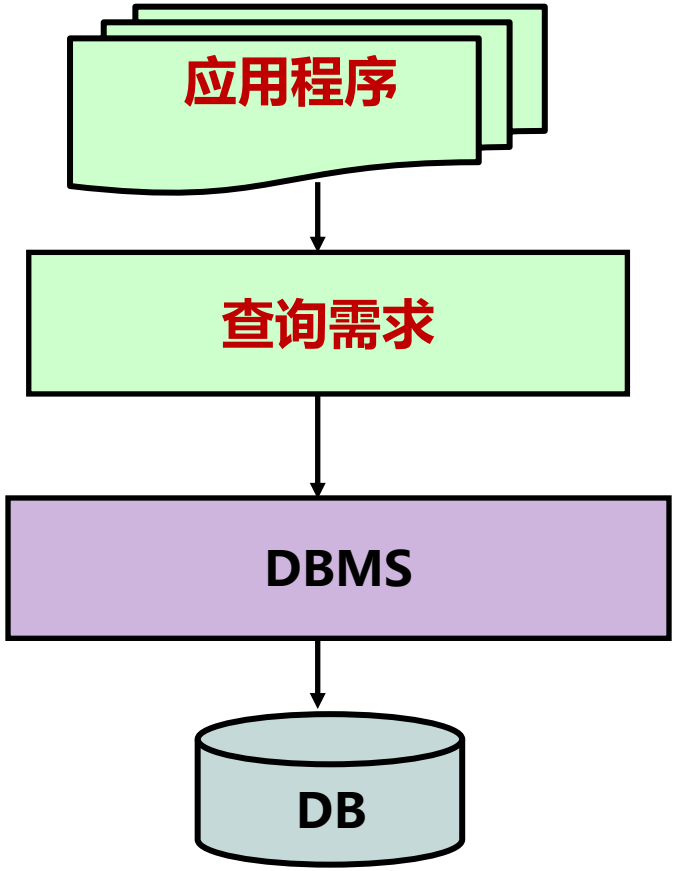
2.1.1 关系
2.1.2 关系模式
2.1.3 关系数据库

如何用离散数学中的代数系统来定义关系数据库？

如何定义代数运算，用数学语言描述查询需求？

关系代数系统的引入动机

关系代数产生动机



问题：为什么需要代数语言?

如何描述查询需求?

查询需求描述自动化
查询需求描述自然化

- 关系数据库是表集
- 定义完备的表操作
- 用表的运算表达式描述查询需求
- 系统解决表达式与查询语言的转化、执行

2.1.1 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。
- 关系模型建立在集合代数的基础上，关系数据结构的形式化定义：
  - 域（Domain）
  - 笛卡儿积（Cartesian Product）
  - 关系（Relation）

1. 域（Domain）

- 域是一组具有相同数据类型的值的集合。又称为值域。（一般用D表示）

  如：整数的集合、字符串的集合、{0,1}、……

- 域中所包含的值的个数称为域的基数（用m表示）。
- 关系中用域表示属性的取值范围。
- 例如：
  - D1 = { 李力，王平，刘伟 }　　m1=3
  - D2 = { 男，女 }　　m2=2
  - D3 = { 47，28，30 }　　m3=3

其中，D1，D2，D3为域名，分别表示教师关系中姓名、性别、年龄的取值集合。

2. 笛卡儿积（Cartesian Product）

- 给定一组域D1，D2，…，Dn，这些域中可以有相同的，则D1，D2，…，Dn的笛卡儿积为：

  $D1 \times D2 \times \ldots \times Dn = \{(d1,d2,\ldots, dn) \mid di \in Di，i = 1,2,\ldots,n\}$

- 笛卡儿积中每一个元素(d1,d2,…, dn)叫作一个n元组（n-tuple）或简称元组(Tuple)。
- 笛卡儿积元素(d1,d2,…, dn)中的每一个值di叫作一个分量(Component)。
- 若Di（i = 1,2,…,n）为有限集，其基数为mi（i = 1,2,…,n），则D1×D2×…×Dn的基数M 为：

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

2. 笛卡儿积

- 笛卡儿积是一个集合；是所有域的所有取值的一个组合。
- 不能重复。
- 例：设D1为学生集合（T）={张群，徐晶，王刚}

  D2为性别集合（S）={男，女}

  则D1×D2是个二元组集合，元组个数为3×2，是所有可能的（学生，性别）元组集合。

$D_1 \times D_2 =$

T	S
张群	男
张群	女
徐晶	男
徐晶	女
王刚	男
王刚	女

2. 笛卡儿积

- 如果 D1为教师集合 ={张清玫，刘逸}, D2为专业集合= {计算机专业，信息专业}, D3为研究生集合= {李勇，刘晨，王敏}

  则D1×D2×D3 =?

