设计篇
第五章 逻辑数据库设计

主讲：程思瑶
海量数据计算研究中心

逻辑数据库设计

- 逻辑数据库设计的任务
  - 把概念数据库设计阶段产生的概念数据库模式变换为逻辑数据库模式　　ER图->关系表
- 逻辑数据库设计的目标
  - 满足用户的完整性和安全性要求；
  - 动态关系至少具有第三规范形式，静态关系至少具有第一规范形式；
  - 能够在逻辑级上高效率地支持各种数据库事务的运行；
  - 存储空间利用率高

逻辑数据库设计

- 逻辑数据库设计的步骤
  - 形成初始关系数据库模式
  - 关系模式规范化
  - 关系模式优化
  - 定义关系上的完整性和安全性约束
  - 子模式定义
  - 性能估计

逻辑数据库设计

- 逻辑数据库设计的步骤
  - 形成初始关系数据库模式
  - 关系模式规范化
  - 关系模式优化
  - 定义关系上的完整性和安全性约束
  - 子模式定义
  - 性能估计

5.1形成初始关系数据库模式

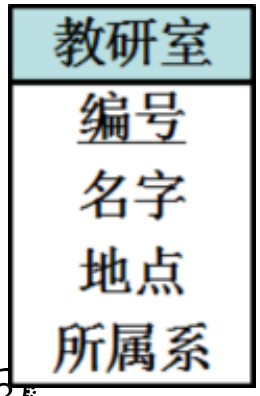
- 初始关系数据库模式是指直接由概念数据库模式生成的关系数据库模式。
- 初始关系数据库模式生成的目的是把概念数据库模式的实体、实体间联系等模型结构变换为关系模式。

5.1形成初始关系数据库模式

- 由概念数据库模式生成初始关系数据库模式的方法：
  - 普通实体集的变换
  - 弱实体的变换
  - 多值属性的变换
  - 实体间联系的变换
  - 确定函数依赖集

5.1形成初始关系数据库模式

- 普通实体集的变换
  - 为概念数据库模式中的每个普通实体集E建立一个关系S。
  - S包含E的所有简单属性和E的复合属性的简单子属性。
  - E的码是S的主码。



教研室关系G(编号,名字, 地点,所属系)

5.1形成初始关系数据库模式

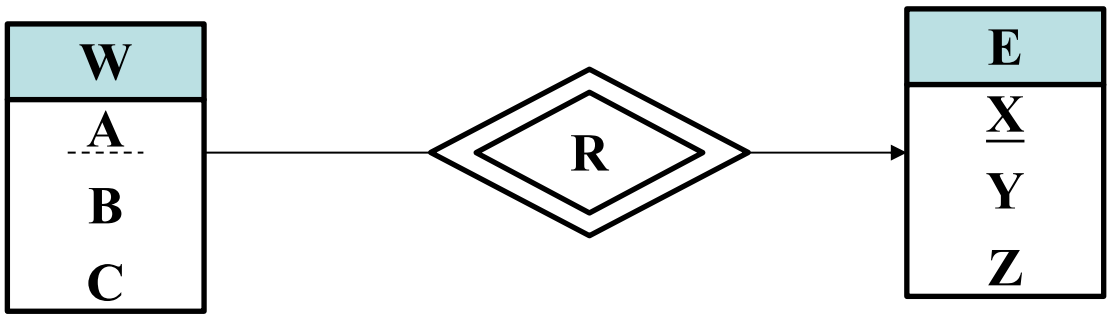
- 普通实体集的变换
  - 例

教师关系T(身份证号,名字, 工资,生日,职称,性别,邮编,市,区,学校,信箱)

