第7讲 关系模式的范式和规范化

函数依赖、正则化范式与分解理论

基本内容

1. 关系模式的范式
2. 对关系模式进行分解来实现规范化
3. 分解结果的验证方法

重点与难点

概念：关系模式的范式：1NF, 2NF, 3NF, BCNF等

方法：判断数据库设计的正确性和可能存在的问题，掌握对关系模式进行分解和结果验证的方法

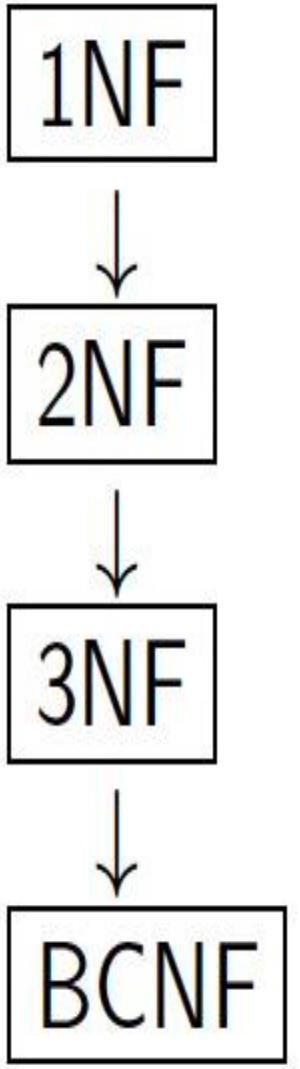
什么是关系的范式(Normal Form, NF)

回顾: 不合理的数据库关系（表）模式设计会导致一些问题:

1. **插入异常（无法插入必要信息）**
2. **删除异常（需保留的信息丢失）**
3. **更新异常（不必要的重复更新）**

范式：关系模式设计的规范和约束，避免出现以上问题。

- 第一范式(1NF)
- 第二范式(2NF)
- 第三范式(3NF)
- Boyce-Codd范式(BCNF)



第一范式(1NF)

定义：若关系模式R(U)中关系的每个分量都是不可分的数据项(值、原子)，则称 R(U)属于第一范式，记为：R(U)∈1NF。

解读：1NF要求表中不存在多值属性和复合属性

示例： Star(name, address(street, city))

Star不属于1NF, 因为属性address仍包含了street和city两个属性，其分量不是原子。

✓ 将非1NF转换为 1NF情况

示例： Star(name, address(street, city)) →

Star(name, address) 或者 Star(name, street, city)

将复合属性处理为简单属性；将多值属性与关键字单独组成一新的关系

第一范式(1NF)

一个表如果能够称为一个关系（relation）那它至少是1NF。

示例： 多值属性

Name	Address
张三	A大街101号 B胡同202号 C小区303号

✓ 将非1NF转换为 1NF情况。注意该关系的主键将发生变化

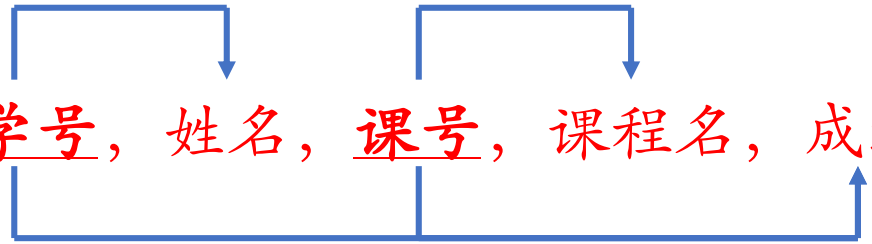
Name	Address
张三	A大街101号
张三	B胡同202号
张三	C小区303号

第二范式(2NF)

[Definition] 2NF:

