



SQL SERVER 2005

# 数据库系统概论

参考：第六章 关系数据库理论P<sub>177</sub>—P<sub>204</sub>

# 内容回顾

## ❖ 关系

- 笛卡尔积、关系、候选码、主码、主属性
- 关系6个性质

## ❖ 关系的完整性

- 实体完整性、参照完整性、自定义完整性

## ❖ 关系操作

- 并、交、差、笛卡尔积
- 选择、投影、连接

# 本节教学目标

## ❖ 掌握

- 函数四种依赖，重点掌握函数的部分依赖、传递依赖
- 规范化：2NF, 3NF, BCNF，能够利用投影法熟练对关系进行规范化

## ❖ 了解

- 多值依赖，4NF、5NF

## ❖ 重点

- 规范化：2NF, 3NF, BCNF

## ❖ 难点

- 规范化：3NF, BCNF

# 本节内容

- 第一节 问题的提出
- 第二节 规范化



# 第一节 问题的提出

- ❖ 什么是数据依赖
- ❖ 关系模式的简化定义
- ❖ 数据依赖对关系模式影响

# 一、什么是数据依赖

## ❖ 1. 完整性约束的表现形式

- 限定属性取值范围：例如学生成绩必须在0-100之间
- 定义属性值间的相互关连

## ❖ 2. 数据依赖

- 一个关系内部属性与属性之间的约束关系
- 现实世界属性间相互联系的抽象
- 数据内在的性质
- 语义的体现

### ❖ 3. 数据依赖的类型

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



## 二、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

# 三、数据依赖对关系模式的影响

[例1] 建立一个描述学校教务的数据库：

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：  $\text{Student} \langle U, F \rangle$

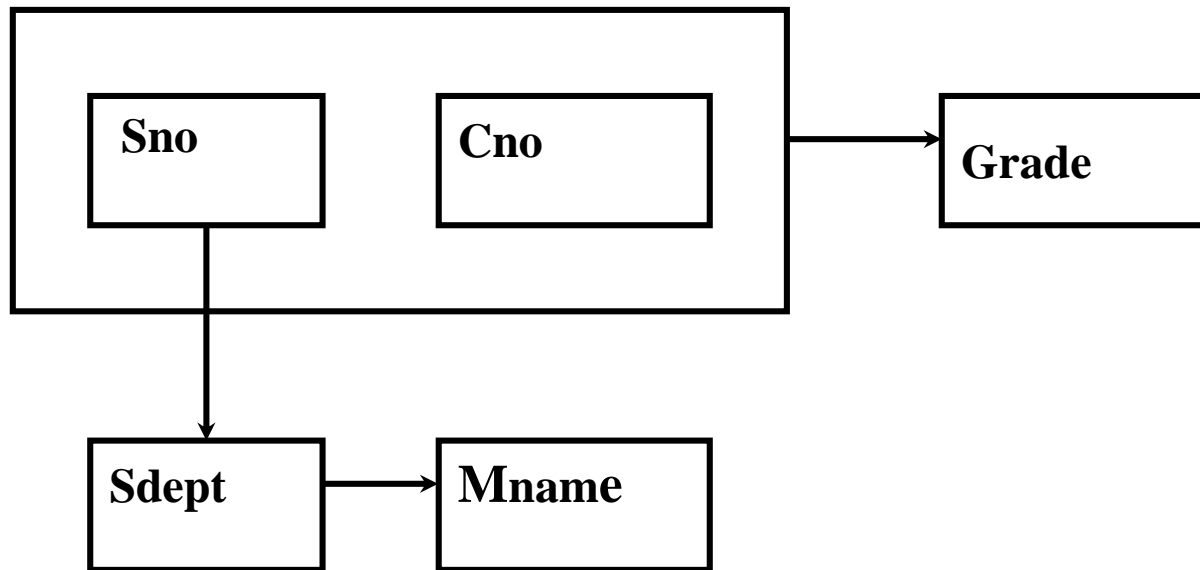
$U = \{ Sno, Sdept, Mname, Cno, Grade \}$

