课程内容

- 数据库系统基本概念（数据模型，体系结构）
- 关系数据库
- 关系数据库标准语言SQL
- 数据库保护
- 关系数据理论
- 数据库设计
- 存储管理与索引
- 查询处理和查询优化
- 事务处理技术
- 数据库技术新发展

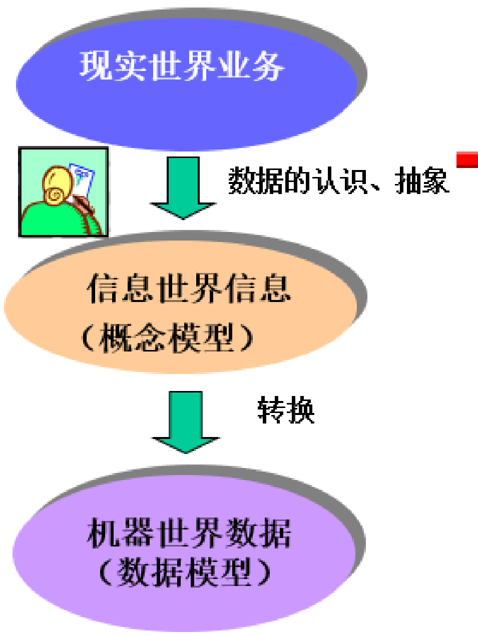
第一部分
基础理论

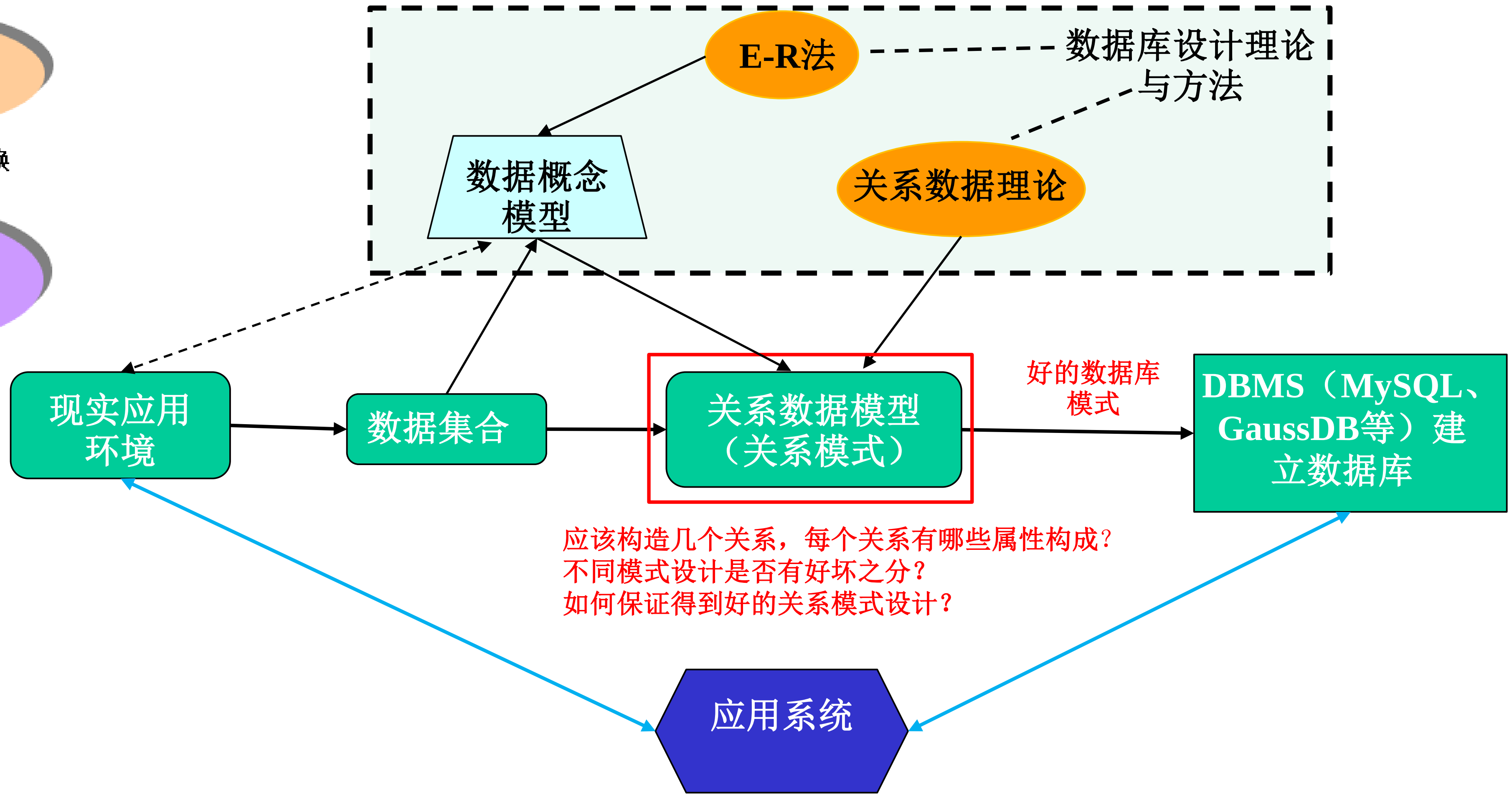
第二部分
设计理论

第三部分
实现技术

第四部分
新技术

数据库设计理论概述


现实世界业务
数据的认识、抽象
信息世界信息
（概念模型）
转换
机器世界数据
（数据模型）


E-R法
数据库设计理论
与方法
数据概念
模型
关系数据理论
现实应用
环境
数据集合
关系数据模型
（关系模式）
好的数据库
模式
DBMS（MySQL、
GaussDB等）建
立数据库
应该构造几个关系，每个关系有哪些属性构成？
不同模式设计是否有好坏之分？
如何保证得到好的关系模式设计？
应用系统

第五章 关系数据理论

- 本章主要内容
  - 关系模式设计中的问题
  - 数据依赖（函数依赖、多值依赖）
  - 规范化
  - 模式分解理论
- 本章要阐述的问题
  - 1. 什么是好的关系模式设计，什么是不好的设计？
  - 2. 关系的属性间存在怎样的依赖关系？
  - 3. 不同类型的数据依赖如何影响关系模式的好坏？
  - 4. 怎样把不好的模式设计变成好的？

关系模式设计中的问题

例：为学校设计一个关系数据库，管理的信息包括学生学号、选修课程名称、成绩、所在系以及系主任名。

现实世界的语义：

- 一个系有若干个学生，但一个学生只属于一个系。
- 一个系只有一名系主任。
- 一个学生可以选修多门课程，每门课程可有若干学生选修。
- 每个学生学习每门课程有一个成绩

设计名称为UN的关系模式：

UN (S#，CN，G，SDN，MN)，

其中：S# —学号，CN — 课程名，G — 成绩，

SDN — 系名，MN — 系负责人

关系模式设计中的问题

- 对UN进行操作时的问题：

UN (S#， CN， G， SDN， MN)

插入：插入异常！

修改：数据冗余大，修改异常！

删除：删除异常！

- UN是“不好”的关系模式！

关系模式设计中的问题

将学生管理数据库进行如下设计：

SD (S#， SDN)
SG (S#， CN， G)
DM (SDN， MN)

其中： S# —学号， SDN — 系名， CN — 课程名，G — 成绩， MN — 系负责人

SD、SG、DM不会发生插入异常和删除异常，冗余最少。
SD、SG、DM是“好的”数据库模式！

问题的分析

UN(S#，CN，G，SDN，MN)

不好的关系模式

关系内部属性之间关系：

S# → SDN；

SDN → MN；

S# →MN；

(S#，CN)→G；

(S#，CN)→SDN；

(S#，CN)→MN

？因素

因素：关系内部属性之间的约束关系

好的关系模式

SD (S#，SDN)

SG (S#，CN，G)

DM (SDN，MN)

关系内部属性之间关系：

S# → SDN；

(S#，CN)→G；

SDN → MN；

关系模式定义：R（U，D，dom，F）

数据依赖的概念

- 数据依赖：一个关系内部属性值之间相互依赖又相互制约的关系。如：

  S# → SDN；　SDN → MN；　(S#，CN)→SDN

- 数据依赖有许多种类型，其中最重要的有两种：
  - 函数依赖(Functional Dependency);
  - 多值依赖(Multivalued Dependency)。

