第八章 关系数据库设计理论

陶煜波
计算机科学与技术学院

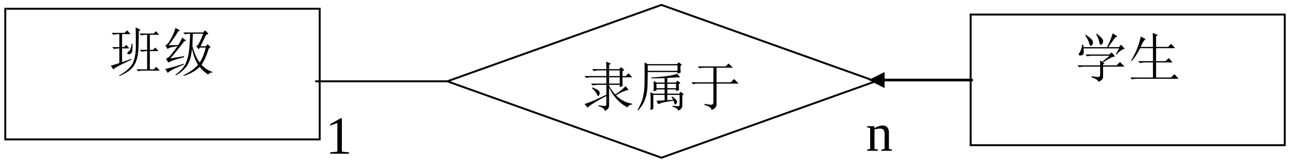
关系数据库设计

数据库设计一般要经过以下几个步骤：

- 需求分析阶段 → 数据流图和数据字典 (软工)
- 概念结构设计阶段 → E-R图/UML图
- 逻辑结构设计阶段 → 关系数据库模式 (规范化)
- 数据库物理设计阶段 → 存储方式、索引和用户权限
- 数据库实施阶段
- 数据库运行和维护阶段

关系数据库设计

- 从E-R图到关系模式



方法一： Student (<u>Sno</u>, Sname, ….)

Class(<u>ClassID</u>, num, …)

Class_Student (<u>Sno</u>, ClassID)

方法二： Student (<u>Sno</u>, Sname, …, ClassID)

Class(<u>ClassID</u>, num, …)

转换得到的关系模式是设计合理或“好”的关系模式?

关系数据库设计

Conceptual Model:

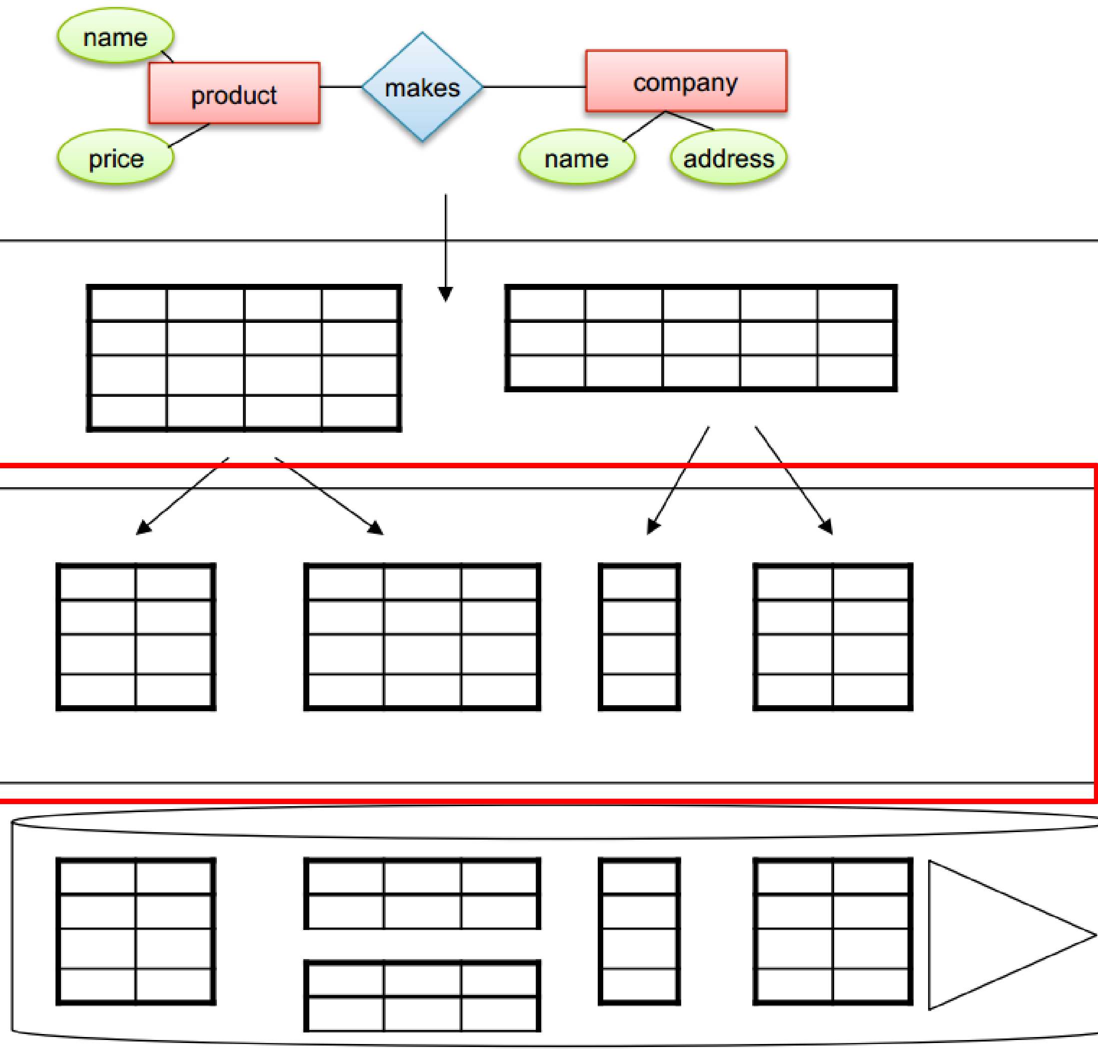
Relational Model:
Tables + constraints
And also functional dep.