## ❖ 学校数据库的语义：

- 1. 一个系有若干学生， 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。

**属性组U上的一组函数依赖F:**

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



假设存在这样一个关系：

**Student**(Sno, Sdept, Mname, Cno, Grade)

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
⋮	⋮	⋮	⋮	⋮



请问该关系模式好吗？



Sno	Sdept	Mname	Cno	Grade
-----	-------	-------	-----	-------


- 系名、系主任名重复出现
  - “张明”退休，李四接替
  - 一个新系刚成立，尚无学生
  - 一个系的学生全部毕业
- 数据冗余太大
  - 更新异常
  - 插入异常
  - 删除异常

# 小结

## 结论：

- Student关系模式不是一个好的关系模式。
- “好”的关系模式：

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

**原因：** 由存在于关系模式中的**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

# 本节内容

- 第一节 问题的提出
- 第二节 规范化





# 第二节 规范化

❖ 函数依赖

❖ 码

❖ 范式

- 1NF

- 2NF

- 3NF

- BCNF

# 函数依赖

**定义6.1** 设 $R(U)$ 是一个属性集 $U$ 上的关系模式， $X$ 和 $Y$ 是 $U$ 的子集。

若对于 $R(U)$ 的任意一个可能的关系 $r$ ， $r$ 中不可能存在两个元组在 $X$ 上的属性值相等，而在 $Y$ 上的属性值不等，则称“ **$X$ 函数确定 $Y$** ”或“ **$Y$ 函数依赖于 $X$** ”，记作 $X \rightarrow Y$ 。

$X$ 称为这个函数依赖的**决定属性集**(Determinant)。

$$Y=f(x)$$

# 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“姓名 $\rightarrow$ 年龄”这个函数依赖只有在不允许有同名人的条件下成立
3. 数据库设计者可以对现实世界作强制的规定。例如规定不允许同名人的出现，函数依赖“姓名 $\rightarrow$ 年龄”成立。所插入的元组必须满足规定的函数依赖，若发现有同名人的存在，则拒绝装入该元组。

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

假设不允许重名, 则有:

$Sno \rightarrow Ssex, \quad Sno \rightarrow Sage, \quad Sno \rightarrow Sdept,$

$Sno \longleftrightarrow Sname, \quad Sname \rightarrow Ssex, \quad Sname \rightarrow Sage$

$Sname \rightarrow Sdept$

但  $Ssex \nrightarrow Sage$

若  $X \rightarrow Y$ , 并且  $Y \rightarrow X$ , 则记为  $X \longleftrightarrow Y$ 。

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

# 函数依赖（cont）

❖ 函数依赖可以从不同角度分为：

- 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

# 平凡函数依赖与非平凡函数依赖

在关系模式 $R(U)$ 中，对于 $U$ 的子集 $X$ 和 $Y$ ，

如果 $X \rightarrow Y$ ，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**

例：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： $(Sno, Cno) \rightarrow Sno$

$(Sno, Cno) \rightarrow Cno$

# 完全函数依赖与部分函数依赖

定义6.2 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 $X \rightarrow Y$ , 但 $Y$ 不完全函数依赖于 $X$ , 则称 $Y$ 部分函数依赖于 $X$ , 记作 $X \xrightarrow{p} Y$ 。

例：在关系SC(Sno, Cno, Grade)中，

由于：  $Sno \twoheadrightarrow Grade$ ,  $Cno \twoheadrightarrow Grade$ ,

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

在关系Student(Sno, Sdept, Mname, Cno, Grade)

由于：  $Sno \rightarrow Sdept$ , Sno是(Sno, Cno)的真子集

因此：  $(Sno, Cno) \xrightarrow{p} Sdept$



# 传递函数依赖

定义6.3 在关系模式 $R(U)$ 中, 如果 $X \rightarrow Y$ ,  $(Y \not\subseteq X)$ ,  $Y \not\rightarrow X$ ,

$Y \rightarrow Z$ , 则称 $Z$ 传递函数依赖于 $X$ , 记为:  $X \xrightarrow{\text{传递}} Z$

**注:** 如果 $Y \rightarrow X$ , 即 $X \longleftrightarrow Y$ , 则 $Z$ 直接依赖于 $X$ 。

例: 在关系 $Std(Sno, Sdept, Mname)$ 中, 有:

$Sno \rightarrow Sdept$ ,  $Sdept \rightarrow Mname$ ,  $Sdept \not\rightarrow Sno$

