浙江大学管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

分布式文件系统及数据库技术
第八讲

主讲人：曹仔科 or 彭希羡
浙江大学管理学院
数据科学与工程管理系

AACSB ACCREDITED
EFMD EQUIS ACCREDITED
ASSOCIATION AMBA ACCREDITED
CAMEA 中国高质量MBA教育认证
IQA CEEMAN Accredited

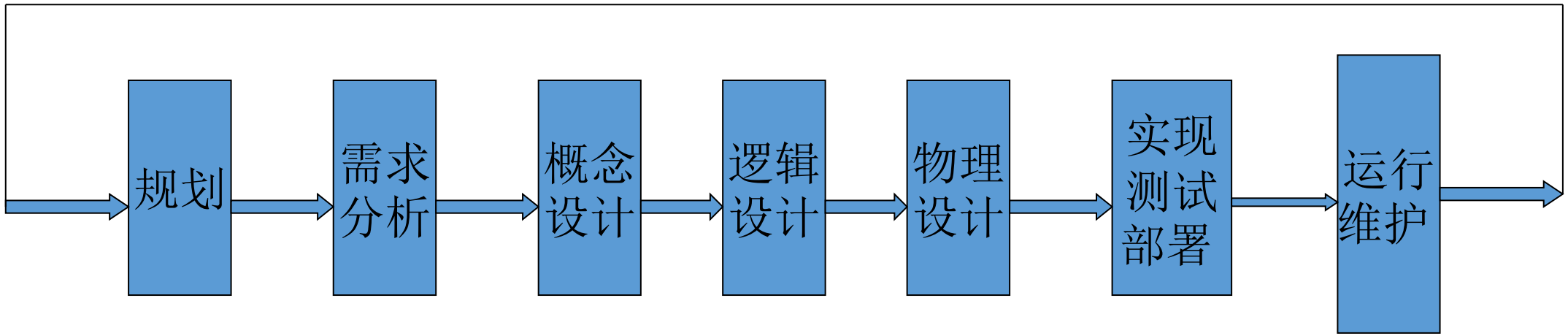
1. 问题的提出

2. 规范化

3. 模式的分解

4. 小结

数据库系统开发生命周期



设计阶段

浙江大学管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

这是整个生命周期中最关键的技术阶段，它将需求转化为一个详细的、可实现的数据库蓝图。通常分为三个子阶段：

（1）概念设计：

- **目标：** 创建一个高层次、技术中立的的数据模型，专注于数据本身及其关系，而不考虑具体的DBMS实现。
- 使用**实体-关系模型**（E-R model）。
- 产出**E-R图**，展示了实体、实体的属性以及实体之间的关系（1:1, 1:N, M:N）。

设计阶段

（2）逻辑设计

- **目标：**将概念模型映射到所选数据模型（最常用的是**关系模型**）的结构上。
  - 将E-R图转换为**关系模式**（即一系列的表结构）。
- 应用**规范化理论**（通常是第三范式3NF或巴斯-科德范式BCNF）来消除数据冗余和插入/更新/删除异常，确保数据完整性。
- 定义**主键、外键。**
- 产出完整的数据库逻辑模式（表结构清单），包括每个表的字段名、数据类型、主外键关系。

设计阶段

（3）物理设计：

- **目标：**为特定的 **数据库管理系统**（如MySQL，Oracle，SQL Server）设计底层的存储结构和访问机制，以优化性能。
- 决定文件的存储位置、分区策略。
- 设计**索引**（哪些列需要创建索引，是什么类型的索引）以加速查询。
- 确定磁盘存储结构和访问方法。

问题的提出

关系数据库逻辑设计

- 针对具体问题，如何构造一个适合于它的关系模式（即对一个表属性的设计）
- 重要的理论工具：关系数据库的规范化理论

例子

- 有三个属性（姓名, 级别, 工资）的工资关系模式。
  对应此模式建立的某个表如下所示：

姓名	级别	工资
A	9	110
B	9	110
C	6	35
D	7	50
E	8	80
F	8	80

存在的问题：

- 数据冗余
  - ✓浪费存储空间
- 更新异常
  - ✓ 某级别工资增加需要一一更新，容易造成数据的不一致
- 插入异常
  - ✓无法单独插入新的级别和工资信息
- 删除异常
  - ✓删除员工C就导致级别6的工资信息不存在了

解决方法

分解为两个关系模式表达： 职工级别（姓名，级别）,级别工资（级别, 工资）

姓名	级别
A	9
B	9
C	6
D	7
E	8
F	8

级别	工资
9	110
6	35
7	50
8	80

改进后，

✓数据无冗余

✓表达能力强

✓修改方便

数据依赖

- **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：　属性向域的映射集合

  F：　属性间数据的依赖关系集合

- 简化的关系模式三元组 R(U,F)

