13

第十三讲 关系数据库规范化理论

教学内容

- 好关系模式判定标准
  - 尽可能少的数据冗余
  - 没有插入异常
  - 没有删除异常
  - 没有更新异常
- 函数依赖
- 范式理论
  - 第一范式
  - 第二范式
  - 第三范式
  - **BCNF范式**
- 关系模式分解方法

13.1 规范化问题的提出

- 关系数据库的规范化理论
  - 最早是由关系数据库的创始人**E.F.Codd**提出
  - 后经许多专家学者深入的研究和发展
  - 形成了一整套有关关系数据库设计的理论

规范化问题的提出

- 系统运行效率、成败的决定因素
  - 如何设计一个适合的关系数据库系统，关键是关系数据库模式的设计
  - 一个好的关系数据库模式应该包括多少关系模式
  - 每一个关系模式又应该包括哪些属性
  - 如何将这些相互关联的关系模式组建一个适合关系模型
- 必须在关系数据库的规范化理论的指导下逐步完成

13.1.1 关系数据库规范化理论的内容

- 主要包括三个方面
  - 函数依赖
  - 范式（Normal Form）
  - 模式设计
- 函数依赖起着核心的作用
- 函数依赖是模式分解和模式设计的基础
- 范式是模式分解的标准

关系模式的存储异常问题

- 数据库的逻辑设计为什么要遵循一定的规范化理论？
- 什么是好的关系模式？
- 某些不好的关系模式可能导致哪些问题？

13.1.2 存储异常示例

- 教学管理数据库，关系模式SCD
  - SCD(StudNo,StudName,StudSex,Deptment,DMasterNM, CourseID,StudScore)

属性名	含义
StudNo	学生学号
StudName	学生姓名
StudSex	学生性别
Deptment	学生所在的系别
DMasterNM	系主任姓名
CourseID	课程号
StudScore	成绩

语义规定

- 根据实际情况，数据有如下语义规定：
  - 1. 一个系有若干个学生，但一个学生只属于一个系；
  - 2. 一个系只有一名系主任，但一个系主任可以同时兼几个系的系主任；
  - 3. 一个学生可以选修多门功课，每门课程可有若干学生选修；
  - 4. 每个学生学习课程有一个成绩。

关系SCD示例数据

StudNo	StudName	StudSex	Deptment	DMasterNM	CourseID	StudScore
20191152001	赵亦	男	计算机	刘伟	SJKJC01	90
20191152001	赵亦	男	计算机	刘伟	GDSX01	85
20191151001	钱尔	男	信息	王平	YY01	57
20191151001	钱尔	男	信息	王平	DXYW02	80
20191151001	钱尔	男	信息	王平	DXPLL02	70
20191151001	钱尔	男	信息	王平	GDSX02	70
20191151002	孙珊	女	信息	王平	SJKJC01	0
20191151002	孙珊	女	信息	王平	GDSX01	70
20191151002	孙珊	女	信息	王平	SJJG02	85
20191153001	李思	女	自动化	刘伟	SJKC01	93

13.1.3 异常问题

- 主关系键：(StudNo,CourseID)属性的组合能唯一标识一个元组
- 在进行数据库的操作时，会出现以下几方面的问题
  - 1. 数据冗余
    - 系名、系主任姓名、学生姓名、性别重复
    - 数据的冗余度很大，浪费了存储空间
  - 2. 插入异常
    - 新系没有招生，则系名和系主任的信息无法插入
    - 当某个学生尚未选课，根据实体完整性约束，不能进行插入操作

异常问题

- **3.** 删除异常
  - 某系学生全部毕业而没有招生时，删除全部学生的记录则系名、系主任也随之删除。
  - 如果某个学生不再选修**SJKJC01**课程，将整个元组一起删掉，有关该学生的其它信息也随之丢失
- **4.** 更新异常
  - 学生改名，则该学生的所有记录都要逐一修改**StudName**
  - 某系更换系主任，则属于该系的学生记录都要修改**DMasterNM**的内容，稍有不慎，就有可能漏改某些记录，这就会造成数据的不一致性，破坏了数据的完整性

13.1.4 关系模式SCD分解

- 由于存在以上问题，SCD是一个不好的关系模式。
- 产生上述问题的原因，是因为关系中“包罗万象”，内容太杂。
- 分解关系模式SCD为三个结构简单的关系模式
  - 学生关系StudInfo(StudNo,StudName,StudSex,Deptment)
  - 选课关系StudScoreInfo(StudNo,CourseID,StudScore)
  - 系关系Deptmentinfo(Deptment,DMasterNM)

分解后的关系模式

StudInfo

StudNo	StudName	StudSex	Deptment
20191152001	赵亦	男	计算机
20191151001	钱尔	男	信息
20191151002	孙珊	女	信息
20191153001	李思	女	自动化