$Mname$ 传递函数依赖于 $Sno$

# 第二节 规范化

❖ 函数依赖

❖ 码

❖ 范式

- 1NF

- 2NF

- 3NF

- BCNF

# 码

**定义6.4** 设 $K$ 为 $R\langle U, F \rangle$ 中的属性或属性组合。

若 $K \xrightarrow{F} U$ ，则 $K$ 称为 $R$ 的**候选码** (Candidate Key)。

若候选码多于一个，则选定其中的一个做为

**主码** (Primary Key)。

- 主属性和非主属性
- 全码

### [例2]

关系模式S (Sno, Sdept, Sage), 单个属性Sno是码,  
SC (Sno, Cno, Grade) 中, (Sno, Cno) 是码

### [例3]

关系模式R (P, W, A)

P: 演奏者      W: 作品      A: 听众

一个演奏者可以演奏多个作品

某一作品可被多个演奏者演奏

听众可以欣赏不同演奏者的不同作品

码为(P, W, A), 即All-Key

# 外码

**定义6.5** 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

- 如在  $SC(\underline{Sno}, Cno, Grade)$  中， $Sno$  不是码，但  $Sno$  是关系模式  $S(\underline{Sno}, Sdept, Sage)$  的码，则  $Sno$  是关系模式  $SC$  的外部码
- **主码**与**外码**一起提供了表示关系间联系的手段

# 第二节 规范化

❖ 函数依赖

❖ 码

❖ 范式

- 1NF

- 2NF

- 3NF

- BCNF

# 范式

- ❖ 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)

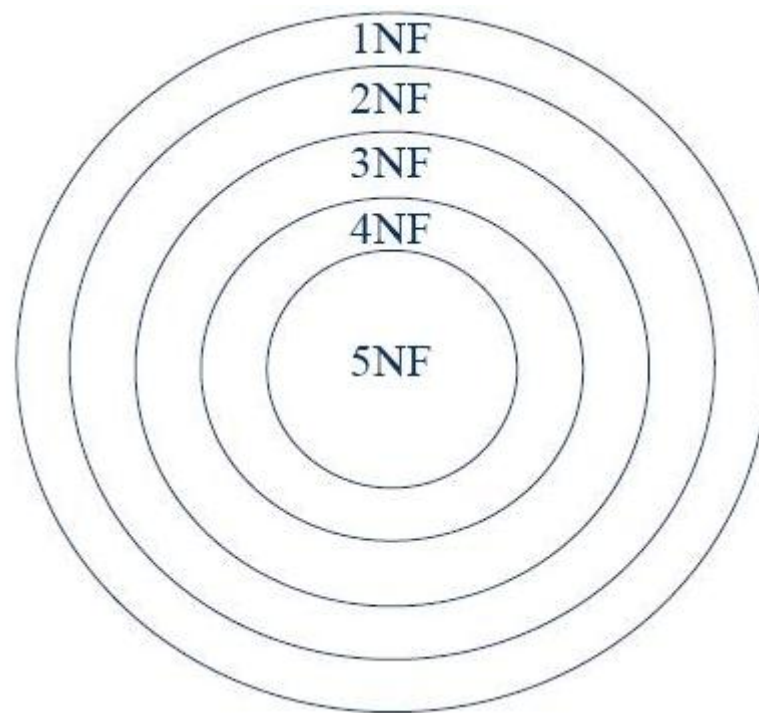
第五范式(5NF)

