



◆ **教育无非是一切已学过的东西都忘掉后所剩下的东西。**

◆ **重要的不是获得知识，而是发展思维能力。**

——1914年诺贝尔奖得主德国物理学家：劳厄



第六章

关系数据理论



主要内容

6.1 问题的提出

6.2 规范化 (重点)

6.3 数据依赖的公理系统

6.4 *模式的分解 (了解)





6.1 问题的提出

- ◆ 针对一个具体问题，应**如何构造一个适合于它的数据模式**，即应该构造几个关系，每个关系由哪些属性组成等。
- ◆ 数据库逻辑设计的工具——关系数据库的规范化理论



关系数据库的规范化理论主要包括三个方面内容：

- 函数依赖
- 范式 (Normal Form)
- 模式设计

其中，**函数依赖**起着核心的作用，是模式分解和模式设计的基础，范式是模式分解的**标准**。



一、概念回顾

- ❖ **关系**
- ❖ **关系模式**
- ❖ **关系数据库**
- ❖ **关系数据库的模式**



二、关系模式的形式化定义

关系模式由五部分组成，即它是一个五元组：

$R(U, D, DOM, F)$

- R:** 关系名
- U:** 组成该关系的属性名集合
- D:** 属性组U中属性所来自的域
- DOM:** 属性向域的映象集合
- F:** 属性间数据的依赖关系集合。即限定了组成关系的各个元组必须满足的完整性约束条件。



三、关系模式的简化表示

❖ 关系模式 $R (U, D, DOM, F)$

简化为一个三元组：

$R (U, F)$

❖ 当且仅当 U 上的一个关系 r 满足 F 时， r 称为关系模式 $R (U, F)$ 的一个关系



四、什么是数据依赖

1. 定义一个关系内部属性与属性之间的一种约束关系（主要体现于值的**相等与否**），这就是**数据依赖**，它是数据库模式设计的关键。

例如：学号对应一个学生，一个学生在一个系学习。

属性间的这种依赖关系类似于 $y=f(x)$ 。



2. 数据依赖

- ❖ 是通过一个关系中**属性间值的相等与否**体现出来的数据间的相互关系;
- ❖ 是现实世界属性间相互联系的抽象;
- ❖ 是数据内在的性质;
- ❖ 是**语义的体现**。



3. 数据依赖的主要类型

- **函数依赖 (Functional Dependency, 简记为FD)**
- **多值依赖 (Multivalued Dependency, 简记为MVD)**
- **其他, 如连接依赖**



五、数据依赖对关系模式的影响

[例] 建立一个描述学校教务的数据库。

涉及的对象包括：

学生的学号 (Sno)

所在系 (Sdept)

系主任姓名 (Mname)

课程号 (Cno)

成绩 (Grade)

单一的关系模式： Student $\langle U, F \rangle$

$U = \{ Sno, Sdept, Mname, Cno, Grade \}$



现实世界的已知事实告诉我们：

- 1. 一个系有若干学生， 但一个学生只属于一个系；**
- 2. 一个系只有一名主任；**
- 3. 一个学生可以选修多门课程， 每门课程有若干学生选修；**
- 4. 每个学生所学的每门课程都有一个成绩。**



思考：这个关系中数据有问题吗？
码是什么？

Sno	Sdept	Mname	Cno	Grade
S1	计算机系	张明	C1	95
S2	计算机系	张明	C1	90
S3	计算机系	张明	C1	88
S4	计算机系	张明	C1	70
S5	计算机系	张明	C1	78
...

某一时刻关系模式student的一个实例



关系模式Student $\langle U, F \rangle$ 中存在的问题:

1 数据冗余太大

-----浪费大量的存储空间

例：每一个系主任的姓名重复出现，重复次数与该系所有学生的所有课程成绩出现次数相同。



2 更新异常 (Update Anomalies)

数据冗余，更新数据时，维护数据完整性代价大。

例：某系更换系主任后，系统必须修改与该系学生有关的每一个元组。



3 插入异常 (Insertion Anomalies)

----该插入的数据插入不进去

例，如果一个系刚成立，尚无学生，我们就
无法把这个系及其系主任的信息存入数据库。



4 删除异常 (Deletion Anomalies)

----不该删除的数据不得不删

例，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，把这个系及其系主任的信息也丢掉了。



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。

原因： 由存在于模式中的某些数据依赖引起的。

解决方法： 通过分解关系模式来消除其中不合适的数据依赖。



把这个单一模式分成3个关系模式：

$S (\mathbf{Sno}, \mathbf{Sdept}, Sno \rightarrow Sdept) ;$

$SC (\mathbf{Sno}, \mathbf{Cno}, \mathbf{Grade}, (Sno, Cno) \rightarrow Grade) ;$

$DEPT (\mathbf{Sdept}, \mathbf{Mname}, Sdept \rightarrow Mname)$



思考：这个模式有四个“毛病” ???

为什么会发生插入异常和删除异常呢？

----存在不合适的函数依赖

解决之道：分解! 分解!! 再分解!!!

基本原则： 一事一地



如 student(Sno, Sdept, Dname, Cno, Grade)

涉及到:

学生

系

学生选课 三件事



规范化理论正是用来改造关系模式，通过分解关系模式来消除其中不合适的数据依赖，以解决插入异常、删除异常、更新异常和数据冗余问题。



要设计一个“好”的关系模式方案，其根本方法是要弄清属性间的内在语义联系。

也即两种依赖联系

(1) 函数依赖

(2) 多值依赖



主要内容

6.1 问题的提出

6.2 规范化 (重点)

6.3 数据依赖的公理系统

6.4 *模式的分解 (难)





6.2 规范化

关系数据库中的每个关系模式的属性间一定要满足某种**内在联系**，而这种联系又可将**对关系的不同要求**分为若干等级，这就叫做**关系的规范化** (normalization)。



规范化的目的主要是为：

克服数据库逻辑结构中的插入异常，删除异常以及冗余度大的缺陷，从而增强数据库结构的稳定性和灵活性。

规范化的过程：模式分解



一、函数依赖

定义6.1: 设 $R(U)$ 是属性集 U 上的关系模式。 X, Y 是 U 的子集。若对于 X 上的每一个具体值, Y 有唯一的具体值与之对应, 也即:

X 上的属性值确定后, Y 上的属性值必确定

则称 X 函数确定 Y 或 Y 函数依赖于 X , 记作 $X \rightarrow Y$



或说: r 是 $R(U)$ 上的任意关系, 如果成立

对 $\forall t, s \in r,$

若 $t[X] = s[X],$

则 $t[Y] = s[Y]$

X 称为决定因素, Y 称为依赖因素



比如描述课程的关系，有：

课程号(Cno),课程名(Cname),先修课(Cpno) ,学分(Credit)

Course(Cno, Cname, Cpno , Credit)

称 Cno函数决定Cname, Cpno , Credit

或者说Cname, Cpno , Credit函数依赖于Cno,

记为：

$Cno \rightarrow Cname$, $Cno \rightarrow Cpno$, $Cno \rightarrow Credit$



再如：

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

存在函数依赖：

Sno \rightarrow Sname

Sno \rightarrow Ssex

Sno \rightarrow Sdept

Sno \rightarrow Sage

思考：存在函数依赖**Sname \rightarrow Sage**吗？



- 注意：
1. 函数依赖不是指关系模式R的某个或某些关系实例满足的约束条件，而是指R的所有关系实例均要满足的约束条件。
 2. 函数依赖是语义范畴的概念。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不允许有同名人的条件下成立



定义6.2: 若有 $X \rightarrow Y$, 但 $Y \not\subseteq X$, 则称 $X \rightarrow Y$ 是**非平凡**的函数依赖。

若 $X \rightarrow Y$, 但 $Y \subseteq X$ 则称 $X \rightarrow Y$ 是**平凡**的函数依赖。



思考：在关系SC(Sno, Cno, Grade)中
非平凡函数依赖？？

----- (Sno, Cno) \rightarrow Grade

平凡函数依赖：

---(Sno, Cno) \rightarrow Sno

----- (Sno, Cno) \rightarrow Cno

注意：若不特别声明，总是讨论非平凡的函数依赖



定义6.3: 在 $R(U)$ 中, 如果 $X \rightarrow Y$, 并且对于 X 的任何一个真子集 X' 有 $X' \not\rightarrow Y$, 则称 **Y 对 X 完全函数依赖**

记作: $X \xrightarrow{f} Y$ 。

由定义知, 当 X 是**单个**属性时, 由于 X 不存在任何真子集, $X \xrightarrow{f} Y$



若 $X \rightarrow Y$, 但 Y 不完全函数依赖于 X (即存在 X 的真子集 X' , 满足 $X' \rightarrow Y$), 则称 Y 对 X 部分函数依赖, 记作 $X \xrightarrow{p} Y$ 。



例3：设有关系模式选课

SC1(S#,C#,GRADE,CREDIT)

在SC1中，由于一个学生可选多门课，一门课程可由多个学生选修，因此，S#,C#,都不能单独决定GRADE,只能由组合 (S#,C#) 决定。由此可知

$(S\#,C\#) \xrightarrow{f} \text{GRADE}$

$(S\#,C\#) \xrightarrow{P} \text{CREDIT}$

(因为 $C\# \rightarrow \text{CREDIT}$)



定义6.4 在 $R(U)$ 中, 如果 $X \rightarrow Y (Y \subseteq X)$, $Y \not\rightarrow X$,

$Y \rightarrow Z (Z \subseteq Y)$

则称 Z 传递函数依赖于 X 。

记为: $X \xrightarrow{\text{传递}} Z$



如: $S1(S\#, Sname, D\#, D_name, Location)$

存在传递函数依赖

$S\# \rightarrow location \quad (S\# \rightarrow D\# \rightarrow location)$



二、码

定义6.5 设 K 为 $R\langle U, F \rangle$ 中的属性或属性组合,
若 $K \xrightarrow{F} U$, 则 K 为 R 的候选关键字 (码) (Candidate key)。

如: Student (**Sno**, **Sname**, Ssex, Sdept, Sage)



若**U**部分依赖于**K**

$$K \xrightarrow{P} U$$

则 **K**称为超码，如

Student (Sno, Sname, Ssex, Sdept, Sage)
(**Sno, Sname**) 为超码

因此，候选码是最小的超码



候选码有两个性质：

- (1) **标志的唯一性**：对 $R(U)$ 中的每一个元组， K 确定后，元组就相应确定了；
- (2) **无冗余性**：当 K 是属性组时， K 的任一部分不能确定元组。



- ◆ 若候选码多于一个，则选定其中的一个为主关键字（码）（Primary key）
- ◆ 包含在任何一个候选码中的属性，叫做主属性 (Prime attribute)。
- ◆ 不包含在任何码中的属性称为非主属性 (Nonprime attribute) 或非码属性 (Non-key attribute)。



在例3关系

SC1(S#, C#, GRADE, CREDIT)中, 属性组

(S#, C#) 是候选码

S#, C#是主属性,

GRADE, CREDIT是非主属性。



最简单的情况，单个属性是码；最极端的情况，
所有属性都为码

如 S (SNO , SDEPT , SAGE)

SC(SNO , CNO , GRADE)

R(P, W, A) ——(P182例6.3)



定义6.6 关系模式R中属性或属性组X并非R的码，
但X是另一个关系模式的码，则称 X是R的外部码
(Foreign key)也称**外码**。

主码与外部码提供了一个**表示关系间联系**的手段。



三、范式(Normal Form)

范式是符合**某一种级别**的关系模式的集合。

关系数据库中的关系必须满足一定的要求。

满足不同程度要求的为不同范式。



范式的种类:

第一范式(1NF)

第二范式(2NF)

第三范式(3NF)

BC范式(BCNF)