DeptmentInfo

Deptment	DMasterNM
计算机	刘伟
信息	王平
自动化	刘伟

StudScoreInfo

StudNo	CourseID	StudScore
20191152001	SJKJC01	90
20191152001	GDSX01	85
20191151001	YY01	57
20191151001	DXYW02	80
20191151001	DXPLL02	50
20191151001	GDSX02	70
20191151002	SJKJC01	0
20191151002	GDSX01	70
20191151002	SJJG02	85
20191153001	SJKJC01	93

分解后的关系模式分析

- 在以上三个关系模式中，实现了信息的某种程度的分离
  - **StudInfo**中存储学生基本信息，与所选课程及系主任无关；
  - **DeptmentInfo**中存储系的有关信息，与学生无关；
  - **StudScoreInfo**中存储学生选课的信息，而与所学生及系的有关信息无关。
- 经过上述分析，分解后的关系模式是一个好的关系数据库模式。

13.1.5 结论

- 结论：一个好的关系模式应该具备以下四个条件
  - 1. 尽可能少的数据冗余
  - 2. 没有插入异常
  - 3. 没有删除异常
  - 4. 没有更新异常
- 注意
  - 一个好的关系模式不是在任何情况下都是最优的
  - 比如查询某个学生选修课程名及所在系的系主任时，要通过连接，而连接所需要的系统开销非常大，以实际的目标出发进行设计

13.2 关系的规范化

- 关系的规范化
  - 按照一定的规范设计关系模式，将结构复杂的关系分解成结构简单的关系
  - 把不好关系数据库模式转变为好的关系数据库模式
- 规范化又可以根据不同的要求而分成若干级别
- 关系模式中的各属性是相互依赖、相互制约的，在设计关模式时，必须从语义上分析这些依赖关系,这样才构成了一个结构严谨的整体
- 数据库模式好坏和关系中各属性间的依赖关系有关

13.2.1 函数依赖

- 数据依赖
  - 关系模式中各属性之间相互依赖、相互制约的联系
  - 一般分为函数依赖、多值依赖和连接依赖。
- 函数依赖（Functional Dependency）
  - 是最重要的数据依赖
  - 是关系模式中属性之间的一种逻辑依赖关系
  - 分为完全函数依赖、部分函数依赖和传递函数依赖三类
  - 是规范化理论的依据和规范化程度的准则

函数依赖

- 在关系模式SCD中，StudNo与StudName、StudSex、Deptment之间都有一种依赖关系。
- 由于一个StudNo只对应一个学生，而一个学生只能属于一个系，当StudNo的值确定之后，StudName，StudSex，Deptment的值也随之被唯一确定了。
- 这类似于变量之间的单值函数关系。设单值函数Y=F(X)，自变量X的值可以决定一个唯一的函数值Y
- StudNo决定函数（StudName，StudSex，Deptment），或者说（StudName，StudSex，Deptment）函数依赖于StudNo

13.2.2 函数依赖的定义

- 定义：设关系模式R(U，F)，U是属性全集，F是U上的函数依赖集，X和Y是U的子集，如果对于R(U)的任意一个可能的关系r，对于X的每一个具体值，Y都有唯一的具体值与之对应，则称X决定函数Y，或Y函数依赖于X，记作X→Y。我们称X为决定因素，Y为依赖因素。当Y不函数依赖于X时，记作：X $\not\rightarrow$ Y。当X→Y且Y→X时，则记作：X↔Y。

函数依赖示例

- 对于关系模式SCD

  U={StudNo,StudName,StudSex,Deptment,DMaster NM,CourseID,StudScore}

  F={StudNo→StudName，StudNo→StudSex，StudNo→Deptment}

- 一个StudNo有多个StudScore的值与其对应，因此StudScore不能唯一地确定，即StudScore不能函数依赖于StudNo，所以有：StudNo $\not\rightarrow$ StudScore。

- 但是StudScore可以被（StudNo，CourseID）唯一地确定。表示为：（StudNo，CourseID）→StudScore

13.2.3 函数依赖的基本性质

- 1．投影性
  - 一组属性函数决定它的所有子集。
  - 例：在关系SCD中，（StudNo，CourseID）→StudNo和（StudNo，CourseID）→CourseID
- 2．扩张性
  - 若X→Y且W→Z，则（X，W）→（Y，Z）
  - 例：StudNo→（StudName，StudSex），Deptment→DMasterNM，则有（StudNo，Deptment）→（StudName,StudSex, DMasterNM）

函数依赖的基本性质

- 3．合并性
  - 若X→Y且X→Z则必有X→（Y，Z）。
  - 例：在关系SCD中，StudNo→（StudName,StudSex），StudNo→（Deptment,DMasterNM），则有StudNo→（StudName,StudSex，Deptment，DMasterNM）。
- 4．分解性
  - 若X→（Y，Z）,则X→Y且X→Z。很显然，分解性为合并性的逆过程。
  - 由合并性和分解性，很容易得到以下事实：
  - X→A1，A2，…,An成立的充分必要条件是X→Ai（i=1,2,…,n）成立。