什么是数据依赖？

- 不同属性的值间的相关性
- 一个关系内部属性与属性之间的约束关系
- 这些约束关系是现实世界属性间相互联系的抽象

➢数据依赖的类型：

函数依赖 (FD)

多值依赖 (MVF)

数据依赖

- **建立一个描述学校教务的数据库：**

  核心的信息：学生的学号（Sno）、所在系（Sdept）、系主任姓名（Mname）、课程名（Cname）、该课程成绩（Grade）

  如果使用单一的关系模式： **Student （U、F)**

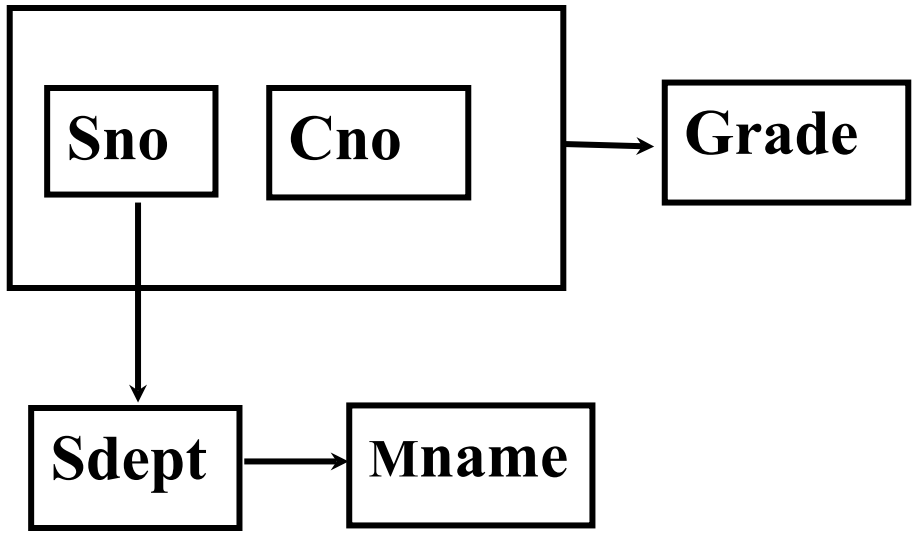
  U = { Sno, Sdept, Mname, Cno, Grade }

- **属性组U上的一组函数依赖F：**

  F = { Sno → Sdept,

  Sdept → Mname,

  (Sno, Cname) → Grade}

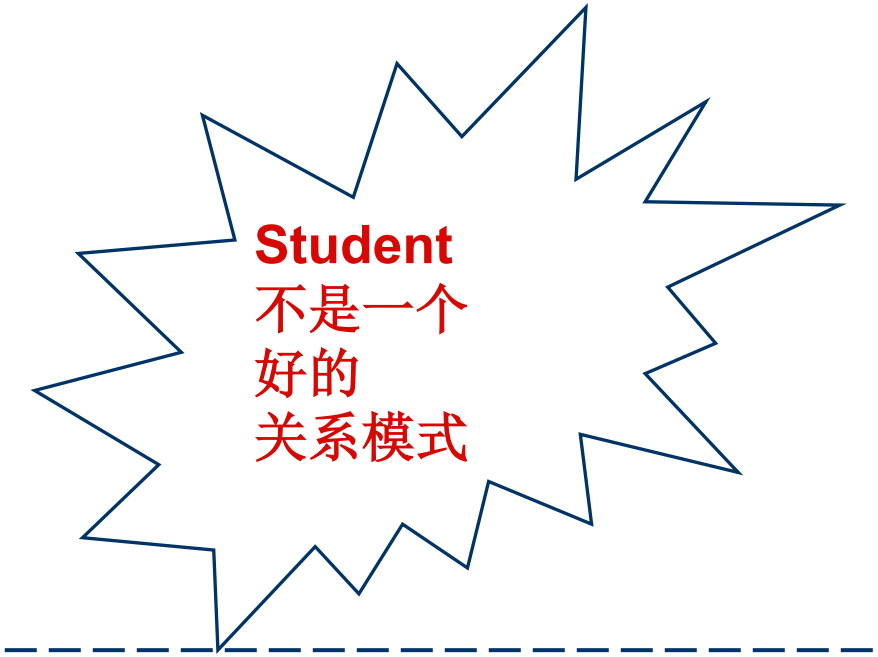


数据依赖

- 关系模式Student(U, F)中存在的问题
  - **数据冗余**：比如每一个系主任名字重复出现，重复次数与该系所有学生所有课程成绩出现次数相同
  - **更新异常**：比如某系更换系主任，必须修改每个有关学生的记录（容易出错，导致数据不一致）
  - **插入异常**：比如一个系刚成立，尚无学生，就可能无法把这个系及其系主任名字录入
  - **删除异常**：比如某个系学生全部毕业了，删除了所有学生信息后也会删除该系的信息