Normalization: 规范化
Eliminates anomalies

Conceptual Schema

Physical storage details

Physical Schema



第八章 关系数据库设计理论

- 8.1 数据依赖对关系模式的影响
- 8.2 函数依赖 (Functional Dependencies)
- 8.3 BC范式 (Boyce-Codd Normal Form)
- 8.4 多值依赖 (Multivalued Dependencies) (自学)
- 8.5 第4范式 (Fourth Normal Form) (自学)
- 8.6 BC范式和第4范式的局限
- 8.7 小结

8.1 数据依赖对关系模式的影响

- 数据依赖
  - 通过关系中属性间值的相等与否体现出的数据间的相互关系
  - 现实世界属性间相互联系的抽象
  - 数据的内在性质
  - 语义的体现
- 分类
  - 函数依赖 (Functional Dependencies)
  - 多值依赖 (Multivalued Dependencies)
  - 其他

8.1 数据依赖对关系模式的影响

- 学校数据库(举例):
  - 学生的学号 (Sno)
  - 所在系 (Sdept)
  - 系主任姓名 (Mname)
  - 课程号 (Cname)
  - 成绩 (Grade)
- E-R图中获得'mega' relation
  - Student(Sno, Sdept, Mname, Cname, Grade)
  - 属性之间的数据依赖

8.1 数据依赖对关系模式的影响

- 学校数据库的语义：
  - 一个系有若干学生， 一个学生只属于一个系
  - 一个系只有一名主任
  - 一个学生可以选修多门课程， 每门课程有若干学生选修
  - 每个学生所学的每门课程都有一个成绩
- 属性之间的函数依赖
  - Sno → Sdept
  - Sdept → Mname
  - (Sno, Cname) → Grade

8.1 数据依赖对关系模式的影响

- 函数依赖 → 设计异常 (Design anomalies)
  - 数据冗余 (Redundancy)
    - 浪费大量的存储空间
    - 例如，每个系主任的姓名重复出现
  - 更新异常 (Update anomalies)
    - 数据冗余，更新数据时，维护数据完整性代价大
    - 例如，某系更换系主任后，系统必须修改与该系学生有关的每个元组

8.1 数据依赖对关系模式的影响

- 函数依赖 → 设计异常 (Design anomalies)
  - 插入异常 (Insert anomalies)
    - 有些数据无法正常插入
    - 例如，如果一个系刚成立，尚无学生，就无法把这个系及其系主任的信息存入数据库
  - 删除异常 (Deletion anomalies)
    - 不该删除的数据不得不删
    - 例如，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，把这个系及其系主任的信息也丢掉了

8.1 数据依赖对关系模式的影响

- 大学申请数据库(举例)
  - SSN(美国社会安全号)和name
  - Colleges applying to
  - High schools attended (with city)
  - Hobbies
- E-R图中获得’mega’ relation
  - Apply(SSN, Sname, Cname, HS, HScity, hobby)
  - 属性之间的数据依赖

Cname，(HS, Hscity), hobby是多值属性 ➔ 多值依赖

8.1 数据依赖对关系模式的影响

- Apply(SSN, Sname, Cname, HS, HScity, hobby)
  - 123 Ann from PAHS (P.A.) and GHS (P.A.) plays tennis and trumpet(喇叭) and applied to Standford, Berkely, and MIT
    - 123, Ann, Stanford, PAHS, P.A., tennis
    - 123, Ann, Berkeley, PAHS, P.A., tennis
    - 123, Ann, Berkeley, PAHS, P.A., trumpet
    - ….