问题  
越  
少



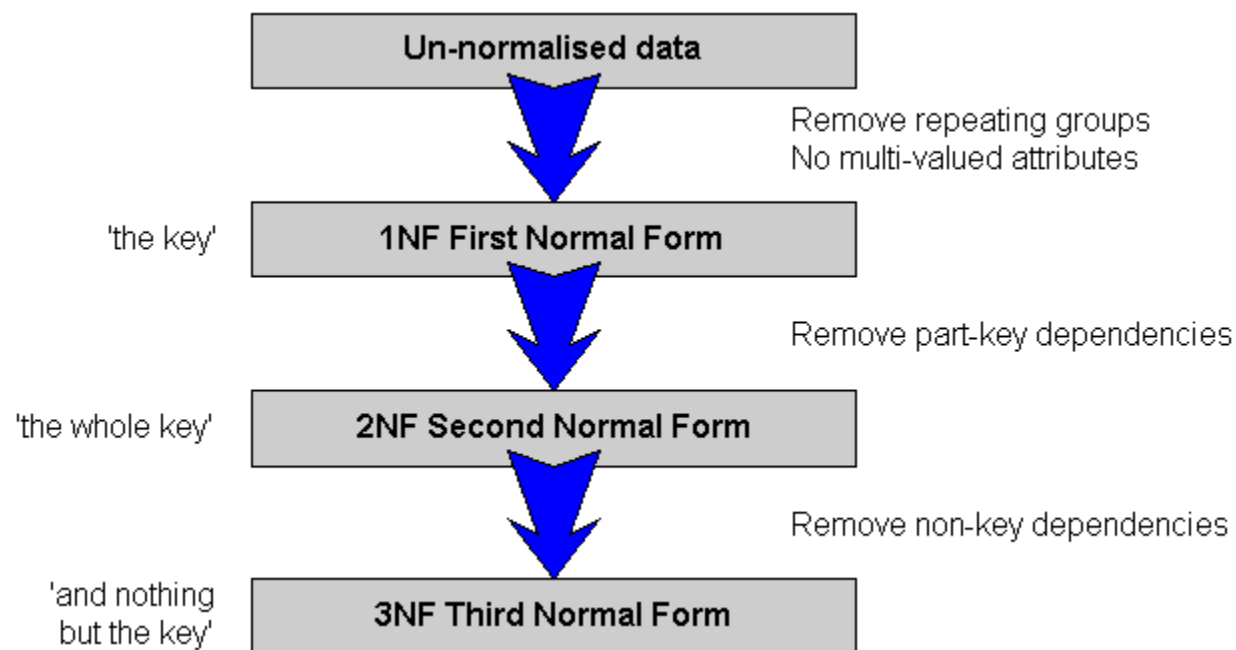
# 范式(cont)

- ❖ 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**可以转换为若干个高一级范式的关系模式的集合，这种过程就叫**规范化**





# 核心范式



# 1NF



## 定义：

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 $R \in 1NF$ 。简单一点来说，符合1范式的关系，就是不存在表中套表的情况

- 关系中不存在重复行、多值列
- 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。



## 实例分析

下面表格是一个不规范化（UNF）学生选课系统的实例，  
我们将从这个实例开始，一步一步将其规范化

表中套表

cno	cname	sno	sname	birthday	Grade	Result
IS701	Database Programming & Web Development	M928642	Bloggs	10/01/1984 01/04/1980 25/08/1975	A	Distinction
		M928268	Smith		C	Pass
		N926484	Green		D	Pass
		N978787	Morris		F	Fail
IS702	Communication Technologies	M928273	Bloggs	01/04/1980 25/08/1975	E	Fail
		N926484	Green		B	Merit
		N978787	Morris		C	PassFail
IS705	Business Systems	M928268	Smith	01/04/1980 25/08/1975	A	Distinction
		N926484	Green		C	Pass
		N978787	Morris		B	Merit

**UNF Un-normalised Form**（不符合1NF的形式、未规范化的形式）



第一步:

将UNF转换成1NF，方法是剔除表中所套的表



SC

Sno	Cno	Sname	Birthday	Grade	Result
M928642	IS701	Bloggs		A	Distinction
M928268	IS701	Smith	10/01/1984	C	Pass
N926484	IS701	Green	01/04/1980	D	Pass
N978787	IS701	Morris	25/08/1975	F	Fail

Course

Cno	Cname
IS701	Database Programming & Web Development
IS702	Communication Technologies
IS705	Business Systems

❖ 转换成1NF后，关系还存在：

 插入异常

 删除异常

 数据冗余度大

 修改复杂

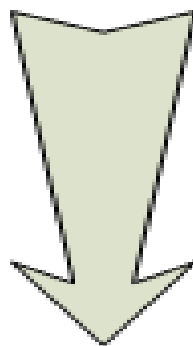




第二步:

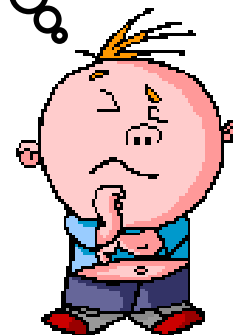
将符合1NF的关系分解成符合2NF的多个关系

1NF



2NF

什么是  
2NF?



