1. 问题的提出
2. 关系模式的组成
3. 五元组

关系模式由五部分组成，是一个五元组：R(U, D, DOM, F)

关系名R是符号化的元组语义

U为一组属性

D为属性组U中的属性所来自的域

DOM为属性到域的映射

F为属性组U上的一组数据依赖

1. 三元组

五元组简化为一个三元组：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

1. 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

1. 数据依赖的类型
2. 函数依赖
3. 多值依赖
4. 函数依赖

函数依赖普遍存在于现实生活中

描述一个学生关系，可以有学号、姓名、系名等属性，一个学号只对应一个学生，一个学生只在一个系中学习，“学号”值确定后，学生的姓名及所在系的值就被唯一确定

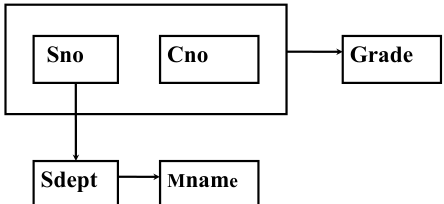
Sname=f(Sno)，Sdept=f(Sno)

Sno函数决定Sname，Sno函数决定Sdept，记作Sno→Sname，Sno→Sdept

示例：假设学校教务的数据库模式用一个单一的关系模式Student来表示，则该关系模式的属性集合为：U ＝{Sno, Sdept, Mname, Cno, Grade}

现实世界的已知事实（语义）：一个系有若干学生， 但一个学生只属于一个系；一个系只有一名（正职）负责人；一个学生可以选修多门课程，每门课程有若干学生选修；每个学生学习每一门课程有一个成绩

由此可得到属性组U上的一组函数依赖F：F={Sno→Sdept, Sdept→ Mname, (Sno, Cno)→ Grade}



假设存在这样一个关系：Student(Sno, Sdept, Mname, Cno, Grade )

“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

Student关系模式不是一个好的关系模式

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

1. 规范化
2. 函数依赖

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→Y

X称为这个函数依赖的**决定属性集**

Y=f(x)

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→年龄”这个函数依赖只有在不允许有同名人的条件下成立

1. 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出现，函数依赖“姓名→年龄”成立。所插入的元组必须满足规定的函数依赖，若发现有同名人存在， 则拒绝装入该元组
2. 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不包含于X，则称X→Y是**非平凡的函数依赖**，若X→Y，但Y包含于X, 则称X→Y是**平凡的函数依赖**

示例：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

1. 完全函数依赖与部分函数依赖

在关系模式R(U)中，如果X→Y，并且对于X的任何一个真子集X’，都有X’不决定Y , 则称**Y完全函数依赖于X**，记作X Y

若X’→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 Y

示例：在关系SC(Sno, Cno, Grade)中，由于：Sno不决定Grade，Cno不决定Grade，因此：(Sno, Cno) Grade

在关系Student(Sno, Sdept, Mname, Cno, Grade)中，由于： Sno → Sdept，Sno是(Sno, Cno)的真子集，因此： (Sno, Cno) Sdept

1. 传递函数依赖

在关系模式R(U)中, 如果X→Y，(Y不包含于X)，Y→X ，Y→Z，则称Z传递函数依赖于X，记为：XZ

如果Y→X， 即X←→Y，则Z直接依赖于X

示例：在关系Std(Sno, Sdept, Mname)中，有：Sno→Sdept，Sdept→Mname，Sdept 不决定Sno，则Mname传递函数依赖于Sno

1. 码

设K为R<U,F>中的属性或属性组合。若K 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）

主属性和非主属性

全码

关系模式S(Sno, Sdept, Sage)，单个属性Sno是码

SC（Sno，Cno，Grade）中，（Sno，Cno）是码

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

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（Sno，Cno，Grade）中，Sno不是码，但Sno是关系模式S（Sno，Sdept，Sage）的码，则Sno是关系模式SC的外部码

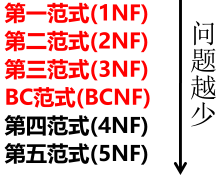
主码与外码一起提供了表示关系间联系的手段

1. 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：



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**

1. 1NF

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。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