13.2.4 完全函数依赖与部分函数依赖

- 定义：设关系模式R(U)，U是属性全集，X和Y是U的子集，如果X→Y，并且对于X的任何一个真子集X′,都有X′ $\not\rightarrow$ Y，则称Y对X完全函数依赖（Full Functional Dependency），记作X $\xrightarrow{f}$ Y。如果对X的某个真子集X′，有X′→Y，则称Y对X部分函数依赖（Partial Functional Dependency），记作X $\xrightarrow{p}$ Y。

完全、部分函数依赖示例

- 例：在关系模式SCD中
  - StudNo $\not\rightarrow$ StudScore
  - 且CourseID $\not\rightarrow$ StudScore
  - 有：(StudNo，CourseID) $\xrightarrow{f}$ StudScore
- 部分函数依赖
  - StudNo→StudSex
  - 有：(StudNo，CourseID) $\xrightarrow{p}$ StudSex

13.2.5 传递函数依赖

- 定义：设有关系模式R（U），U是属性全集，X，Y，Z是U的子集，若$X \rightarrow Y$，但$Y \not\rightarrow X$，而$Y \rightarrow Z$（$Y \notin X$，$Z \notin Y$），则称Z对X传递函数依赖（Transitive Functional Dependency），记作：$X \xrightarrow{t} Z$。如果$Y \rightarrow X$，则$X \leftrightarrow Y$，这时称Z对X直接函数依赖，而不是传递函数依赖。

传递函数依赖示例

- 例：在关系模式SCD中，StudNo→Deptment，但Deptment↛StudNo，而Deptment→DMasterNM，则有 StudNo $\xrightarrow{t}$ DMasterNM。
- 当学生不重名，有StudNo→StudName，StudName→StudNo，StudNo ↔ StudName，StudNo→Deptment，这时Deptment对StudName是直接函数依赖，而不是传递函数依赖

部分函数依赖

- 只有当决定因素是组合属性时，讨论部分函数依赖才有意义
- 当决定因素是单属性时，只能是完全函数依赖
- 例：
  - 在关系模式**S（StudNo，StudName，StudSex，Deptment）**，决定因素为单属性**StudNo**，有**StudNo→（StudName，StudSex,Deptment）**，不存在部分函数依赖。

13.3 范式

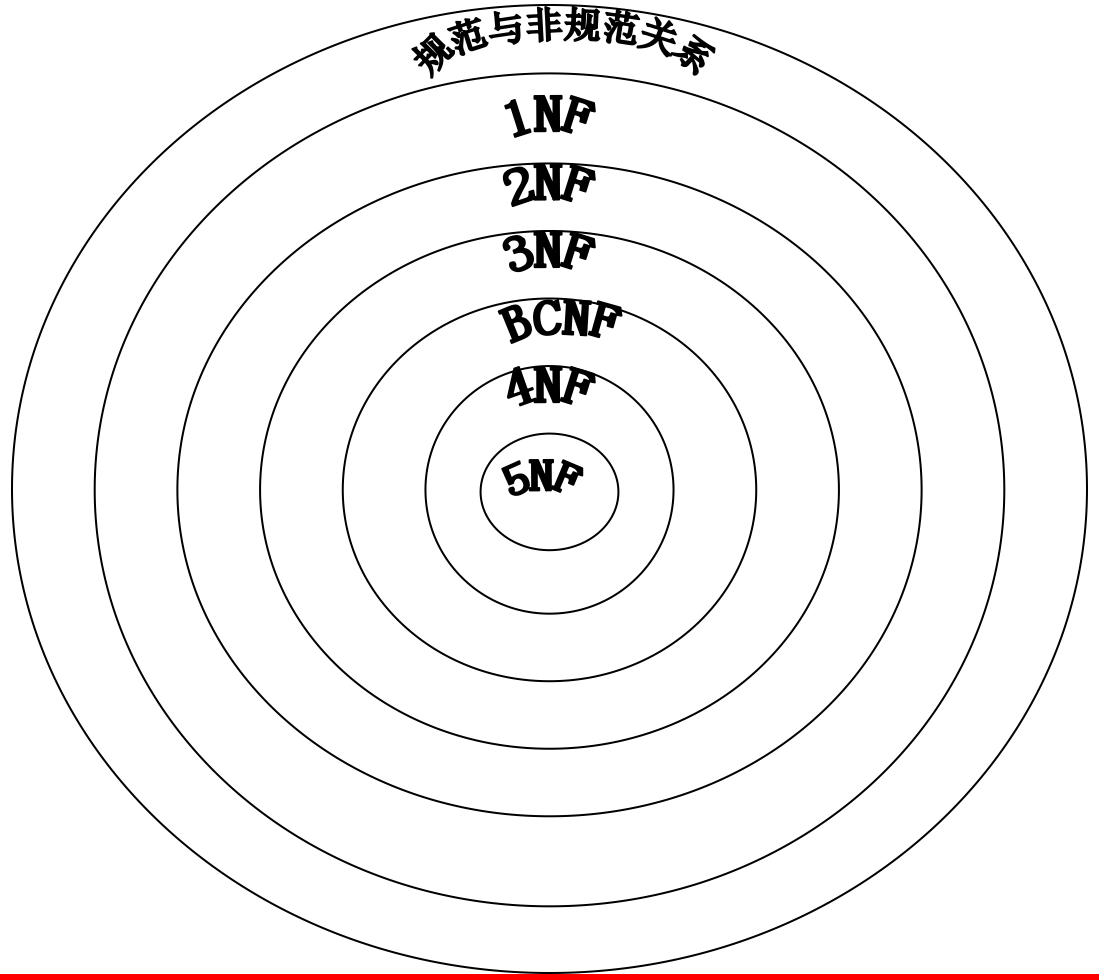
- 规范化的基本思想是消除关系模式中的数据冗余，消除数据依赖中的不合适的部分，解决数据插入、删除时发生异常现象。
- 范式（Normal Form）
  - 关系数据库的规范化过程中为不同程度的规范化要求设立的不同标准。