# 2NF

Course(Cno, Cname)



定义 若 $R \in 1NF$ ，且每一个**非主属性完全**函数依赖于码，则 $R \in 2NF$ 。

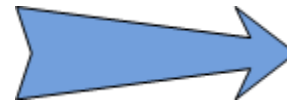
我们可以根据函数的依赖关系来分析SC和Course是不是2NF

SC(Sno, Cno, Sname, Birthdate, Grade, Result)

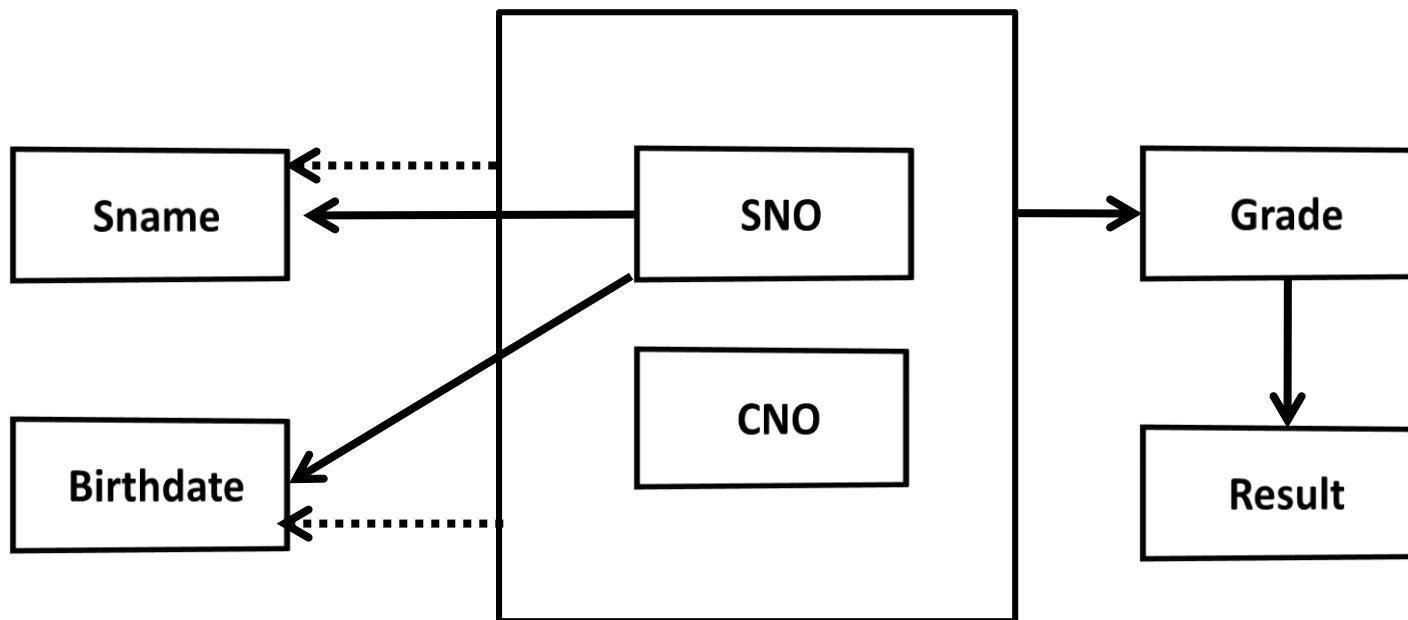
$(Sno, Cno) \xrightarrow{f} Grade$

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

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



**SC  $\notin$  2NF**

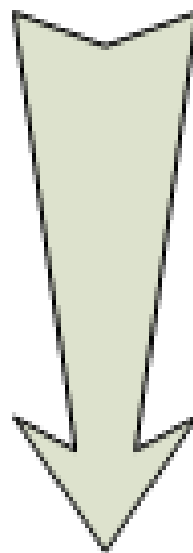


- ❖ SC的码为(Sno, Cno)
- ❖ SC满足第一范式。
- ❖ 非主属性Sname、Birthdate部分函数依赖于码(Sno, Cno)