第四范式(4NF)

第五范式(5NF)



各种范式之间存在联系：

$$1NF \supset 2NF \supset 3NF \supset BCNF \supset 4NF \supset 5NF$$

某一关系模式R为第n范式，可简记为 $R \in nNF$ 。



一个低一级范式的关系模式，通过**模式分解**
(Schema Decomposition)可以转换为若干个
高一级范式的关系模式的集合
-----**规范化 (Normalization)**。



第一范式 (1NF)

❖ 1NF的定义

定义6.7 如果一个关系模式R的**所有属性都是不可分**的基本数据项，则 **$R \in 1NF$** 。

- ❖ 第一范式是对关系模式的**最起码的要求**。不满足第一范式的数据库模式不能称为关系数据库。
- ❖ 满足第一范式的关系模式**并不一定是一个“好”**的关系模式。



[例]: 关系模式 SLC(Sno, Sdept, Sloc, Cno, Grade)
Sloc为学生住处, 假设每个系的学生住在同一个地方。

函数依赖包括:

$(Sno, Cno) \xrightarrow{f} Grade$

$Sno \rightarrow Sdept$

$(Sno, Cno) \xrightarrow{P} Sdept$

$Sno \rightarrow Sloc$

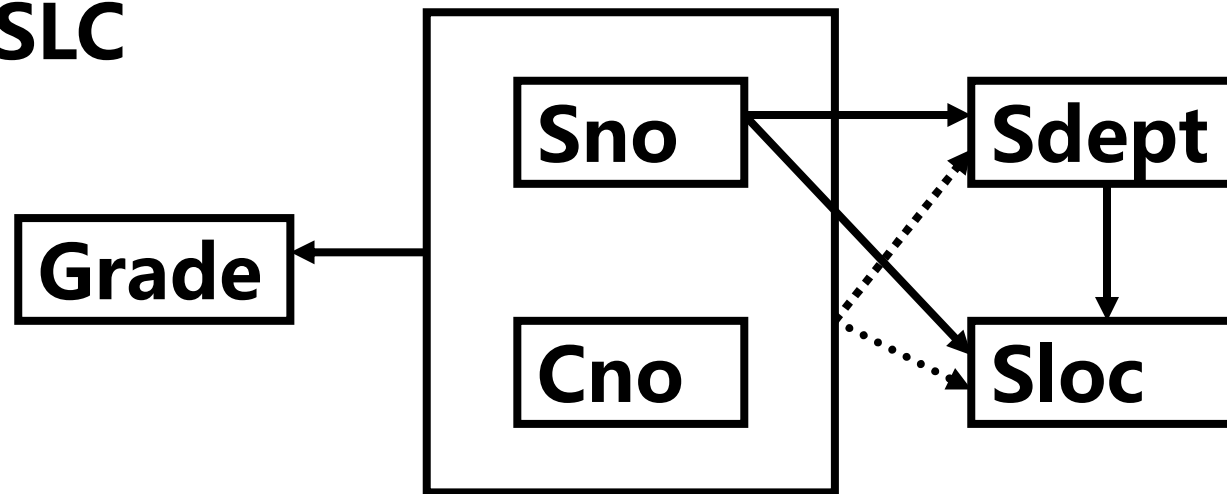
$(Sno, Cno) \xrightarrow{P} Sloc$

$Sdept \rightarrow Sloc$



函数依赖图:

SLC



- ❖ SLC的码为(Sno, Cno)
- ❖ 图中实线表示完全函数依赖，虚线表示部分函数依赖。
- ❖ 结论:
 1. SLC满足第一范式。
 2. 非主属性Sdept和Sloc部分函数依赖于码(Sno, Cno)。



SLC存在的问题

- (1) 插入异常**
- (2) 删除异常**
- (3) 数据冗余度大**
- (4) 修改复杂**

因此, SLC不是一个 “好” 的关系模式。



❖ 原因

Sdept、Sloc**部分函数依赖于码。**

❖ 解决方法

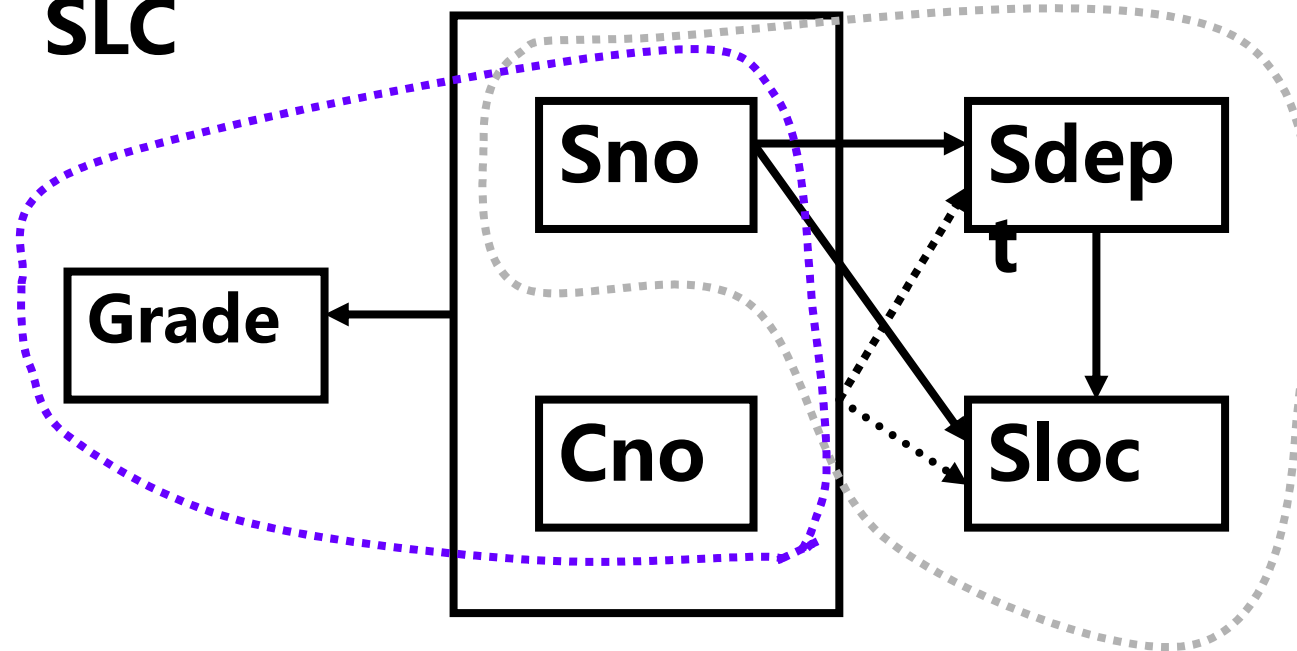
采用**投影分解法**，把SLC分解为两个关系模式，以消除这些部分函数依赖。

SC (Sno, Cno, Grade)

SL (Sno, Sdept, Sloc)



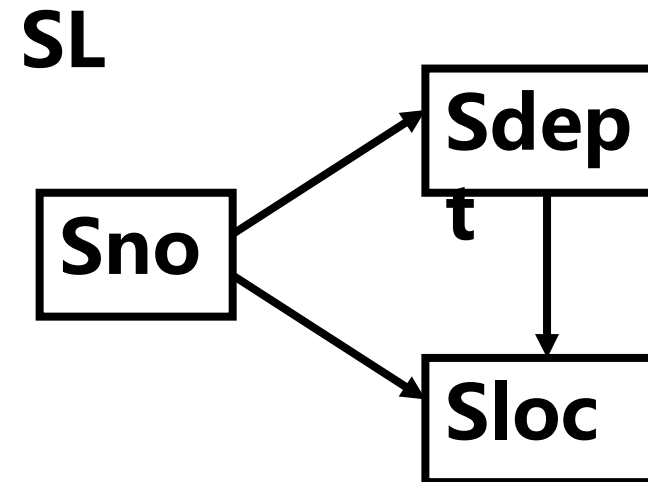
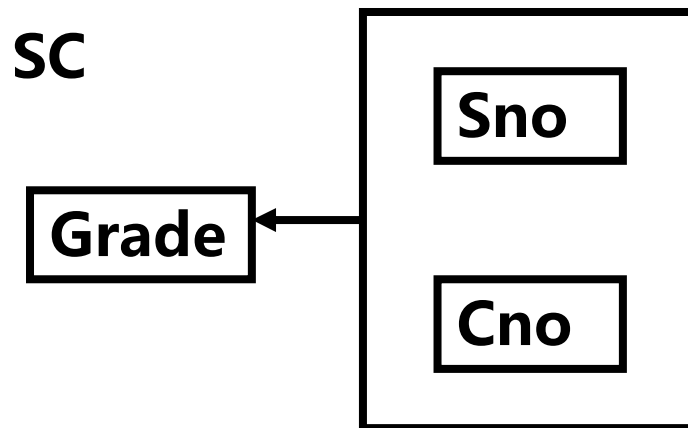
SLC



❖ **SLC的码为(Sno, Cno)**



函数依赖图:



- ❖ 关系模式SC的码为 (Sno, Cno)
- ❖ 关系模式SL的码为Sno
- ❖ 这样非主属性对码都是完全函数依赖



四 第二范式 (2NF)

2NF的定义:

定义6.8 若关系模式 $R \in 1NF$, 并且每一个**非主属性**都**完全**函数依赖于 R 的码, 则 $R \in 2NF$ 。

例: $SLC(Sno, Sdept, Sloc, Cno, Grade) \in 1NF$

$SC(Sno, Cno, Grade) \in 2NF$

$SL(Sno, Sdept, Sloc) \in 2NF$



- ❖ 采用投影分解法将一个1NF的关系分解为多个2NF的关系，可以在一定程度上**减轻**原1NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
- ❖ 将一个1NF关系分解为多个2NF的关系，并**不能完全消除**关系模式中的各种异常情况和数据冗余。



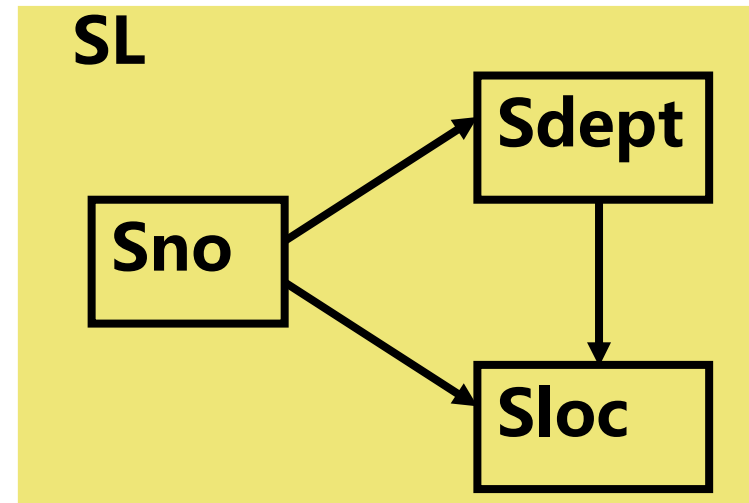
[例]：2NF关系模式SL(Sno, Sdept, Sloc)中

❖函数依赖：

Sno→Sdept

Sdept→Sloc

Sno→Sloc



Sloc传递函数依赖于Sno，即SL中存在非主属性对码的传递函数依赖。



SL关系存在的问题：

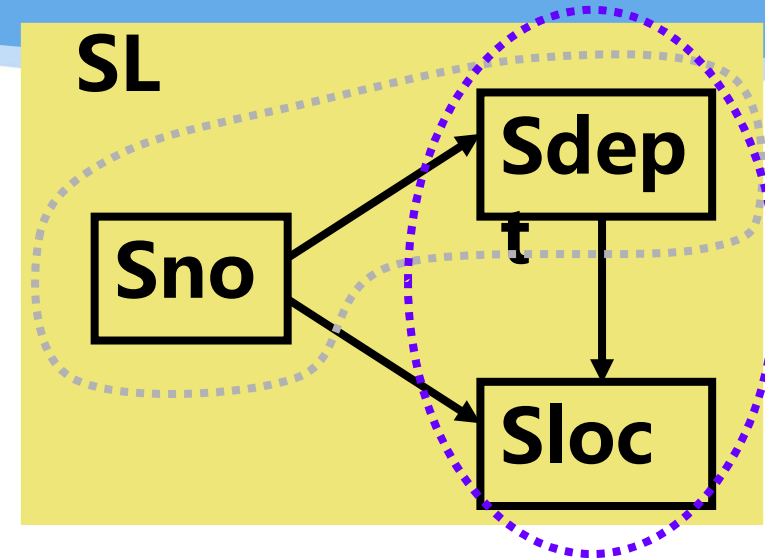
- (1) 插入异常**
- (2) 删除异常**
- (3) 数据冗余度大**
- (4) 修改复杂**