数据依赖

- 关系模式Student(U, F)中存在的问题
  - 数据冗余 (redundance)
  - 更新异常 (Update Anomalies)
  - 插入异常 (Insertion Anomalies)
  - 删除异常 (Deletion Anomalies)

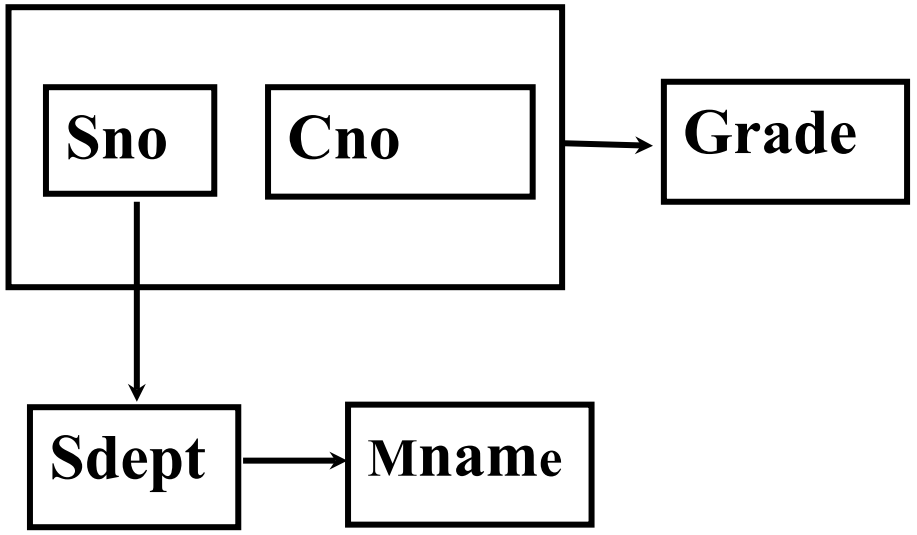


“好”的模式：

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

数据依赖

- 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



- **分解Student关系模式**

把这个单一关系模式分成3个关系模式：

S（Sno，Sdept）：Sno → Sdept

SC（Sno，Cno，Grade）：（Sno，Cno） → Grade

DEPT（Sdept，Mname）： Sdept→ Mname

规范化理论

规范化理论用来改善关系模式，具体是通过**投影分解**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。

函数依赖

码

范式

2NF

3NF

BCNF

多值依赖

4NF

规范化小结

函数依赖

- 定义8.1　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实例r，如果r中不存在两个元组在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→Y。

- 函数依赖说明：
  - 所有关系实例均要满足
  - 设计时可以强加约束以满足函数依赖

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→Y，但Y $\subseteq$ X，则称X→Y是平凡的函数依赖

若X→Y，但Y $\not\subseteq$ X，则称X→Y是非平凡的函数依赖

例：在关系SC(Sno, Cno, Grade)中，

非平凡函数依赖： (Sno, Cno) → Grade

平凡函数依赖： (Sno, Cno) → Sno 、(Sno, Cno) → Cno

- 我们接下来讨论的函数依赖都指非平凡的函数依赖
- 若X→Y，则X称为这个函数依赖的决定属性组，也称为决定因素（Determinant）。
- 若X→Y，Y→X，则记作X←→Y。
- 若Y不函数依赖于X，则记作X$\not\rightarrow$Y。

完全函数依赖与部分函数依赖

- 定义8.2　在R(U)中，
  - 如果X→Y，并且对于X的任何一个真子集X′，都有X′↛Y，则称Y对X完全函数依赖，记作$X \xrightarrow{F} Y$。
  - 若X→Y，但Y不完全函数依赖于X，则称Y对X部分函数依赖，记作$X \xrightarrow{P} Y$。

在关系SC(Sno, Cno, Grade)中：

(Sno, Cno)→Grade 是完全函数依赖，

但，(Sno, Cno)→Sdept是部分函数依赖，因为Sno → Sdept成立，且Sno是(Sno，Cno)的真子集

传递函数依赖

➲ 定义8.3 在R(U)中，如果X→Y, Y→Z，且Y $\not\subseteq$ X ,Y $\not\rightarrow$X, 则称Z对X传递函数依赖。

记为：$X \xrightarrow{传递} Z$

注: 如果Y→X， 即X←→Y，则Z直接依赖于X。

例: 在关系Std(Sno, Sdept, Mname)中，有：

Sno → Sdept，Sdept → Mname

Mname传递函数依赖于Sno

关系的码

- 关系中**不允许出现重复的元组**，因此可以指定一个或多个属性来唯一地标识关系中的每个元组，这样的一个或多个属性称为**超码**

- 本身为超码，但其任何子集都不再是超码则为**候选码**
  - 候选码为最小的超码

- **主码**：从候选码中被选来唯一标识关系中各元组

- **外码**：当一个关系中的某个属性或属性集合与另一个关系的候选码匹配时，就称这个属性或属性集合为外码

码

- 定义8.4 设K为R(U,F)中的属性或属性组合。

  若$K \xrightarrow{F} U$， 则K称为R的侯选码 （Candidate Key）。

  若候选码多于一个，则选定其中的一个做为主码（Primary Key）。

- 主属性与非主属性
  - 包含在任何一个候选码中的属性，称为主属性（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）
- 全码
  - 整个属性组是码，称为全码（All-key）