笛卡儿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。

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

3. 关系（relation）

- 关系（relation）——笛卡儿积

  D1×D2×…×Dn的子集叫做在域D1，D2，…，Dn上的关系，用R (D1，D2，…，Dn)表示。

  - R是关系的名字，n是关系的度或目（Degree）；
  - n=1时，单元关系（Unary relation）/一元关系
  - n=2时，二元关系（Binary relation）
  - 关系中的每个元素称为元组，通常用t表示；
  - 关系也可以表示为二维表的形式，如：学生（姓名，性别）

一般来说，

取笛卡儿积上有意义的子集作一个关系。

姓名	性别
张群	男
徐晶	女
王刚	男

属性

元组

- **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。**

3. 关系

候选码（Candidate key）

若关系中的某一属性组的值能唯一地标识一个元组，若从属性组中去掉任一属性，它就不再具有这一性质，则称该属性组为候选码。

- 简单情况：候选码只包含一个属性。
- 最极端情况：关系模式的所有属性组成它的候选码，称为全码（All-key）。
- 主码（Primary key）：若一个关系有多个候选码，则选定其中一个为主码。

  每个关系必定有且仅有一个主码，通常用较小的属性组合作为主码。选定以后，不能随意改变。
- 主属性（Prime attribute）：候选码的诸属性。
- 非主属性（Non-Prime attribute）或 非码属性（Non-key attribute）：不包含在任何候选码中的属性。

3. 关系

- 例：选课关系（学号，课程号，课程名，成绩），假设课程名不能重复，则（学号，课程号）和（学号，课程名）都可作为候选码。

提问1：（学号，课程号，课程名）是选课关系的候选码吗？ **No**

提问2：若关系中所有属性均为主属性，则其候选码必为全码吗？ **No**

3. 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。

3. 关系

- 关系与二维表的区别：严格地说，关系是规范化了的二维表中行的集合，为了使相应的数据操作简化，在关系模型中对关系作了种种限制。
- 基本关系具有6大特性：
  - 列是同质的（Homogeneous）；
  - 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；
  - 列的顺序无所谓，列的次序可以任意交换；
  - 任意两个元组的候选码不能相同；
  - 行的顺序无所谓，行的次序可以任意交换；
  - 分量必须取原子值，即每个分量是都不可分的数据项。（规范化）

关系是有限集合

2.1.2 关系模式

- 关系模式（Relation Schema）是型，相对静态、稳定
- 关系是值，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
  - 元组语义以及完整性约束条件
  - 属性间的数据依赖关系集合

- 关系模式和关系往往统称为关系；
- 通过上下文加以区别

2.1.2 关系模式

- 关系模式——关系的描述称作关系模式，包括关系名、关系中的属性名、属性向域的映象、属性间的数据依赖关系等，记作 R (U，D，DOM，F)。
  其中：
  - R：关系名
  - U：R中的属性名集合
  - D：属性组U中属性所来自的域（取值范围）
  - DOM：属性到域的映象集（属性类型、长度）
  - F：属性间数据的依赖关系集合。
  - 也可简记为 R(U) 或 R(A1, A2, …, An)。
- 关系——某一时刻对应某个关系模式的内容。

例：
DOM (SUPERVISOR-PERSON)
= DOM (POSTGRADUATE-PERSON)
= PERSON

//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

2.1.3 关系数据库

- 关系数据库模式——关系数据库的型
  - 基于某一应用领域所定义的所有关系模式的集合。
  - 包括若干域的定义，及在这些域上定义的若干关系模式。
- 关系数据库——关系数据库的值
  - 关系数据库模式在某一时刻所对应的关系的集合。
- 从系统思维理解：

  数据字典中存放数据库模式，数据文件中存放数据库值。

2.2 关系操作

2.2.1 基本关系操作

关系操作：对关系表进行运算的运算符，这些运算符描述了应用对数据库的数据处理需求。

- 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

2.2.2 关系数据库语言的分类

- 关系代数语言：用对关系的运算来表达查询要求
  - 代表：ISBL
- 关系演算语言：用谓词来表达查询要求
  - 元组关系演算语言
    - 谓词变元的基本对象是元组变量
    - 代表：ALPHA, QUEL
  - 域关系演算语言
    - 谓词变元的基本对象是域变量
    - 代表：QBE
- 具有关系代数和关系演算双重特点的语言
  - 代表：SQL（Structured Query Language）：高度非过程化语言