所以SL仍不是一个“好”的关系模式。



原因

Sloc传递函数依赖于Sno



解决方法

采用投影分解法，把SL分解为两个关系模式，以消除传递函数依赖：

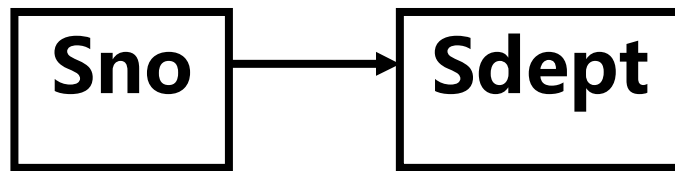
SD (Sno, Sdept)

DL (Sdept, Sloc)

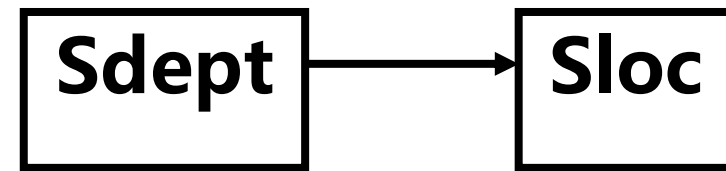
SD的码为Sno， DL的码为Sdept。



SD的码为Sno, DL的码为Sdept.



SD



DL



分解后的关系模式中，既没有非主属性对码的部分函数依赖也没有非主属性对码的传递函数依赖，在**一定程度上解决**了上述四个问题：

- (1) DL关系中可以插入无在校学生的系的信息。
- (2) 某个系的学生全部毕业了，只是删除SD关系中的相应元组，DL关系中关于该系的信息仍存在。
- (3) 关于系的住处的信息只在DL关系中存储一次。
- (4) 当学校调整某个系的学生住处时，只需修改DL关系中一个相应元组的Sloc属性值。



五 第三范式 (3NF)

3NF的定义:

定义6.9 关系模式 $R<U, F>$ 中若**不存在**这样的**码** X 、属性组 Y 及非主属性 Z ($Z \not\subseteq Y$), 使得 $X \rightarrow Y$, $Y \not\rightarrow X$, $Y \rightarrow Z$, 成立, 则称 $R<U, F> \in 3NF$ 。

例: $SL(Sno, Sdept, Sloc) \notin 3NF$
 $SD(Sno, Sdept) \in 3NF$
 $DL(Sdept, Sloc) \in 3NF$



- ❖ 若 $R \in 3NF$ ，则R的每一个非主属性既不部分函数依赖于候选码也不传递函数依赖于候选码。
- ❖ 如果 $R \in 3NF$ ，则R也是2NF。
- ❖ 采用投影分解法将一个2NF的关系分解为多个3NF的关系，可以在一定程度上解决原2NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
- ❖ 将一个2NF关系分解为多个3NF的关系后，并不能完全消除关系模式中的各种异常情况和数据冗余。

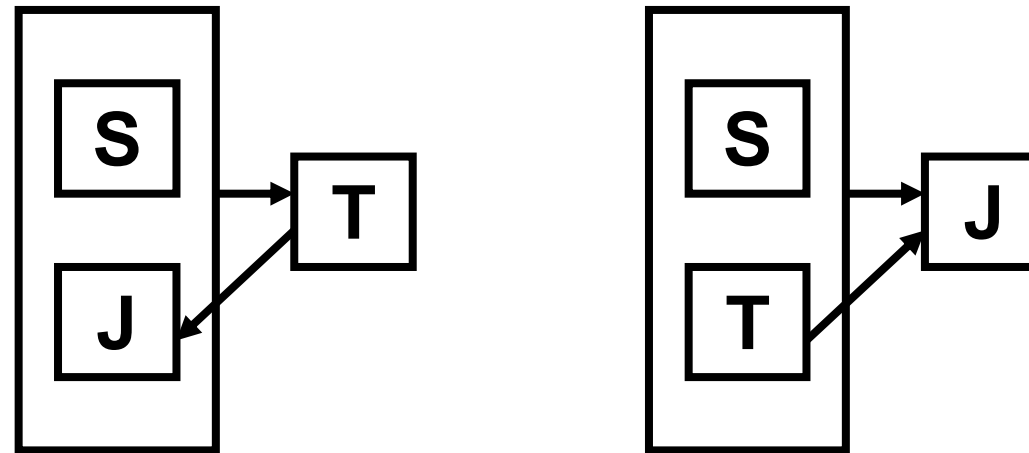


例：在关系模式STJ (S, T, J) 中，S表示学生，T表示教师，J表示课程。

❖ 函数依赖：

假设每一教师只教一门课。每门课由若干教师教，但某一学生选定某门课，就确定了一个固定的教师。某个学生选修某个教师的课就确定了所选课的名称。于是有：

$$(S, J) \rightarrow T, (S, T) \rightarrow J, T \rightarrow J$$



STJ



- ❖ **(S, J)和(S, T)都可以作为候选码。**
- ❖ **$STJ \in 3NF$**
- ❖ **$T \rightarrow J$, 即T是决定属性集, 可是T只是主属性, 它既不是候选码, 也不包含候选码。**



存在的问题：

(1) 插入异常

如果某个教师开设了某门课程，但尚未有学生选修，则有关信息也无法存入数据库中。

(2) 删除异常

如果选修过某门课程的学生全部毕业了，在删除这些学生元组的同时，相应教师开设该门课程的信息也同时丢掉了。



(3) 数据冗余度大

虽然一个教师只教一门课，但每个选修该教师该门课程的学生元组都要记录这一信息。

(4) 修改复杂

某个教师开设的某门课程改名后，所有选修了该教师该门课程的学生元组都要进行相应修改。

因此虽然 $STJ \in 3NF$ ，但它仍不是一个理想的关系模式。



原因：

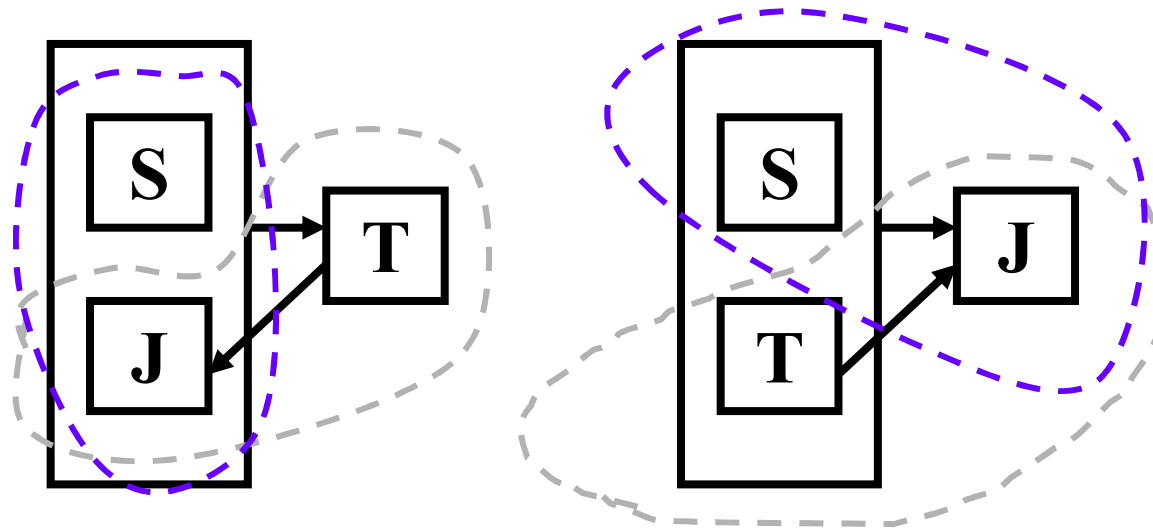
主属性J依赖于T，即**主属性J部分依赖于码**(S, T)。

解决方法：

采用投影分解法，将STJ分解为二个关系模式：

ST(S, T)

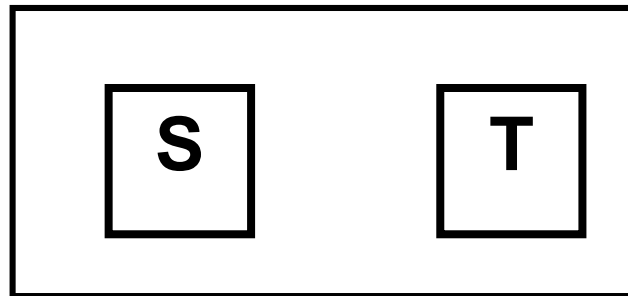
TJ(T, J)



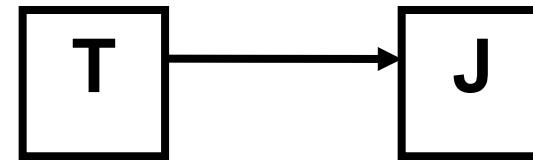
STJ



ST的码为 (**S**, **T**) , **TJ**的码为**T**。



ST



TJ



在分解后的关系模式中没有任何属性对码的部分函数依赖和传递函数依赖。它解决了上述四个问题：

- (1) TJ关系中可以存储所开课程尚未有学生选修的教师信息。**
- (2) 选修过某门课程的学生全部毕业了，只是删除ST关系中的相应元组，不会影响TJ关系中相应教师开设该门课程的信息。**
- (3) 关于每个教师开设课程的信息只在TJ关系中存储一次。**
- (4) 某个教师开设的某门课程改名后，只需修改TJ关系中的一个相应元组即可。**



六 BC范式 (BCNF)

定义6.10 设关系模式 $R\langle U, F \rangle \in 1NF$, 如果对于R的每个函数依赖 $X \rightarrow Y$, 若 $Y \subsetneq X$, 则**X必含有码**, 那么 $R \in BCNF$ 。

等价于: **每一个决定属性集都包含码**

[例]:
STJ (S, T, J) $\in 3NF$
ST (S, T) $\in BCNF$
TJ (T, J) $\in BCNF$



BCNF的关系模式所具有的性质:

- 1 所有非主属性都完全函数依赖于每个候选码。**
- 2 所有主属性都完全函数依赖于每个不包含它的候选码。**
- 3 没有任何属性完全函数依赖于非码的任何一组属性。**



[例] 关系模式C (Cno, Cname, Pcno)

- **$C \in 3NF$**
- **$C \in BCNF$**

[例] 关系模式S (Sno, Sname, Sdept, Sage)

- **假定S有两个码Sno, Sname**
- **$S \in 3NF$ 。**
- **$S \in BCNF$**



[例] 关系模式SJP (S, J, P) , S表示学生, J表示课程, P表示名次。每一个学生选修每门课程的成绩有一定的名次, 每门课程中每一名次只有一个学生 (即没有并列名次) 。