关系数据理论与数据依赖

不好的关系模式

如：UN（S#，CN，G，SDN，MN）

关系的
规范化

规范化方法

- 模式分解

关系模式规范的标准

- 范式理论

数据依赖的概念与公理

- 函数依赖
- 多值依赖

关系数据理论

好的关系模式

如：

SD（S#，SDN）

SG（S#，CN，G）

DM（SDN，MN）

关系数据理论

- 函数依赖
  - 定义、类型以及函数依赖公理系统
- 规范化
  - 范式概念（1NF），与函数依赖相关的范式（2NF, 3NF, BCNF）
- 多值依赖与第四范式
- 模式分解的理论
- 侯选码的求解理论和算法

函数依赖

- 函数依赖的定义
- 三种函数依赖
- 关系键的形式定义
- 函数依赖公理系统

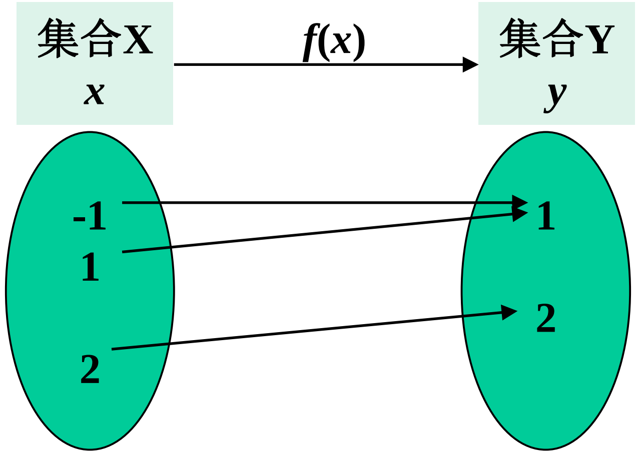
函数依赖

- **函数依赖的定义**

设R(U)是属性集U上的关系模式。X、Y是U的子集。 r是R的任意一个具体关系，t，s 是r中任意两个元组。如果t[X] = s[X]，则t[Y] = s[Y]，则称“X函数确定Y”或“Y函数依赖于X”，记作：X→Y。

- **函数依赖X→Y也可定义为：**

对于X的每个具体值，Y有唯一的值与之对应，则称“X函数确定Y”或“Y函数依赖于X”。



函数 $y = f(x)$

函数依赖

- 函数依赖的定义

  设R(U)是属性集U上的关系模式。X、Y是U的子集。 r是R的任意一个具体关系，t , s 是r中任意两个元组。如果t[X] = s[X]，则t[Y] = s[Y]，则称“X函数确定Y”或“Y函数依赖于X”，记作：$X \rightarrow Y$。

- 函数依赖$X \rightarrow Y$也可定义为：

  对于X的每个具体值，Y有唯一的值与之对应，则称“X函数确定Y”或“Y函数依赖于X”。

UN

S#	CN	G	SDN	MN
S601	数据库	90	CS	张明
S602	数据库	86	CS	张明
S601	编译	85	CS	张明
S602	编译	86	CS	张明
S801	C++	78	IS	李立
S802	C++	80	IS	李立

X　Y

$SDN \rightarrow MN$

函数依赖

- 函数依赖的定义

设R(U)是属性集U上的关系模式。X、Y是U的子集。r是R的任意一个具体关系，t,s 是r中任意两个元组。如果t[X] = s[X]，则t[Y] = s[Y]，则称“X函数确定Y”或“Y函数依赖于X”，记作：$X \rightarrow Y$。

- 函数依赖$X \rightarrow Y$也可定义为：

对于X的每个具体值，Y有唯一的值与之对应，则称“X函数确定Y”或“Y函数依赖于X”。

UN

S# (X)	CN	G	SDN (Y)	MN
S601	数据库	90	CS	张明
S602	数据库	86	CS	张明
S601	编译	85	CS	张明
S602	编译	86	CS	张明
S801	C++	78	IS	李立
S802	C++	80	IS	李立

$SDN \rightarrow MN$

$S\# \rightarrow SDN$

$S\# \rightarrow MN$

函数依赖

- 函数依赖的定义

设R(U)是属性集U上的关系模式。X、Y是U的子集。 r是R的任意一个具体关系，t , s 是r中任意两个元组。如果t[X] = s[X]，则t[Y] = s[Y]，则称“X函数确定Y”或“Y函数依赖于X”，记作：$X \rightarrow Y$。

- 函数依赖$X \rightarrow Y$也可定义为：

对于X的每个具体值，Y有唯一的值与之对应，则称“X函数确定Y”或“Y函数依赖于X”。

UN

S# (X)	CN (X)	G (Y)	SDN	MN
S601	数据库	90	CS	张明
S602	数据库	86	CS	张明
S601	编译	85	CS	张明
S602	编译	86	CS	张明
S801	C++	78	IS	李立
S802	C++	80	IS	李立

$SDN \rightarrow MN$

$S\# \rightarrow SDN$

$S\# \rightarrow MN$

$(S\#,CN) \rightarrow G$

函数依赖相关术语

- **平凡与非平凡的函数依赖**

  ——对于函数依赖X→Y，若Y⊆X，则称X→Y是平凡的函数依赖；若Y⊈X，则称X→Y是非平凡的函数依赖。

- **决定因素**

——对于函数依赖X→Y，则X叫做决定因素。

函数依赖的进一步说明

- 函数依赖是语义范畴的概念
  - 只能根据语义来确定一个函数依赖，而不能形式化证明一个函数依赖成立
- 函数依赖是不随时间而变的
  - 若关系R具有函数依赖X→Y，那么虽然关系R随时间而变化，但X→Y不变

S

S#	SN	SA	SD
s80601	李勇	20	CS
s80201	刘晨	19	IS
s80305	王敏	18	MA
s80202	张立	19	IS

S’

S#	SN	SA	SD
s90601	王姗	19	CS
s90201	章文	19	IS
s90301	李峰	20	MA
s90202	王子涵	18	IS

SN→S#

函数依赖与属性间的联系类型

- 1:1（一对一）联系，如：学生的学号与身份证号
  - 若X与Y是1:1，则
    X→Y，Y→X;
- 1:m（一对多）联系，如:学生所在系的系名与学号
  - X与Y是1: m，则
    只存在Y → X
- n:m （多对多）联系， 如：学号与课程名
  - 若X与Y是 n:m，则
    X与Y之间不存在函数依赖。

三种函数依赖

- **完全函数依赖与部分函数依赖**
  - 定义：在R(U)中，如果$X \rightarrow Y$，且对于任意X的真子集X'，都有$X' \nrightarrow Y$，则称Y对X完全函数依赖，记作$X \xrightarrow{f} Y$，否则称为Y对X部分函数依赖，记作$X \xrightarrow{p} Y$。

- **传递函数依赖**
  - 定义：在R(U)中，如果$X \rightarrow Y$，$Y \rightarrow Z$，且$Y \nrightarrow X$，则称Z对X传递函数依赖，记作$X \xrightarrow{t} Z$。

三种函数依赖示例

例：UN（<u>S#，CN</u>，G，SDN，MN）

属性之间函数依赖关系：

$S\# \xrightarrow{f} SDN$；

$SDN \xrightarrow{f} MN$；

$(S\#，CN) \xrightarrow{f} G$；

$S\# \xrightarrow{t} MN$；

$(S\#，CN) \xrightarrow{p} SDN$；

现实世界的语义：

- 一个系有若干个学生，但一个学生只属于一个系。
- 一个系只有一名系主任。
- 一个学生可以选修多门课程，每门课程可有若干学生选修。
- 每个学生学习每门课程有一个成绩

关系键的形式定义

- 侯选码（键）与主码（键）
  - 定义：设K为R< U , F >中的属性或属性组合，若$K \xrightarrow{f} U$，则称K为R的**侯选码**。若侯选码多于一个，则选定其中的一个作为**主码**。
  - 码的性质：
    - **唯一性**：唯一地标识关系中的元组。
    - **最小性**：若抽去主码中的任意一属性，则主码将失去标识的唯一性。
- 主属性与非主属性：

  包含在任何一个侯选码中的属性，叫**主属性**。

  不包含在任何码中的属性称为**非主属性**。

关系键的形式定义

- 外部码
  - 定义：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的外码。
- 例：

SD（S#，SDN）

SG（S#，CN，G）

DM（SDN，MN）

函数依赖公理系统

- 主要内容
  - Armstrong公理及推论
  - 属性集的闭包
  - 函数依赖集的最小依赖集

UN(S#，CN，G，SDN，MN)

