

# 关系数据库设计 规范化理论





# 同一应用的不同设计方案

---

- 设计方案1:

- `movies(title,year,length,movietype,studioaddr,starname)`

- 设计方案2:

- `movies1(title,year,lenth,movietype,studioaddr)`
- `movies2(title,year,starname)`

- 设计方案3:

- `movies11(title,year,length,movietype,studioaddr)`
- `movies12(studioaddr)`
- `movies2(title,year,starname)`



# 问题的提出

---

- 针对一个具体应用，应该如何构造一个适合于它的关系数据库模式？
- 这是数据库的逻辑设计问题
- 关系数据库的**规范化理论**是数据库逻辑设计的理论基础



# 关系模式的设计问题

---

- 假设以下电影关系模式

- `movies(title,year,length,movietype,studioaddr,starname)`

- 包含了信息：

- 电影： `title,year,length,movietype`
  - 电影公司： `studioaddr`
  - 影星： `starname`

`('Star Wars',1977,124,'sciFic','fox','Fox Boulevard','Harrison Ford')`

`('Star Wars',1977,124,'sciFic','fox','Fox Boulevard','Mark Hamill')`

`('Star Wars',1977,124,'sciFic','fox','Fox Boulevard','Carrie Fisher')`

`('Star Wars',1977,124,'sciFic','fox','Fox Boulevard' .....`

# 关系模式的设计问题

## ■ 以下关系模式

movies(title,year,length,movietype,studio,studioaddr,starname)

- 它存在哪些问题？
- 原因何在？

电影	发行时间	片长	类型	电影公司	公司地址	演员
Empire Strikes Back	1980	111	drama	Fox	Fox Boulevard	Harrison Ford
Harry Potter and the ...	2007	120		Warner Bros.	NY Boulevard	Daniel Radcliffe
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Steve Gawley
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Carrie Fisher
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Anthony Daniels
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Mark Hamill
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Kenny Baker
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Peter Cushing
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Alec Guinness
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Harrison Ford
Terms of Endearment	1983	132	drama	MGM	MGM Boulevard	Debra Winger
Terms of Endearment	1983	132	drama	MGM	MGM Boulevard	Jack Nicholson
The Rescuers	1977	120	Animation	Disney	Disney Boulevard	Jack Nicholson



# 关系模式Movies中存在的问题

---

## 1. 数据冗余太大

- 例：('Star Wars',1977)电影的时长、电影公司地址重复出现

不必要的数据重复称为  
冗余(redundancy);

# 关系模式Movies中存在的问题

## 2. 更新异常 (Update Anomalies)

- 引起数据不一致
- 更新数据时，维护数据完整性代价大

电影	发行时间	片长	类型	电影公司	公司地址	演员
Empire Strikes Back	1980	111	drama	Fox	Fox Boulevard	Harrison Ford
Harry Potter and the ...	2007	120		Warner Bros.	NY Boulevard	Daniel Radcliffe
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Steve Gawley
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Carrie Fisher
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Anthony Daniels
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Mark Hamill
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Kenny Baker
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Peter Cushing
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Alec Guinness
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Harrison Ford
Terms of Endearment	1983	132	drama	MGM	MGM Boulevard	Debra Winger
Terms of Endearment	1983	132	drama	MGM	MGM Boulevard	Jack Nicholson
The Rescuers	1977	120	Animation	Disney	Disney Boulevard	Jack Nicholson

# 关系模式Movies中存在的问题

## 3. 删除异常（Deletion Anomalies）

- 为删除数据，删除了不该删除的信息

电影	发行时间	片长	类型	电影公司	公司地址	演员
Empire Strikes Back	1980	111	drama	Fox	Fox Boulevard	Harrison Ford
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Steve Gawley
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Carrie Fisher
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Anthony Daniels
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Mark Hamill
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Kenny Baker
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Peter Cushing
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Alec Guinness
Star Wars	1977	124	sciFic	Fox	Fox Boulevard	Harrison Ford
Terms of Endearment	1983	132	drama	MGM	MGM Boulevard	Debra Winger
Terms of Endearment	1983	132	drama	MGM	MGM Boulevard	Jack Nicholson
The Rescuers	1977	120	Animation	Disney	Disney Boulevard	Jack Nicholson