教师
身份证号
名字
工资
生日
职称
性别
地址
邮编
市
区
单位
学校
信箱

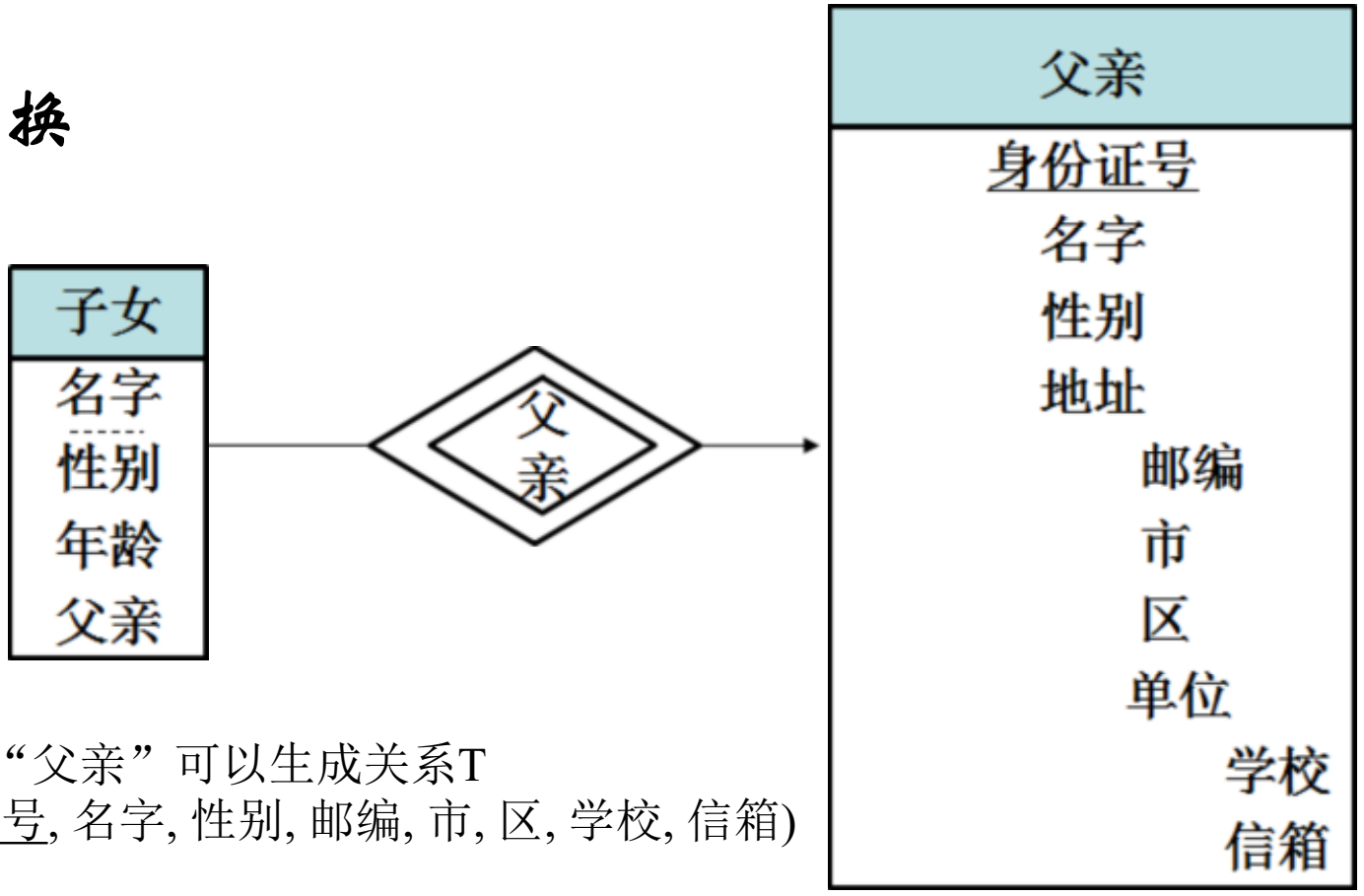
5.1形成初始关系数据库模式

- 弱实体的变换
  - 设W是概念数据库模式中以实体集E为识别实体集的弱实体。
    - 建立一个与W对应的关系R；
    - W的所有简单属性和复合属性的简单子属性映射为R的属性；
    - E的码属性也是R的属性；
    - R的主码由E的码和W的部分码组合而成；
    - E对应的关系的码是R的外码。



5.1形成初始关系数据库模式

- 弱实体变换
  - 例



识别实体“父亲”可以生成关系T
(身份证号, 名字, 性别, 邮编, 市, 区, 学校, 信箱)

由弱实体“子女”和识别实体“双亲”可以生成关系R
(身份证号(父亲)，名字，性别，年龄)