范式

- 范式的概念最早由E.F.Codd提出,从1971年起，Codd相继提出了关系的三级规范化形式，即第一范式（1NF）、第二范式（2NF）、第三范式（3NF）。
- 1974年，Codd和Boyce以共同提出了一个新的范式的概念，即Boyce-Codd范式，简称BC范式
- 1976年Fagin提出了第四范式
- 每种范式都规定了一些限制约束条件。

各种范式之间的关系

- 5NF⊂ 4NF⊂ BCNF⊂ 3NF⊂ 2NF⊂ 1NF



13.3.1 第一范式

- 第一范式（First Normal Form）是最基本的规范形式，即关系中每个属性都是不可再分的简单项。
- 定义:如果关系模式R，其所有的属性均为简单属性，即每个属性都是不可再分的，则称R属于第一范式，简称1NF，记作$R \in 1NF$。
- 在非规范化的关系中去掉组合项就能化成规范化的关系,每个规范化的关系都属于1NF
- 一个关系模式仅仅属于第一范式是不适用的,关系模式SCD属于第一范式，但其具有大量的数据冗余，具有插入异常、删除异常、更新异常等弊端

关系模式SCD中的函数依赖关系

- 关系模式SCD的关系键是（StudNo，CourseID）属性组合

（StudNo，CourseID）$\xrightarrow{f}$ StudScore

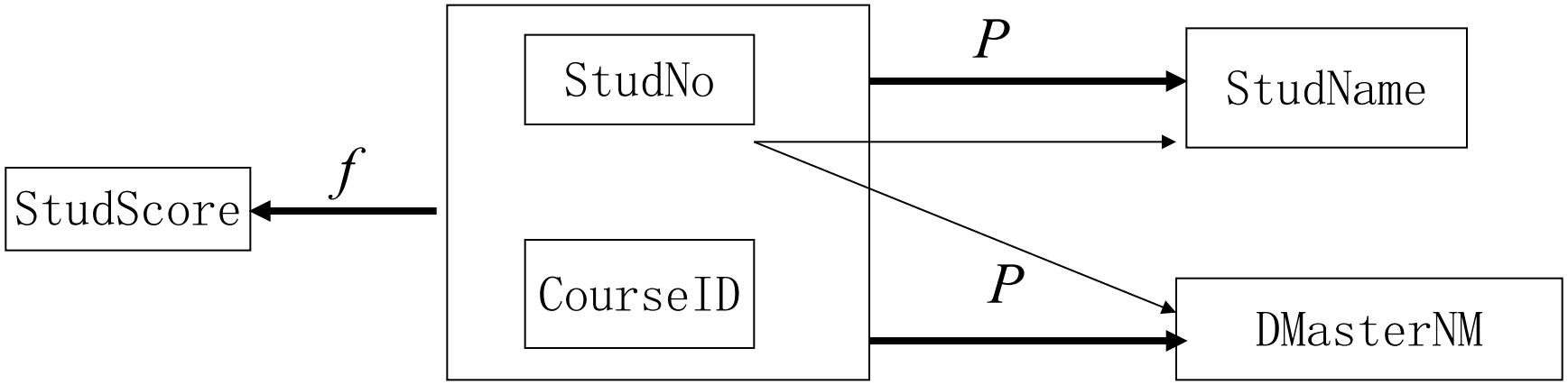
StudNo→StudName,（StudNo,CourseID）$\xrightarrow{p}$ StudName

StudNo→StudSex,（StudNo,CourseID） $\xrightarrow{p}$ StudSex

StudNo→Deptment,（StudNo,CourseID）$\xrightarrow{p}$ Deptment

StudNo $\xrightarrow{t}$ DMasterNM,(StudNo,CourseID) $\xrightarrow{p}$ DMasterNM

函数依赖图



- 在SCD中，既存在完全函数依赖，又存在部分函数依赖和传递函数依赖。这种情况在数据库中是不允许的，由于关系中存在着复杂的函数依赖，导致数据操作中出现了种种弊端
- 克服这些弊端的方法是用投影运算将关系分解，去掉过于复杂的函数依赖关系，向更高一级的范式进行转换。