# 关系模式Movies中存在的问题

---

## 4.插入异常（Insertion Anomalies）

- 添加数据时无法执行
  - 例:如果一个影视公司刚成立，尚无电影，我们就无法把这个公司及其地址信息存入数据库



# 什么是“好的”关系模式

---

## ■ 结论:

- **Movies**关系模式不是一个好的模式
- “好”的模式不会发生插入异常、删除异常、更新异常，数据冗余应尽可能少



# 要解决的内容

---

- 到底什么样的模式才最佳？
- 怎么分解才能达到要求？
- 标准是什么？
- 如何实现？

# 数据依赖对关系模式的影响

## (续)



---

- 引起异常的原因：由存在于模式中的某些数据依赖引起的  
如： `studio name`→`studio addr`



# 关系模式的设计问题

---

## ■ 概念回顾

### ■ 直观

- 关系：一张二维表
- 元组：表中一行
- 属性：表中一列

### ■ 抽象

- 属性所对应的域——属性集
- 元组：属性集的笛卡尔积的一个元素
- 关系：元组的集合，即笛卡尔积的一个子集
- 关系模式：元组集合的结构上的描述（关系的集合）

## ■ 问题：关系模式的形式描述？



# 关系模式的设计问题

---

## ■ 关系模式的形式描述

- 关系模式由五部分组成，即关系模式是一个五元组：
- $R(U, D, DOM, F)$ 
  - 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: 属性到域的映射
  - F: 属性间的数据依赖集合。它限定了组成关系的各个元组必须满足的完整性约束条件。

属性间的数据依赖完全来自于现实世界的语义，而不是凭主观臆断的。



# 数据库设计方法之规范化理论

规范化理论正是用来分解关系模式来**消除其中不合适的数据依赖**，使所有关系模式最终达到某种**范式形式**，解决异常和数据冗余

**数据依赖**是各属性间的关联，包括

**函数依赖**、**多值依赖**和**连接依赖**

■ 数学中的函数 $y=f(x)$ ，自变量 $x$ 确定之后，相应的函数值 $y$ 也就唯一地确定了。

## 函数依赖 (FD) 定义

✧ 定义 1： 设 $R(U)$ 是属性集 $U$ 上的关系模式。 $X, Y$ 是 $U$ 的子集。对于 $R(U)$ 的任意一个关系 $r$ ， $X$ 的每一个具体值， $Y$ 都有唯一的值与之对应，则称“ $X$ 函数决定 $Y$ ”或“ $Y$ 函数依赖于 $X$ ”，记作 $X \rightarrow Y$ 。

✧  $X$ 为决定因子， $Y$ 是依赖因子。 $X \rightarrow Y$ 为模式 $R(U)$ 的一个函数依赖。

✧ 若 $Y$ 不函数依赖于 $X$ ，则记为  $X \nrightarrow Y$

违背了 $Sno \rightarrow Sname$			
Sno	Sname	Ssex	Sage
S1	张三	男	20
S1	李四	女	21
S3	王五	男	20
S4	赵六	男	21
S5	田七	男	20
⋮	⋮	⋮	⋮



由下面的关系表, 能否得出 **Sname**  $\rightarrow$  **Sno** ?

Sno	Sname	Ssex	Sage	Sdept
S1	张三	男	20	计算机系
S2	李四	女	21	自动化系
S3	王五	男	20	计算机系
S4	赵六	男	21	计算机系
S5	田七	男	20	计算机系
⋮	⋮	⋮	⋮	⋮

函数依赖不是指关系模式R的**某个或某些关系实例r**满足的约束条件, 而是指R的**所有关系实例r**均要满足的约束条件。

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**。

■ 如**Sname**  $\rightarrow$  **Sno**函数依赖只有在“学生不允许有重名”的条件下成立。

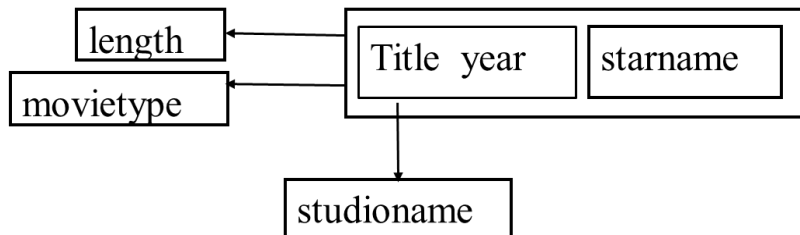
# 示例

- Movies(title,year,length,movietype,starname,studio name)
- Movies存在以下函数依赖：
  - Title,year->length
  - Title,year->movietype
  - Title,year->studio name

但是函数依赖

**Title,year->starname**不成立

**Title,year,starname->starname**成立





# 平凡函数依赖 (FD) 定义

- ☞ 若 $X \rightarrow Y$ ，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。

对于任一关系模式，平凡函数依赖都是必然成立的，它不反映新的语义，因此若不特别声明，**我们总是讨论非平凡函数依赖。**

# 完全函数依赖

✧ 定义 2：在关系模式  $R(U)$  中，如果  $X \rightarrow Y$ ，并且对于  $X$  的任何一个真子集  $X'$  都有  $X' \not\rightarrow Y$ ，则称  $Y$  完全函数依赖于  $X$ ，记作  $X \xrightarrow{F} Y$ 。