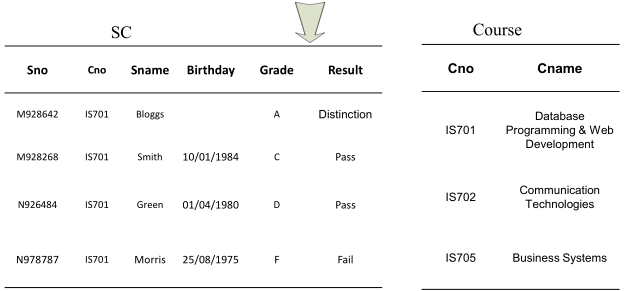
关系中不存在重复行、多值列

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

示例：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





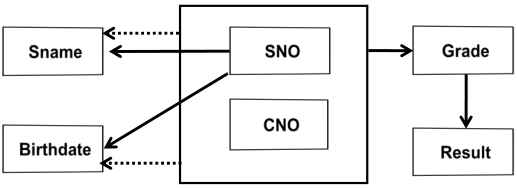
转换成1NF，方法是剔除表中所套的表

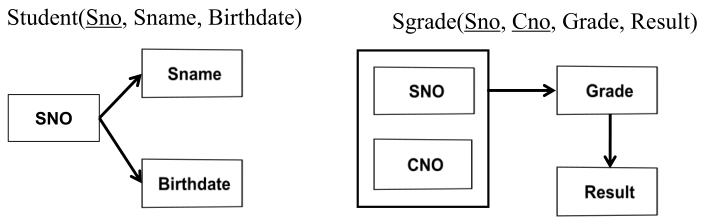
1. 2NF

若R∈1NF，且每一个**非主属性完全函数依赖于码**，则R∈2NF

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

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





1NF—>2NF，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**

1. 3NF

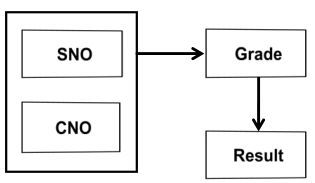
关系模式R<U，F> 中若不存在这样的码X、属性组Y及非主属性Z（Z不包含于Y）, 使得X→Y，Y→X，Y→Z，成立，则称R<U，F> ∈ 3NF

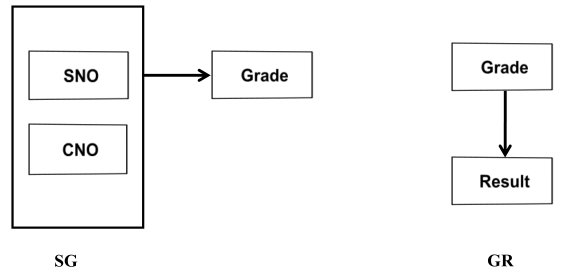
在2NF关系模式Sgrade(Sno, Cno, Grade, Result)中存在以下函数依赖：(Sno, Cno)→Grade，Grade → (Sno, Cno)，Grade→Result，(Sno, Cno) Result

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

会造成：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：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





2NF—>3NF，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在非主属性对主码的传递依赖

注意：

1. 若R∈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**
2. 如果R∈3NF，则R也是2NF
3. 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
4. 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
5. BCNF

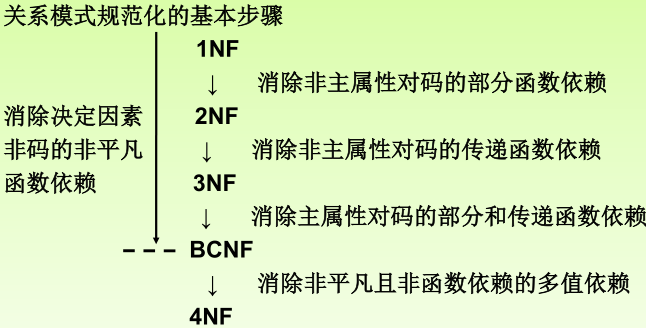
关系模式R<U，F> ∈ 1NF，若X→Y且Y不包含于X时，X必含有码，则R<U，F> ∈BCNF

若R∈BCNF，则

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





1. 数据依赖的公理系统
2. 逻辑蕴含

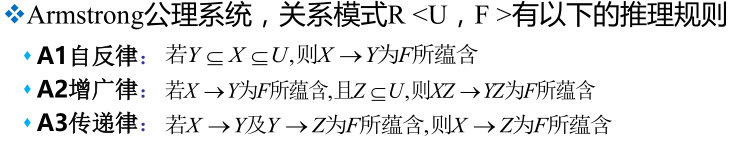
对于满足一组函数依赖F的关系模式R<U, F>，其任何一个关系r，若函数依赖X→Y都成立，则称**F逻辑蕴含X→Y**