13.3.2 第二范式

- 定义: 如果关系模式R ∈ 1NF，且每个非主属性都完全函数依赖于R的每个关系键，则称R属于第二范式（Second Normal Form），简称2NF，记作R ∈ 2NF。
- 在关系模式SCD中，(StudNo,CourseID)为主属性，StudName，StudSex，Deptment，DMasterNM，StudScore均为非主属性，经分析，存在非主属性对关系键的部分函数依赖，所以SCD不属于2NF

全码

- 关系模式TCS（T，C，S），一个教师可以讲授多门课程，一门课程可以为多个教师讲授，同样一个学生可以选听多门课程，一门课程可以为多个学生选听，(T,C,S)三个属性的组合是关系键，T,C,S都是主属性，而无非主属性，所以也就不可能存在非主属性对关系键的部分函数依赖，TCS ∈ 2NF。
- 结论
  - 1．从1NF关系中消除非主属性对关系键的部分函数依赖，则可得到2NF关系。
  - 2．如果R的关系键为单属性，或R的全体属性均为主属性，则R ∈ 2NF。

2NF规范化

- 2NF规范化是指把1NF关系模式通过投影分解转换成2NF关系模式的集合。
- 分解时遵循的基本原则就是“一事一地”，让一个关系只描述一个实体或者实体间的联系。如果多于一个实体或联系，则进行投影分解。

2NF规范化示例

- 关系模式SCD(StudNo,StudName,StudSex,Deptment,DMasterNM,CourseID,StudScore)规范为2NF。
- 由StudNo→StudName，StudNo→StudSex，StudNo→Deptment,(StudNo，CourseID)$\xrightarrow{f}$StudScore
- 可以判断，关系SCD至少描述了两个实体
  - 一个为学生实体，属性有StudNo、StudName、StudSex、Deptment、DMasterNM；
  - 另一个是学生与课程的联系（选课），属性有StudNo、CourseID和StudScore。

分解SCD：SD和StudScoreInfo关系

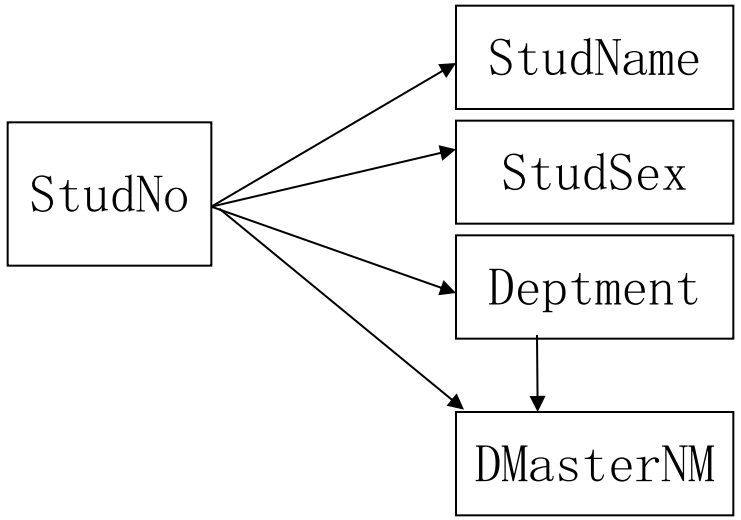
SD

StudNo	StudName	StudSex	Deptment	DMasterNM
20191152001	赵亦	男	计算机	刘伟
20191151001	钱尔	男	信息	王平
20191151002	孙珊	女	信息	王平
20191153001	李思	女	自动化	刘伟

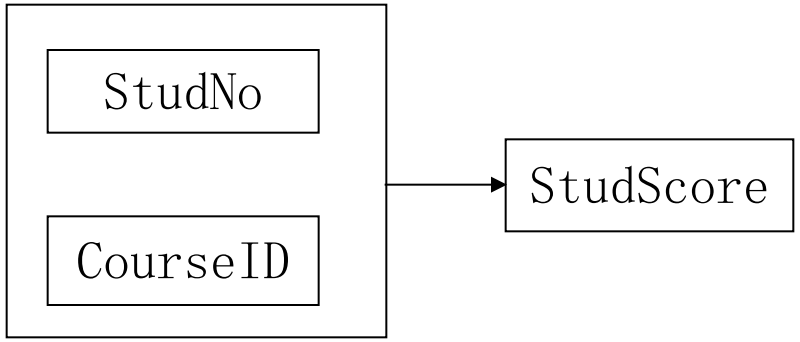
StudScoreInfo

StudNo	CourseID	StudScore
20191152001	SJKJC01	90
20191152001	GDSX01	85
20191151001	YY01	57
20191151001	DXYW02	80
20191151001	DXPLL02	50
20191151001	YY01	70
20191151002	SJKJC01	0
20191151002	GDSX01	70
20191151002	SJJG02	85
20191153001	SJKJC01	93

SD和StudScoreInfo关系



SD中的函数依赖关系



StudScoreInfo中的函数依赖关系

- 对于分解后的两个关系SD和Studscoreinfo，主键分别为StudNo和（StudNo，CourseID），非主属性对主键完全函数依赖。因此，SD∈2NF,Studscoreinfo∈2NF。