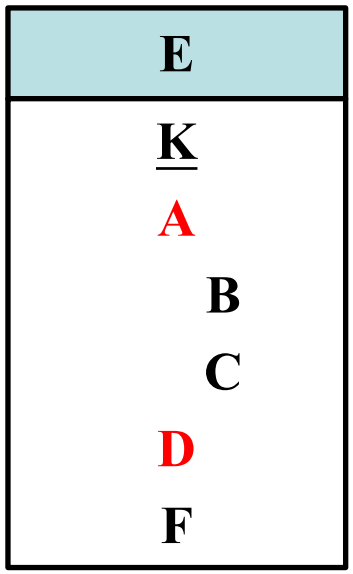
5.1形成初始关系数据库模式

- 多值属性的变换
  - 设实体集E具有多值属性，S是E对应的关系
    - 为E的每个多值属性A建立一个关系T，用T表示A。
    - 如果A是简单属性，T的属性为A与S的主码K。A和K形成T的主码。
    - 如果A是复合属性，T包含A的简单子属性和S的码K。A的简单子属性和K形成T的码。
    - S关系中忽略属性A。
    - 对联系R的多值属性 类似处理

E
K
A
C
D

5.1形成初始关系数据库模式

- 多值属性的变换
  - 例



S(K,F)
T_1(K,B,C)
T2(K,D)

5.1形成初始关系数据库模式

- 实体间联系的变换
  - 1:1联系的变换
    - 设R是实体集E_1和E_2之间的1:1联系，S和T是E_1和E_2对应的关系
    - 方法1：通过在S或T中增加有关信息来表示联系R
    - 方法2：建立一个单独的关系W表示R
      - T和S的主码作为主码添入W；
      - R的简单属性和复合属性的简单子属性作为简单属性添入W。

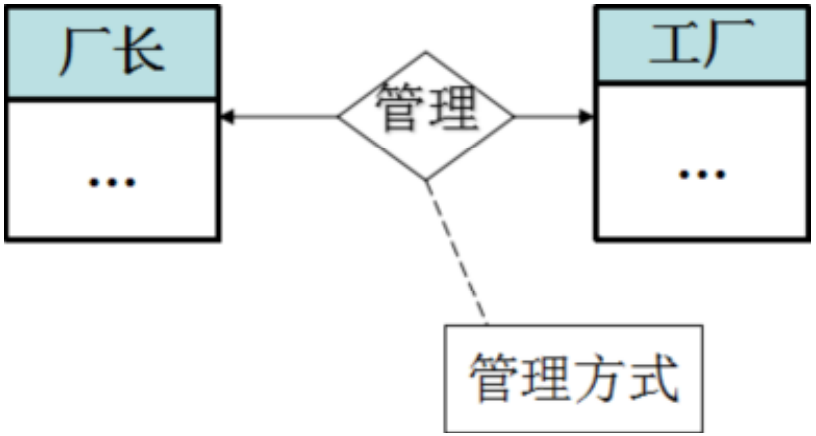
5.1形成初始关系数据库模式

- 实体间联系的变换
  - 1:1联系的变换
    - 例

- 产品(产品号，产品名，职工号)
- 职工(职工号，姓名)

- 产品(产品号，产品名)
- 职工(职工号，姓名，产品号)

- 职工(职工号，姓名)
- 产品(产品号，产品名)
- 负责(职工号，产品号)

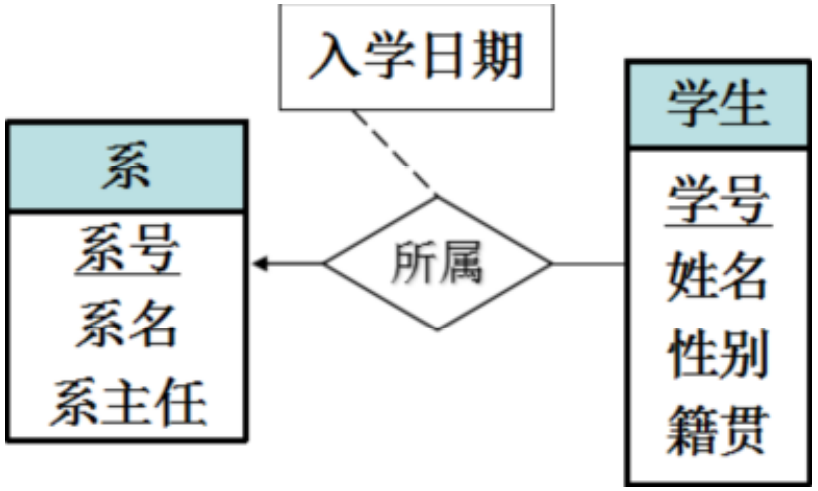


5.1形成初始关系数据库模式

- 实体间联系的变换
  - 1:n联系的变换
    - 设R是从实体集E_1到实体集E_2的1:N联系，S和T是E_1和E_2对应的关系。
    - 方法1：不需建立新关系。由于T的每个实体至多与S的一个实体对应，因此用T来表示R
      - S的主码作为外码添入T;
      - R的简单属性和复合属性的简单子属性作为简单属性添入T。
    - 方法2：建立一个单独的关系W表示R，同1:1联系。