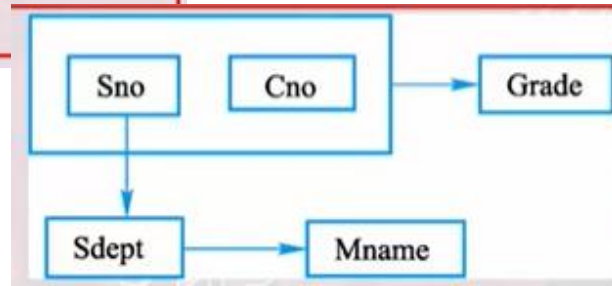
✧ 若  $X \rightarrow Y$ ，并且对于  $X$  的某一个真子集  $X'$  ( $X' \subset X$ )，有  $X' \rightarrow Y$ ，则称  $Y$  部分函数依赖于  $X$ ，记作： $X \xrightarrow{P} Y$ 。

✧ 如果  $Y$  部分函数依赖于  $X$ ，则  $X$  必定是组合属性。

[例] 在关系  $STUDENT(Sno, Sdept, Mname, Cno, Grade)$  中，

$(Sno, Cno) \xrightarrow{F} Grade$  是完全函数依赖

$(Sno, Cno) \xrightarrow{P} Sdept$  是部分函数依赖，因为  $Sno \rightarrow Sdept$ ，





# 讨论

[例] 关系模式 S-L-C(Sno, Cno, Sdept, Sloc, Grade)

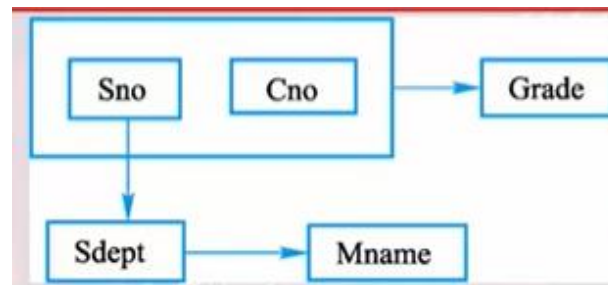
Sloc为学生住处，假设每个系的学生住在同一个楼。

$(Sno, Cno) \xrightarrow{F} Grade$   
 $Sno \rightarrow Sdept,$   
 $(Sno, Cno) \xrightarrow{P} Sdept$   
 $Sno \rightarrow Sloc$   
 $(Sno, Cno) \xrightarrow{P} Sloc$   
 $Sdept \rightarrow Sloc$

# 传递函数依赖

✧ 定义 3：在关系模式  $R(U)$  中，如果  $X \rightarrow Y$ , ( $Y \not\subseteq X$ ),  $Y \not\rightarrow X$ ,  $Y \rightarrow Z$  ( $Z \not\subseteq Y$ ), 则称  $Z$  传递函数依赖于  $X$ , 记作  $X \xrightarrow{\text{传递}} Z$ 。

[例] 在关系 **STUDENT** (**Sno**, **Sdept**, **Mname**, **Cno**, **Grade**) 中,  
 $Sno \rightarrow Sdept$ ,  $Sdept \rightarrow Mname$ ,  $Sno \xrightarrow{\text{传递}} Mname$







# 候选码

---

定义 4： 设有关系模式 $R(U)$ ， $U$ 是 $R$ 的全部属性集合， $K \subseteq U$ 是一个属性或是一个属性组合。

■ 若 $K \rightarrow U$ ，则 $K$ 为 $R$ 的**码**。

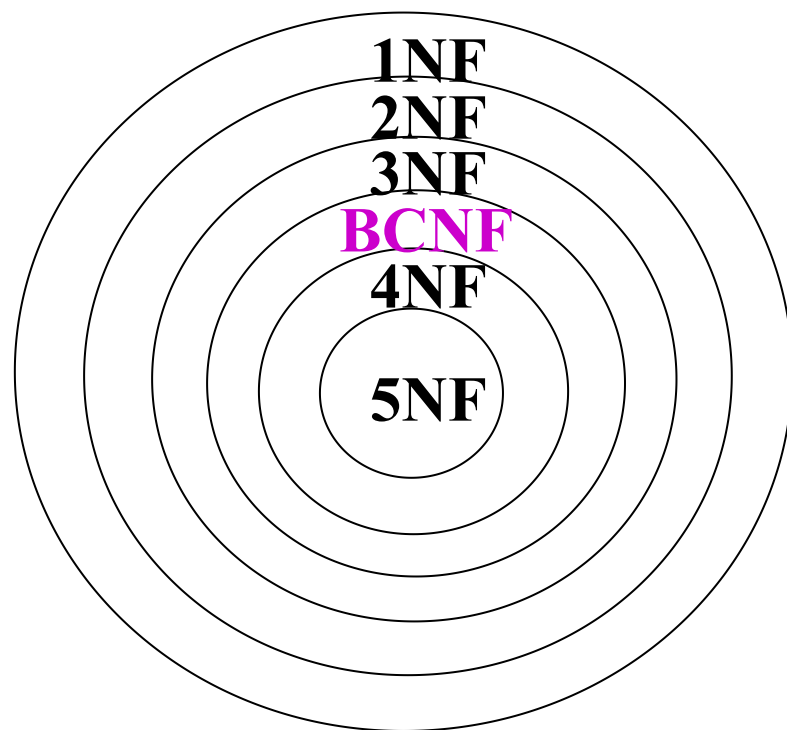
■ **候选码**： $K \xrightarrow{F} U$  ( $K \rightarrow U$ ，且不存在任何 $K' \subset K$ 使得 $K' \rightarrow U$ 成立)。