2NF结论

- 1NF的关系模式经过投影分解转换成2NF后，消除了一些数据冗余
- 分析SD和StudScoreInfo中的数据，存储的冗余度比关系模式SCD有了较大辐度的降低。
- 学生的姓名、性别不需要重复存储多次。
- 在一定程度上避免数据更新所造成的数据不一致性的问题
- 由于学生的基本信息与选课信息分开存储，则学生基本信息因没选课而不能插入的问题得到了解决，插入异常得到了部分改善
- 如果某个学生不再选修YY01课程，在选课关系StudScoreInfo中删去该该学生选修YY01的记录即可，而SD中有关该学生的信息不会受到影响，解决了部分删除异常问题
- 因此关系模式SD和SC在性能上比SCD有了显著提高。

2NF的缺点

- 2NF的关系模式解决了1NF中存在的一些问题，但仍然存在着一些问题：
  - 1．数据冗余。每个系名和系主任的名字存储的次数等于该系的学生人数。
  - 2．插入异常。当一个新系没有招生时，有关该系的信息无法插入。
  - 3．删除异常。某系学生全部毕业而没有招生时，删除全部学生的记录也随之删除了该系的有关信息。
  - 4．更新异常。更换系主任时，仍需改动较多的学生记录。

2NF存在问题的原因

- 存在这些问题是由于在SCD中存在着非主属性对主键的传递依赖
- 分析SCD中的函数依赖关系，
  StudNo→StudName，StudNo→StudSex，
  StudNo→Deptment，Deptment→DMasterNM，
  StudNo $\xrightarrow{t}$ DMasterNM，非主属性DMasterNM对主键StudNo传递依赖。
- 对关系模式SCD还需进一步简化，消除这种传递依赖，得到3NF

13.3.3 第三范式

- 定义：如果关系模式R ∈ 2NF，且每个非主属性都不传递依赖于R的每个关系键，则称R属于第三范式（Third Normal Form），简称3NF，记作R ∈ 3NF。
- 第三范式具有如下性质：
  - 1．如果R ∈ 3NF，则R ∈ 2NF
  - 2．如果R ∈ 2NF，则R不一定是3NF。
- 例：关系模式SCD分解而得到的SD和StudScoreInfo都为2NF，其中，Studscoreinfo ∈ 3NF，但在SD中存在着非主属性DMasterNM对主键StudNo传递依赖，SD ∉ 3NF。

3NF规范化

- 3NF规范化是指把2NF关系模式通过投影分解转换成3NF关系模式的集合。
- 和2NF的规范化时遵循的原则相同，即“一事一地”，让一个关系只描述一个实体或者实体间的联系。

3NF规范化示例

- 例：将SD(StudNo,StudName,StudSex, Deptment,DMasterNM)规范到3NF
- 分析SD的属性组成，关系SD描述了两个实体：
  - 一个为学生实体，属性有StudNo，StudName，StudSex，Deptment
  - 一个是系的实体，其属性Deptment和DMasterNM
- 根据分解的原则，将SD分解成如下两个关系
  - StudInfo(StudNo,StudName,StudSex,Deptment)
  - DeptmentInfo(Deptment，DMasterNM)

StudInfo和DeptmentInfo关系

- 分解后的关系不存在非主属性对主键的传递函数依赖

StudInfo

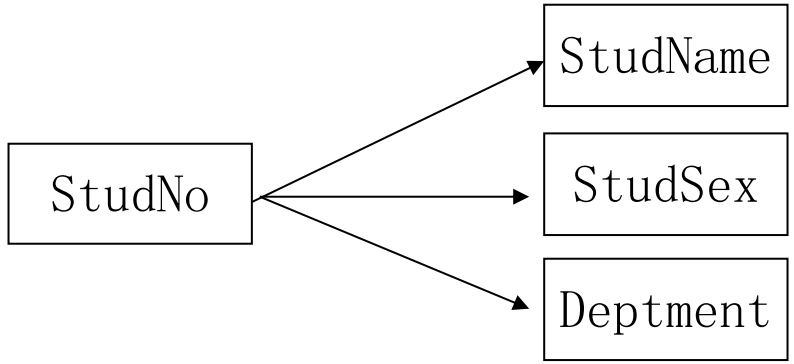
StudNo	StudName	StudSex	Deptment
20191152001	赵亦	男	计算机
20191151001	钱尔	男	信息
20191151002	孙珊	女	信息
20191153001	李思	女	自动化