5.1形成初始关系数据库模式

- 实体间联系的变换
  - 1:n联系的变换
    - 例



- 学生(学号，姓名，性别，籍贯)
- 系(系号，系名，系主任)
- 所属(学号，系号，入学日期)

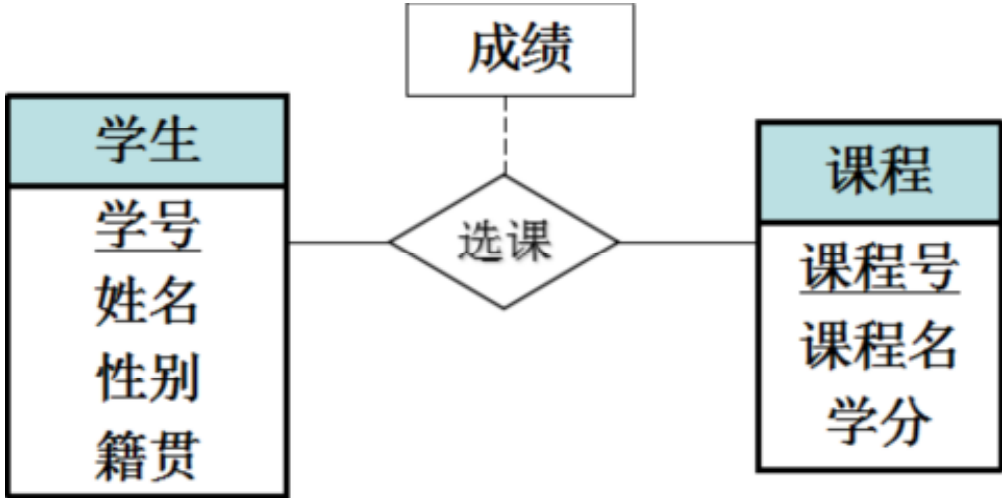
- 学生(学号，姓名，性别，籍贯，系号，入学日期)
- 系(系号，系名，系主任)

5.1形成初始关系数据库模式

- 实体间联系的变换
  - m:n联系的变换
    - 设R是从实体集E_1到实体集E_2的M:N联系，S和T是E_1和E_2对应的关系。
    - 建立一个新关系W来表示R。
    - S和T的主码添入W，既作为外码，也组合起来作为W的主码。
    - W还需要包含R的简单属性和复合属性的简单子属性。