码

- 定义8.5 关系模式 *R* 中属性或属性组*X* 并非 *R*的候选码，但 *X* 是另一个关系模式的候选码，则称 *X* 是*R* 的外码（Foreign key）

  如在SC（Sno，Cno，Grade）中, Sno不是候选码，但Sno是关系模式S（Sno，Sdept，Sage）的候选码（主码），则Sno是关系模式SC的外码

- 主码与外码一起提供了表示关系间联系的手段

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
- 范式的种类及联系：

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

- 某一关系模式R为第n范式，可简记为R∈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

第一范式

- 1NF的定义

  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∈1NF

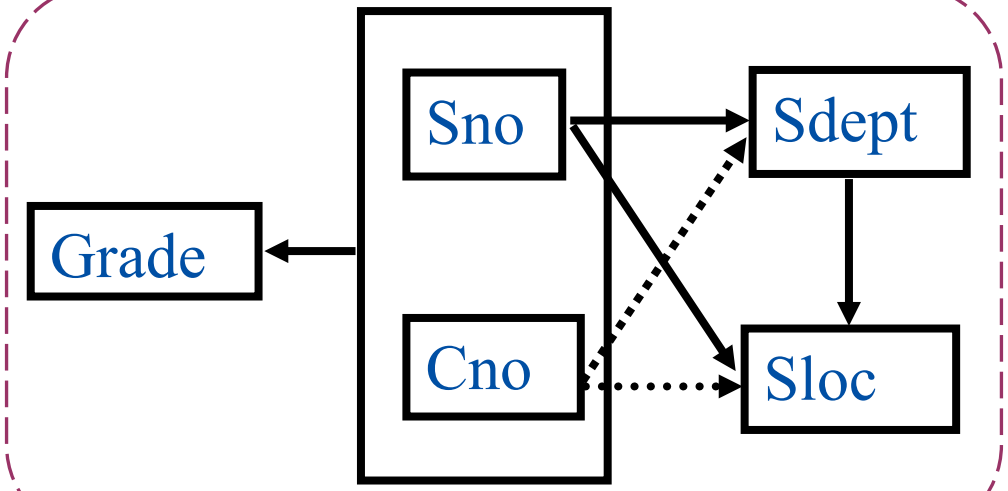
- 第一范式是对关系模式的最起码的要求。不满足第一范式的数据库模式不能称为关系数据库

- 但是满足第一范式的关系模式并不一定是一个好的关系模式

第一范式

[例4] 关系模式 S-L-C(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 → Sdept
- (Sno, Cno) $\xrightarrow{P}$ Sdept
- Sno → Sloc
- (Sno, Cno) $\xrightarrow{P}$ Sloc
- Sdept → Sloc

- S-L-C的码为(Sno, Cno)
- S-L-C满足第一范式。
- 非主属性Sdept和Sloc部分函数依赖于候选码(Sno, Cno)

第一范式不能解决的问题

(1) 插入异常

假设Sno = 95102，Sdept = IS，Sloc = N的学生还未选课，因课程号是主属性，因此该学生的信息无法插入SLC。

(2) 删除异常

假定某个学生本来只选修了3号课程这一门课。现在因身体不适，他连3号课程也不选修了。因课程号是主属性，此操作将导致该学生信息的整个元组都要删除。

(3) 数据冗余度大

如果一个学生选修了10门课程，那么他的Sdept和Sloc值就要重复存储了10次。

(4) 修改复杂

例如学生转系，在修改此学生元组的Sdept值的同时，还可能需要修改住处（Sloc）。如果这个学生选修了K门课，则必须无遗漏地修改K个元组中全部Sdept、Sloc信息。

第一范式

- S-L-C(Sno, Sdept, Sloc, Cno, Grade)不是一个好的关系模式
  - 插入异常
  - 删除异常
  - 数据冗余度大
  - 修改复杂

函数依赖包括：

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

Sno → Sdept

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

Sno → Sloc

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

Sdept → Sloc

原因：Sdept、Sloc部分函数依赖于候选码

第一范式

- 解决方法

S-L-C分解为两个关系模式，以消除这些部分函数依赖

SC（Sno， Cno， Grade）

S-L（Sno， Sdept， Sloc）

第二范式

浙江大学管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

- **2NF的定义**

  定义8.6　若R∈1NF，且每一个非主属性完全函数依赖于候选码，则R∈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。