DeptmentInfo

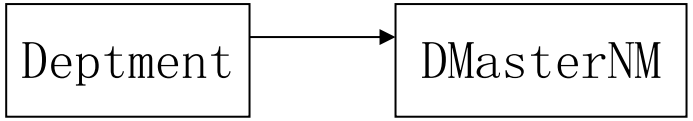
Deptment	DMasterNM
计算机	刘伟
信息	王平
自动化	刘伟

Studinfo和DeptmentInfo的函数依赖

- 关系模式SD由2NF分解为3NF后，函数依赖关系变得更加简单，既没有非主属性对键的部分依赖，也没有非主属性对键的传递依赖，解决了2NF中存在的四个问题。



Studinfo中的函数依赖关系图



Deptmentinfo中的函数依赖关系图

3NF结论

- 将SCD规范到3NF后，所存在的异常现象已经全部消失。
- 3NF只限制了非主属性对键的依赖关系，而没有限制主属性对键的依赖关系。仍有可能存在数据冗余、插入异常、删除异常和修改异常。
- 对3NF进一步规范化，消除主属性对键的依赖关系，Boyce与Codd共同提出了一个新范式的定义，这就是Boyce-Codd范式，通常简称BCNF或BC范式。它弥补了3NF的不足。

13.3.4 BCNF范式

- 定义：如果关系模式R ∈ 1NF，且所有的函数依赖X→Y（Y∉X）,决定因素X都包含了R的一个候选键，则称R属于BC范式（Boyce-Codd Normal Form），记作R ∈ BCNF。
- BCNF具有如下性质：
  - 1．满足BCNF的关系将消除任何属性（主属性或非主属性）对键的部分函数依赖和传递函数依赖，如果R ∈ BCNF，则R ∈ 3NF。
  - 2．如果R ∈ 3NF，则R不一定是BCNF。

BCNF范式示例

- 设关系模式StudCourse（StudNo，StudName，CourseID，StudScore）
- 注：StudName代表学生姓名并假设没有重名
- 两个候选键（StudNo，CourseID）和（StudName，CourseID），其函数依赖如下：
  - StudNo ↔ StudName
  - （StudNo，CourseID）→StudScore
  - （StudName，CourseID）→StudScore
- 唯一非主属性StudScore对键不存在部分函数依赖，也不存在传递函数依赖，StudCourseinfo ∈ 3NF

主属性对键的部分函数依赖存在的问题

- 因**StudNo ↔ StudName**，即决定因素**StudNo**或**StudName**不包含候选键，从另一个角度说，存在着主属性对键的部分函数依赖：

  （**StudNo，CourseID**）$\xrightarrow{p}$ **StudName**，（**StudName，CourseID**）$\xrightarrow{p}$ **StudNo**，所以**StudCourse**不是**BCNF**。

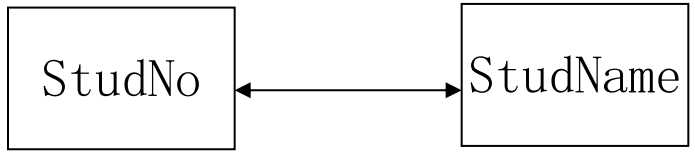
- 因存在主属性对键的部分函数依赖关系，造成了关系**StudCourse**中存在着较大的数据冗余，学生姓名的存储次数等于该生所选的课程数，从而会引起修改异常。
- 例：当更改某个学生的姓名时，则必须搜索出现该姓名的每个学生选课记录，并对其姓名逐一修改，容易造成数据的不一致问题
- 解决这一问题的办法是通过投影分解进一步提高**StudCourse**的范式等级，规范到**BCNF**。

BCNF规范化

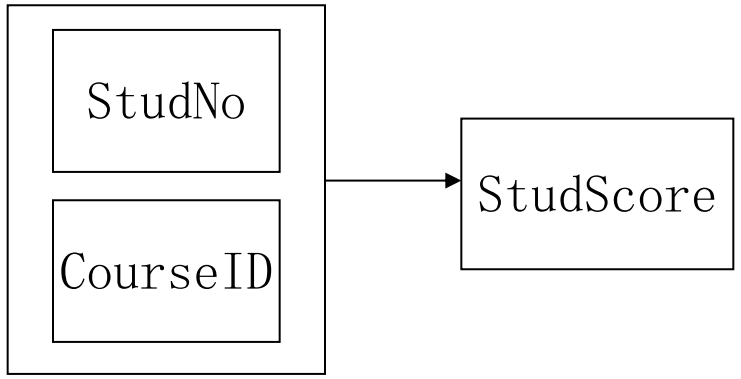
- BCNF规范化是指把3NF关系模式通过投影分解转换成BCNF关系模式的集合。
- 例：将StudCourse(StudNo,StudName,CourseID，StudScore)规范到BCNF。
- 分析StudCourse数据冗余的原因是在这一个关系中存在两个实体，一个为学生实体，属性有StudNo、StudName；另一个是选课实体，属性有StudNo、CourseID和StudScore。
- 根据分解的原则，将StudCourse分解成如下两个关系：
  - StudInfo(StudNo,StudName)
  - StudScoreInfo(StudNo,CourseID,StudScore)
- 在这两个关系中，无论主属性还是非主属性都不存在对键的部分依赖和传递依赖，StudInfo ∈ BCNF，StudScoreInfo ∈ BCNF。

Studinfo和Studscoreinfo的函数依赖

- 关系StudCourse转换成BCNF后，数据冗余度明显降低。
- 学生的姓名只在关系S1中存储一次，学生要改名时，只需改动一条学生记录中的相应的StudName值，从而不会发生修改异常。