若R(U)∈1NF 且 U中的每一非主属性完全函数依赖于候选键，则称R(U)属于第二范式，记为：R(U)∈2NF。

解读：（1）满足第一范式（2）不存在非主属性对候选键的部份依赖



示例：学生(学号，姓名，课号，课程名，成绩)　　主键为：{学号，课号}

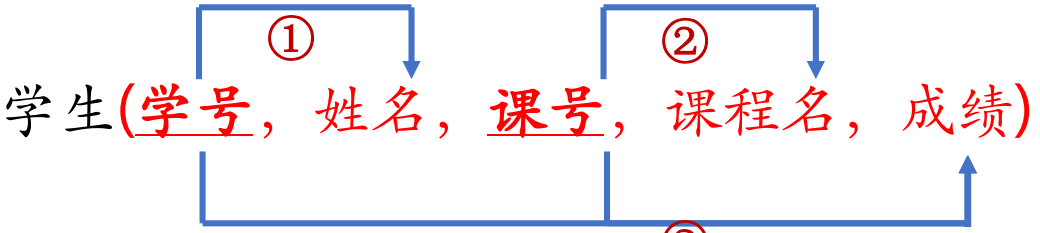
学号→姓名, 课号→课程名 均为非主属性对候选键的部份依赖而非完全依赖!

- 该关系模式不属于第二范式!
- 潜在风险：插入异常（没有选课的学生无法录入，没人选过的课无法录入）
  删除异常（选某课的所有学生都毕业后，该课程信息丢失）

第二范式(2NF)

如何使一个关系模式符合第二范式?

- **拆分为多个关系，每个被依赖的主属性（组合）作为主键单独建立一个表**



关系1：学生（**学号**，姓名） 2NF

关系2：课程（**课号**，课程名） 2NF

关系3：选课（**学号**，**课号**，成绩） 2NF

学生

学号	姓名	课程号	课程名	成绩
98030101	张三	001	数据库	92
98030101	张三	002	计算机原理	85
98030101	张三	003	高等数学	88
98040202	李四	002	计算机原理	90
98040202	李四	003	高等数学	80
98040202	李四	001	数据库	55
98040203	王五	003	高等数学	56
98030102	周六	001	数据库	54
98030102	周六	002	计算机原理	85
98030102	周六	003	高等数学	48

学生

学号	姓名
98030101	张三
98040202	李四
98040203	王五
98030102	周六

课程

课程号	课程名
001	数据库
002	计算机原理
003	高等数学

选课

学号	课程号	成绩
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

规律1：如果一个1NF的关系模式，所有**候选键都是单属性的**，则它一定是2NF

规律2：如果一个1NF的关系模式，**候选键包含所有属性**，则它一定是2NF

第二范式(2NF)

练习：**下列模式是否满足第2范式?　　怎样使其满足第2范式?**

- 学生(学号, 姓名, 班级, 课号, 课程名, 成绩, 教师, 教师职务)

  主属性？非主属性？部分依赖？还是完全依赖

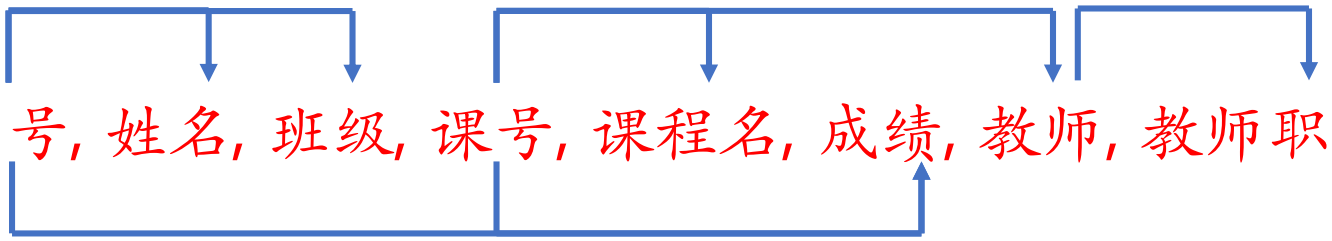
- 员工(员工码, 姓名, 出生日期, 联系电话, 最后学历, 毕业学校, 培训日期, 培训内容)