关系内部属性之间关系：
S# → SDN；
SDN → MN；
S# →MN；
(S#，CN)→G；
(S#，CN)→SDN；
(S#，CN)→MN

SD (S#，SDN)
SG (S#，CN，G)
DM (SDN，MN)

关系内部属性之间关系：
S# → SDN；
(S#，CN)→G；
SDN → MN；

函数依赖的逻辑蕴涵

- 定义：关系模式R< U, F >中，X、Y是R的属性集合，如果从F中的函数依赖能够推出X→Y，则称F逻辑蕴涵X→Y。
- 函数依赖集F的闭包
  - 定义：在关系模式R< U, F >中，为F所逻辑蕴涵的函数依赖的全体称作F的闭包，记作F^+。

Armstrong公理系统

- **Armstrong公理系统**

  对于R<U,F>，有如下规则：

  – A1自反律：若$Y \subseteq X \subseteq U$，则$X \to Y$为F所蕴含。

  – A2增广律：若$X \to Y$为F所蕴含，且$Z \subseteq U$则$XZ \to YZ$为F所蕴含。

  – A3传递律：若$X \to Y$，$Y \to Z$为F所蕴含，则$X \to Z$为F所蕴含。

Armstrong公理系统的正确性

对R<U, F>的任一关系r中任意两个元组t，s：

- 证明自反律：即若$Y \subseteq X$，则$X \rightarrow Y$。

$$\left.\begin{array}{l}\text{若 } \underline{t[X] = s[X]} \\ \text{且 } Y \subseteq X\end{array}\right\} \Longrightarrow \underline{t[Y] = s[Y]} \Longrightarrow X \rightarrow Y$$

- 证明增广律：即若$X \rightarrow Y$，则$XZ \rightarrow YZ$。

$$\text{若 } \underline{t[XZ] = s[XZ]} \Longrightarrow \left\{\begin{array}{l}\underline{t[Z] = s[Z]} \\ \left.\begin{array}{l}t[X] = s[X] \\ \text{又因 } X \rightarrow Y\end{array}\right\} \Longrightarrow \underline{t[Y] = s[Y]}\end{array}\right\}$$

$$\Longrightarrow \underline{t[YZ] = s[YZ]} \Longrightarrow XZ \rightarrow YZ$$

Armstrong公理系统的正确性

对R<U, F>的任一关系r中任意两个元组t，s：

- **证明传递律：**即若$X \rightarrow Y$，$Y \rightarrow Z$，则$X \rightarrow Z$。

$$\left.\begin{array}{l}\text{若 } t[X] = s[X] \\ \text{由于 } X \rightarrow Y\end{array}\right\} \Longrightarrow \left.\begin{array}{l} t[Y] = s[Y] \\ \text{又因 } Y \rightarrow Z\end{array}\right\} \Longrightarrow t[Z] = s[Z]$$

$$\Longrightarrow X \rightarrow Z$$

Armstrong公理的推论

- 由Armstrong公理系统得到的三条推理规则
  - 合并规则：由$X \rightarrow Y$，$X \rightarrow Z$，有$X \rightarrow YZ$。
  - 伪传递规则：由$X \rightarrow Y$，$WY \rightarrow Z$，有$WX \rightarrow Z$。
  - 分解规则：由$X \rightarrow Y$及$Z \subseteq Y$，有$X \rightarrow Z$。

- 从合并规则和分解规则得出如下定理：

定理1：

$$X \rightarrow A_1 A_2 \ldots A_k \text{成立} \Leftrightarrow X \rightarrow A_i \text{成立} (i=1, 2, \ldots, k)$$

属性集的闭包

- 属性集X关于函数依赖集F的闭包
  - 定义：在R<U，F>中，$X \subseteq U$，

    $X_F^+ = \{A \mid X \to A$ 能由F根据Armstrong公理导出$\}$

    称 X_F^+ 为属性集X关于函数依赖集F的闭包。

- 定理：

  $X \to Y$ 能够由F根据Armstrong公理导出

  $\Leftrightarrow Y \subseteq X_F^+$

闭包的计算

- 求 X_F^+ 的算法

Input：X，F

Output： X_F^+

X_F^+ **:= X;**

do

for any A $\subseteq X_F^+$ **do**

if 在F中存在函数依赖 A→B

then $X_F^+ = X_F^+ \cup$ **B**

while (X_F^+ **发生变化且** $X_F^+ \neq$ **U)**

示例（1）

- 求 X_F^+ 的示例1：

  R< U, F >, U = (A, B, C, D, E), F = {AB→C, B→D, C→E, CE→B, AC→B},计算$(AB)_F^+$ 。

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

$(AB)_F^+$ = ABCDE

示例（2）

- 求 X_F^+ 的示例2：

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

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

$(AG)_F^+ = AGBCH\ I$

示例（3）

- **示例3**

  $R\langle U, F\rangle$, $U = (A, B, C, D, E, G)$, $F = \{A \to E, BE \to AG, CE \to A, G \to D\}$，计算 $(AB)_F^+$。

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

$$(AB)_F^+ = ABEGD$$

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：指由F出发根据Armstrong公理推导出来的每个函数依赖一定在F所蕴含的函数依赖的全体之中。
  - 完备性：F所蕴含的函数依赖的全体中的每一个函数依赖，必定可以由F根据Armstrong公理导出。
- Armstrong公理系统的有效性由Armstrong公理系统的正确性得到证明，需要进一步证明Armstrong公理系统的完备性。

Armstrong公理完备性证明

- Armstrong公理完备性的证明
  - 公理的完备性：F所蕴含的函数依赖全体（F^+）中的每一个函数依赖，必定可以由F根据Armstrong公理导出
  - 证明逆否命题：若X→Y不能用Armstrong公理从F中导出，那么它必然不被F逻辑蕴涵。

  或者说，对于R<U,F>, 存在一个具体关系r，F中所有的函数依赖都满足r，而不能用公理推出的X→Y不满足r，即X→Y不被F逻辑蕴涵。

  - 设X→Y不能用Armstrong公理导出，并建立关系r：

	X_F^+	$U - X_F^+$
t	11…1	00 …0
s	11…1	11 …1

公理完备性需证明：

（1）在r中F的所有函数依赖都成立；

（2）在r中，X→Y不能成立。

Armstrong公理完备性证明

	X_F^+	$U - X_F^+$
t	11…1	00 …0
s	11…1	11…1

（1）证明在r中，F的所有函数依赖都成立。

设V→W是F中任一函数依赖，则有下列两种情况：

a) $V \subseteq X_F^+$。因为$V \subseteq X_F^+$，所以有X→V；由于V→W，于是X →W成立，所以$W \subseteq X_F^+$。因为r中X_F^+的值全相等，所以V→W在r上成立。

b) $V \not\subseteq X_F^+$。如果V不完全属于X_F^+，则V在两元组t和s上的属性值必不相等，则V→W在r上成立。

因此，在关系r中，F的任一函数依赖都成立。

Armstrong公理完备性证明

	X_F^+	$U - X_F^+$
t	11…1	00 …0
s	11…1	11…1

（2）在r中， X→Y不能成立。

因为X→Y不能用公理从F推出，则$Y \nsubseteq X_F^+$，而 $X \subseteq X_F^+$，那么r中元组t, s在X上的值相等，而在Y上的值不等，则X→Y在r上不成立，即X→Y不被F逻辑蕴涵。

结论：凡不能用公理推出的函数依赖都不被F逻辑蕴涵，即凡是F逻辑蕴涵的函数依赖都能用Armstrong公理从F导出。

—Armstrong公理是完备的。

函数依赖集等价与覆盖

- 函数依赖集等价
  - 函数依赖集F，G，若$F^+ = G^+$，则称F与G等价。
  - 如果F和G等价，则称F覆盖G，同时G也覆盖F。
  - $F^+ = G^+ \Leftrightarrow F \subseteq G^+$，$G \subseteq F^+$

函数依赖集的最小依赖集

- 最小依赖集
  - 定义：若函数依赖集F满足下列条件，则称F为一个极小函数依赖集，也称为最小依赖集或最小覆盖：
    - F中任一函数依赖$X \to A$，A必是单属性。（右部单属性化）
    - F中不存在这样的函数依赖$X \to A$，使得F与$F - \{X \to A\}$等价。（没有多余的FD）
    - F中不存在这样的函数依赖$X \to A$，在X中有真子集Z，使得F与$F - \{X \to A\} \cup \{Z \to A\}$等价。（每个FD左部没有多余属性）