例：S-L-C(Sno, Sdept, Sloc, Cno, Grade) ∈1NF

S-L-C(Sno, Sdept, Sloc, Cno, Grade) ∉ 2NF

SC（Sno， Cno， Grade） ∈ 2NF

S-L（Sno， Sdept， Sloc） ∈ 2NF

- 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。

第二范式不能解决的问题

(1) 插入异常

如果某个系因种种原因（例如刚刚成立），目前暂时没有在校学生，我们就无法把这个系的信息存入数据库。

(2) 删除异常

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，把这个系的信息也丢掉了。

(3) 数据冗余度大

每一个系的学生都住在同一个地方，关于系的住处的信息却重复出现，重复次数与该系学生人数相同。

(4) 修改复杂

当学校调整学生住处时，由于关于每个系的住处信息是重复存储的，修改时必须同时更新该系所有学生的Sloc属性值。

第三范式

- **3NF的定义**

  定义8.7 关系模式$R(U，F)$中若不存在这样的候选码X、属性组Y及非主属性Z（$Z \nsubseteq Y$），使得$X \rightarrow Y$ ($Y \not\rightarrow X$)，$Y \rightarrow Z$成立，则称$R(U，F) \in 3NF$。

  - 若$R \in 3NF$，则每一个非主属性既不部分依赖于候选码也不传递依赖于候选码。
  - 3NF在2NF的基础上进一步消除非主属性对候选码的传递函数依赖。

第三范式

例：2NF关系模式S-L(Sno, Sdept, Sloc)中

- 函数依赖：

  Sno→Sdept

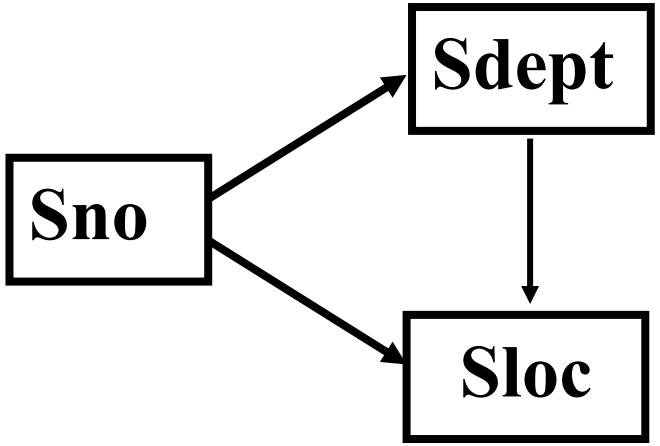
  Sdept ↛ Sno

  Sdept→Sloc

  可得：

Sno $\xrightarrow{\text{传递}}$ Sloc，即S-L中存在非主属性对码的传递函数依赖，S-L $\notin$ *3NF*

S-L函数依赖图：



第三范式

- 解决方法

采用投影分解法，把S-L分解为两个关系模式，以消除传递函数依赖：

S-D（Sno， Sdept）　　D-L（Sdept，Sloc）

- S-D的码为Sno， D-L的码为Sdept
- 分解后的关系模式S-D与D-L中不再存在传递依赖

- S-L(Sno, Sdept, Sloc) ∈ 2NF
  S-L(Sno, Sdept, Sloc) ∉ 3NF
  S-D(Sno，Sdept) ∈ 3NF
  D-L(Sdept， Sloc)∈ 3NF

第三范式

- 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，仍然有可能不能完全消除关系模式中的各种异常情况和数据冗余

第三范式不能解决的问题

例：在关系模式STC(S，T，C)中，S表示学生，T表示教师，C表示课程。

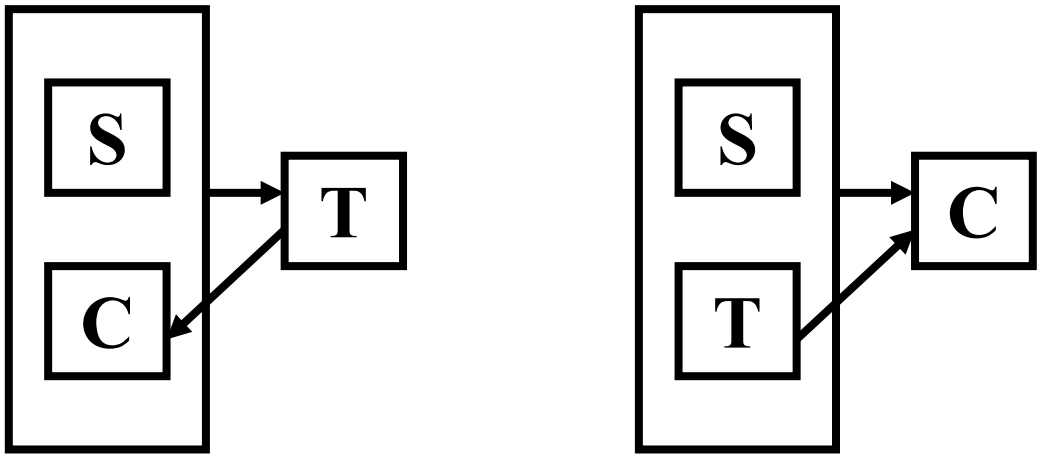
- 函数依赖：

  假设每一教师只教一门课，每门课由若干教师教，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。于是有如下函数依赖：