2.3 关系的完整性

- 实体完整性机制——这条机制要求关系中元组在组成主码的属性上不能有空值和重复值。
- 参照完整性机制——这条规则要求“不引用不存在的实体”。

前2者称为关系的两个不变性，由关系系统自动支持。

- 用户定义的完整性机制——这是针对某一具体数据的约束条件，由应用环境决定。体现了具体领域中的语义约束。

2.3.1 实体完整性（Entity Integrity）

- 在关系中的所有元组在码上的取值满足以下条件，则说该关系具有实体完整性：
  - 主属性非空——若属性A是基本关系R的主属性，则属性A不能取空值。
  - 主码各不相同——不会出现主码相同的两个记录。
- 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) 主码中的属性不能取空值。

2.3.1 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
- 意义：关系对应到现实世界中的实体集，元组对应到实体，实体是相互可区分的，通过主码来唯一标识，若主码为空，则出现不可标识的实体，这是不容许的

学号	姓名	性别	系名
0101	张	男	CS
0102	李	女	CS
0203	赵	男	MA

向关系中插入新行，下列哪些行能够插入？

A. （~~'0203'~~，'张'，男, null)

B. (~~null,~~ '吴'，女，'IS'）

C. （'0301'，null, null, null)

D. （'0105'，'张'，男，'MA'）

2.3.2 参照完整性（Reference Integrity）

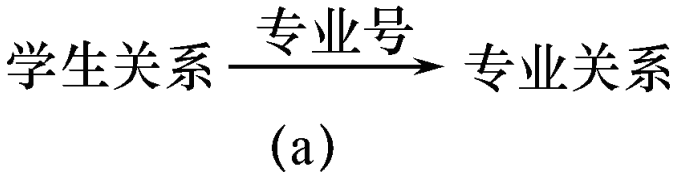
- 关系间的引用

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1：

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

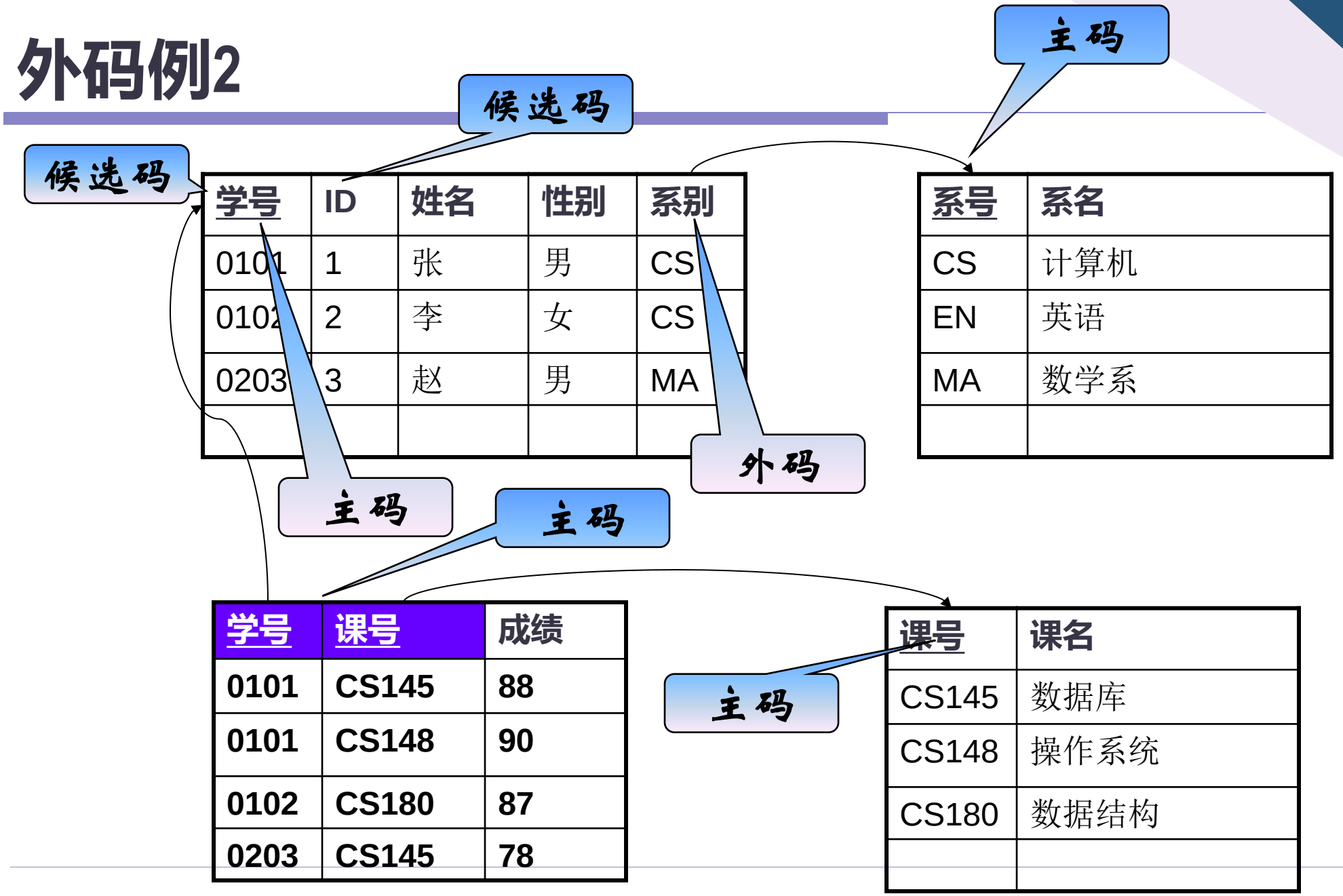


- 设F 是基本关系R 的一个或一组属性，但不是关系 R 的码。如果 F 与基本关系S 的主码 Ks 相对应，则称 F 是基本关系R 的外码。

R 称为参照关系（Referencing Relation）

S 称为被参照关系（Referenced Relation）或 目标关系（Target Relation）

外码例2


候选码
候选码
主码
主码
外码
主码
主码
学号
ID
姓名
性别
系别
0101
1
张
男
CS
0102
2
李
女
CS
0203
3
赵
男
MA
系号
系名
CS
计算机
EN
英语
MA
数学系
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