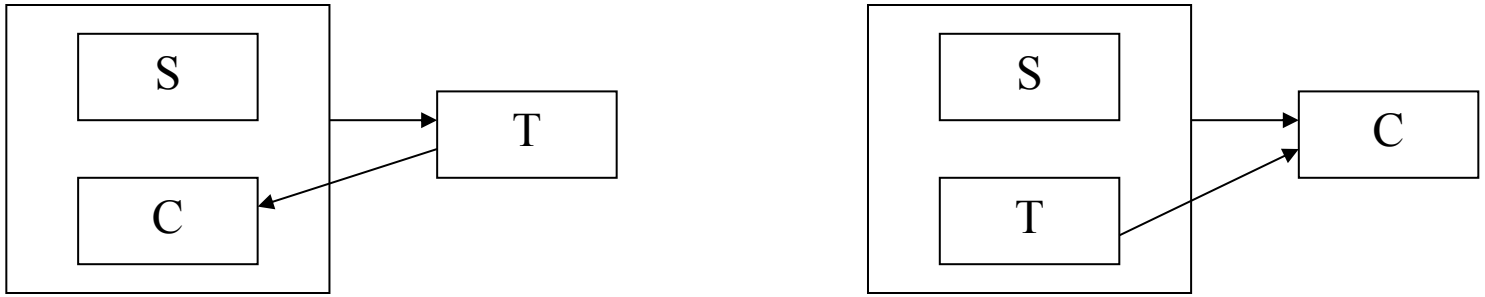
StudInfo中的函数依赖关系



StudScoreInfo中的函数依赖关系

BCNF范式示例

- 例：设关系模式TCS（T，C，S），T表示教师，C表示课程，S表示学生。
- 语义假设：每一位教师只讲授一门课程；每门课程由一个教师讲授；某一学生选定某门课程，就对应于一确定的教师。
- 根据语义假设，TCS的函数依赖是：
  - （S，C）→T，（S，T）→C，T→C。



TCS中的函数依赖关系

BCNF范式示例

- 对于TCS，（S，C）和（S，T）都是候选键，两个候选键相交，有公共的属性S。TCS中不存在非主属性，也就不可能存在非主属性对键的部分依赖或传递依赖，所以TCS ∈ 3NF

TCS一个关系实例

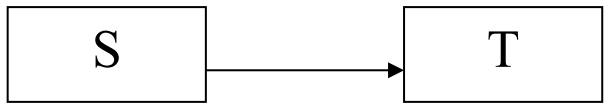
T	C	S
T1	C1	S1
T1	C1	S2
T2	C1	S3
T2	C1	S4
T3	C2	S2
T4	C2	S2
T4	C3	S2

关系TCS存在的问题

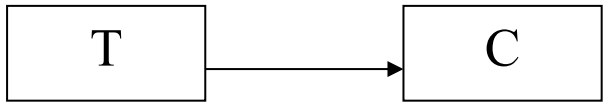
- 1. 数据冗余
- 2. 插入异常
- 3. 删除异常
- 4. 更新异常

BCNF范式示例

- 分析出现上述问题的原因在于主属性部分依赖于键，（S，T）→ C，因此关系模式还继续分解，转换成更高一级的范式BCNF，以消除数据库操作中的异常现象。
- 将TCS分解为两个关系模式ST（S，T）和TC（T，C），消除函数依赖（S，T）C。其中ST的键为S，TC的键为T。ST ∈ BCNF，TC ∈ BCNF。



ST中的函数依赖关系



TC中的函数依赖关系

BCNF结论

- 关系模式TCS由规范到BCNF后，解决原来存在的四个异常问题
  - 1．数据冗余降低
  - 2．不存在插入异常
  - 3．不存在删除异常
  - 4．不存在更新异常
- 如果一个关系数据库中所有关系模式都属于BCNF，那么在函数依赖的范畴内，已经实现了模式的彻底分解，消除了产生插入异常和删除异常的根源，而且数据冗余也减少到极小程度。

13.3.5 关系模式的规范化

- 规范化的基本原则
  - 遵从概念单一化“一事一地”的原则，即一个关系只描述一个实体或者实体间的联系。
- 若多于一个实体，就把它“分离”出来。
- 所谓规范化，实质上是概念的单一化，即一个关系表示一个实体

关系模式规范化的步骤

- 规范化就是对原关系进行投影，消除决定属性不是候选键的任何函数依赖。具体可以分为以下几步：
  - 1．对1NF关系进行投影，消除原关系中非主属性对键的部分函数依赖，将1NF关系转换成若干个2NF关系。
  - 2．对2NF关系进行投影，消除原关系中非主属性对键的传递函数依赖，将2NF关系转换成若干个3NF关系。
  - 3．对3NF关系进行投影，消除原关系中主属性对键的部分函数依赖和传递函数依赖，也就是说使决定因素都包含一个候选键。得到一组BCNF关系。

关系模式规范化过程