- **函数依赖: $(S, J) \rightarrow P; (J, P) \rightarrow S$**
- **(S, J) 与 (J, P) 都可以作为候选码, 属性相交**
- **$SJP \in 3NF,$**
- **$SJP \in BCNF$**

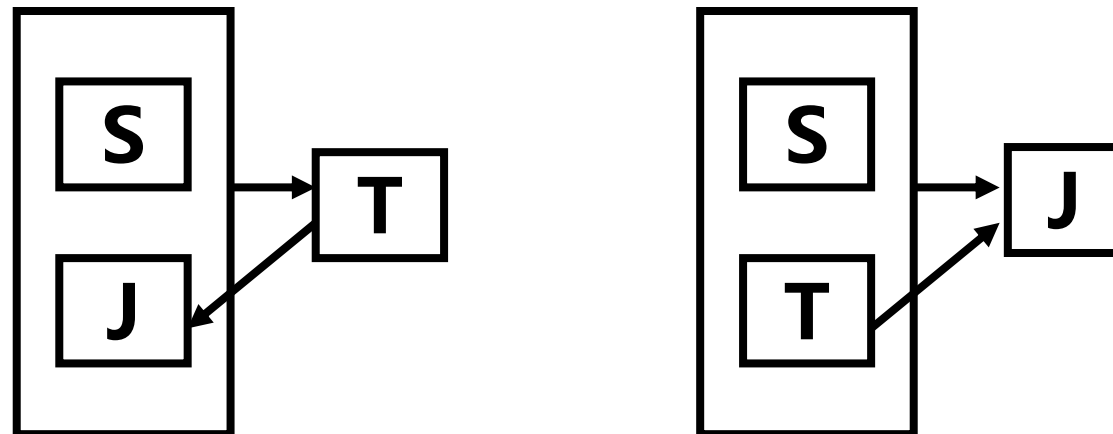


[例] 在关系模式STJ (S, T, J) 中, S表示学生, T表示教师, J表示课程。

- **函数依赖:**

$(S, J) \rightarrow T, (S, T) \rightarrow J, T \rightarrow J$

- **(S, J) 和 (S, T) 都是候选码**



STJ中的函数依赖



❖ **$STJ \in 3NF$**

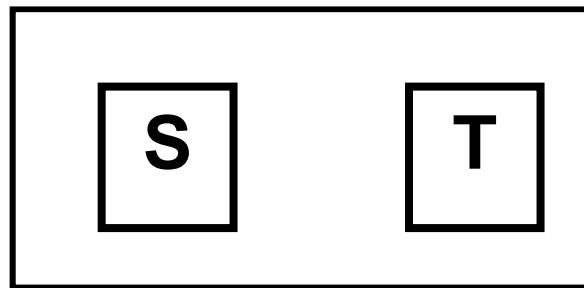
- 没有任何非主属性对码传递依赖或部分依赖

❖ **$STJ \notin BCNF$**

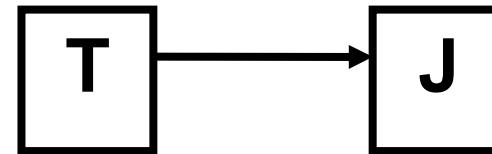
- T是决定因素，T不包含码



❖ 解决方法：将STJ分解为二个关系模式：
 $ST(S, T) \in \text{BCNF}$, $TJ(T, J) \in \text{BCNF}$



ST



TJ

没有任何属性对码的部分函数依赖和传递函数依赖



3NF与BCNF的关系

❖ $R \in \text{BCNF} \xrightleftharpoons[\text{不必要}]{\text{充分}} R \in \text{3NF}$

❖ 如果 $R \in \text{3NF}$, 且 R 只有一个候选码

$R \in \text{BCNF} \xrightleftharpoons[\text{必要}]{\text{充分}} R \in \text{3NF}$



- ❖ 采用投影分解法将一个3NF的关系分解为多个BCNF的关系，可以**进一步解决**原3NF关系中存在的插入异常、删除异常、数据冗余度大、修改复杂等问题。
- ❖ **3NF和BCNF是以函数依赖为基础**的关系模式规范化程度的测度。



练习

(1) 假设有关系模式仓库保管WPE

WPE(W# , P#, E#, QNT)

其中各属性分别表示:

仓库号W#、零件号P#、职工号E#、零件数量QNT

它是3NF吗?



它所反映的语义是：

- ①一个仓库有多个职工
- ②一个职工仅在一个仓库工作
- ③每个仓库里一种零件由专人负责

但一个人可以管理几种零件

W#	p#	E#	QNT
W1	P1	E1	20
W1	P2	E3	15
W1	P3	E2	18
W1	P4	E1	10
W1	P5	E3	20
W1	P7	E3	18
W1	P6	E2	15
W2	P4	E4	12
W2	P1	E5	20
W2	P3	E4	18
W2	P6	E5	15
W3	P2	E6	20
W3	P3	E6	15
W3	P4	E7	18



根据语义可以分析出下面函数依赖

$(W\#, P\#) \rightarrow QNT$ $(W\#, P\#) \rightarrow E\#$

$E\# \rightarrow W\#$ $(E\#, P\#) \rightarrow QNT$

因此这个关系模式有两个候选关键字

$(W\#, P\#)$, $(E\#, P\#)$

是BCNF吗？

----不是

只有一个非主属性QNT，它对任何一个候选关键字都是完全依赖的，并且是直接函数依赖。故该关系模式属于3NF



(2) 存在异常吗?

注意: 两个候选关键字

(W#, P#), (E#, P#)

W#	p#	E#	QNT
W1	P1	E1	20
W1	P2	E3	15
W1	P3	E2	18
W1	P4	E1	10
W1	P5	E3	20
W1	P7	E3	18
W1	P6	E2	15
W2	P4	E4	12
W2	P1	E5	20
W2	P3	E4	18
W2	P6	E5	15
W3	P2	E6	20
W3	P3	E6	15
W3	P4	E7	18



会出现:

如果一个新职工被分配到仓库工作，但暂时处于实习阶段，没有独立承担任务，由于缺少关键字的一部分P#而无法插入到该关系中去。

原因是:主属性E#是W#的决定因素，但它本身不是关键字，只是组合关键字的一部分，这就造成主属性W#对另一个不包含它候选关键字 (E# , P#) 的部分依赖。



(3) 将上述的仓库保管WPE分解成BCNF范式

WPE(W# , P# , E# , QNT)

保管: EP (E# , P# , QNT)

码是 (E# , P#)

工作: EW (E# , W#)

码是E#



教书是一场.....

- ❖ 教书是一场初恋，
学生厌我千百遍，我待学生如初恋；
- ❖ 教书是一场暗恋，
我全心全意爱一群人，最终只感动了自己；
- ❖ 教书是一场单恋，
我费心爱的那一群人，却爱上了手机和游戏；
- ❖ 教书是一场苦恋，
期末了，曾经怕自己一个人考不好，现在却要担心一群人考不好；
- ❖ 教书是一场群恋，
学生们相恋成片，老师却在讲台上自恋；



一个模式中的关系模式如果都属于BCNF，
那么在函数依赖范畴内，它已实现了彻底的分
离，已消除了插入和删除的异常。

属于BCNF的关系模式是否就很完美了呢？



七、多值依赖与第四范式 (4NF)

例子

(一) 多值依赖

(二) 第四范式 (4NF)



例: 设学校中某一门课程由多个教师讲授, 他们使用相同的一套参考书。每个教员可以讲授多门课程, 每种参考书可以供多门课程使用。

用关系模式Teaching(C, T, B)来表示课程C、教师T和参考书B之间的关系。



Teaching(C# , T# , B#)

C#	T#	B#
C1	{T1,T2}	{B1,B2}
C2	{T1,T3}	{B3,B4}

非规范化关系

C#	T#	B#
C1	T1	B1
C1	T1	B2
C1	T2	B1
C1	T2	B2
C2	T1	B3
C2	T1	B4
C2	T3	B3
C2	T3	B4



非规范化关系

课 程 C	教 员 T	参 考 书 B
物理	{ 李 勇 王 军 }	{ 普通物理学 光学原理 物理习题集 }
数学	{ 李 勇 张 平 }	{ 数学分析 微分方程 高等代数 }
计算数学 ⋮	{ 张 平 周 峰 ⋮ }	{ 数学分析 ⋮ }



用二维表表示：表6.4 Teaching

课程C	教员T	参考书B
物理	李勇	普通物理学
物理	李勇	光学原理
物理	李勇	物理习题集
物理	王军	普通物理学
物理	王军	光学原理
物理	王军	物理习题集
数学	李勇	数学分析
数学	李勇	微分方程
数学	李勇	高等代数
数学	张平	数学分析
数学	张平	微分方程
数学	张平	高等代数
...



❖ **Teaching** \in **BCNF**

❖ **Teaching** 具有唯一候选码(**C, T, B**), 即全码。



但它仍有不良特性:

数据冗余: 同一门课的教员与参考书信息被反复存储多次

更新异常: 当一门课程的教员或参考书作出改变时, 需要修改多个元组



插入异常：当某门课程增加一名教员时，该门课程有多少本参考书就必须插入多少个元组；同样当某门课程需要增加一本参考书时，它有多少个教员就必须插入多少个元组。

删除异常：当删除一门课程的某个教员或者某本参考书时，需要删除多个元组。

原因？？
存在多值
依赖



课程C#(X) 教员T#(Y) 参考书B#(Z)

对于一个
(物理, 普通物理学)

有一组T#值{李勇, 王军}

这组值仅与课程“物理”
有关, 而与参考书无关

物理	李	勇	普通物理学
物理	李	勇	光学原理
物理	李	勇	物理习题集
物理	王	军	普通物理学
物理	王	军	光学原理
物理	王	军	物理习题集
数学	李	勇	数学分析
数学	李	勇	微分方程
数学	李	勇	高等代数
数学	林	静	数学分析
数学	林	静	微分方程
数学	林	静	高等代数