1NF

消除非主属性对  
主码的**部分依赖**

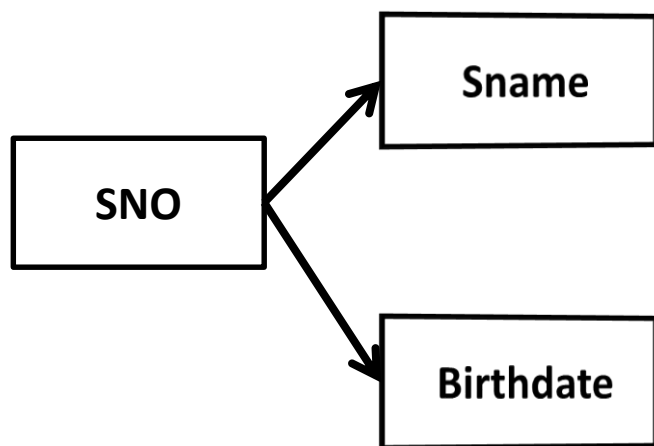


方法：将一个模式分解为多个  
模式，直至每个模式里都不存  
在非主属性对主码的**部分依赖**

2NF

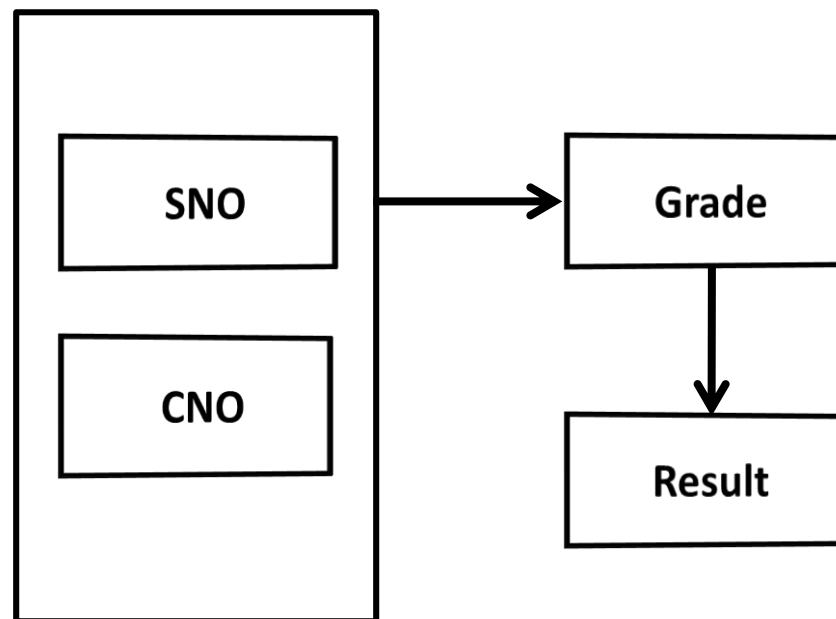
## ❖ 将SC分解为:

Student(Sno, Sname, Birthdate)



Student中函数依赖

Sgrade(Sno, Cno, Grade, Result)



Sgrade中函数依赖

这样Student和Sgrade都不存在非主属性的部分依赖，都属于2NF

在2NF关系模式Sgrade (Sno, Cno, Grade, Result)  
中存在以下函数依赖:

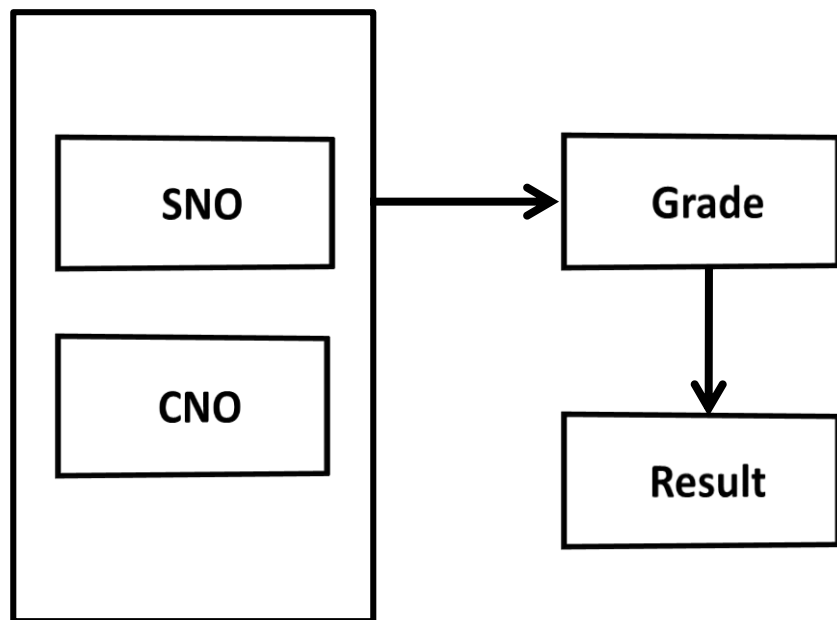
$$(Sno, Cno) \rightarrow Grade$$

$$Grade \rightarrow Result$$

$$(Sno, Cno) \xrightarrow{\text{传递}} Result$$

Result传递函数依赖于(Sno, Cno) , 即Sgrade中存在  
非主属性对码的传递函数依赖。

## 函数依赖图：



### ■ 插入异常

如果没有学生得A则Result中Distinction无法插入到数据库。

### ■ 删除异常

如果学生只有一个学生得A，那么删除这个学生信息时Result中Distinction也丢失了。

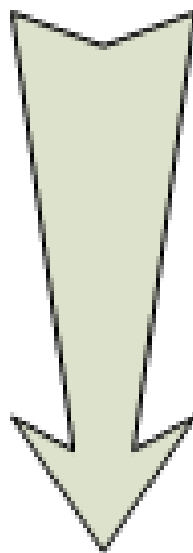
### ■ 数据冗余

每个Grade都对应一个Result，则Result列重复存储。

# 解决方法

2NF

消除非主属性对  
主码的传递依赖



方法：将一个模式分解为多个  
模式，直至每个模式里都不存  
之非主属性对主码的传递依赖

3NF

# 3NF



定义

关系模式  $R\langle U, F \rangle$  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  $R\langle U, F \rangle \in 3NF$ 。

例，  $Sgrade(Sno, Cno, Grade, Result) \in 2NF$

$(Sno, Cno) \xrightarrow{\text{传递}} Result$

$Sgrade(Sno, Cno, Grade, Result) \notin 3NF$

## ❖ 解决方法

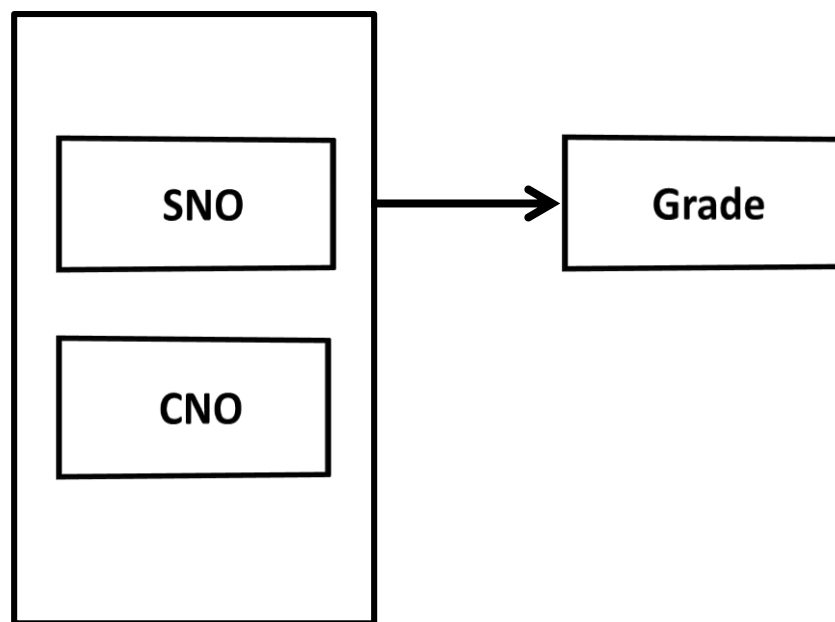
采用投影分解法，把Sgrade分解为两个关系模式，以消除传递函数依赖：

SG (Sno, Cno, Grade)