函数依赖集F的极小化处理

- 函数依赖集F的极小化处理
  - 定理：每个函数依赖集F均等价于一个极小函数依赖集Fm，此Fm为F的最小依赖集。
  - F的极小化算法：
    - 逐个检查F中各函数依赖FD_i：$X\rightarrow Y$，若$Y=A_1A_2\ldots A_k$，$k\geq 2$，则用$\{X\rightarrow A_i|i=1,2,\ldots k\}$代替$X\rightarrow Y$。
    - 逐个检查F中各函数依赖$X\rightarrow A$，设$X = B_1\ldots B_m$，逐个考查B_i，若$A \in (X-B_i)_F^+$，则以（$X - B_i$）取代X。直到F不再改变。
    - 逐个检查F中各函数依赖$X\rightarrow A$，令$G = F-\{X\rightarrow A\}$，若$A\in (X)_G^+$，则从F中去掉该函数依赖，直到F不再改变。

示例

- 示例1

  $F = \{A \to B，B \to A，A \to C，B \to C\}$，求Fm。

  $Fm = \{A \to B，B \to A，B \to C\}$

  或者

  $Fm = \{A \to B，B \to A，A \to C\}$

- 示例2

  $F = \{C \to A，A \to G，CG \to B，B \to A\}$，求Fm。

  $Fm = \{A \to G，C \to B，B \to A\}$

范式的概念（1）

- 如果一个关系满足某个指定的约束集，则称它属于某种特定的**范式（Normal Form）**；
- 满足最低要求约束的称为**第一范式**，简称1NF,

  当一个关系只包含**原子值**这一约束时，称为1NF。原子值即为二维表的每一行和列的交叉位置上总是精确地存在一个值，而不是值集。也就是不能“表中有表”；
- 满足“原子值”这一约束条件的关系称为**规范化关系**，简称**范式**。在关系数据库中，都是规范化的关系。

S

S#	SN	SA	SD
s80601	李勇	20	CS
s80201	刘晨	19	IS
s80305	王敏	18	MA
s80202	张立	19	IS

范式的概念（1）

- 范式理论的发展过程：
  - 1971- 1972 CODD系统提出1NF，2NF，3NF的概念，讨论了进一步规范化的问题。
  - 1974- CODD 和BOYCE提出BCNF。
  - 1976- FAGIN 提出4NF，后来又提出了“投影-连接范式”PJNF，也称5NF。
- 各级范式间的联系：

  1NF ⊃ 2NF ⊃ 3NF ⊃BCNF ⊃ 4NF ⊃ 5NF
- 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级范式的关系模式的集合，这一过程称作规范化。

2NF（1）

- 定义：若R∈1NF，且每个非主属性完全依赖于码，则称R∈2NF；
- 注意：
  - 如果关系R的全体属性都是R的主属性，那么R ∈2NF；
  - 从1NF中消除非主属性对码的部分函数依赖，则可获得2NF关系；
  - 在2NF中，允许主属性部分函数依赖于码。

2NF（2）

- 2NF的规范化
  - 把1NF关系模式规范提高到2NF关系模式的集合。

例：关系 UN（S#，CN，G，SDN，MN）∈1NF，其属性之间的函数依赖关系，用函数依赖图表示：

UN属性之间函数依赖关系：

S# → SDN；

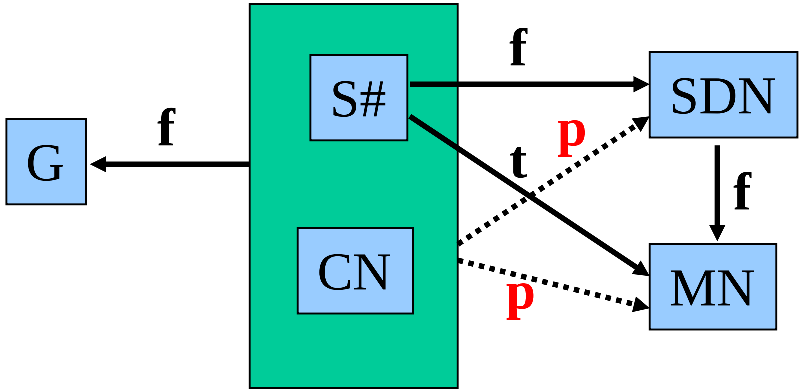
SDN → MN；

S# → MN；

(S#，CN) → G；

(S#，CN)→SDN；

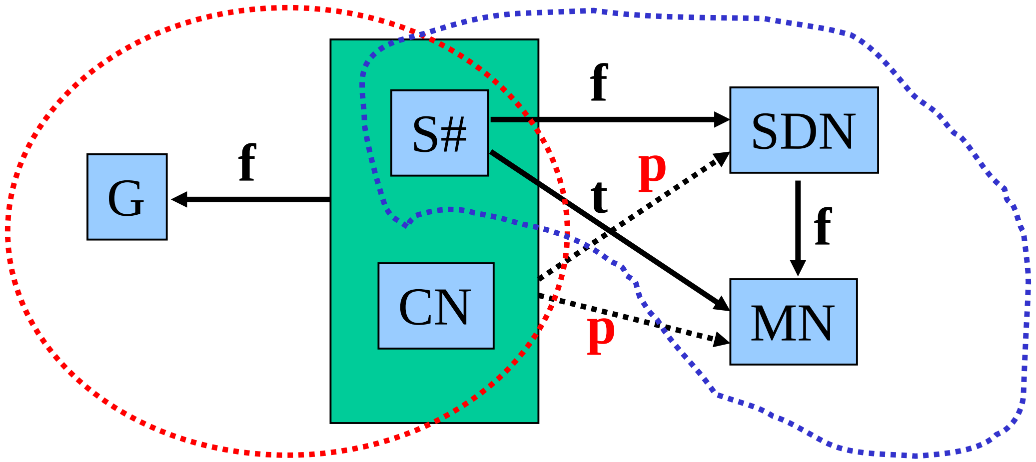
(S#，CN)→MN



所以UN∉2NF。

2NF（3）

- 采取**投影分解**方法，消除UN中的非主属性对码的部分函数依赖。



UN ➡ SG=UN[S#，CN，G] ∈2NF
SDM=UN[S#，SDN，MN] ∈2NF

2NF（4）

- UN（S#，CN，G，SDN，MN）∈1NF
- 2NF存在的弊病

  SG=UN[S#，CN，G] ∈2NF

  SDM=UN[S#，SDN，MN] ∈2NF

  – 插入异常有所改善，但还是存在：如果系中没有学生，则有关系的信息就无法插入。
  – 删除异常：如果删除学生的信息，所在系的信息也随之删除了。
  – 数据冗余得到一定改善：每个学生都存储了所在系的系主任的信息。

3NF（1）

- 定义：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$Z(Z\not\subset Y)$，使得下式成立，$X\rightarrow Y$，$Y\rightarrow Z$，$Y\not\rightarrow X$，则称$R\in 3NF$。