- 主属性：包含在任何一个候选码中的属性。
- 非主属性：不包含在任何一个候选码中的属性。

# 范式

## 定义

- 范式是对关系的不同数据依赖程度的要求，即符合某种级别的关系模式的集合，满足不同程度要求的为不同范式。
- 通过模式分解将一个低级范式转换为若干个高级范式的过程称作规范化（概念的纯粹化）





# 1NF

定义5: 设有关系模式R(U), 其所有的属性都是不可分的基本数据项, 则称R属于第一范式, 简称1NF, 记作 $R \in 1NF$ 。

- 在任何一个关系数据库系统中, 第一范式是对关系模式的一个最起码的要求。
- 不满足第一范式的数据库模式不能称为关系数据库。

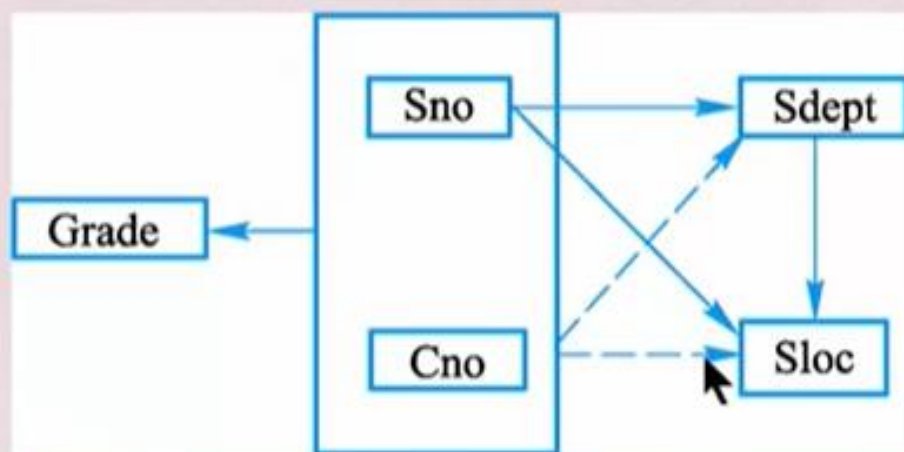
科目名称	科目余额表					
	期初余额		本期发生额		期末余额	
	借方	贷方	借方	贷方	借方	贷方
现金	950		4,360	4,350	960	
银行存款	2,690		14,910	7,460	10,140	
应收账款	16,660		1,740	18,400	0	
原材料	5,000		1,720	2,620	4,100	
预付账款	2,500		5,000	3,500	4,000	
待摊费用	500		200	100	600	
固定资产	5,400		5,000	0	10,400	
短期借款		2,000	2,000			0
应付账款		3,700	4,400			2,000
应付票据	0	5,000	4,000	2,000	0	3,600
预提费用	0	660	0	300	0	1,000
实收资本		20,000			0	20,000
未分配利润		2,340	0			3,600
	33,700	33,700	43,330	43,330	30,200	30,200

编号	职工姓名	性别	出生年月	籍贯	民族	工作时间	技术职务
1	李淑玉	女	09-01-69	北京市	汉族	07-01-90	讲师
2	王清照	女	11-01-51	山东济南	汉族	09-01-99	教授
3	辛加虎	男	08-01-54	河北无极	汉族	12-01-73	副教授
4	柳长亭	男	06-01-60	河南光山	回族	01-01-84	副教授
5	张煜	男	07-01-72	福建厦门	汉族	07-01-95	教授
6	周春花	女	01-01-63	上海市	汉族	10-01-99	副教授
7	李雨	男	02-01-68	山东青岛	汉族	06-01-99	助理研究员
8	欧阳太白	男	01-01-47	广西桂林	壮族	06-01-99	研究员

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 S-L-C(Sno, Cno, Sdept, Sloc, Grade)