规范化小结

(1) 规范化的目的

在关系数据库中，对关系模式的基本要求是满足第一范式，但有些关系模式存在插入删除异常，**规范化的目的就是解决插入、修改、删除异常以及数据冗余**



规范化小结

(2) 规范化的方法（思想）

逐步消除数据依赖中不合适的部分，使模式中的各关系模式达到某种程度的“分离”，即“**一事一地**”的模式设计原则，尽量作到每个模式表示客观世界中的一个事物



规范化小结

(3) 规范化的方法（过程）

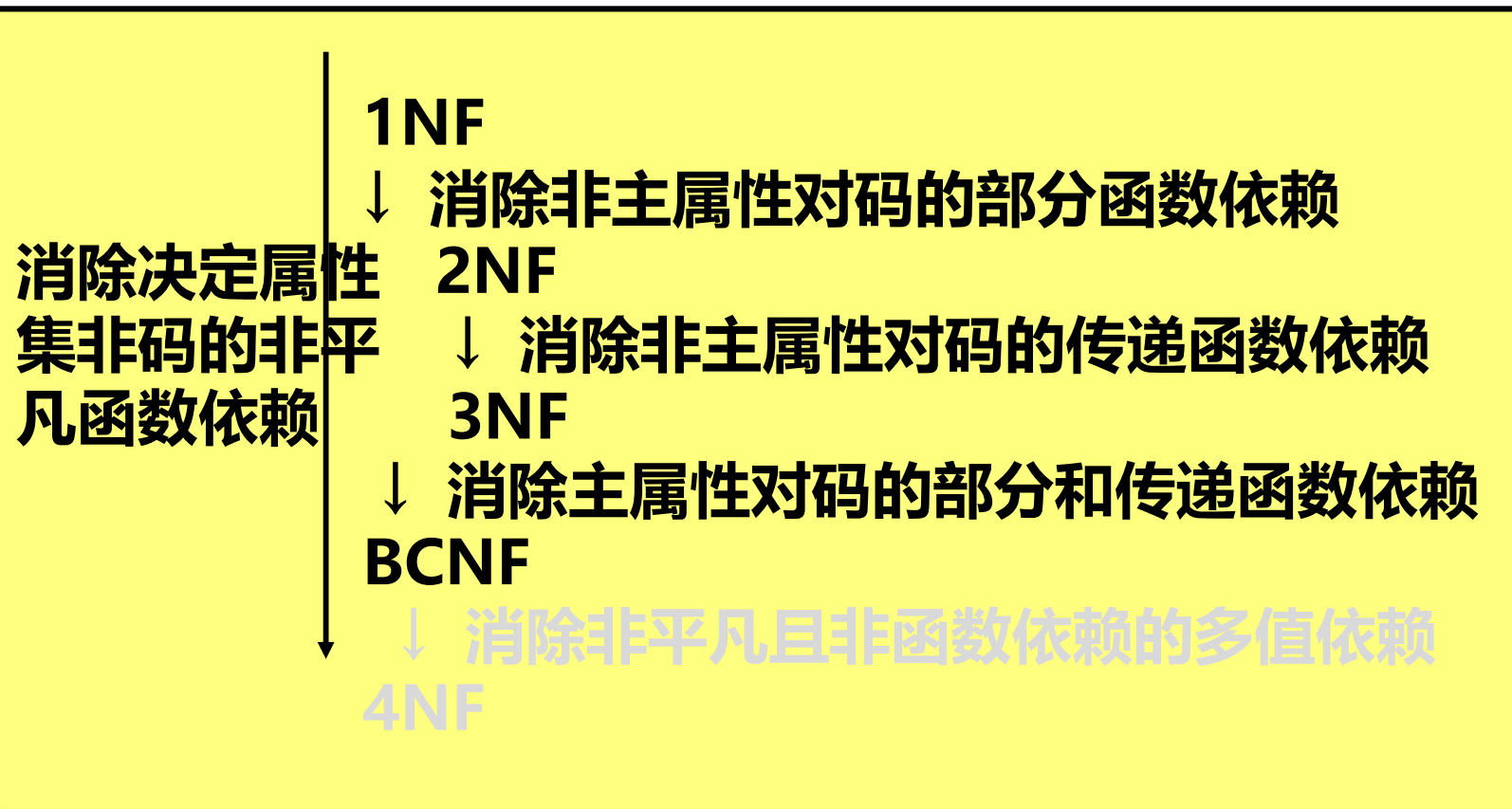
规范化的方法（过程）是用模式分解的方法。

通过对关系模式的分解，把低一级的关系分解成若干个高一级的关系。



规范化小结

❖ 关系模式规范化的基本步骤





规范化小结

要求

- (1) 了解：什么是“不好”的数据库模式；什么是插入异常和删除异常；规范化理论的重要意义
- (2) **牢固掌握**：函数依赖的有关定义（函数依赖，平凡函数依赖，非平凡函数依赖，部分函数依赖，完全函数依赖，传递函数依赖，码，候选码，外码，多值依赖），范式，从1NF到BCNF的定义和举例；规范化的含义和作用



规范化小结

**(3) 举一反三：四个范式的理解和应用，各个级别
范式中存在的问题和解决办法；根据应用语义，
写出关系模式中的数据依赖集合，会进行模式分
解，会判断几范式**



主要内容

6.1 问题的提出

6.2 规范化 (重点)

6.3 数据依赖的公理系统

6.4 *模式的分解 (难)





引例： $R(X, Y, Z) \quad F = \{Y \rightarrow Z, XZ \rightarrow Y\}$

是第几范式？

R的候选关键字为**XY(??)**和XZ，R中所有属性都为主属性，不存在非主属性对候选关键字的传递依赖，但存在主属性Z对关键字XY的部分依赖（也可看成传递依赖（ $XY \rightarrow Y, Y \rightarrow Z$ ）或者说决定因素Y被关键字包含或决定因素不是关键字
所以 仅为3NF



如何判定关系的范式?

--找出所有函数依赖, 确定码

如何判定关系的码?

问题

给定一组函数依赖, 是否能导出另外一些函数依赖,
或另外的函数依赖是否成立。



如:

$R (X, Y, Z)$

$F = \{Y \rightarrow Z, XZ \rightarrow Y\}$ 是否隐含

$XY \rightarrow Z$

再如 $FD = \{A \rightarrow B, B \rightarrow C\}$, $A \rightarrow C$ 是否成立?



本节主要内容

- 逻辑蕴涵
- Armstrong公理系统
- 闭包的计算(重点)
- 函数依赖的等价和覆盖(极小覆盖计算)



逻辑蕴涵

定义6.13

关系模式R，F是其函数依赖集，X，Y是其属性子集，如果从F函数依赖集中能够推导出 $X \rightarrow Y$ ，则称F**逻辑蕴涵** $X \rightarrow Y$ ，记作 $F \vdash X \rightarrow Y$ 。

如: $R(X, Y) \quad F = \{X \rightarrow Y\}$
能推出: $X \rightarrow X, X \rightarrow XY$



定义6.14

被F所逻辑蕴涵的函数依赖的全体所构成的集合

称作F的 **闭包**，记作 $F^+ = \{X \rightarrow Y \mid F \vdash X \rightarrow Y\}$

如： $R(X, Y), F = \{X \rightarrow Y\}$

$F^+ = \{X \rightarrow \varnothing, X \rightarrow X, X \rightarrow Y, X \rightarrow XY,$

$Y \rightarrow \varnothing, Y \rightarrow Y$

$XY \rightarrow \varnothing, XY \rightarrow X, XY \rightarrow Y, XY \rightarrow XY\}$



Armstrong公理系统

Armstrong公理

对于 $R \langle U, F \rangle$ ， X, Y, Z 是属性集， F 是函数依赖集。

- **自反律**(reflexivity): 若 $Y \subseteq X \subseteq U$ ，则 $X \rightarrow Y$ 为 F 所蕴含。
- **增广律**(augmentation): 若 $X \rightarrow Y$ 为 F 所蕴含，且 $Z \subseteq U$ ，则 $XZ \rightarrow YZ$ 为 F 所蕴含。
- **传递律**(transitivity): 若 $X \rightarrow Y$ ， $Y \rightarrow Z$ 为 F 所蕴含，则 $X \rightarrow Z$ 为 F 所蕴含。

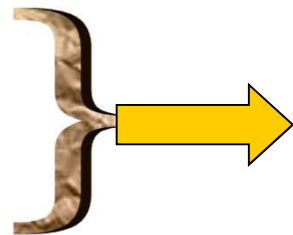


定理 6.1 Armstrong推理规则是正确的

按函数依赖定义 r 是 $R\langle U, F \rangle$ 上的任一关系, $t, s \in r$

自反律

$$\begin{array}{l} t[X] = s[X] \\ Y \subseteq X \end{array}$$



$$t[Y] = s[Y]$$



$$X \rightarrow Y$$

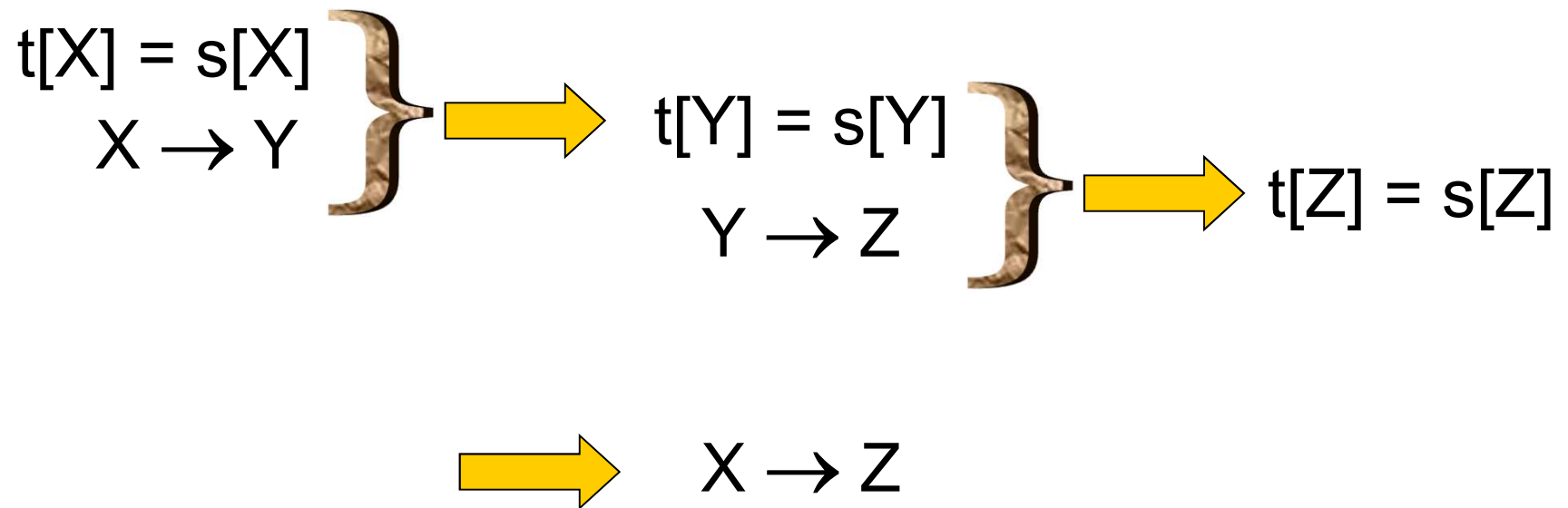


增广律

$$\begin{array}{l} t[XZ] = s[XZ] \xrightarrow{\quad} t[X] = s[X] \\ \quad \quad \quad X \rightarrow Y \end{array} \left. \vphantom{\begin{array}{l} t[XZ] = s[XZ] \xrightarrow{\quad} t[X] = s[X] \\ \quad \quad \quad X \rightarrow Y \end{array}} \right\} \xrightarrow{\quad} \left. \begin{array}{l} t[Y] = s[Y] \\ t[XZ] = s[XZ] \xrightarrow{\quad} t[Z] = s[Z] \end{array} \right\} \xrightarrow{\quad} t[YZ] = s[YZ] \xrightarrow{\quad} XZ \rightarrow YZ$$



传递律





Armstrong公理系统导出规则

❖ 由Armstrong公理导出的推理规则

- **合并律**(union rule)

- 若 $X \rightarrow Y$, $X \rightarrow Z$, 则 $X \rightarrow YZ$

- **分解律**(decomposition rule)

- 若 $X \rightarrow Y$, $Z \subseteq Y$, 则 $X \rightarrow Z$ 。
若 $X \rightarrow YZ$, 则 $X \rightarrow Y$, $X \rightarrow Z$