5.1形成初始关系数据库模式

- 实体间联系的变换
  - m:n联系的变换
    - 例



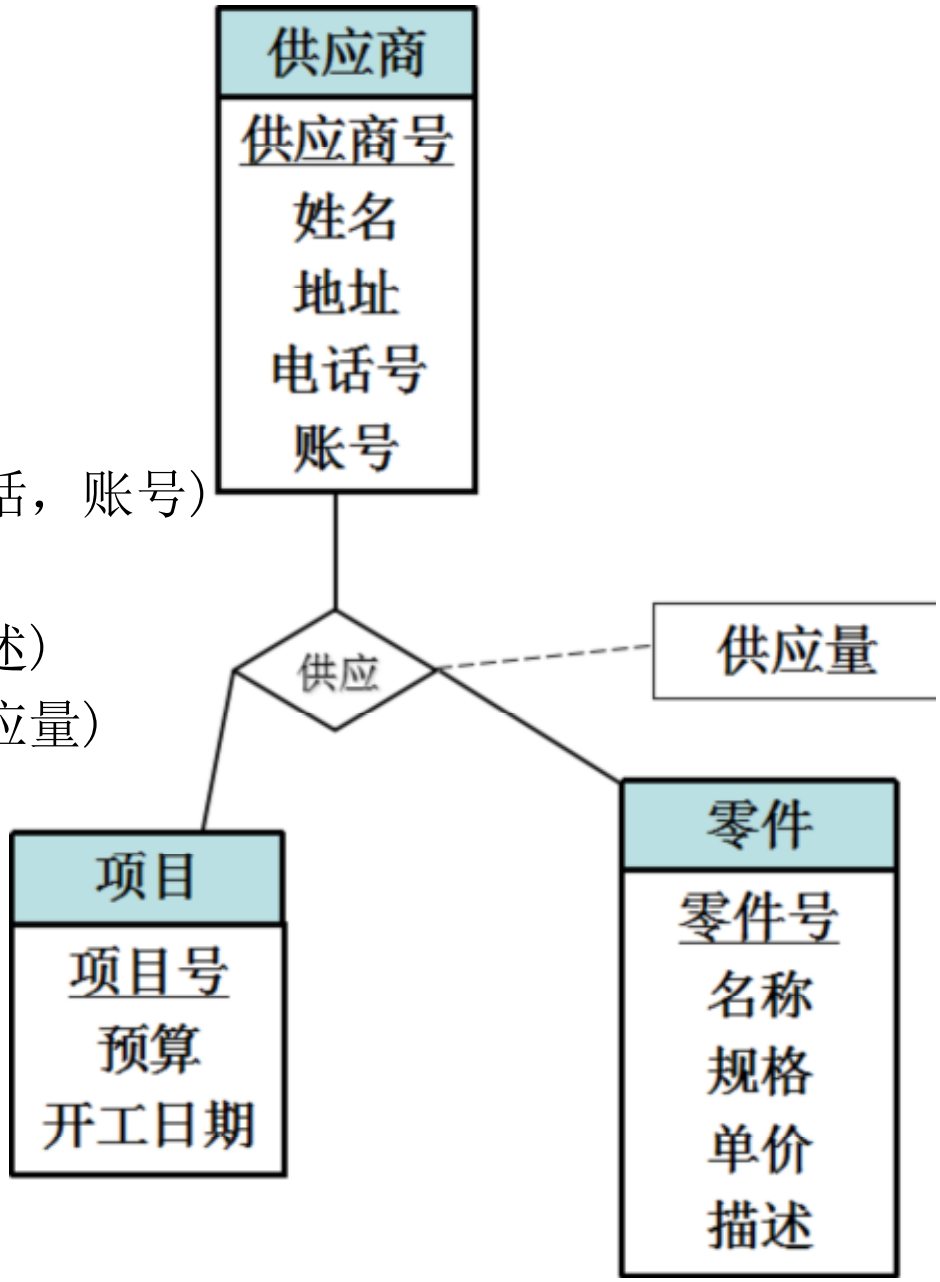
- 学生(学号，姓名，性别，籍贯)
- 课程(课程号，课程名，学分)
- 选课(学号，课程号，成绩)

5.1形成初始关系数据库模式

- 实体间联系的变换
  - n元联系的变换
    - 设R是关联实体集E_1、E_2、…、E_n的n元联系。
    - 类似于M:N联系的表示方法：
      - 需建立一个关系T，用T来表示R。
      - 所有E_i的主码都是T的外码，也组合起来作为T的主码。
      - T还包含R的简单属性和复合属性的简单子属性。

- 实体间联系的变换
  - n元联系的变换
    - 例

- 供应商(供应商号，姓名，地址的，电话，账号)
- 项目(项目号，预算，开工日期)
- 零件(零件号，名称，规格，单价，描述)
- 供应(供应商号，项目号，零件号，供应量)



5.1形成初始关系数据库模式

- 确定函数依赖集
  - 通过前面的步骤，初始关系数据库模式已经形成。
  - 最后，对初始关系数据库模式中的每个关系模式进行深入地分析，与用户协商，确定每个初始关系的函数依赖集，使用关系数据库设计理论，对关系模式进行规范化处理。

逻辑数据库设计

- 逻辑数据库设计的步骤
  - 形成初始关系数据库模式；
  - 关系模式规范化；
  - 关系模式优化；
  - 定义关系上的完整性和安全性约束；
  - 子模式定义；
  - 性能估计。

5.2关系数据库设计理论

- 问题的提出
  - 初始关系模式不是逻辑设计的最终结果，其中某些关系模式可能存在由属性间的函数依赖引起的冗余问题、插入问题、更新问题和删除问题。

5.2关系数据库设计理论

- 问题的提出
  - 例如，建立一个描述学校的数据库。

    涉及的对象包括：

    学生的学号（S#），所在系（SD），

    系主任姓名（MN），课程名（CN），

    成绩（G）
  - 假设学校的数据库模式由一个单一的关系模式Student构成，则该关系模式的属性集合为：

    U ={S#, SD, MN, CN, G}

5.2关系数据库设计理论

一例：

U ={S#, SD, MN, CN, G}

- 现实世界的已知事实：
  - 一个系有若干学生，一个学生只属于一个系
  - 一个系只有一个系主任
  - 一个学生可以选修多门课，每门课有若干学生选修
  - 每个学生所学的每门课程都有一个成绩
- 得到属性集U上的函数依赖
  - F={S#→SD, SD→MN, (S#,CN)→G}