Sloc为学生住处，假设每个系的学生住在同一个楼。



$(Sno, Cno) \xrightarrow{F} Grade$   
 $Sno \rightarrow Sdept$   
 $(Sno, Cno) \xrightarrow{P} Sdept$   
 $Sno \rightarrow Sloc$   
 $(Sno, Cno) \xrightarrow{P} Sloc$   
 $Sdept \rightarrow Sloc$

1. S-L-C满足第一范式。
- 2 S-L-C的码为(Sno, Cno)，主属性：Sno, Cno。  
非主属性：Grade，Sdept和Sloc。
3. 非主属性 Sdept 和 Sloc 部分函数依赖于码(Sno, Cno)。

[例] 关系模式 S-L-C(Sno, Cno, Sdept, Sloc, Grade)

Sloc为学生住处，假设每个系的学生住在同一个楼。

Sno	Sdept	Sloc	Cno	Grade
2014101	IS	N	3	89
2014101	IS	N	2	97
2014101	IS	N	5	88
2014103	IS	N	1	86
2014103	IS	N	3	92
2014104	IS	N	3	79
2014102	IS	N	null	null

插入异常

Sno	Sdept	Sloc	Cno	Grade
2014101	IS	N	3	89
2014101	IS	N	2	97
2014101	IS	N	5	88
2014103	IS	N	1	86
2014103	IS	N	3	92
2014104	IS	N	3	79

删除异常

Sno	Sdept	Sloc	Cno	Grade
2014101	IS	N	3	89
2014101	IS	N	2	97
2014101	IS	N	5	88
2014103	IS	N	1	86
2014103	IS	N	3	92
2014104	IS	N	3	79
2014101	IS	N	1	72
2014101	IS	N	4	65
2014101	IS	N	6	99

更新异常  
数据冗余



## 2NF

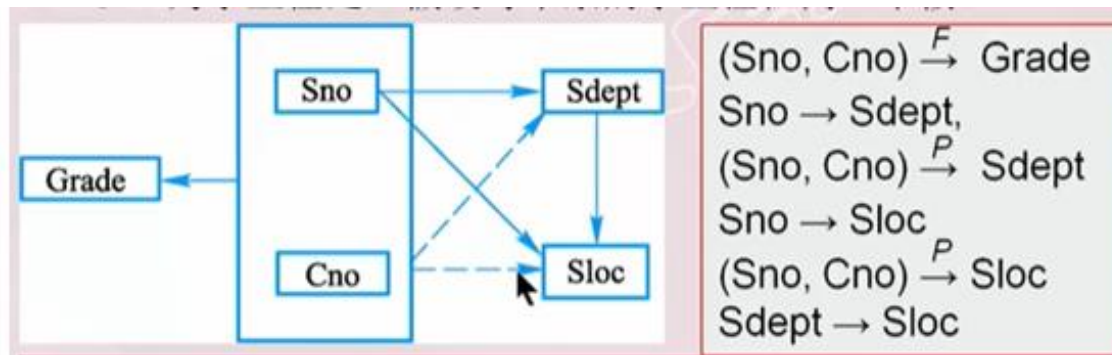
---

定义6: 设有关系模式 $R(U) \in 1NF$ ，如果其所有非主属性都完全函数依赖于码，则称 $R$ 属于第二范式，简称2NF，记作 $R \in 2NF$ 。

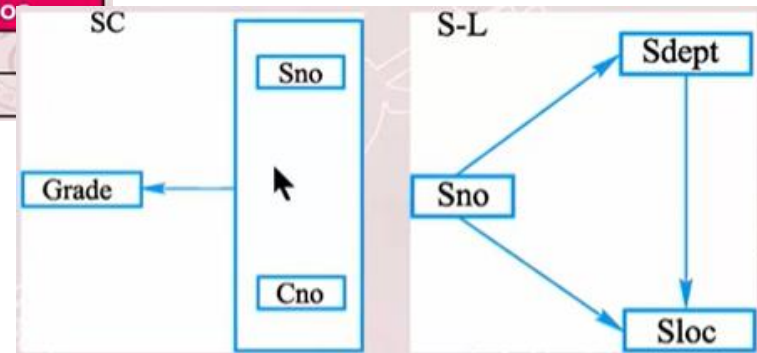
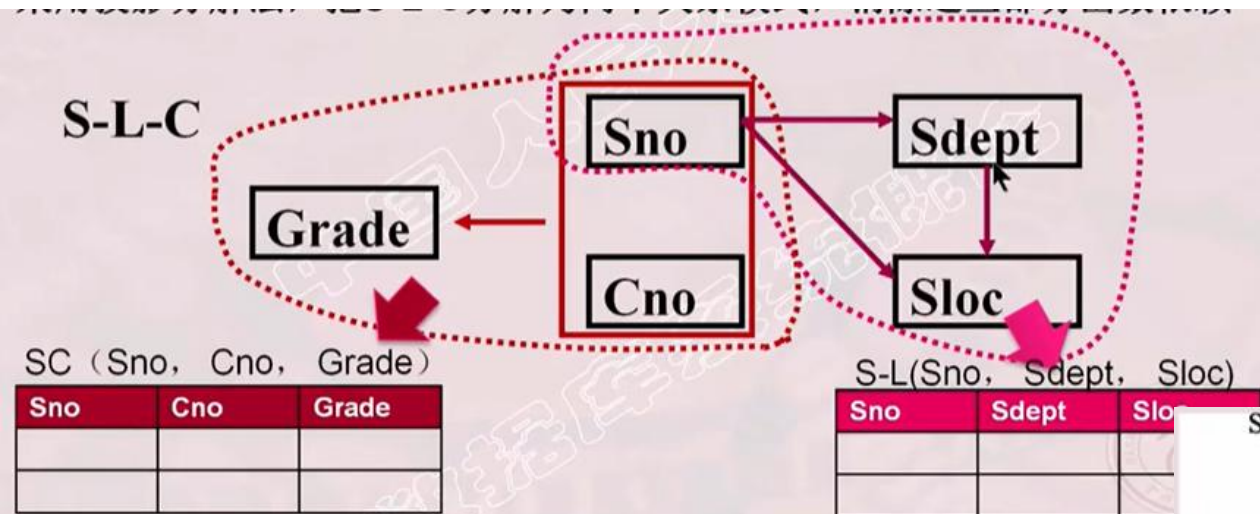
- 关系模式中不存在非主属性对码的部分函数依赖问题。