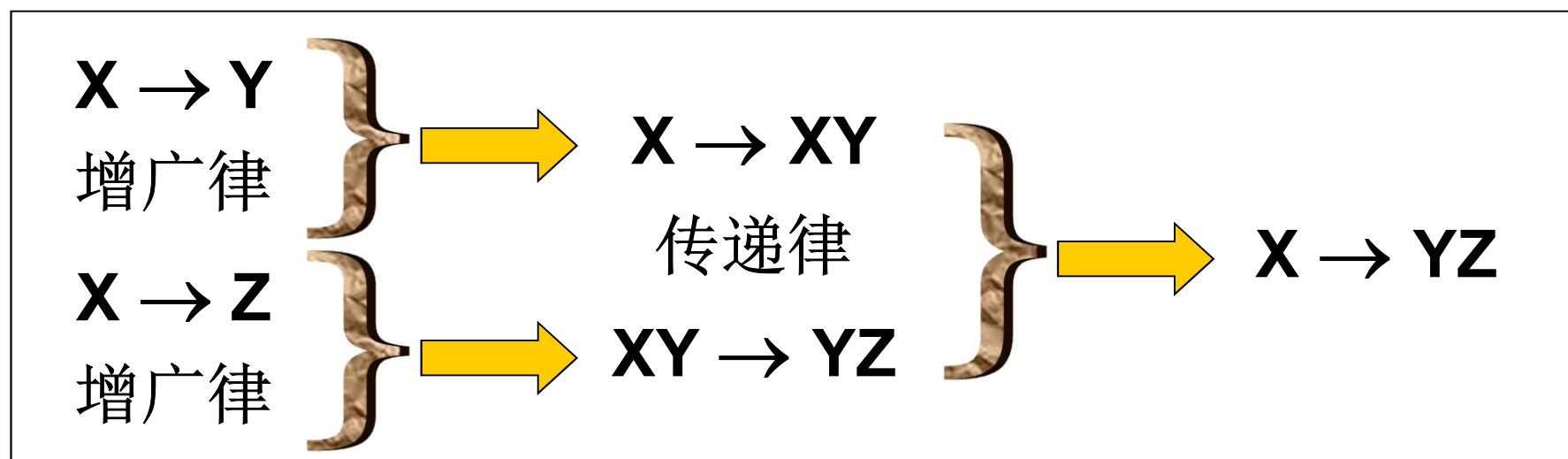
或者,

- **伪传递律**(pseudotransitivity rule)

- 若 $X \rightarrow Y$, $WY \rightarrow Z$, 则 $WX \rightarrow Z$

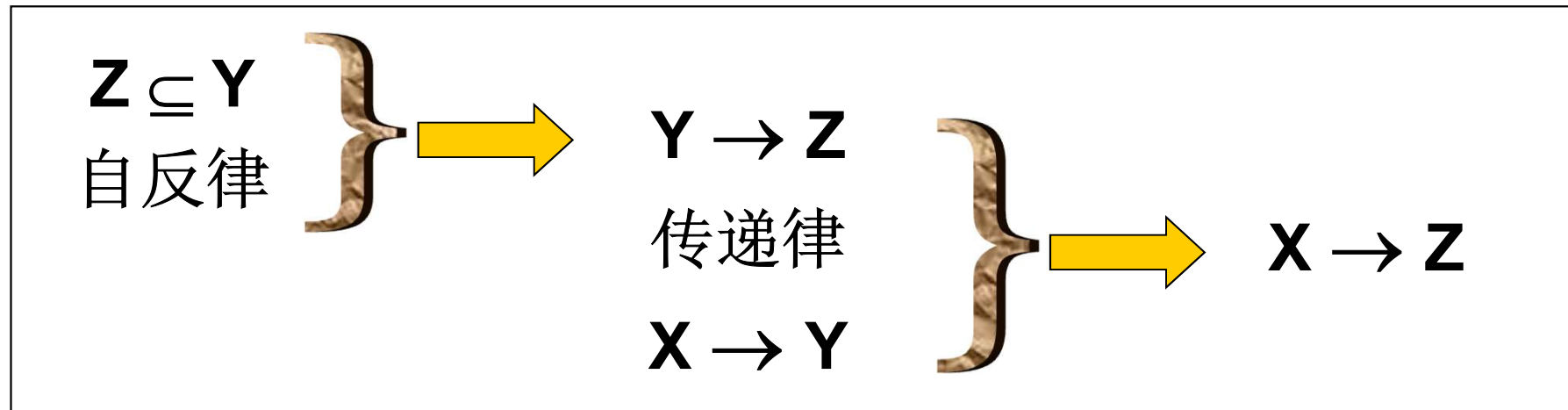


合并律





分解律





伪传递律





Armstrong公理系统的有效性与完备性

❖ Armstrong公理系统是有效的、完备的

- 有效性：由 F 出发根据Armstrong公理推导出来的每一个函数依赖一定在 F^+ 中； /* Armstrong正确
- 完备性： F^+ 中的每一个函数依赖，必定可以由 F 出发根据Armstrong公理推导出来。 /* Armstrong公理够用，完全



Armstrong公理系统的有效性与完备性

- ❖ 建立公理系统体系目的：从已知的 F 推导出未知的 F 。
- ❖ 明确：
 1. 公理系统推导出来的 f 正确？
 2. F^+ 中的每一个 f 都能推导出来？



Armstrong公理系统的有效性与完备性

Armstrong公理的完备性及有效性说明:

“蕴含” == “导出” 等价的概念

F^+ == 由 F 出发借助Armstrong公理导出的函数依赖的集合



Armstrong公理系统的有效性与完备性

❖ **定理6.2** Armstrong公理系统是有效的、完备的

❖ **证明:**

1. 有效性

可由定理6.1得证

2. 完备性

只需证明**逆否命题**: 若函数依赖 $X \twoheadrightarrow Y$ 不能由 F 从Armstrong公理导出, 那么它必然不为 F 所蕴含



Armstrong公理系统完备性证明

(1) 引理: 若 $V \rightarrow W$ 成立, 且 $V \subseteq X_F^+$, 则 $W \subseteq X_F^+$

(2) 构造一张二维表 r , 它由下列两个元组构成, 可以证明 r 必是 $R(U, F)$ 的一个关系, 即 F^+ 中的全部函数依赖在 r 上成立。

X_F^+	$U - X_F^+$
11.....1	00.....0
11.....1	11.....1

(3) 若 $X \rightarrow Y$ 不能由 F 从 Armstrong 公理导出, 则 Y 不是 X_F^+ 的子集。



示例

$R < U, F >, U = (A, B, C, G, H, I), F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$

• $A \rightarrow H?$

✓

传递律

• $CG \rightarrow HI?$

✓

合并律

• $AG \rightarrow I?$

✓

伪传递律



❖ **引理一：** $X \rightarrow A_1 A_2 \dots A_k$ 成立 $\Leftrightarrow X \rightarrow A_i$ 成立
($i=1, 2, \dots, k$)

证明： 由合并律 \Leftarrow
由分解律 \Rightarrow



函数依赖闭包

回顾：

被F所逻辑蕴涵的函数依赖的全体所构成的集合

称作F的闭包，记作 $F^+ = \{X \rightarrow Y \mid F \vdash X \rightarrow Y\}$ 。



$F = \{X \rightarrow Y, Y \rightarrow Z\}$

$F^+ = \{$

$X \rightarrow \varphi,$	$Y \rightarrow \varphi,$	$Z \rightarrow \varphi,$	$XY \rightarrow \varphi,$	$XZ \rightarrow \varphi,$	$YZ \rightarrow \varphi,$	$XYZ \rightarrow \varphi,$
$X \rightarrow X,$	$Y \rightarrow Y,$	$Z \rightarrow Z,$	$XY \rightarrow X,$	$XZ \rightarrow X,$	$YZ \rightarrow Y,$	$XYZ \rightarrow X,$
$X \rightarrow Y,$	$Y \rightarrow Z,$		$XY \rightarrow Y,$	$XZ \rightarrow Y,$	$YZ \rightarrow Z,$	$XYZ \rightarrow Y,$
$X \rightarrow Z,$	$Y \rightarrow YZ,$		$XY \rightarrow Z,$	$XZ \rightarrow Z,$	$YZ \rightarrow YZ,$	$XYZ \rightarrow Z,$
$X \rightarrow XY,$			$XY \rightarrow XY,$	$XZ \rightarrow XY,$		$XYZ \rightarrow XY,$
$X \rightarrow XZ,$			$XY \rightarrow YZ,$	$XZ \rightarrow XZ,$		$XYZ \rightarrow YZ,$
$X \rightarrow YZ,$			$XY \rightarrow XZ,$	$XZ \rightarrow XY,$		$XYZ \rightarrow XZ,$
$X \rightarrow ZYZ,$			$XY \rightarrow XYZ,$	$XZ \rightarrow XYZ,$		$XYZ \rightarrow XYZ \}$

$F = \{X \rightarrow A_1, \dots, X \rightarrow A_n\}$ 的闭包 F^+ 计算是一个复杂且困难的问题
(NP完全问题、指数级时间问题)



属性集的闭包

❖ 设 F 为属性集 U 上的一组函数依赖, $X \subseteq U$, X_F^+
= $\{ A \mid X \rightarrow A \text{ 能由 } F \text{ 根据 Armstrong 公理导出} \}$,
 X_F^+ 称为属性集 X 关于函数依赖集 F 的闭包。



关于闭包的引理

❖ **引理二** $X \rightarrow Y$ 能由 Armstrong 公理导出 $\Leftrightarrow Y \subseteq X_F^+$

必要性:

$Y \subseteq X_F^+ \longrightarrow$ 对 Y 中任意属性 A , 有 $X \rightarrow A$
 引理一 $\} \longrightarrow X \rightarrow Y$

充分性:

若 $Y \not\subseteq X_F^+ \longrightarrow Y$ 中有属性 $A \notin X_F^+ \longrightarrow$

$X \rightarrow A$ 不能由 F 导出
 引理一 $\} \longrightarrow X \rightarrow Y$ 也不能由 F 导出
 F 逻辑蕴涵 $X \rightarrow Y$ $\} \longrightarrow$ 矛盾



闭包的计算

问题：有没有一般性的算法判定 $X \rightarrow Y$ 是否能由 F 根据Armstrong公理导出？

如果计算出 F^+ ，再判断 $X \rightarrow Y$ 是否属于 F^+ ，则由于 F^+ 的计算非常复杂，实际上是不可行的。

由引理二，判定 $X \rightarrow Y$ 是否能由 F 根据Armstrong公理导出，可转化为求 X_{F^+} ，判定 $Y \subseteq X_{F^+}$ 是否成立。



闭包的计算

算法（求属性集的闭包）

Input: X, F

Output: X_F^+

开始: $X_F^+ := X;$

while (X_F^+ 发生变化) **do**

for each 函数依赖 $A \rightarrow B$ **in** F **do**

begin

if $A \subseteq X_F^+$ **then** $X_F^+ := X_F^+ \cup B$

end



示例1: $R < U, F >, U = (A, B, C, G, H, I), F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$, 计算 $(AG)_F^+$

所用依赖	$(AG)_F^+ = AG$
$A \rightarrow B$	AGB
$A \rightarrow C$	AGBC
$CG \rightarrow H$	AGBCH
$CG \rightarrow I$	AGBCH I

所以: $(AG)_F^+ = AGBCH I$



示例2: $R \langle U, F \rangle$, $U = (A, B, C, D, E)$, $F = \{AB \rightarrow C, B \rightarrow D, C \rightarrow E, CE \rightarrow B, AC \rightarrow B\}$, 计算 $(AB)_F^+$ 。

所用依赖	$(AB)_F^+$
$AB \rightarrow C$	ABC
$B \rightarrow D$	ABCD
$C \rightarrow E$	ABCDE
$(AB)_F^+ = ABCDE$	



示例3: $R < U, F >, U = (A, B, C, D, E, G), F = \{A \rightarrow E, BE \rightarrow AG, CE \rightarrow A, G \rightarrow D\}$, 计算 $(AB)_F^+$ 。

所用依赖	$(AB)_F^+$
$A \rightarrow E$	ABE
$BE \rightarrow AG$	ABEG
$G \rightarrow D$	ABEGD
$(AB)_F^+ =$	ABEGD