- 图书(书号, 书名, 出版日期, 出版社, 书架号, 房间号)

第二范式(2NF)

练习：**下列模式是否满足第2范式?　　怎样使其满足第2范式?**

- 学生(学号, 姓名, 班级, 课号, 课程名, 成绩, 教师, 教师职务)



假设每门课只有一位教师授课

候选键：{学号，课号}，　非主属性：姓名，班级，课程名，成绩，教师，教师职务

部份依赖：学号→{姓名，班级}；课号→{课程名，成绩，教师，教师职务}

完全依赖：{学号，课号} $\xrightarrow{f}$ U

关系1：学生（**学号**，姓名，班级）　　2NF

关系2：课程（**课号**，课程名，教师，教师职务）　　2NF

关系3：选课（**学号**，**课号**，成绩）　　2NF

第二范式(2NF)

练习：**下列模式是否满足第2范式?　　怎样使其满足第2范式?**

- 员工(员工码, 姓名, 出生日期, 联系电话, 最后学历, 毕业学校, 培训日期, 培训内容)

  候选键：{员工码，培训日期}

  部分函数依赖：员工码→{姓名, 出生日期, 联系电话, 最后学历, 毕业学校}

  分解结果:

  关系1：员工（员工码，姓名, 出生日期, 联系电话, 最后学历, 毕业学校）

  关系2：培训（员工码，培训日期，培训内容）

- 假设每个员工每天只能参加一个培训，每天可以有多个培训内容。
- 如果每个员工每天可以参加多个培训，则培训内容也是主属性。
- 关系模式定义与业务规则和假设密切相关

第二范式(2NF)

- 图书(书号, 书名, 出版日期, 出版社, 书架号, 房间号)

  候选键：书号（唯一单属性候选键），不存在对候选键部分函数依赖

  根据规律1，该关系模式已经属于2NF

第三范式(3NF)

- 学生(学号, 姓名, 班级, 课号, 课程名, 成绩, 教师, 教师职务)

分解为三个2NF关系模式:

关系1：学生（学号，姓名，班级）

关系2：课程（**课号**，课程名，教师，教师职务）

关系3：选课（学号，课号，成绩）

“教师职务”并非直接依赖于“课号”，而是通过“教师” 传递依赖

课号→ 教师→教师职务

潜在问题：更新异常。如果某教师职务变更，其所有教过的课程记录都需要更新。

原因：没有直接关联的属性（课号，教师职务）出现在了同一个关系模式中。

解决方案：继续优化关系模式设计，使该关系属于第三范式。

第三范式(3NF)

[Definition]3NF

若R(U，F)∈2NF 且 R的每个非主属性都不传递函数依赖于R的候选键，则称R为第三范式关系模式，记为：R(U)∈3NF。

解读：所有非主属性必须直接依赖于候选键，而不是通过其他属性传递依赖。

课程（**课号**，课程名，教师，教师职务）　课号→ 教师→教师职务　非3NF

解决方案：根据依赖关系拆分成多个关系模式。作为传递者的中间非主属性作为主键单独建立一个新关系。同时，保留该属性在原关系中作为一个外键。

关系1：课程（课号，课程名，教师）　3NF

关系2：教师（教师，教师职务）　3NF

其中教师为关系1的外键(Foreign Key)，对应关系2的主键。

第三范式(3NF)

练习：**下列模式是否满足第3范式？怎样使其满足第3范式？**

- 学生(学号，系号，系主任)　　2NF not 3NF
  - 候选键/主键：学号。非主属性：系号，系主任
  - 单属性主键，不可能存在部分依赖，属于2NF
  - 存在传递依赖：学号→系号，系号→系主任，不属于3NF