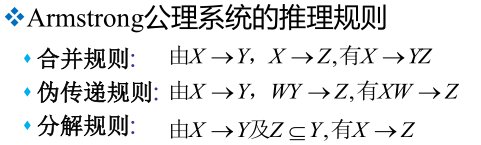
示例：已知R(X,Y,Z)，F={X→Y，Y→Z}, 则X→Z成立，X→Z被F逻辑蕴含

1. Armstrong公理系统

一套推理规则，是模式分解算法的理论基础

用途：从一组函数依赖求得蕴含的函数依赖、求给定关系模式的码





闭包：在关系模式R<U, F>中为F所逻辑蕴含(或推导)的**函数依赖的全体**叫做F的闭包，记为**F+**

Armstrong公理系统的有效性：由F出发根据Armstrong公理导出的每一个函数依赖一定在F + 中

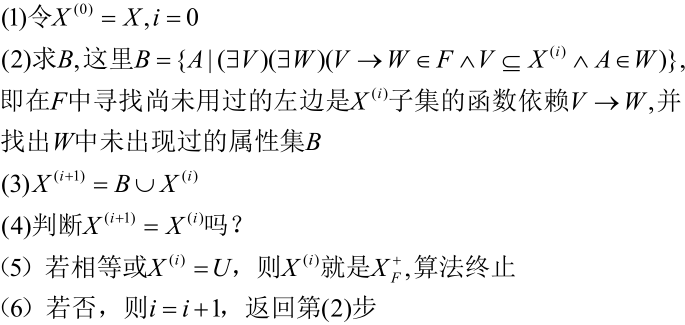
Armstrong公理系统的完备性：F+中的每一个函数依赖，必定可以由F出发根据Armstrong公理导出

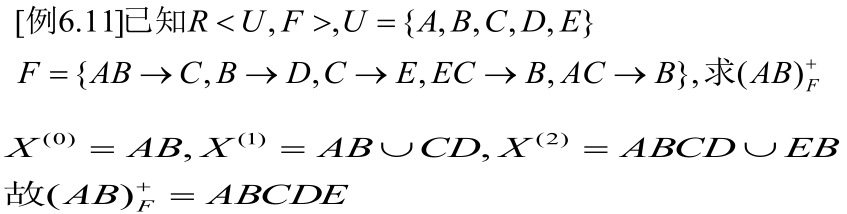
1. 闭包

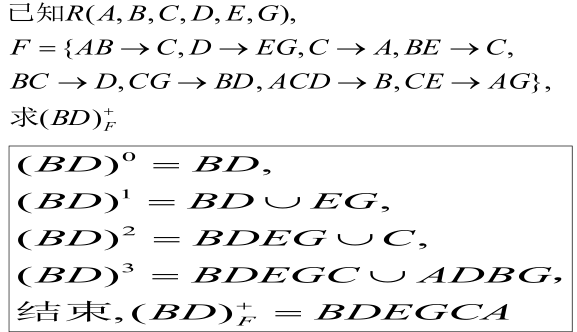
设F为属性集U上的一组函数依赖，X ⊆U，XF+ ={ A|X→A能由F 根据Armstrong公理导出}，X F + 称为属性集X关于函数依赖集F的闭包

示例：U={A, B, C, D}; F={A → B, BC → D}; XF+= {A，B}

1. 求属性集X关于函数依赖集F的闭包：







1. 利用属性组的闭包求关系的候选码

已知：R(X,Y,Z)，F={X→Y , Y→Z}，求关系的候选码

过程：首先，找出**所有没有在任何一个函数依赖右侧出现的属性**，把他们组成一个属性组K，候选码一定含有K; 计算KF+ ，如果KF+ = U，则K为候选码，而且只有这一个;否则，基于K扩充属性，形成新的属性组（含有K但不含候选码的所有可能的属性组合），计算新属性组的闭包，判断其是否为候选码

示例：已知关系模式R<U，F>，其中U={A，B，C，D，E}; F={AB→C，B→D，C→E，EC→B，AC→B}，求关系的候选码

A是没有在任何一个函数依赖右侧出现的属性，候选码一定含有A，AF+ ≠ U，扩充属性，(AB)0 = AB ∪ CD , (AB)1 = ABCD ∪ E ，(AB)F+ = ABCDE , (AB)F+ = U，(AC)0 = AC ∪ EB , (AC)1 = ACEB ∪ D ，(AC)F+ = ABCDE , (AC)F+ = U，关系的候选码是AB和AC