练习： $R < U, F >, U = (C, T, H, R, S), F = \{C \rightarrow T, HR \rightarrow C, HT \rightarrow R, HS \rightarrow R\}$, HR是码吗？HS呢？

$$(HR)_F^+ = CTHR$$

$$(HS)_F^+ = CTHRS = U$$



假设某个关系模式R上有**F**，那么在该关系作**任何更新**时，数据库系统都要保证F中的所有依赖在新数据库状态下仍然满足，否则就取消这种更新操作。因此数据库系统要对更新进行检测，开销较大。

为了降低检测开销，在**不改变闭包**的情况下，可以简化给定的函数依赖集。



函数依赖集等价

定义6.15 如果 $G^+ = F^+$ ，就说函数依赖集 F **覆盖** G (F 是 G 的覆盖，或 G 是 F 的覆盖)，或 F 与 G **等价**。

引理三 $F^+ = G^+$ 的充分必要条件是 $F \subseteq G^+$ ，和 $G \subseteq F^+$
证：必要性显然，只证充分性。

(1) 若 $F \subseteq G^+$ ，则 $X_F^+ \subseteq X_{G^+}^+$ 。

(2) 任取 $X \rightarrow Y \in F^+$ 则有 $Y \subseteq X_F^+ \subseteq X_{G^+}^+$ 。

所以 $X \rightarrow Y \in (G^+)^+ = G^+$ 。即 $F^+ \subseteq G^+$ 。

(3) 同理可证 $G^+ \subseteq F^+$ ，所以 $F^+ = G^+$ 。



最小依赖集

定义6.16 如果函数依赖集F满足下列条件，则称F为一个极小函数依赖集。亦称为最小依赖集或最小覆盖。

- (1) 单属性化：F中任一函数依赖的右部仅含有一个属性。
- (2) 无冗余化：F中不存在这样的函数依赖 $X \rightarrow A$ ，使得F与 $F - \{X \rightarrow A\}$ 等价。
- (3) 既约化：F中不存在这样的函数依赖 $X \rightarrow A$ ，X有真子集Z使得 $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ 与F等价。



[例] 关系模式 $S<U, F>$, 其中:

$U=\{ Sno, Sdept, Mname, Cno, Grade \}$,

$F=\{ Sno \rightarrow Sdept, Sdept \rightarrow Mname, (Sno, Cno) \rightarrow Grade \}$

设 $F' = \{ Sno \rightarrow Sdept, Sno \rightarrow Mname, Sdept \rightarrow Mname, (Sno, Cno) \rightarrow Grade, (Sno, Sdept) \rightarrow Sdept \}$

F 是最小覆盖, 而 F' 不是。

因为: $F' - \{ Sno \rightarrow Mname \}$ 与 F' 等价

$F' - \{ (Sno, Sdept) \rightarrow Sdept \}$ 也与 F' 等价



极小化过程

定理6.3 每一个函数依赖集 F 均等价于一个极小函数依赖集 F_m 。此 F_m 称为 F 的最小依赖集。

证明: 构造性证明, 找出 F 的一个最小依赖集。



- (1) 逐一检查 F 中各函数依赖 $FD_i: X \rightarrow Y$, 若 $Y = A_1 A_2 \dots A_k$, $k > 2$,
则用 $\{ X \rightarrow A_j \mid j=1, 2, \dots, k \}$ 来取代 $X \rightarrow Y$ 。
- (2) 逐一检查 F 中各函数依赖 $FD_i: X \rightarrow A$, 令 $G = F - \{ X \rightarrow A \}$,
若 $A \in X_G^+$, 则从 F 中去掉此函数依赖。
- (3) 逐一取出 F 中各函数依赖 $FD_i: X \rightarrow A$, 设 $X = B_1 B_2 \dots B_m$,
逐一考查 B_i ($i=1, 2, \dots, m$), 若 $A \in (X - B_i)_F^+$,
则以 $X - B_i$ 取代 X 。



[例] $F = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, A \rightarrow C, C \rightarrow A\}$

F_{m1} 、 F_{m2} 都是 F 的最小依赖集:

$$F_{m1} = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$$

$$F_{m2} = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, C \rightarrow A\}$$

- ❖ **F 的最小依赖集 F_m 不唯一**
- ❖ **极小化过程(定理6.3的证明)也是检验 F 是否为极小依赖集的一个算法**



示例

$F = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, B \rightarrow C\}$, 求 F_m

- 检查 $A \rightarrow B$, 令 $G = F - \{A \rightarrow B\} = \{B \rightarrow A, A \rightarrow C, B \rightarrow C\}$

$(A)_G^+ = \{A, C\}$, 因 $B \notin \{A, C\}$

- 检查 $A \rightarrow C$, 令 $G = F - \{A \rightarrow C\} = \{A \rightarrow B, B \rightarrow A, B \rightarrow C\}$

$(A)_G^+ = \{A, B, C\}$, 因 $C \in \{A, B, C\}$

所以从 F 中删除 $A \rightarrow C$,

$F_m = \{A \rightarrow B, B \rightarrow A, B \rightarrow C\}$

或者

$F_m = \{A \rightarrow B, B \rightarrow A, A \rightarrow C\}$



示例（注意计算的顺序）

$F = \{C \rightarrow A, A \rightarrow G, CG \rightarrow B, B \rightarrow A\}$, 求 F_m

判断 $CG \rightarrow B$,

$$(CG - C)_F^+ = (G)_F^+ = \{G\}, B \notin (CG - C)_F^+$$

$$(CG - G)_F^+ = (C)_F^+ = \{C, A, G, B\}$$

$B \in (CG - G)_F^+$, 以 C 代替 CG

此时, $F = \{C \rightarrow A, A \rightarrow G, C \rightarrow B, B \rightarrow A\}$

检查无冗余性, $C \rightarrow A$ 是冗余的, 从 F 中删除。

最后, $F_m = \{A \rightarrow G, C \rightarrow B, B \rightarrow A\}$



习 题

❖ 设关系模式R (ABCDE) , F是R上成立的FD集
 , $F = \{AB \rightarrow C, CD \rightarrow E, DE \rightarrow B\}$, 试判断AB是
R的候选键吗? ABD呢? 请作出解释。



❖ 设关系模式R (ABCD) , F是R上成立的FD集,

$$F = \{ A \rightarrow B, B \rightarrow C \}$$

① 试写出属性集BD的闭包 $(BD)^+$ 。

② 试写出所有左部是B的函数依赖 (即形为 “ $B \rightarrow ?$ ”) 。



习题：关系模式 **$R(U, F)$** 中， **$U=ABCDE$** ， **$F=\{A \rightarrow B, BC \rightarrow E, ED \rightarrow AB\}$**

求：等价于 **F** 的最小值函数依赖集 **F_{\min}**

答案： **$F_{\min}=\{A \rightarrow B, BC \rightarrow E, ED \rightarrow A\}$**



主要内容

6.1 问题的提出

6.2 规范化 (重点)

6.3 数据依赖的公理系统

6.4 *模式的分解 (难)





❖ 关系模式的规范化过程是通过对关系模式的分解来实现的

- 把低一级的关系模式分解为若干个高一级的关系模式的方法并不是唯一的
- 在这些分解方法中，只有能够**保证分解后的关系模式与原关系模式等价**的方法才有意义



关系模式的分解（续）

- ❖ 将一个关系模式 $R\langle U, F \rangle$ 分解为若干个关系模式 $R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle$ （其中 $U = U_1 \cup U_2 \cup \dots \cup U_n$ ，且不存在 $U_i \subseteq U_j$ ， F_i 为 F 在 U_i 上的投影），意味着相应地将存储在一个二维表 t 中的数据分散到若干个二维表 t_1, t_2, \dots, t_n 中去（其中 t_i 是 t 在属性集 U_i 上的投影）。



例：对于关系模式SL (Sno, Sdept, Sloc) , 同一个系的学生住在一起, SL中有下列函数依赖：

Sno→Sdept

Sdept→Sloc

Sno→Sloc

已知 $SL \in 2NF$, 该关系模式存在插入异常、删除异常、数据冗余度大和修改复杂的问题。

**因此需要分解该关系模式, 使成为更高范式的关系模式。
分解方法可以有很多种。**



假设下面是该关系模式的一个关系：

SL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	IS	B
95005	PH	B



■ 第一种分解方法

将SL分解为下面三个关系模式:

SN(Sno)

SD(Sdept)

SO(Sloc)



分解后的关系为:

SN	<u>Sno</u>	SD	<u>Sdept</u>	SO	<u>Sloc</u>
	95001		CS		A
	95002		IS		B
	95003		MA		C
	95004		PH		
	95005				



SN、SD和SO都是规范化程度很高的关系模式(5NF)。但分解后的数据库丢失了许多信息，例如无法查询95001学生所在系或所在宿舍。因此这种分解方法是不可取的。

如果分解后的关系可以通过自然连接恢复为原来的关系，那么这种分解就没有丢失信息。



■ 第二种分解方法

将SL分解为下面二个关系模式：

NL(Sno, Sloc)

DL(Sdept, Sloc)

分解后的关系为：

NL

Sno	Sloc
95001	A
95002	B
95003	C
95004	B
95005	B

DL

Sdept	Sloc
CS	A
IS	B
MA	C
PH	B



对NL和DL关系进行自然连接的结果为：

NL \bowtie DL

Sno	Sloc	Sdept
95001	A	CS
95002	B	IS
95002	B	PH
95003	C	MA
95004	B	IS
95004	B	PH
95005	B	IS
95005	B	PH



**NL \bowtie DL比原来的SL关系多了两个元组
(95002, B, PH)和(95005, B, IS)。因此我们也无
法知道原来的SL关系中究竟有哪些元组，**从这个意
义上说，此分解方法仍然丢失了信息。****



■ 第三种分解方法

将SL分解为下面二个关系模式:

ND(Sno, Sdept)

NL(Sno, Sloc)

分解后的关系为:

ND

<hr/>	
Sno	Sdept
<hr/>	
95001	CS
95002	IS
95003	MA
95004	IS
95005	PH
<hr/>	

NL

<hr/>	
Sno	Sloc
<hr/>	
95001	A
95002	B
95003	C
95004	B
95005	B
<hr/>	



对ND和NL关系进行自然连接的结果为：

ND \bowtie NL

Sno	Sdept	Sloc
95001	CS	A
95002	IS	B
95003	MA	C
95004	CS	A
95005	PH	B

它与SL关系完全一样，因此第三种分解方法没有丢失信息。



❖ 具有无损连接性的模式分解

- 设关系模式 $R\langle U, F \rangle$ 被分解为若干个关系模式 $R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle$ (其中 $U = U_1 \cup U_2 \cup \dots \cup U_n$, 且不存在 $U_i \subseteq U_j$, F_i 为 F 在 U_i 上的投影), **若 R 与 R_1, R_2, \dots, R_n 自然连接的结果相等, 则称关系模式 R 的这个分解具有无损连接性 (Lossless join)。**
- 只有具有无损连接性的分解才能够保证不丢失信息。
- **无损连接性不一定**能解决插入异常、删除异常、修改复杂、数据冗余等问题



❖ 第三种分解方法

- 具有无损连接性，保证了不丢失原关系中的信息
- 但它并没有解决插入异常、删除异常、修改复杂、数据冗余等问题。

例如：95001学生由CS系转到IS系，ND关系的(95001, CS)元组和NL关系的(95001, A)元组必须同时进行修改，否则会破坏数据库的一致性。

- **原因：分解得到的两个关系模式不是互相独立的。SL中的函数依赖 $Sdept \rightarrow Sloc$ 既没有投影到关系模式ND上，也没有投影到关系模式NL上，而是跨在这两个关系模式上。也就是这种分解方法没有保持原关系中的函数依赖。**