分解：学生（**学号**，系号），系别（**系号**，系主任），系号为学生的外键

- 员工(员工码，姓名，部门，部门经理)　　2NF not 3NF
  - 候选键/主键：员工码。非主属性：姓名，部门，部门经理
  - 单属性主键，不可能存在部分依赖，属于2NF
  - 存在传递依赖：员工码→部门，部门→部门经理，不属于3NF

分解：员工（员工码，部门），部门（部门，部门经理），部门为员工的外键

Boyce-Codd范式(BCNF)

[Definition]BCNF

（1）若R(U，F)∈1NF，且对F中任意非平凡函数依赖X→Y，X都是R的一个超键，则称R(U)属于Boyce-Codd范式，记为：R(U)∈BCNF

或者

（2）若R(U，F)∈3NF，且任意主属性都只直接完全函数依赖于R的候选键，而不存在对任何其他属性（组合）的函数依赖，则称R(U)属于Boyce-Codd范式。

示例：邮编(城市, 街道, 邮政编码)

函数依赖：{城市, 街道} → 邮政编码; 邮政编码→城市.

候选键：{城市,街道} $\xrightarrow{f}$ U

因主属性依赖一个非候选键：邮政编码 → 城市；所以不满足BCNF

因无传递依赖，所以满足第3范式；

Boyce-Codd范式(BCNF)

- 其他满足3NF但不满足BCNF的例子?

- 公司项目（项目，部门，对接人）
- {项目，部门}→对接人 （每个项目在每个部门只有一位对接人）
- 对接人→部门 （每个对接人，也是员工，只隶属于一个部门）
- 存在主属性对非主属性的函数依赖，不满足BCNF。但是满足3NF。
- 非BCNF有什么问题? 更新异常（比如一个对接人换部门…）

- BCNF有时无法通过分解实现，因为对3NF关系进行分解可能导致约束丢失。
- 分解的关系过多导致查询时需要大量连接操作，影响效率
- 3NF总能够达到，一般可满足应用中的多数需求。

范式之间的递进关系

1NF

↓

2NF

↓

3NF

↓

BCNF

消除非主属性对候选键的部分函数依赖

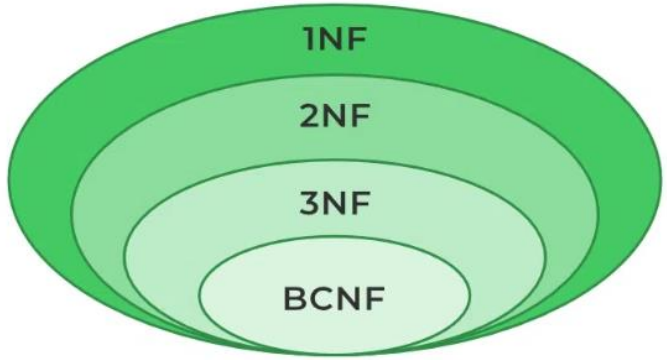
满足第二范式的一定满足第一范式，反过来不一定

消除非主属性对候选键的传递函数依赖

满足第三范式的一定满足第二范式，反过来不一定

消除主属性对非候选键的函数依赖

满足第BCNF范式的一定满足第三范式，反过来不一定



模式分解方法及检验

模式分解的经验性方法（即正规化, Normalization）：

1NF → 2NF:

1. 针对每一个被部分依赖的主属性（或主属性组合），建立一个新的关系模式，并将被依赖的主属性（或主属性组合）作为新模式的主键，将依赖该主属性（或组合）的非主属性添加进来。
2. 原关系模式中只保留完整的候选键，以及所有完全依赖于该候选键的属性。将每一个被部份依赖的主属性（或主属性组合）定义为原关系模式的外键，与对应的新模式相关联。

2NF → 3NF:

1. 将作为传递依赖中介的非主属性作为主键，建立一个新的关系模式，并将对其依赖的非主属性都移至新关系模式中。原关系模式保留该中介属性作为一个外键。

模式分解问题的形式化定义

[Definition]模式分解

关系模式R(U)的分解是指用R的一组子集$\rho=\{R_1(U_1),\cdots,R_k(U_k)\}$来代替它。其中$U=U_1\cup U_2\cup\cdots\cup U_k$；$U_i\not\subset U_j\ (i\neq j)$。

注：为便于后面叙述，我们用R_i代替$R_i(U_i)$，R代替R(U)。

分解后的模式需要保证

（1）仍然可以通过连接操作获得与R完全等价的数据内容（无损连接性）

（2）分解后仍然保持与R完全等价的数据依赖约束（保持依赖性）

$m_\rho(r)=\pi_{R_1}(r)\bowtie\ \ldots\ \bowtie\pi_{R_k}(r)=\bowtie_{(i=1,\ldots,k)}\pi_{R_i}(r)$　　这里：$\pi_{Ri}(r)$是分解后关系模式R_i里的数据

$m_\rho(r)$为R向ρ的投影连接，即通过连接(Join)操作把分解后的表再连起来的结果

无损连接分解(Lossless join decomposition)

- 对关系模式R的分解ρ，可以通过对ρ＝{R_1,…,R_k}成员的自然连接操作（Natural Join）将R中数据恢复（不会缺失，也不会产生额外数据）。

  即： $m_\rho(r) = r$

- 自然连接操作：用R_1与R_2中共有的属性作为连接条件，且该属性必须为R_1 R_2至少一方的候选键或超键。

- 正例：R= 课程（**课号**，课程名，教师，教师职务）

  分解为： R_1 = 课程（**课号**，课程名，教师），R_2 = 教师（教师，教师职务）

课号	课程名	教师
001	线性代数	李老师
002	数据结构	王老师

⋈

教师	教师职务
李老师	系主任
王老师	无

=

课号	课程名	教师	教师职务
001	线性代数	李老师	系主任
002	数据结构	王老师	无

无损连接分解(Lossless join decomposition)

- 反例：R= 课程（**课号**，课程名，教师，教师职务）

  分解为： R_1 = 课程（**课号**，课程名），R_2 = 教师（**教师**，教师职务）

课号	课程名
001	线性代数
002	数据结构

⋈

教师	教师职务
李老师	系主任
王老师	无

=

课号	课程名	教师	教师职务
001	线性代数	李老师	系主任
001	线性代数	王老师	无
002	数据结构	李老师	系主任
002	数据结构	王老师	无

有损分解（产生额外数据！）

原因：教师作为外键没有保留在课程里，导致课程与教师的对应关系丢失。

如何判断无损连接分解

[定理] 设F是关系模式R上的一个函数依赖集合。ρ={R_1,R_2}是R的一个分解，则：当且仅当$R_1 \cap R_2 \rightarrow R_1 - R_2$或者$R_1 \cap R_2 \rightarrow R_2 - R_1$属于$F^+$时，ρ是关于F无损连接的。

- 正例：R= 课程（**课号**，课程名，教师，教师职务），

  F = {课号→课程名，课号→教师， 教师→教师职务}

分解为： R_1 = 课程（**课号**，课程名，教师），R_2 = 教师（教师，教师职务）

$R_1 \cap R_2$ =教师， $R_2 - R_1$ =教师职务，教师→教师职务 存在于F^+中

故为ρ无损链接分解。

如何判断无损连接分解

- 另一个正例：R_1 = 课程（课号，课程名，教师），R_2 = 教师（课号，教师，教师职务）

  $R_1 \cap R_2$ ={课号，教师}，$R_2 - R_1$ =教师职务， $R_1 - R_2$ =课程名

  {课号，教师}→教师职务，{课号，教师}→课程名 均存在于F^+中

  故该分解也是无损连接分解。但是R_2仍存在传递依赖，故不是3NF。

- **反例**：R_1 = 课程（课号，课程名），R_2 = 教师（教师，教师职务）

  $R_1 \cap R_2$ ={}，$R_2 - R_1$ ={教师，教师职务}， $R_1 - R_2$ ={课号，课程名}

  {}空集不能决定任何其他属性，也不存在于函数依赖集合的左端。

  故该分解是有损连接分解。

保持依赖分解(Dependency-Preserving Decomposition)