# 1NF→2NF



两类非主属性: grade  
sdept, sloc



SC

Sno	Cno	Grade

S-L

Sno	Sdept	Sloc

- (1) 由于学生选修课程的情况与学生的基本情况是分开存储在两个关系中的，在S-L关系中可以插入尚未选课的学生。
- (2) 删除一个学生的所有选课记录，只是SC关系中没有关于该学生的记录了，S-L关系中关于该学生的记录不受影响。
- (3) 不论一个学生选多少门课程，他的Sdept和Sloc值都只存储1次。这就大大降低了数据冗余。
- (4) 学生转系只需修改S-L关系中该学生元组的Sdept值和Sloc值，由于Sdept、Sloc并未重复存储，因此减化了修改操作。

2NF关系模式一定程度上减轻了插入异常，删除异常，更新异常以及数据冗余。



# 讨论

---

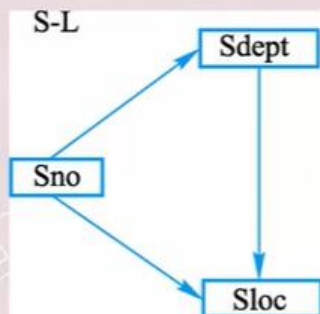
S1(SNO, SNAME, SAGE, SSEX, SDEPT, DNAME, CNO, GRADE)  
S1 ∈ 1NF

2NF?

S\_1(SNO, SNAME, SAGE, SSEX, SDEPT, DNAME) ∈ 2NF  
S\_2(SNO, CNO, GRADE) ∈ 2NF

# ■ 2NF关系模式S-L(Sno, Sdept, Sloc)中

函数依赖:



Sloc传递函数依赖于Sno，即S-L中存在非主属性对码的传递函数依赖 $Sno \xrightarrow{\text{传递}} Sloc$ 。

Sno	Sdept	Sloc
2014101	IS	N
2014102	IS	N
2014103	IS	N
2014104	IS	N
null	MA	S

Sno	Sdept	Sloc
2014101	IS	N
2014102	IS	N
2014103	IS	N
2014104	IS	N
2014105	PH	S
2014106	PH	S

Sno	Sdept	Sloc
2014101	IS	N
2014102	IS	N
2014103	IS	N
2014104	IS	N
2014105	PH	S
2014106	PH	S
2014107	PH	S
2014108	PH	S
.....	.....	.....

2NF关系模式不能消除插入异常，删除异常，更新异常以及数据冗余。

Sno	Sdept	Sloc
2014101	IS	N
2014102	IS	N
2014103	IS	N
2014104	IS	N
2014105	PH	S
2014106	PH	S
2014107	PH	S
2014108	PH	S
.....	.....	.....