  (S，C)→T，(S，T)→C，T→C

- (S，C)和(S，T)都可以作为候选码。
- STC∈3NF（因为不存在非主属性）
- T→C，即T是决定属性集，

  但它既不是候选码，也不包含候选码。



STC

第三范式的问题

(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。

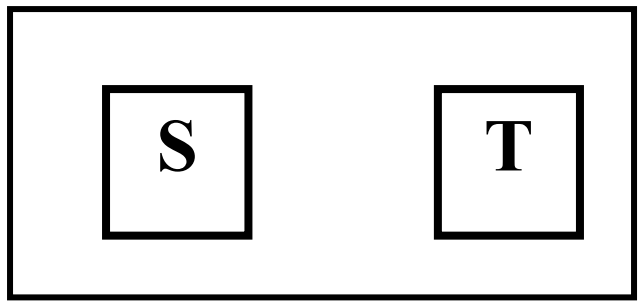
解决方法

- 原因：
  主属性C依赖于T，即主属性C部分依赖于候选码(S, T)。

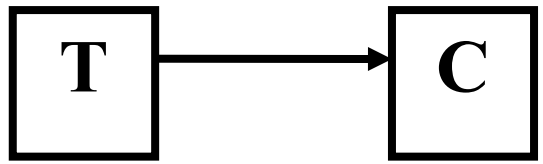
- 解决方法：
  采用投影分解法，将STC分解为二个关系模式：

  ST(S，T)

  TC(T，C)



ST



TC

BC范式

浙江大学 管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

- BCNF（Boyce Codd Normal Form）是由Boyce和Codd提出的，比3NF更进了一步。通常认为BCNF是修正的第三范式，所以有时也称为第三范式。
- BCNF的定义

  定义8.8 设关系模式R(U，F)∈1NF，如果对于R的每个函数依赖X→Y，若Y不属于X，则X必含有候选码，那么R∈BCNF。

  换句话说，在关系模式R(U，F)中，如果每一个决定属性集都包含候选码，则R∈BCNF。

- 例： STC(S，T，C)∈ 3NF 但不属于BCNF，因为T是决定属性，但不是（不包含）候选码
- ST(S，T)∈ BCNF
- TC(T，C)∈ BCNF

BC范式和第三范式

- 若R∈BCNF
  - 所有非主属性对每一个候选码都是完全函数依赖
  - 所有的主属性对每一个不包含它的候选码，也是完全函数依赖
  - 没有任何属性完全函数依赖于非候选码的任何一组属性

- R ∈BCNF ⇄ R ∈3NF（充分；不必要）

多值依赖

- 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。

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 … …
⋮	⋮	⋮

多值依赖

浙江大学 管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

❖ 单一关系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∈BCNF
- Teaching具有唯一候选码(C，T，B)， 即全码
- Teaching模式中仍然存在之前的问题：
  - 数据冗余度大
  - 插入异常
  - 删除异常
  - 修改异常

存在多值依赖

多值依赖

浙江大学 管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

➲ 定义8.9

设R(U)是一个属性集U上的一个关系模式， X、 Y和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*任一关系实例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

➲平凡多值依赖和非平凡的多值依赖

- 若X→→Y，而 Z＝φ（空集），则称X→→Y为平凡的多值依赖
- 否则称X→→Y为非平凡的多值依赖
- 需要关注的是非平凡的多值依赖

多值依赖

例： Teaching(C,T,B) 存在非平凡的多值依赖C→→T及C→→B。

- 在Teach关系中，每个（C, B）上的值对应一组T值，而且这种对应与B无关。同样地，每个（C, T）上的值对应一组B值，而且这种对应与T无关。

- 在这个表中，存在这样一个事实：一门课程，对应着一组独立的教师值，也对应着一组独立的教材值。
  - 即：课程多值决定教师，同时课程也多值决定教材。并且，教师和教材之间是相互独立的。
- 它描述的是一个属性（课程）独立地决定另外两组属性（教师和教材）的“组合”情况。

多值依赖

- 函数依赖：“一对一”或“多对一” 的约束；知道了**X**，就能唯一确定一个**Y**。
- 多值依赖：“一对多”且独立” 的约束；知道了**X**，就能确定一组**Y**的值，并且这组**Y**的值与其他的属性**Z**完全独立。