保持函数依赖的模式分解

- 设关系模式 $R\langle U, F \rangle$ 被分解为若干个关系模式 $R_1\langle U_1, F_1 \rangle, R_2\langle U_2, F_2 \rangle, \dots, R_n\langle U_n, F_n \rangle$ (其中 $U = U_1 \cup U_2 \cup \dots \cup U_n$, 且不存在 $U_i \subseteq U_j$, F_i 为 F 在 U_i 上的投影), 若 F 所逻辑蕴含的函数依赖一定也由分解得到的某个关系模式中的函数依赖 F_i 所逻辑蕴含, 则称关系模式 R 的这个**分解是保持函数依赖的** (Preserve dependency)。



第四种分解方法

将SL分解为下面二个关系模式：

ND(Sno, Sdept)

DL(Sdept, Sloc)

这种分解方法就保持了函数依赖。



判断对关系模式的一个分解是否与原关系模式等价的标准：

- 1 分解具有无损连接性**
- 2 分解要保持函数依赖**
- 3 分解既要保持函数依赖，又要具有无损连接性**



- ❖ 如果一个分解具有无损连接性，则它能够保证不丢失信息。
- ❖ 如果一个分解保持了函数依赖，则它可以减轻或解决各种异常情况。
- ❖ 分解具有无损连接性和分解保持函数依赖是两个**互相独立的**标准。具有无损连接性的分解不一定能够保持函数依赖。同样，保持函数依赖的分解也不一定具有无损连接性。



例：上例中

第一种分解方法既不具有无损连接性，也未保持函数依赖，它不是原关系模式的一个等价分解。

第二种分解方法未保持了函数依赖，也不具有无损连接性。

第三种分解方法具有无损连接性，但未持函数依赖。

第四种分解方法既具有无损连接性，又保持了函数依赖。



规范化理论提供了一套完整的模式分解算法，可将关系模式R分别转换为：

- ✓ **3NF保持函数依赖的分解**
- ✓ **3NF的既有无损连接性又保持函数依赖的分解**
- ✓ **BCNF的无损连接分解**
- ✓ **达到4NF的具有无损连接性的分解**



小 结

- ❖ 函数依赖
- ❖ 多值依赖
- ❖ 关系模式规范化的基本步骤
- ❖ Armstrong公理系统

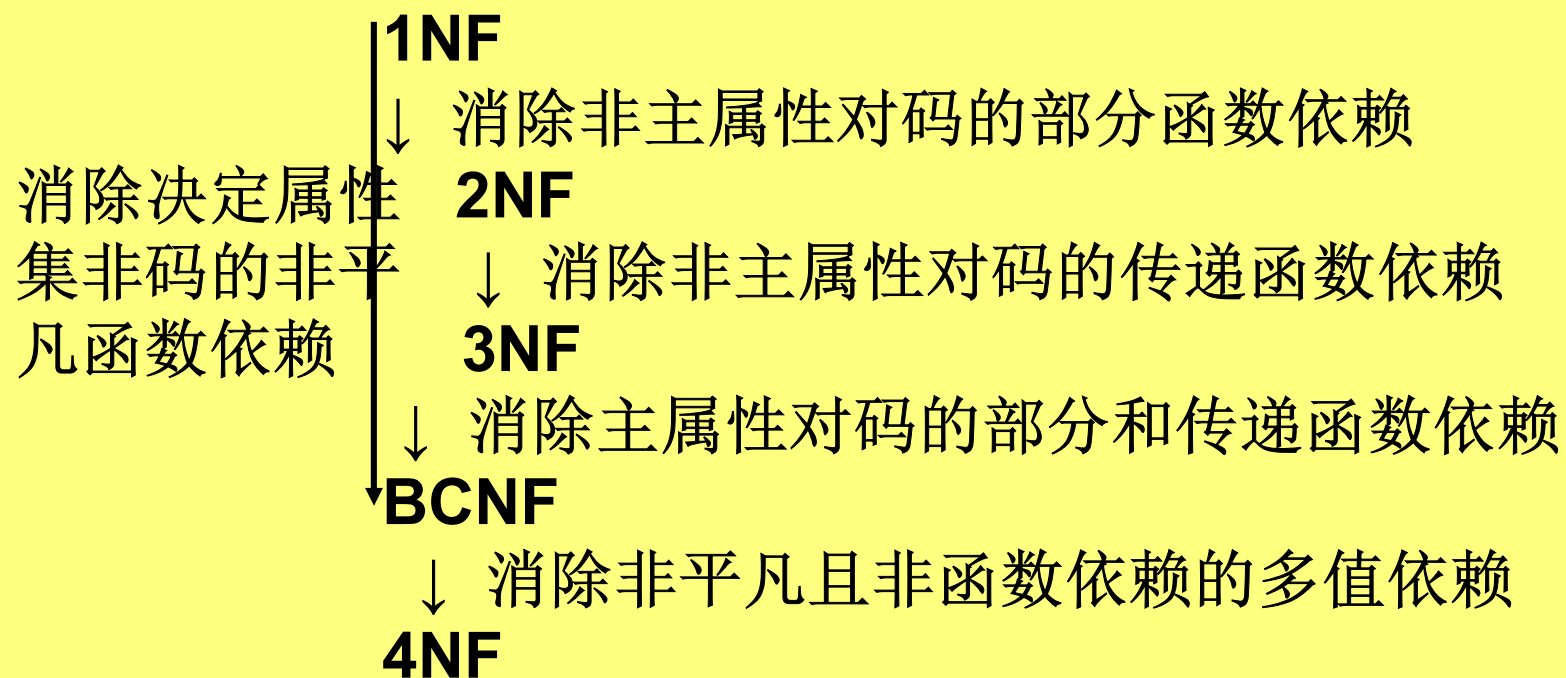


一、函数依赖

- 函数依赖
- 平凡函数依赖与非平凡函数依赖
- 完全函数依赖与部分函数依赖
- 传递函数依赖
- 码



❖ 关系模式规范化的基本步骤





- ❖ 若要求分解具有无损连接性，那么模式分解一定能够达到4NF
- ❖ 若要求分解保持函数依赖，那么模式分解一定能够达到3NF，但不一定能够达到BCNF
- ❖ 若要求分解既具有无损连接性，又保持函数依赖，则模式分解一定能够达到3NF，但不一定能够达到BCNF



- ❖ **规范化理论为数据库设计提供了理论的指南和工具**
 - 也仅仅是指南和工具
- ❖ **并不是规范化程度越高，模式就越好**
 - 必须结合应用环境和现实世界的具体情况合理地选择数据库模式



❖ 关系模式规范化理论

- 语义层
 - 函数依赖
- 规范层
 - 范式
- 实现层
 - 模式分解



- ❖ **Armstrong公理**
- ❖ **属性集闭包**
- ❖ **极小依赖集**



习题

1. 规范化理论是关系数据库进行逻辑设计的理论依据，根据这个理论，关系数据库中的关系必须满足：每一个属性都是（ ）。
A. 长度不变的
☒ B. 不可分解的
C. 互相关联的
D. 互不相关的
2. 已知关系模式R (A, B, C, D, E) 及其上的函数依赖集合 $F = \{A \rightarrow D, B \rightarrow C, E \rightarrow A\}$ ，该关系模式的候选码是（ ）。
A. AB
☒ B. BE
C. CD
D. DE
3. 关系模式中，满足2NF的模式（ ）。
A. 可能是1NF
☒ B. 必定是1NF
C. 必定是3NF
D. 必定是BCNF



4. 关系模式R中的属性全是主属性，则R的最高范式必定是（ ）。
- A.1NF
 - B.2NF
 - C.BCNF
 - ☒ D.3NF
5. 消除了部分函数依赖的1NF的关系模式，必定是（ ）。
- A.1NF
 - ☒ B.2NF
 - C.3NF
 - D.BCNF
6. 关系模式的候选码可以有1个或多个，而主码有（ ）。
- A.多个
 - B.0个
 - ☒ C.1个
 - D.1个或多个
7. 候选码的属性可以有（ ）。
- A.多个
 - B.0个
 - C.1个
 - ☒ D.1个或多个



8. 设有关系模式R (S, D, M) , 其函数依赖集: $F = \{S \rightarrow D, D \rightarrow M\}$, 则关系模式R的规范化程度最高达到 () 。
- A. 1NF
B. 2NF
C. 3NF
D. BCNF
9. 在关系模式R (A, B, C, D) 中, 存在函数依赖关系 $\{A \rightarrow B, A \rightarrow C, A \rightarrow D, (B, C) \rightarrow A\}$, 则候选码是 A, (B, C), 关系模式R (A, B, C, D) 属于 3NF 。
10. 在关系模式R (D, E, G) 中, 存在函数依赖关系 $\{E \rightarrow D, (D, G) \rightarrow E\}$, 则候选码是 (E, G), (D, G), 关系模式R (D, E, G) 属于 3NF 。
11. 在关系模式R (A, C, D) 中, 存在函数依赖关系 $\{A \rightarrow C, A \rightarrow D\}$, 则候选码是 A, 关系模式R (A, C, D) 最高可以达到 BCNF 。



A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

■ 检验： $A \rightarrow C, C \rightarrow A, AB \rightarrow D$ 在此关系上是否成立。



判断:

- (1) 平凡函数依赖必定是部分函数依赖 ()
- (2) 非平凡函数依赖也可能是部分函数依赖()

[例]: Student(Sno, Sname, Ssex, Sage, Sdept)
Sno \xrightarrow{f} Sname, Sno \xrightarrow{f} Ssex, Sno \xrightarrow{f} Sage,
Sno \xrightarrow{f} Sdept
(Sno, Sname) \xrightarrow{p} Sdept, (Sno, Ssex) \xrightarrow{p} Sdept



证明若 $R \in \text{BCNF}$, 则 $R \in 3\text{NF}$ 。

证明：假设 $R \in \text{BCNF}$, 但 $R \notin 3\text{NF}$

因为 $R \notin 3\text{NF}$, 所以 R 中必有一非主属性 A_i 传递依赖于码 X , 即:

$X \rightarrow Y$ (Y 不决定 X) , $Y \rightarrow A_i$

因为 $R \in \text{BCNF}$, 所以 Y 含有候选码

$Y \rightarrow U$, $Y \rightarrow X$, 这与假设相矛盾,

所以 $R \in \text{BCNF}$, 则 $R \in 3\text{NF}$ 。



作业

已知：关系模式R (U, F) 中, $U=ABCDE$, $F=\{A \rightarrow B, BC \rightarrow E, ED \rightarrow AB\}$ →

求：

- (1)、计算 A_F^+ 、 $(AB)_F^+$ 、 $(ABC)_F^+$ 、 $(BCD)_F^+$;
- (2)、等价于F的最小值函数依赖集Fmin ;
- (3)、求R的所有候选码, 并说明理由;
- (4)、R至少满足第几范式? 为什么?



❖ 练习：设有一关系模式 $R(A, B, C, D, E, F)$, 其函数依赖为 $\{A \rightarrow C, (A, B) \rightarrow D, C \rightarrow E, D \rightarrow (B, F)\}$, 请按下述要求进行模式分解，并给出分解后的结果关系模式上存在的函数依赖。

- (1) 给出关系模式 R 的候选键
- (2) 分解 R ，使之满足2NF
- (3) 将上题的结果分解，使之满足3NF



设有关系模式 R (**运动员编号, 比赛项目, 成绩, 比赛类别, 比赛主管**) 存储运动员比赛成绩及比赛类别、主管等信息。

如果规定：**每个运动员每参加一个比赛项目, 只有一个成绩; 每个比赛项目只属于一个比赛类别; 每个比赛类别只有一个比赛主管。**

试回答下列问题：

- (1) 写出模式R的基本函数依赖和所有码;
- (2) 说明R不是2NF的理由, 并把R分解成2NF;
- (3) 进而分解成3NF模式集。