## 3NF

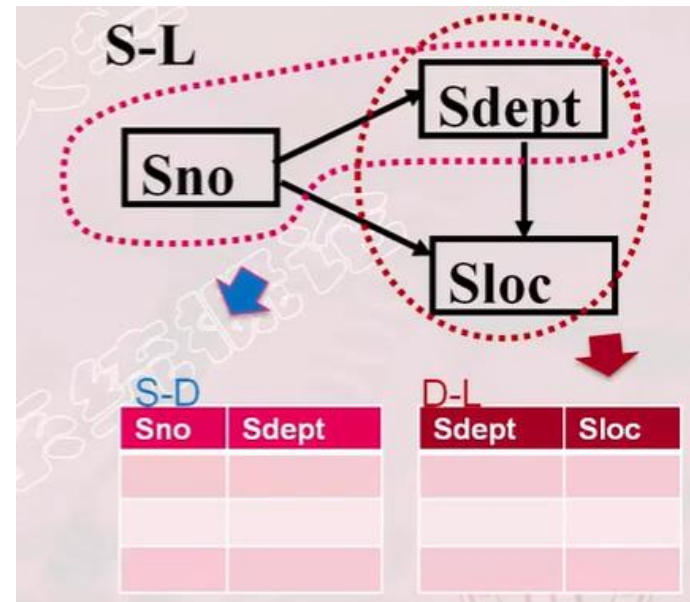
---

定义7: 设有关系模式 $R(U) \in 2NF$ , 如果其**所有非主属性都不传递函数依赖于码**, 则称 $R$ 属于第三范式, 简称**3NF**, 记作 $R \in 3NF$ 。

□ 解决关系模式中**非主属性对码的传递函数依赖**问题。

# 2NF → 3NF

S-L中Sloc传递函数依赖于Sno，  
即：非主属性传递函数依赖码



# 异常进一步缓解

S-D	Sno	Sdept	D-L	Sdept	Sloc

(1) D-L关系中可以插入系的信息，即使还没有在校学生。

(2) 某个系的学生全部毕业了，只是删除S-D关系中的相应元组，D-L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D-L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D-L关系中一个元组的Sloc属性值。



# 讨论

---

S1(SNO, SNAME, SAGE, SSEX, SDEPT, DNAME, CNO, GRADE)     S1 ∈ 1NF

2NF?

S\_1(SNO, SNAME, SAGE, SSEX, SDEPT, DNAME) ∈ 2NF  
S\_2(SNO, CNO, GRADE) ∈ 2NF

3NF?

S\_1(SNO, SNAME, SAGE, SSEX, SDEPT) ∈ 3NF  
S\_3(SDEPT, DNAME) ∈ 3NF  
S\_2(SNO, CNO, GRADE) ∈ 3NF



# 讨论

---

关系模式  $R(S, C, P)$ ,  $S$ 表示学号,  $C$ 表示课程号,  $P$ 表示成绩排名 (一门课程没有并列名次)。

- (1) 候选码
- (2) 函数依赖
- (3)  $R$ 属于第几范式?



## 一个3NF的例子

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

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

**STJ (S,T,J)  $\in$  3NF,**

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

**(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

**原因：**

**主属性J部分依赖于码(S, T)。因为 $T \rightarrow J$**





# BCNF

---

BCNF 设关系模式 $R\langle U, F \rangle \in 1NF$ ，如果对于 $R$ 的每个函数依赖 $X \rightarrow Y$ ，且 $X \not\supseteq Y$ 时， $X$ 必含有码，那么 $R \in BCNF$ 。

□ 每一个决定因素都包含候选码

定义7： 设有关系模式 $R(U) \in 3NF$ ，如果其所有主属性都不部分和传递依赖于码，则称 $R$ 属于BCNF范式，记作 $R \in BCNF$ 。





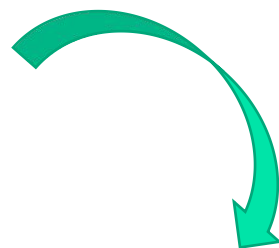
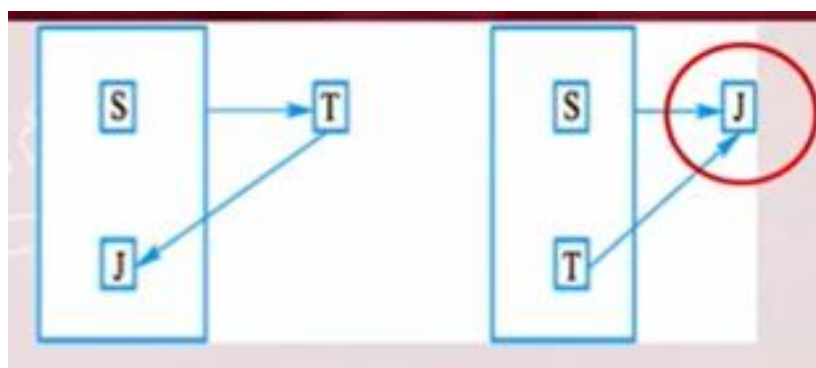
# BCNF关系模式的性质