- 想象一个“家庭-孩子-宠物”的关系：
  - 一个家庭有多个孩子（小明、小红）。
  - 一个家庭也养了多只宠物（小狗、小猫）。
  - 孩子和宠物是独立的（每个孩子都喜欢所有的宠物，而不是某个孩子只属于某只宠物）。
  - 这就构成了多值依赖：家庭 →→ 孩子 | 宠物。

第四范式

例： Teaching(C,T,B) $\notin$ 4NF， 因为存在非平凡的多值依赖C→→T，且C不是候选码

- 用投影分解法把Teaching分解为如下两个关系模式：

$$CT(C, T) \in 4NF$$

$$CB(C, B) \in 4NF$$

  - 参考书只需要在CB关系中存储一次。
  - 当某一课程增加一名任课教师时，只需要在CT关系中增加一个元组。
  - 某一门课要去掉一本参考书，只需要在CB关系中删除一个相应的元组。

第四范式

- 定义8.10　设R(U)是一个属性集U上的一个关系模式， X、 Y和Z是U的子集，并且Z＝U－X－Y，如果对于R的每个非平凡多值依赖X→→Y，X都含有候选码，则R∈4NF。

- 如果R ∈ 4NF， 则R ∈ BCNF

- 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。因为根据定义，对于每一个非平凡的多值依赖X→→Y（Y不属于X），而且X都含有侯选码，于是当然有X→Y，所以4NF所允许的是非平凡多值依赖实际上是决定属性集都包含候选码的函数依赖。

数据依赖

- 函数依赖和多值依赖是两种最重要的数据依赖。
- 如果只考虑函数依赖，则属于**BCNF**的关系模式已经很完美了。如果考虑多值依赖，则属于**4NF**的关系模式已经很完美了。
- 事实上，函数依赖是多值依赖的一种特殊情况。
- 数据依赖中还有一种连接依赖，多值依赖实际上又是连接依赖的一种特殊情况。
  - 这需要涉及**5NF**（本课程不再讨论）。

规范化小结

- 关系数据库的规范化理论是数据库逻辑设计的工具
- 目的：尽量消除插入、删除异常，修改复杂，数据冗余
- 基本思想：逐步消除数据依赖中不合适的部分
  - 实质：概念的单一化

规范化的步骤

- 关系模式规范化的基本步骤

消除决定属性集非码的非平凡函数依赖

1NF
↓ 消除非主属性对码的部分函数依赖
2NF
↓ 消除非主属性对码的传递函数依赖
3NF
↓ 消除主属性对码的部分和传递函数依赖
BCNF
↓ 消除非平凡且非函数依赖的多值依赖
4NF

规范化小结

- 不能说规范化程度越高的关系模式就越好
- 在设计数据库模式结构时，必须对现实世界的实际情况和用户应用需求作进一步分析，确定一个合适的、能够反映现实世界的模式
- 上面的规范化步骤可以在其中任何一步终止

模式的分解

模式分解的标准

三种模式分解等价的定义：

- 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

模式的分解

例：SL(Sno,Sdept,Sloc)，

其中Sno是学号，Sdept是学生所在系，Sloc是学生住处，每个系的学生住在同一个地方。

F={Sno→Sdept, Sno→Sloc, Sdept→Sloc}

∵存在传递函数依赖，∴ SL∈2NF。

模式的分解

- 第一种分解：SN(Sno)，SC(Sdept)，SO(Sloc)
- 分解后都是规范化程度很高的关系模式，但分解后丢失了许多信息。
- 此种分解方法不可取。要使分解有意义，起码的要求是后者不能丢失前者的信息.

Sno	Sdept	Sloc
98001	CS	A
98002	IS	B
98003	MA	C
98004	IS	B
98005	PH	B

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

模式的分解

➲第二种分解：NL(Sno, Sloc)，DL(Sdept, Sloc)

NL

Sno	Sloc
98001	A
98002	B
98003	C
98004	B
98005	B

DL

Sdept	Sloc
CS	A
IS	B
MA	C
IS	B
PH	B

$NL \bowtie DL \Rightarrow$

Sno	Sloc	Sdept
98001	A	CS
98002	B	IS
98002	B	PH
98003	C	MA
98004	B	IS
98004	B	PH
98005	B	IS
98005	B	PH

连接后多了三个元组，将无法知道原来关系中究竟有哪些元组，从这个意义上讲，这个分解仍然丢失了信息。

模式的分解

- 第三种分解：ND(Sno, Sdept)，NL(Sno, Sloc)