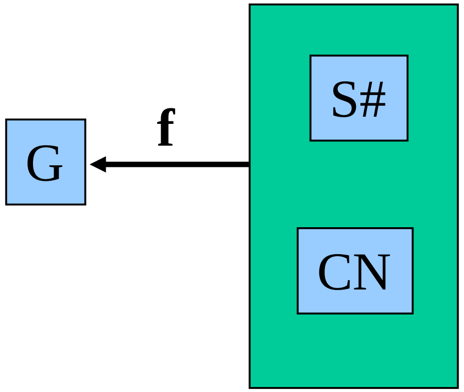
- 或定义为：

  若关系模式$R\in 2NF$，且每个非主属性都不传递依赖于R的任何码，则$R\in 3NF$。

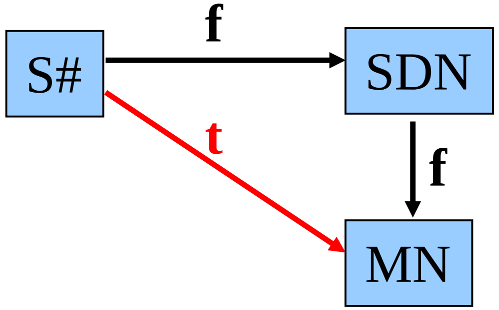
3NF（2）

- **3NF规范化**

UN ⟶ {
SG=UN[<u>S#，CN</u>，G] $\in$2NF
SDM=UN[<u>S#</u>，SDN，MN] $\in$2NF



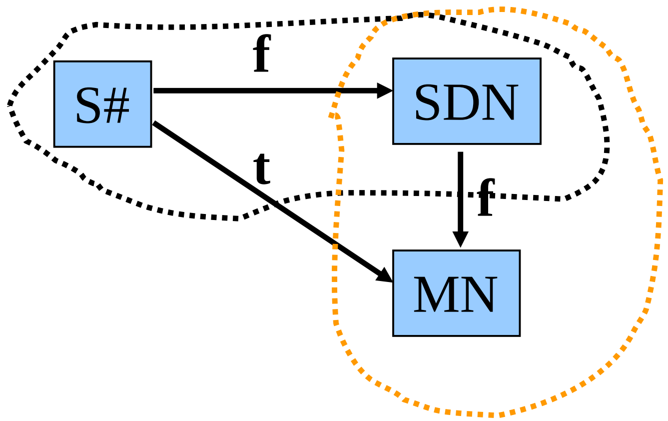
SG $\in$3NF



SDM $\notin$3NF

3NF（3）

- 采用投影分解的方法，将SDM规范到3NF。



SDM ⇒ SD=SDM[S#，SDN] ∈3NF
SDM=SDM[SDN，MN] ∈3NF

3NF（4）

- 所以有如下结果：

UN（S#，CN，G，SDN，MN） → SG（S#，CN，G）∈3NF
SD（S#，SDN）∈3NF
SDM（SDN，MN）∈3NF

•SG，SD，DM均是单个关系表示单个实体或联系，所以，所有的“异常”、“毛病”都消失了。

BCNF（1）

- 3NF的不完善
  - 3NF没有限制主属性对码的部分与传递函数依赖。如果发生这些依赖，仍可能存在插入异常、删除异常、修改异常。
  - 例：
    - STC(S, T, C)，S表示学生，T表示教师，C表示课程。

      每位老师只教授一门课,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，因此具有以下函数依赖：

      T→C，（S，C）→T

      （S，T），（S，C）为候选码。

    STC ∈3NF。

BCNF（2）

STC(S, T, C)，其中
S—学生，T—教师，C—课程

- STC中存在的弊病
  - 插入异常：如果没有学生选修某位老师的任课，则该老师担任课程的信息就无法插入。
  - 删除异常：删除学生选课信息，会删除掉老师的任课信息。
  - 数据冗余：每位学生都存储了有关老师所教授的课程的信息。
  - 更新异常：如果老师所教授的课程有所改动，则所有选修该老师课程的学生元组都要做改动。

BCNF（3）

- BCNF的定义：
  - 若关系模式R< U , F >∈1NF，如果对于R的每个函数依赖X→Y，且Y$\not\subseteq$X时，X必含有码，则R< U , F >∈BCNF。
  - 由BCNF的定义可以看到，每个BCNF的关系模式都具有如下三个性质：
    - 所有非主属性都完全函数依赖于每个候选码。
    - 所有主属性都完全函数依赖于每个不包含它的候选码。
    - 没有任何属性完全函数依赖于非码的任何一组属性。

BCNF（4）

- **BCNF的规范化**
  - STC（S，T，C），{T→C，（S，C）→T}，因为T → C，而T不是码。所以，STC $\notin$ BCNF。
  - 将S分解为TC（T，C），ST（S，T）。

多值依赖与第四范式

- 属性之间的函数依赖反映了现实世界实体一些特性之间的相互约束。
- 现实世界一些特性之间的还有其他类型的约束：
  - 多值依赖(Multivalue Dependency)
  - 连接依赖(Join Dependency)
  - 分层依赖(Hierarchical Dependency)
  - 相互依赖(Mutual Dependency)。

多值依赖的定义

- 定义：设R(U)是属性集U上的一个关系模式，X、Y、Z是U的子集，并且Z＝U－X－Y，关系模式R(U)中多值依赖X →→ Y成立，当且仅当对R(U)的任一关系r，给定的一对（x，z）值有一组Y的值，这组值仅仅决定于x值而与z值无关。
- 形式化定义：在R(U)的任一关系r中，如果存在元组t，s使得t[X]=s[X]，那么就必然存在元组w，v∈r，（w，v可以与s，t相同），使得：

  w[X] = s[X] = v[X] = t[X]

  w[Y] = t[Y]，v[Y] = s[Y]

  w[Z] = s[Z]， v[Z] = t[Z]

  则称Y多值依赖与X，记作X →→ Y。

多值依赖与函数依赖的比较

- 有效性范围
  - $X \rightarrow Y$的有效性仅决定于X、Y属性集上的值，它在任何属性集W（$XY \subseteq W \subseteq U$）上都成立；
  - $X \rightarrow\rightarrow Y$在U上成立，则 $X \rightarrow\rightarrow Y$在属性集W（$XY \subseteq W \subseteq U$）上成立；
  - $X \rightarrow\rightarrow Y$在属性集W（$XY \subseteq W \subseteq U$）上成立，但在U上不一定成立；

  - 若$X \rightarrow Y$在R(U)上成立，则对于任何$Y' \subseteq Y$，均有$X \rightarrow Y'$成立；
  - 若$X \rightarrow\rightarrow Y$在R(U)上成立，$Y' \subseteq Y$，则不一定有$X \rightarrow\rightarrow Y'$成立。

示例（1）

- 关系模式TEACH（C，T，B），C,T,B分别表示课程、教师和参考书。一门课程由多个教师担任，每个教师可以讲授多门课程；一门课程使用相同的一套参考书，每种参考书可被多门课程使用。

  它的码是（C，T，B），所以属于BCNF。

C	T	B
物理	{张明，张平}	{普通物理学，光学原理}
化学	{李勇，王微}	{无机化学，有机化学}

T多值依赖于C，记作

C →→ T，

同样有C →→ B

C	T	B
物理	张明	普通物理学
物理	张明	光学原理
物理	张平	普通物理学
物理	张平	光学原理
化学	李勇	无机化学
化学	李勇	有机化学
化学	王微	无机化学
化学	王微	有机化学

多值依赖的性质

- 多值依赖有对称性
  - 若X→→Y，则X→→Z，其中Z=U-X-Y
- 若X→Y,则X→→Y。即函数依赖可以看作多值依赖的特殊情况。
- 若X→→Y，而Z=∅，则称X→→Y为平凡的多值依赖，否则，

  若X→→Y，而Z≠∅，则称X→→Y为非平凡的多值依赖。
- 若X→→Y，X→→Z，则 X→→YZ。
- 若X→→Y，X→→Z，则 X→→Y∩Z。
- 若X→→Y，X→→Z，则 X→→Y-Z，X→→Z-Y 。

4NF的定义

- 定义
  - 关系模式R<U,F>∈1NF，如果对于R的每个非平凡的多值依赖X→→Y（Y⊈X），X都含有码，则称R∈4NF。
  - 4NF所允许的非平凡的多值依赖实际上是函数依赖（左部含有码的）。4NF就是限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。

- 或定义
  - 关系模式R∈BCNF，且不存在非平凡的非函数依赖的多值依赖，则R∈4NF。
  - 含义：若R∈BCNF，当R中只存在函数依赖，则R∈4NF；或当R中存在平凡的多值依赖时，R∈4NF。

4NF的规范化

C	T	B
物理	{张明，张平}	{普通物理学，光学原理}
化学	{李勇，王微}	{无机化学，有机化学}

C	T	B
物理	张明	普通物理学
物理	张明	光学原理
物理	张平	普通物理学
物理	张平	光学原理
化学	李勇	无机化学
化学	李勇	有机化学
化学	王微	无机化学
化学	王微	有机化学

- 非4NF的关系存在的弊病是数据冗余太大
  - 例： TEACH（C，T，B），由于C →→ T，C →→ B，码为（C，T，B）。所以TEACH ∉4NF。
  - 如果一门课有m个教师，n本参考书，则同一门课将有m×n个元组。
- 采用模式分解的方法消去非平凡且非函数依赖的多值依赖
  - 例：将CTB分解为CT（C，T），CB（C，B）。在CT、CB中，CT ∈4NF， CB∈4NF。
  - 如果一门课有m个教师，n本参考书，则同一门课将有m+n个元组。