2. 外码

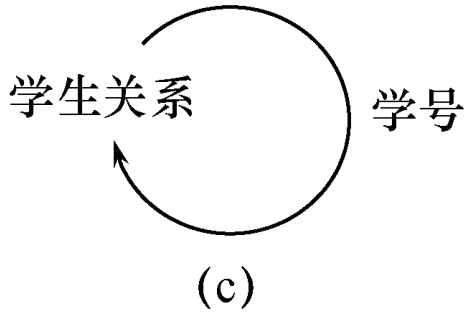
- 关系R 和S 不一定是不同的关系；
- 目标关系S 的主码Ks 和参照关系的外码F 必须定义在同一个（或一组）域上；
- 外码并不一定要与相应的主码同名：

  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3：学生(学号，姓名，…，班长)

其中：学号是“主码”，“班长”是外码

学生关系既是参照关系也是被参照关系



(c)

3. 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 Ks 相对应（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

外码例6

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

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

现要向关系P中插入新的元组，下列元组中哪些能够成功的插入？

A．(037，‘绿’，null)

B．~~(null，~~‘黄’，‘T20’)

C．~~(201，~~‘红’，‘T20’)

D．(105，‘蓝’，‘B01’)

E．(101 ‘黄’，~~‘T11’~~)

2.3.3 用户定义的完整性

- 用户针对具体的应用环境定义的完整性约束条件。
- 意义：反映某一具体应用所涉及的数据必须满足的语义要求，以便用统一的、系统的方法处理它们，而不需要由应用程序来承担这一功能。
- 例如：
  - “成绩”不能为负数，
  - “学号”要求是8位整数，
  - “性别”要求取值为“男”或“女”，
  - “工龄”应该小于“年龄”等。

2.3 关系的完整性

- 问题：以上三类完整性是如何实现的?
- DBMS 提供接口，用户负责定义完整性约束；

  如：实体完整性约束通过说明关系的主码来定义，参照完整性通过说明关系的外码来定义
- DBMS 提供完整性约束的自动检查。

  如：向关系中插入新元组时自动检查主码属性值是否唯一和非空，否则拒绝插入

小结

- 关系数据结构
  - 关系
    - 域
    - 笛卡儿积
    - 关系
      - 关系，属性，元组
      - 候选码，主码，主属性
      - 基本关系的性质
  - 关系模式
  - 关系数据库

- 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

- 关系的完整性约束
  - 实体完整性
  - 参照完整性
    - 外码
  - 用户定义的完整性