ND

Sno	Sdept
98001	CS
98002	IS
98003	MA
98004	IS
98005	PH

NL

Sno	Sloc
98001	A
98002	B
98003	C
98004	B
98005	B

⇒

ND⋈NL

Sno	Sdept	Sloc
98001	CS	A
98002	IS	B
98003	MA	C
98004	IS	B
98005	PH	B

没有丢失信息，恢复为原来的关系，这种分解称为:
“无损连接分解”。

模式的分解

- 定义8.11：设关系模式R(U,F)被分解为若干个关系模式 $R_1(U_1,F_1)$，$R_2(U_2,F_2)$，…，$R_n(U_n,F_n)$ (其中$U=U_1 \cup U_2 \cup \ldots \cup U_n$，且不存在$U_i \subseteq U_j$，$F_i$为F在$U_i$上的投影)，若R与$R_1$，…，$R_n$自然连接的结果相等，则称R的这个分解*具有无损连接性*。

- 自然连接是一种特殊的等值连接，它自动寻找两个表中列名相同且数据类型兼容的列作为连接条件，然后在这些列的值相等的基础上，将两个表合并成一个新表，并且自动去掉重复的列。

➲只有具有无损连接性的分解才能保证不丢失信息

模式的分解

- 第三种分解：ND(Sno, Sdept)，NL(Sno, Sloc)
- 第三种分解虽然具有无损连接性，保证了不丢失原关系中的信息，但它并没有完全解决修改麻烦的现象。如：98001转系，则两个关系中相应的元组必须同时修改。
- 因为这种分解没有保证原关系中的函数依赖，SL中的Sdept→Sloc即没有投影到ND上，也没有投影到NL上，而是跨在两个关系上。

模式的分解

定义8.12 设关系模式R(U,F)被分解为若干个关系模式$R_1(U_1,F_1)$，$R_2(U_2,F_2)$，…，$R_n(U_n,F_n)$ (其中$U=U_1 \cup U_2 \cup \ldots \cup U_n$,且不存在$U_i \subseteq U_j$,$F_i$为F在$U_i$上的投影)，若F所逻辑蕴涵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所逻辑蕴涵，则称关系模式R的这个分解是保持函数依赖的。

➲第四种分解：ND(Sno, Sdept)，DL(Sdept, Sloc)

这种分解保持了函数依赖。

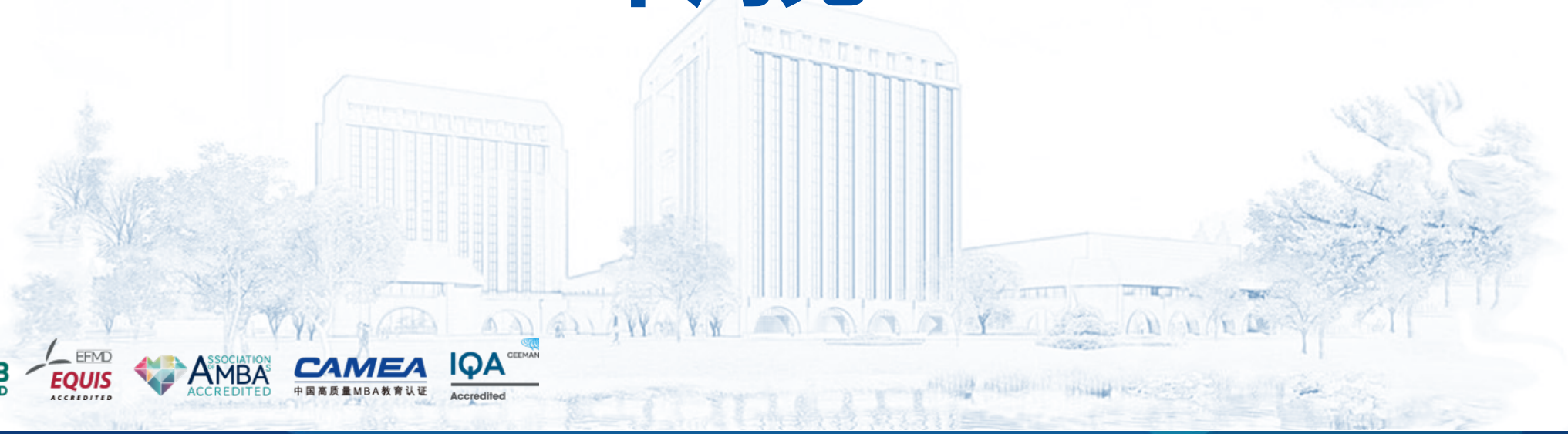
模式的分解

- 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。
- 如果一个分解具有无损连接性，则它能保证不丢失信息。
- 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。
- 最后应当注意的是，规范化理论为数据库设计提供了理论的指南和工具，但仅仅是指南和工具。并不是规范化程度越高，模式就越好，实际应用中还必须结合应用环境和具体情况，合理地选择数据库模式。

浙江大学管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

谢谢！

下周见！



AACSB ACCREDITED
EFMD EQUIS ACCREDITED
ASSOCIATION AMBA ACCREDITED
CAMEA 中国高质量MBA教育认证
IQA CEEMAN Accredited