规范化目的与基本思想

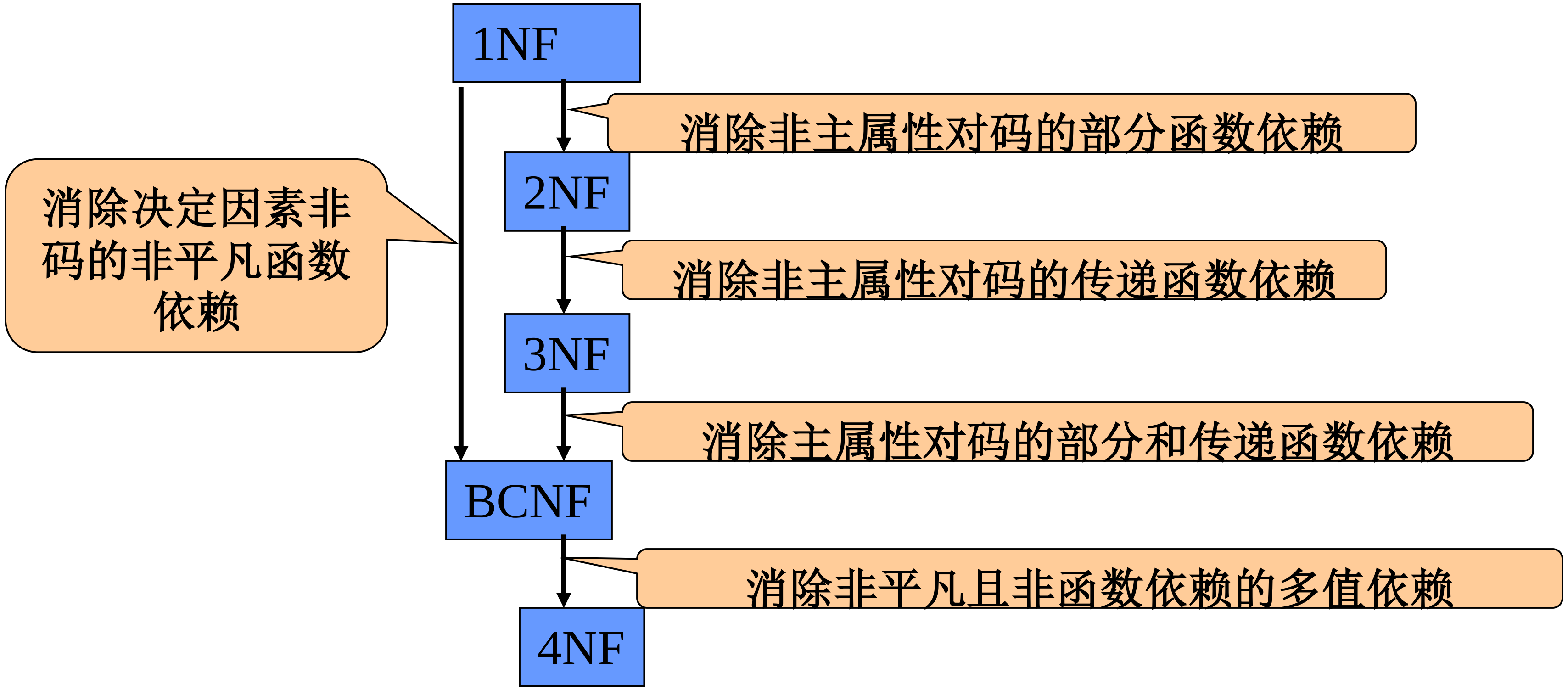
- 在关系数据库中，对关系的最基本要求是满足第一范式。这些关系常有一些异常或冗余等弊病。规范化的目的就是要消除这些弊病。
- 规范化的基本思想是逐步消除数据依赖中不合适的部分，使数据库模式中各关系模式达到某种程度的“分离”，使一个关系只描述一个实体或者实体间的一种联系。即“一事一地”的设计原则。规范化的实质是概念的单一化。

UN（S#，CN，G，SDN，MN）∈1NF ⟶

- SG（S#，CN，G）∈3NF
- SD（S#，SDN）∈3NF
- SDM（SDN，MN）∈3NF

规范化的过程

- 规范化的过程概括如下：



范式之间的关系（1）

- $3NF \subset 2NF$

反证：设$R \in 3NF$，但$R \notin 2NF$。则按2NF定义，一定有非主属性部分依赖于码，

设X为R的码，则存在X的真子集X′，以及非主属性Z（$Z \not\subset X'$），使得$X' \to Z$。

于是在R中存在码X，属性组X′，以及非主属性Z（$Z \not\subset X'$），使得$X \to X'$，$X' \to Z$，$X' \not\to X$成立，这与$R \in 3NF$矛盾。所以$R \in 2NF$。

范式之间的关系（2）

- $BCNF \subset 3NF$

反证：设$R \in BCNF$， 但$R \notin 3NF$。则按3NF定义，一定有非主属性对码的传递函数依赖，于是存在：

R的码X， 属性组Y，以及非主属性Z（$Z \not\subset Y$），使得$X \rightarrow Y$， $Y \rightarrow Z$，$Y \not\rightarrow X$成立 。

由$Y \rightarrow Z$，按BCNF定义，Y含有码，于是$Y \rightarrow X$成立，这与$Y \not\rightarrow X$矛盾。 所以$R \in 3NF$。

- $4NF \subset BCNF$

模式分解理论

- 模式分解的定义
- 分解的无损连接性
- 分解的保持函数依赖性
- 模式分解的原则
- 模式分解的算法

模式分解的定义

- 关系模式分解
  - 函数依赖集合$F_i = \{X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq U_i\}$，称$F_i$为F在$U_i$上的投影。
  - 关系模式R<U , F>的一个分解ρ是指

$$\rho = \{R_1<U_1, F_1>, R_2<U_2, F_2>, \ldots, R_n<U_n, F_n>\}$$

其中$U = \bigcup_{i=1}^{n} U_i$，并且没有$U_i \subseteq U_j$，$1 \le i, j \le n$，$F_i$是F在$U_i$上的投影。

例：UN(S#，CN，G，SDN，MN)的一个分解：

$$\begin{aligned}\rho = \{ & SG<(S\#, CN, G), \{(S\#,CN) \rightarrow G\}>, \\ & SD<(S\#, SDN), \{S\# \rightarrow SDN\}>, \\ & SDM<(SDN, MN), \{SDN \rightarrow MN\}> \}\end{aligned}$$

分解的无损连接性

- 设$\rho = \{R_1\langle U_1, F_1\rangle, R_2\langle U_2, F_2\rangle, \ldots, R_k\langle U_k, F_k\rangle\}$是$R\langle U, F\rangle$的一个分解，r是$R\langle U, F\rangle$的一个关系，定义$m_\rho(r) = \bowtie_{i=1}^{k} \Pi_{R_i}(r)$，其中$\Pi_{R_i}(r)=\{t[U_i] \mid t \in r\}$，即$m_\rho(r)$是r在ρ中各关系模式投影上的连接。若对于$R\langle U, F\rangle$的任何一个关系r，都有$r = m_\rho(r)$，则称分解ρ具有无损连接性，简称ρ为无损分解。

无损分解的判定算法（1）

- **算法：（判别一个分解的无损连接性）**

  $\rho = \{R_1<U_1, F_1>, R_2<U_2, F_2>, \ldots, R_k<U_k, F_k>\}$, $U=\{A_1, A_2, \ldots, A_n\}$, $F=\{FD_1, FD_2, \ldots FD_p\}$, FD_i为 $X_i \rightarrow A_i$, $X_i \subseteq U$。

(1) 建立n列k行的表TB：

- 每一列对应一个属性A_i；
- 每一行对应分解中的一个关系模式R_i 。
- 分量的取值： $C_{ij} = \begin{cases} a_j, & A_j \in U_i \\ b_{ij}, & A_j \notin U_i \end{cases}$