[Definition]**保持依赖分解**

对于关系模式R(U，F)，U是属性全集，F是函数依赖集合，ρ={$R_1(U_1)$,..., $R_n(U_n)$}是R的一个分解，如在π_{Ri}(F)中的所有依赖之并集(i=1,...,k),逻辑蕴涵F的每个依赖，则称分解ρ保持依赖集F。

其中π_{Ri}(F)是F在R_i上的投影，即F中的任一投影X→Y，如果X, Y均包含于R_i,则X→Y∈π_R(F)。

解读：该类分解不会导致原有的函数依赖丢失。注：如果可以通过新的函数依赖推导出原有函数依赖，则不算丢失。

正例：R= 课程（**课号**，课程名，教师，教师职务）

保持依赖分解！

- 原有函数依赖：课号→{课程名，教师}，教师→教师职务

分解为：R_1 = 课程（**课号**，课程名，教师），R_2 = 教师（教师，教师职务）

- 新的函数依赖：课号→{课程名，教师}，　　教师→教师职务

保持依赖分解(Dependency-Preserving Decomposition)

反例：R= 课程（**课号**，课程名，教师，教师职务）

- 原有函数依赖：课号→{课程名，教师}，教师→教师职务

  分解为：R_1 = 课程（**课号**，课程名），R_2 = 教师（**教师**，教师职务）

- 新的函数依赖：课号→课程名，　　教师→教师职务

原有函数依赖 **课号→教师** 丢失！且无法根据新的依赖推断出来。

因此，该分解不是保持依赖分解。

保持依赖分解(Dependency-Preserving Decomposition)

例子：R(C, S, Z), C是城市，S是街道, Z是邮政编码

F={ CS→Z, Z→C }　ρ={R_1(SZ), R_2(CZ)}

R_1={街道，邮编}，　R_2 = {城市，邮编}

分解后是否有约束丢失了?

R_1中只有城市和街道的组合；R_2中只有城市和邮编的组合，CS→Z的依赖关系丢失。所以不是保持依赖分解！

那么该分解是否是无损连接分解呢？应用刚刚讲的判定定理。

注意：(1)保持依赖的分解可能不是无损连接的。 (2)无损连接的分解可能不是保持依赖的。

对范式进行连接无损且保持依赖的分解以达到BCNF有时候无法做到。

如何进行无损连接分解

- **对模式进行无损连接分解成BCNF的方法**
- **输入模式：R = (U, F) ∉ BCNF**
- **输出模式：无损链接分解ρ ={R_1, R_2, … R_n}，其中R_i ∈ BCNF**

1. 令ρ ={**R**}。
2. 对每个模式s∈ρ，若s∉BCNF，则s上必有X→A成立且X不是s的超键且 A∉X，用模式s_1，s_2替代s。s_1={X, A}，s_2={s - A} (可以发现，s_1∈BCNF)。X为s_2外键。
3. 重复步骤(2)，直至ρ中全部关系模式达到BCNF 。

对任意关系模式R，总可以通过无损连接分解达到BCNF（但不保证保持依赖）

如何进行无损连接分解

示例：R(A, B, C, D, E, F, G) 函数依赖集合{ A→B, A→C, C→D, C→E, E→FG }

候选键：A　　有传递依赖，R不满足3NF。

使用前述算法进行无损连接分解。

首先按左端合并函数依赖:{A→BC, C→DE, E→FG}

0. ρ={ R(A, B, C, D, E, F ,G} }　(2NF)