5.2关系数据库设计理论

- 例： U = {S#, SD, MN, CN, G}
  - 存在的问题：
    - 如果一个系刚刚成立，尚无学生，我们无法把这个系及其主任的信息存入数据库，称为**插入异常**。
    - 反过来，如果某个系的学生全部毕业了，我们在删除该系学生信息的同时，把这个系及其主任的信息也删掉了，这称为**删除异常**。
    - 每个系主任的名字在关系中重复出现，出现次数与该系学生人数相同，称为**数据冗余**。
    - 若某个系的系主任更换了，需要修改与该系学生有关的每个元组，称为**更新问题**。

5.2关系数据库设计理论

结论：Student关系模式不是一个好的模式。

- 一个“好”的模式应当不会发生插入异常、删除异常
- 更新异常、数据冗余应尽可能少。

1. 怎样评价一个关系模式的优劣；
2. 怎样将一个不太好的关系模式分解为一组较理想的关系模式。

相关概念

- 函数依赖(*)
  - 定义1：
    - 设R是一个关系模式，U是R的属性集合，X和Y是U的子集。对于R的任意实例 r，r 中任意两个元组 t_1 和 t_2，如果 $t_1[X]=t_2[X]$，则 $t_1[Y]=t_2[Y]$，我们称X函数地确定Y，或Y函数依赖于X，记作 $X \rightarrow Y$。

只能根据数据的语义来确定函数依赖

例如，“姓名”→“年龄”这个函数依赖只有在没有同名人的条件下成立

相关概念

- 函数依赖
  - 如果X→Y而且Y不是X的子集，则称X→Y是非平凡函数依赖。若不特别声明，我们总是讨论非平凡函数依赖。
  - 如果X→Y，我们称X为这个函数依赖的决定属性集。

例：在关系SC(Sno, Cno, Grade)中，
非平凡函数依赖：(Sno, Cno) → Grade
平凡函数依赖：(Sno, Cno) → Sno
(Sno, Cno) → Cno

相关概念

- 函数依赖
  - 定义2：
    - 设R是一个具有属性集合U的关系模式，如果X→Y，并且对于X的任何一个真子集Z，Z→Y都不成立，则称Y完全函数依赖于X。若X→Y，但Y不完全函数依赖于X，则称Y部分函数依赖于X。
  - 定义3：
    - 设R是一个具有属性集合U的关系模式，$X \subseteq U$，$Y \subseteq U$，$Z \subseteq U$，Y→X不成立，Z-X、Z-Y、Y-X不空。如果X→Y，Y→Z，则称Z传递地函数依赖于X。

- 例

Student(Sno, Sname, Ssex, Sage, Sdept)

Sno →Sname, Sno→Ssex, Sno → Sage, Sno → Sdept

(Sno, Sname) → Sdept,

(Sno, Ssex) → Sdept

例：在关系Std(Sno, Sdept, Mname)中，有：

Sno → Sdept，Sdept → Mname，

Mname传递函数依赖于Sno。

- 函数依赖
  - 定义4：
    - 设R是一个具有属性集合U的关系模式，$K \subseteq U$。若$K \rightarrow U$，则称K是R的一个超码(superkey)
    - 超码可以唯一地识别关系的元组。
    - 设R是一个具有属性集合U的关系模式，$K \subseteq U$。如果K是R的一个超码满足下列两个条件，且不存在K的真子集Z使得$Z \rightarrow U$。则称K是R的一个候选码：
    - 一个关系模式中可能具有多个候选码，指定其中的一个作为识别关系元组的主码(主键)。
    - 包含在任何一个候选码中的属性称为键属性。
    - 不包含在任何候选码中的属性称为非键属性。
    - 在最简单的情况下，候选码只包含一个属性。
    - 在最复杂的情况下，候选码包含关系模式的所有属性，称为全键。

- 函数依赖
  - 定义5：
    - 设X是关系模式R的属性子集合。如果X是另一个关系模式的主码，则称X是R的外码。