无损分解的判定算法（2）

(2)对FD_i中每一个函数依赖$X \to Y$，若TB中存在元组t_1，t_2，使得 $t_1[X]=t_2[X]$，则对每一个$A_i \in Y$：

①若$t_1[A_i]$，$t_2[A_i]$中有一个等于a_i，则另一个也改为a_i；

②若①不成立，则取$t_1[A_i] = t_2[A_i]$（t_1的行号小于t_2）。

(3)反复执行(2)，直至：

①TB中出现一行为$a_1, a_2, \ldots, a_n$的一行。

② TB不再发生变化，且没有一行为$a_1, \ldots, a_n$。

在①情况下，ρ为无损分解，否则为有损分解。

无损分解的判定算法（3）

- 例： U={A,B,C,D,E}, F={AB→C, C→D,D→E}

  ρ ={(A, B, C), (C, D), (D, E)}

A	B	C	D	E
a_1	a_2	a_3	b_{14}	b_{15}
b_{21}	b_{22}	a_3	a_4	b_{25}
b_{31}	b_{32}	b_{33}	a_4	a_5

AB→C　C→D

A	B	C	D	E
a_1	a_2	a_3	b_{14}	b_{15}
b_{21}	b_{22}	a_3	a_4	b_{25}
b_{31}	b_{32}	b_{33}	a_4	a_5

A	B	C	D	E
a_1	a_2	a_3	a_4	b_{15}
b_{21}	b_{22}	a_3	a_4	b_{25}
b_{31}	b_{32}	b_{33}	a_4	a_5

D→E

A	B	C	D	E
a_1	a_2	a_3	a_4	a_5
b_{21}	b_{22}	a_3	a_4	a_5
b_{31}	b_{32}	b_{33}	a_4	a_5

无损分解的判定准则

- 定理：R<U，F>的一个分解$\rho=\{R_1<U_1, F_1>, R_2<U_2, F_2>\}$具有无损连接性的充分必要条件是$U_1 \cap U_2 \rightarrow U_1 - U_2 \in F^+$ 或 $U_1 \cap U_2 \rightarrow U_2 - U_1 \in F^+$。
  - 即R_1，R_2的共同属性至少构成R_1、R_2 二者之一的候选码。

分解的保持函数依赖性

- 定义：若$F^+ = (\bigcup_{i=1}^{n} F_i)^+$，则称R< U , F >的分解ρ = {$R_1$<$U_1$, F_1> , … , R_n<U_n , F_n>}保持函数依赖。
- 保持函数依赖性的判定方法

  设$G= (\bigcup_{i=1}^{n} F_i)$，则

  $F^+ = G^+ \Leftrightarrow F \subseteq G^+$，且 $G \subseteq F^+$

  要判定$F \subseteq G^+$，只需逐一对F中函数依赖X→Y，考察Y是否属于X_G^+。若有F中的函数依赖不满足该条件，则$F^+ \neq G^+$，ρ未保持函数依赖。

  – R中的每个函数依赖都能够从R_1 … R_n函数依赖的并集中逻辑导出。

示例

- 例：$R=<ABC，\{A \to B, B \to C\}>$,

  $\rho=\{<AB, \{A \to B\}>，<AC, \{A \to C\}>\}$

则$G=\{A \to B，A \to C\}$，$G^+=\{A \to B，A \to C，A \to BC\}$, B关于G^+的闭包为（B），因为对于F中$B \to C$，$C \notin B$关于G^+的闭包，所以$F \not\subseteq G^+$，$F^+ \neq G^+$，ρ未保持函数依赖。ρ是无损分解。

模式分解的原则

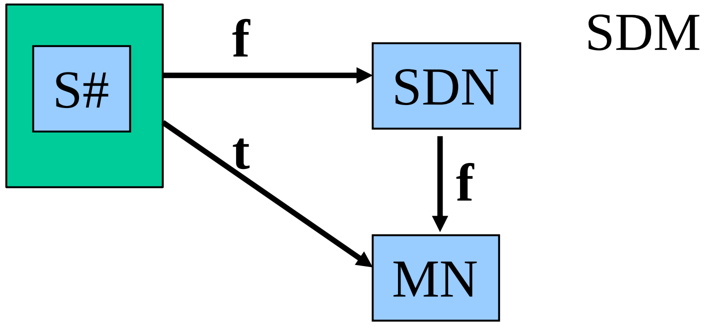
- 规范化中的问题
  - 规范化通过投影分解来完成。投影分解不是唯一的。并且结果大不相同。
  - 例如，对于SDM（S#，SDN，MN）到3NF的投影分解：

（1）SD（S#，SDN）
SM（S#，MN）
–具有无损连接性；
–未保持函数依赖

（2）SM（S#，MN）
DM（SDN，MN）
–不具有无损连接性；
–未保持函数依赖

（3）SD（S#，SDN）
DM（SDN，MN）
–具有无损连接性；
–保持函数依赖

- 投影分解中应遵循的原则：
  - 具有无损连接性
  - 保持函数依赖



模式分解能够达到的范式等级

- 模式分解能够达到的范式等级：
  - 若要求分解保持函数依赖，那么模式分解总可以达到3NF，但不一定能达到BCNF；
  - 若要求分解具有无损连接性，那一定可以达到4NF或更高；
  - 若要求分解既保持函数依赖，又具有无损连接性，可以达到3NF，但不一定能达到BCNF。

关系模式的分解算法

- 关系模式的分解算法
  - 达到3NF且保持函数依赖的分解算法
  - 达到3NF且同时保持无损连接与函数依赖的分解算法
  - 达到BCNF无损连接分解算法

达到3NF的等价模式分解（1）

- 达到3NF且保持函数依赖的分解算法：

  1.对F进行极小化处理，仍记为F。

  2.找出不在F中出现的属性，将它们构成一个关系模式，并从U中去掉它们(剩余属性仍记为U)。

  3.若有$X \rightarrow A \in F$，且$XA=U$，则$\rho=\{R\}$，算法终止。

  4.对F按具有相同左部的原则进行分组（设为k组），每一组函数依赖所涉及的属性全体为U_i，若有$U_i \subseteq U_j$ $(i \neq j)$,则去掉U_i。令F_i为F在U_i上的投影，则$\rho = \{R_1<U_1, F_1>, \ldots, R_k<U_k, F_k>\}$ 是$R<U, F>$的一个保持函数依赖的分解，并且每个$R_i<U_i, F_i> \in 3NF$。

示例

– 示例1：

U={S#，SDN，MN，C#，G}

F={S#→SDN，S#→MN，SDN→MN，(S#,C#)→G}

1. F_m={S#→SDN，SDN→MN，(S#,C#)→G}

2. 保持函数依赖的分解：ρ={

{(S#，SDN)，S#→SDN}

{(SDN，MN)，SDN→MN}

{(S#，C#，G)，(S#,C#)→G} }

分解具有无损连接性。

示例

– 示例2： R（ABC；A→C，B→C）

保持函数依赖分解：

ρ ={ {AC；A→C}，{BC；B→C}}。

分解是有损的。

达到3NF的等价模式分解（2）

- 达到3NF且同时保持无损连接与函数依赖的分解
  - 算法：

设$\rho = \{R_1<U_1, F_1>, \ldots, R_k<U_k, F_k>\}$ 是 $R<U, F>$的一个保持函数依赖的3NF分解。

设X为$R<U, F>$的码，

（1）若有某个U_i，$X \subseteq U_i$，则ρ即为所求，

（2）否则令$\tau = \rho \cup \{R^* <X, F_X>\}$，$\tau$即为所求。

示例

例1：求R（ABC；A→C，B→C）的保持无损连接和函数依赖的3NF分解。

（1）按保持函数依赖分解

进行分组， ρ={{AC；A→C}，{BC；B→C}}。

（2）码为AB

$\tau = \rho \cup \{AB\}$

最后的分解为：

{{AC；A→C}，{BC；B→C}，{AB}}

达到BCNF无损连接分解算法

- 算法：

给定关系模式R<U , F> ,

(1) 令$\rho = \{R<U , F>\}$

(2) 检查ρ中各关系模式是否属于BCNF，若是，则算法终止。

(3) 设ρ 中$R_i<U_i , F_i>$不属于BCNF，

则存在函数依赖$X \rightarrow A \in F_i^+$ （$A \notin X$），且X不是R_i的码，则XA是U_i的真子集，将R_i分解为$\sigma=\{S_1, S_2\}$，