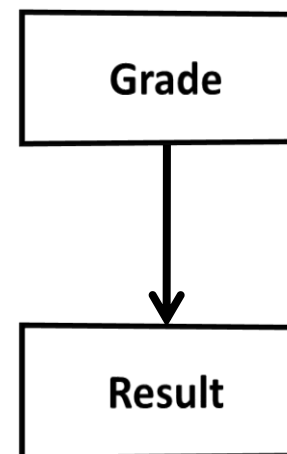
GR (Grade, Result)

SG的码为(Sno,Cno)， GR的码为Grade。

## 函数依赖图



SG



GR



- ❖ 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。

# BCNF

❖ **定义6.8** 关系模式 $R\langle U, F \rangle \in 1NF$ ，若 $X \rightarrow Y$ 且 $Y \not\subseteq X$ 时 $X$ 必含有码，则 $R\langle U, F \rangle \in BCNF$ 。

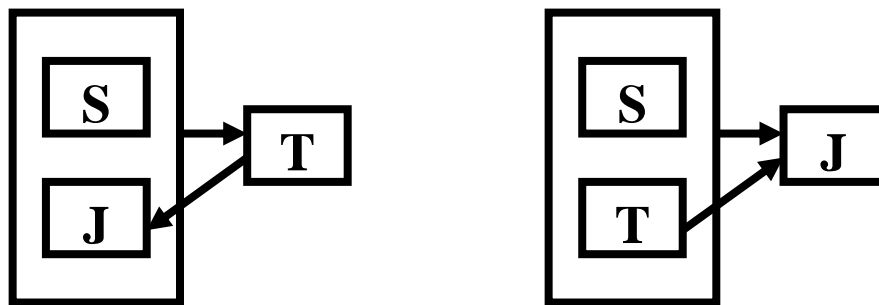
❖ 若 $R \in BCNF$

- 所有非主属性对每一个码都是完全函数依赖
- 所有的主属性对每一个不包含它的码，也是完全函数依赖
- 没有任何属性完全函数依赖于非码的任何一组属性

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

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。

- 每一教师只教一门课。每门课由若干教师教，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： $(S, J) \rightarrow T$ ,  $(S, T) \rightarrow J$ ,  $T \rightarrow J$



STJ函数依赖

❖  $STJ \in 3NF$

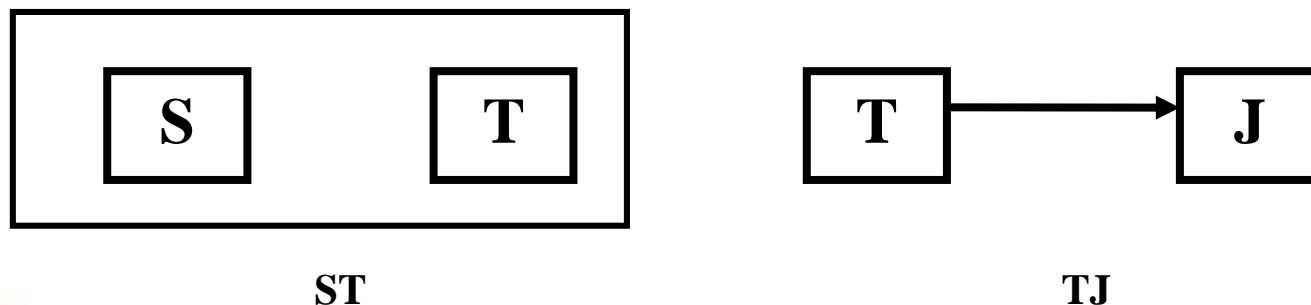
- 没有任何非主属性对码传递依赖或部分依赖

❖  $STJ \notin BCNF$

- 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：

- $SJ(S, J) \in BCNF, TJ(T, J) \in BCNF$



# 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

# Q & A

❖ 按照规范化理论设计的关系模式是最优的吗？



# 这次课我们学到了...

## ❖ 函数依赖

- 完全函数依赖、部分函数依赖

## ❖ 规范化

- 2NF、3NF
- 投影分解法