1. ρ={ R1(A, B, C, D, E), R2(E, F ,G} }　　//提取出E→FG

　　2NF(继续分解)　　BCNF

2. ρ={ R1(A, B, C), R2(C, D, E), R3(E, F ,G} }　　//提取出C→DE

　　BCNF　　BCNF　　BCNF

最终结果，每一个关系模式都属于BCNF， ρ同时即使无损连接的也是保持依赖的

如何进行保持依赖分解

- **对模式进行保持依赖分解成的方法**
- **输入模式：R = (U, F)，F是函数依赖最小覆盖集**
- **输出模式：保持依赖分解 $\rho=\{R_1, R_2, \cdots R_n\}$，其中$R_i \in$ 3NF**

1. 把R中不出现在F中的属性去掉并单独组成一模式。　　//摘除完全独立的信息
2. 对F中的每一组左端相同的函数依赖$X\rightarrow A_1$, $X\rightarrow A_2$, …, $X\rightarrow A_m$, 以$(X, A_1, A_2, \cdots, A_m)$组成一模式，其中X为主键/候选键。
3. 取ρ为上述新模式之集合，则ρ即为所求之分解。

对任意关系模式R, 总可以通过保持依赖分解达到3NF，但不一定达到BCNF!

如何进行保持依赖分解

示例：R(A, B, C, D) 函数依赖集合F = {AB→C, C→D, C→B }

候选键：AB　　有传递依赖，R不满足3NF。

- 使用前述算法进行保持依赖分解。首先合并左端：{AB→C, C→BD}
- 对每一组左端相同的函数依赖，建立一个单独的关系：

- ρ={ R1(A, B, C), R2(C, B, D) }

R2为BCNF，但是R1不符合BCNF (因为存在C→B，主属性对非主属性的函数依赖)

- 最终结果，每一个关系模式都属于3NF，ρ是保持依赖且同时无损连接的
- 如果继续强行对R1进行无损链接分解来达到BCNF，则会破坏保持依赖性：

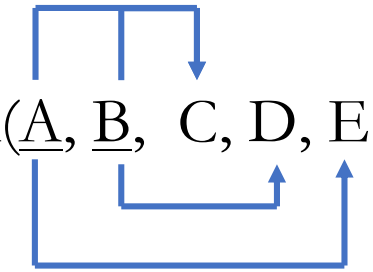
  ρ'={ R1(A, C), R2(B, C)}　依赖AB→C丢失

  之前的例子：A = 城市，B = 街道，C = 邮编 就是这种情况。

一般性模式分解策略

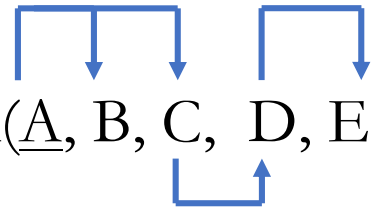
模式分解的常用方法（同时满足无损链接性和依赖保持性）：

- 1NF → 2NF: 每组左端相同的非冗余函数依赖单独建立一个新模式。



R(A, B, C, D, E)　$\mathbf{R}_1$(A, B, C), $\mathbf{R}_2$(A, E), $\mathbf{R}_3$(B, D), A,B为R_1外键也是候选键

- 2NF → 3NF: 每组左端相同的非冗余函数依赖单独建立一个新模式。



R(A, B, C, D, E)　$\mathbf{R}_1$(A, B, C), $\mathbf{R}_2$(C, D), $\mathbf{R}_3$(D, E), C 为R_1外键, D为$\mathbf{R}_2$外键。

总结

- 关系模式范式
  - 1NF → 2NF → 3NF → BCNF (还有第四第五范式，一般用不到，可自己看书)
  - 如何判断一个关系模式属于哪一级范式

- 关系模式的正规化和分解
  - 经验性方法1NF→2NF, 2NF→3NF
  - 保持依赖分解与无损连接分解：互不排斥，互不包含。
  - 两种分解的判定方法
  - 将一个关系模式转换为BCNF的无损连接且保持依赖分解有可能不存在。