其中$U_{S_1} = XA$， $U_{S_2} = U_i - \{A\}$

以σ代替R_i，返回到（2）。

示例

例1：有R〈U，F〉，其中U={S#，SD，MN，C#，G}，

F={S#→SD，S#→MN，SD→MN，(S#,C#)→G}，将R无损分解到BCNF。

（1）U_1={S#，SD} , F_1={S#→SD}

U_2={S#, MN, C#, G}, F_2={S#→MN, (S#,C#)→G}

（2）U_1 = {S#, SD}, F_1={S#→SD}

U_2 = {S#, MN}, F_2={S#→MN}

U_3 = {S#, C#, G}, F_3 = {(S#,C#)→G}

ρ={R1〈U1，F1〉，R2〈U2，F2〉，R3〈U3，F3〉},且R1，R2，R3均属于BCNF。

候选码的求解理论和算法

- 快速求解候选码的充分条件
- 左边为单属性的函数依赖集候选码成员的图论判定方法
- 多属性依赖集候选码求解法

候选码的求解理论和算法

对于关系R<U,F >，其属性可分为4类：

- **L类**：仅出现在F的函数依赖左部的属性。
- **R类**：仅出现在F的函数依赖右部的属性。
- **N类**：在F的函数依赖左右两边均未出现的属性。
- **LR类**：在F的函数依赖左右两边均出现的属性。

快速求解候选码

- 定理1：对于关系R<U,F>，若X（X $\subseteq$ U）是L类属性，则X必为R的任一候选码成员。
  - 推论1.1：对于关系R<U,F>，若X（X $\subseteq$ U）是L类属性，且 X_F^+ =U，则X必为R的唯一候选码。
- 定理2：对于关系R<U,F>，若X（X $\subseteq$ U）是R类属性，则X不在R的任何候选码中。
- 定理3：对于关系R<U,F>，若X（X $\subseteq$ U）是N类属性，则X必包含在R的任一候选码中。
  - 推论3.1：对于关系R<U,F>，若X（X $\subseteq$ U）是N类和L类组成的属性集，且 X_F^+ =U，则X为R的唯一候选码。

示例

- 例1：设有关系模式R（A，B，C，D），其函数依赖集F={D →B, B →D, AD →B,AC →D},求R的所有候选码。

- 例2：设有关系模式R（A，B，C，D，E，P），其函数依赖集F={A → D，E → D，D → B，BC →D，DC →A}，求R的所有候选码。

候选码的图论判定方法（1）

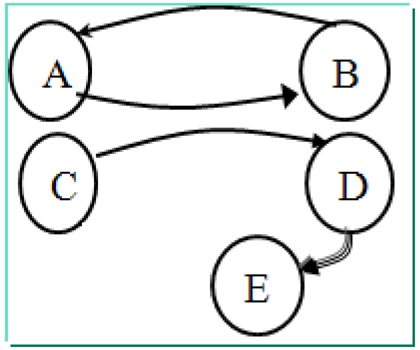
- 定义1：函数依赖关系图G是一个有序二元组（U，F），记作G＝（U，F），其中：

  (1)U=（A1，A2，...An)是有限非空集，Ai(i= 1,2, ...,n)是G的结点，它们表示对应关系模式R（A1，A2，...An)的属性。

  (2)F是G的边集，其元素是G的一条有向边，每条边(Ai,Aj)表示一个函数依赖 Ai → Aj，则F是R的单属性最小依赖集。

候选码的图论判定方法（2）

- 定义2：在一个函数依赖图中有如下术语：
  - 原始点：只有引出线而无引入线的结点，表示L类属性；
  - 终结点：只有引入线而无引出线的结点，表示R类属性；
  - 途中点：既有引出线又有引入线的结点，表示LR类属性；
  - 孤立点：既无引出线又无引入线的结点，表示N类属性；
  - 关键点：原始点和孤立点统称为关键点；
  - 关键属性：关键点对应的属性；
  - 独立回路：不能由其他结点到达的回路。
    - 回路：有向图G=(V,E)中,若边序列P=(e_{i1}, e_{i2}, …, e_{iq}), 如果e_{iq}的终点也是e_{i1}的始点，则称P是G的一条有向回路。



候选码的图论判定方法（3）

- 定理4：关系模式R的函数依赖图G中若存在关键点，则关键点对应的属性必在R的任何候选码中，而所有终结点必不在R的任何候选码中。
- 定理5：属性集X是R的唯一候选码的充要条件是X能到达G中的任一结点。
  – 推论5.1:在单属性情况下，R具有唯一候选码的充要条件是G中不存在独立的回路。

候选码的图论判定方法（4）

- **定理6**：设Y是途中点，则Y必在某个候选码中的充要条件是Y为某一独立回路中的结点。

- **定理7**：设F是单属性依赖集， X 是R的关键属性集，G中存k（k>=1）个独立回路$r_1,r_2,\ldots,r_k$，则：

  （1）R的候选码必不唯一；

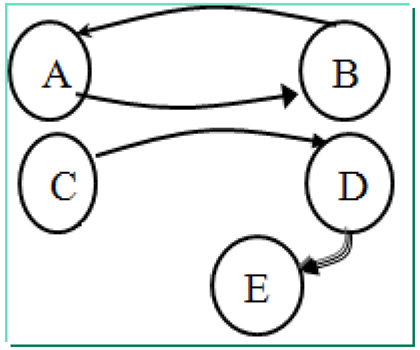
  （2）R的候选码均由两部分构成：

    – 关键属性集X（X可为空）；

    – k个独立回路结点集的笛卡儿积的任一元素；

  （3）候选码的个数等于k个独立回路中结点个数的乘积。

  （4）每个候选码所含属性个数是一个常数，等于关键属性个数加上独立回路个数k。



候选码的图论判定方法（5）

- **算法1：单属性依赖集候选码图论求解法**
  - 输入：关系模式R，R的单属性函数依赖集F。
  - 输出：R的所有候选码。

算法：

（1）求F的最小依赖集Fm；

（2）构造函数依赖图G；

（3）从G中找出关键属性集X（X可为空）；

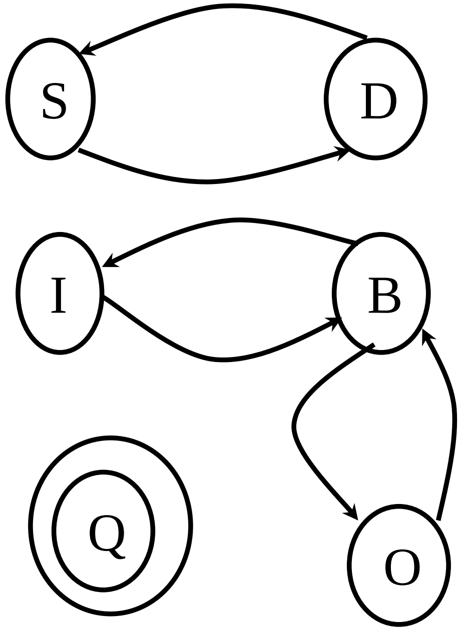
（4）查看G中有无独立回路，若无则输出X即为R的唯一候选码，结束；若有则继续（5）；

（5）从各独立回路中各取一结点对应的属性与X组成一个候选码，并重复这一过程，直至取尽所有可能组合，即为R的全部候选码。结束。

示例

- 例：设R（O，B，I，S，Q，D），F={S→D，D→S，I→B，B→I，B→O，O→B}，求R的所有候选码。

解：



– F_m=F；

– 构造函数依赖图；

–关键属性集{Q}；

–有四条回路，两条独立回路，
所以每个候选码有3个属性，
共有6个候选码；

–R的所有候选码为QSI，QSB，QSO，QDI，QDB，QDO。

多属性依赖集候选码求解法

- 算法2：多属性依赖集候选码求解法。

  输入：关系模式R<U,F>；

  输出：R的所有候选码。

  算法：

  (1) 将R的所有属性分为L、R、N和LR四类，并令X代表L、N两类，Y代表LR类。

  (2) 求X_F^+，若 X_F^+=U，则X即为R的唯一候选码，结束。否则，继续。

  (3) 对于Y中的任一属性A，求$(XA)_F^+$，若$(XA)_F^+$=U，则XA为一候选码；否则在Y中依次取两个、三个、…，求其属性闭包，直到其闭包包含R的全部属性。

小结

- 函数依赖
  - 定义，三种类型函数依赖，函数依赖的公理系统，函数依赖集的闭包，属性关于函数依赖集的闭包，最小函数依赖集
- 范式
  - 1NF,2NF,3NF,BCNF
- 多值依赖与第四范式
- 模式分解的理论
  - 模式分解遵循的原则，到3NF和BCNF分解算法
- 候选码的求解理论和算法