---

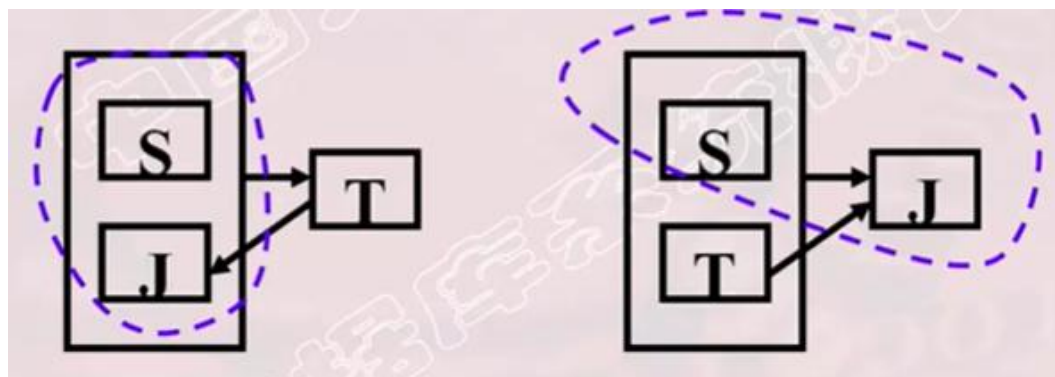
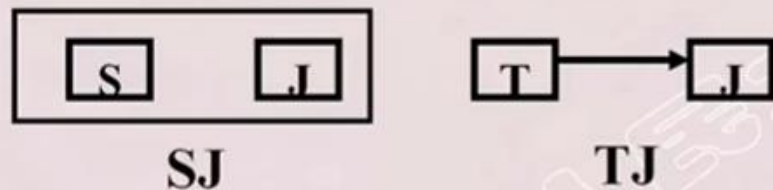
- 1.所有非主属性对每一个码都是完全函数依赖。
- 2.所有主属性对每一个不包含它的码也是完全函数依赖。
- 3.没有任何属性完全函数依赖于非码的任何一组属性。

如果一个关系模式属于 $BCNF$ ，则在函数依赖范畴内已经实现了彻底的分离。

- (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SJ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。



❖ SJ的码为 (S, J)， TJ的码为T。



# 关系模式的规范化过程

0NF

非规范化的关系

1NF

消除属性的组合性，并指定关键字

2NF

消除非主属性对关键字的部分函数依赖

3NF

消除非主属性对关键字的传递函数依赖

BCNF

消除主属性对关键字的部分和传递函数依赖



# 总结说明

---

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 “**studio name**→**studio addr**” 这个函数依赖只有在不允许电影公司有多个地址条件下成立



# 讨论

---

【例】：已知关系模式 $R(A, B, C, D, E)$ ， $R$ 上存在的函数依赖有 $F = \{AB \rightarrow E, B \rightarrow C, C \rightarrow D\}$ ，求：

- ① 关系模式候选码；
- ② 该关系模式满足几范式，为什么？
- ③ 将 $R$ 分解为高一级的范式，并说明理由。

**【例】：** 已知关系模式 $R(A, B, C, D, E)$ ， $R$ 上存在的函数依赖有 $F = \{AB \rightarrow E, B \rightarrow C, C \rightarrow D\}$ ，求：

① 关系模式的码；

解：① 根据函数依赖集 $F = \{AB \rightarrow E, B \rightarrow C, C \rightarrow D\}$

由于 $(A, B)$ 能够决定所有的属性集合，且没有任一真子集能够决定所有的属性集合，所以 $R$ 的码为 $(A, B)$ 。

【例】：已知关系模式 $R(A, B, C, D, E)$ ， $R$ 上存在的函数依赖有 $F = \{AB \rightarrow E, B \rightarrow C, C \rightarrow D\}$ ，求：

② 该关系模式满足几范式，为什么？

解：

② 已知 $R$ 的码为 $(A, B)$ ，所以非主属性为 $C, D, E$ 。

$\because$   $F$ 中存在 $B \rightarrow C$

$\therefore$  存在非主属性 $C$ 对码 $(A, B)$ 的部分函数依赖，所以 $R \in 1NF$ 。



**【例】：** 已知关系模式 $R(A, B, C, D, E)$ ， $R$ 上存在的函数依赖有 $F = \{AB \rightarrow E, B \rightarrow C, C \rightarrow D\}$ ，求：

③将 $R$ 分解为高一级的范式，并说明理由。

③ 将 $R$ 分解为 $R_1, R_2$ ：

$R_1(A, B, E)$ ， $F_1 = \{(A, B) \rightarrow E\}$ ；

$R_2(B, C, D)$ ， $F_2 = \{B \rightarrow C, C \rightarrow D\}$ 。

$R_1$ 中：不存在非主属性对候选码的部分函数依赖，所以 $R_1 \in 2NF$ 。

$R_2$ 中：码为 $B$ ，则非主属性为 $C, D$ ，不存在非主属性对候选码的部分函数依赖，所以 $R_2 \in 2NF$ 。