思考：Ann在Apply关系中有多少个元组？

8.1 数据依赖对关系模式的影响

- Apply(SSN, Sname, Cname, HS, HScity, hobby)
- 设计异常
  - 数据冗余
    - HS, HScity
    - Cname, hobby
  - 更新异常
    - Trumpet → Cornet
  - 插入异常
    - 尚未申请学校的学生信息无法插入数据库
  - 删除异常

8.1 数据依赖对关系模式的影响

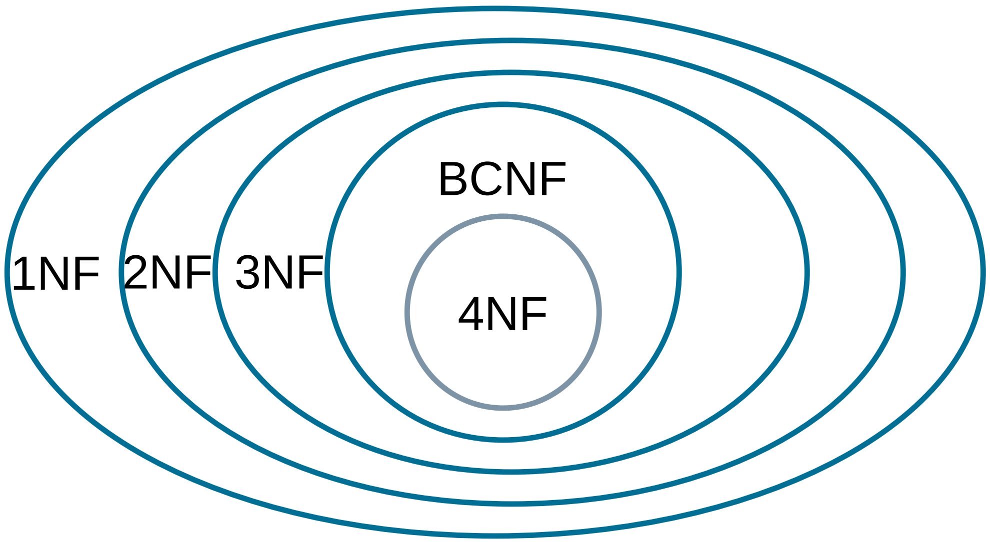
- 结论
  - Student(Sno, Sdept, Mname, Cname, Grade)
  - Apply(SSN, Sname, Cname, HS, HScity, hobby)
  - Student和Apply关系模式不是“好”的模式
  - “好”的模式不会发生插入异常、删除异常、更新异常，数据冗余应尽可能小
- 原因：关系模式中**数据依赖**
- 解决方法：通过**分解**关系模式来消除其中不合适的数据依赖，且能重构原有模式

8.1 数据依赖对关系模式的影响

- Design by decomposition
  - Start with “mega” relations containing everything
  - Decompose into smaller, better relations with the same information
  - Can do decomposition automatically
- Automatic decomposition
  - “Mega” relations + properties of the data
  - System decomposes based on properties
  - Final set of relations satisfies normal form
    - No anomalies, no lost information

8.1 数据依赖对关系模式的影响

- Properties of the data (数据依赖) and Normal Forms (范式)
  - 函数依赖 → BC范式 (Boyce-Codd Normal Forms)
  - 多值依赖 → 第4范式 (Forth Normal Form)



8.1 数据依赖对关系模式的影响

- 1st Normal Form (1NF) = All tables are flat
- *2nd Normal Form = disused*
- **Boyce-Codd Normal Form (BCNF)**
- **3rd Normal Form (3NF)**

DB designs based on ***functional dependencies***, intended to prevent *data anomalies*

- ***4th Normal Forms***

DB designs based on ***multivalued dependencies***

- *5th Normal Forms = see text books*

8.1 数据依赖对关系模式的影响

- 1st Normal Form (1NF)

Student	Majors
Mary	{CS, EE}
Joe	{GIS, SE}
…	…

Violates 1NF

Student	Major
Mary	CS
Mary	EE
Joe	GIS
Joe	SE

In 1st NF

1NF Constraint: Types must be atomic!

8.1 数据依赖对关系模式的影响

- Apply(SSN, Sname, Cname)
  - 数据冗余；数据更新和删除异常
  - 每个大学都存储了SSN-Sname信息

思考：Apply的key？

- 函数依赖 SSN→ Sname
  - 相同的SSN始终得到相同的Sname
  - f(SSN) = Sname
  - SSN的Sname应该只存储一次
- BC范式：if A→B, then A is a key
- 分解：Student(SSN, Sname), Apply(SSN, Cname)

8.1 数据依赖对关系模式的影响

- Apply(SSN, Cname, HS)
  - 数据冗余；数据更新和删除异常
  - Multiplicative effect
    - C个大学，H个高中，C×H个元组，期望是C+H个元组
  - BC范式并没有解决这一问题，无函数依赖

思考：Apply的key？

- 多值依赖 SSN →→ Cname (SSN↠HS)
  - Given SSN has every combination of Cname and HS
  - SSN的Cname和HS应该都只存储一次
- 第4范式：if A→→B, then A is a key
- 分解：Apply(SSN, Cname), HighSchool(SSN, HS)

思考：如果C=1和H=1， SSN →→ Cname还成立吗？

8.1 数据依赖对关系模式的影响

- Design by decomposition
  - “Mega” relations + properties of the data
  - System decomposes based on properties
  - Final set of relations satisfies normal form
    - No anomalies, no lost information
  - Functional dependencies ➔ Boyce-Codd Normal Form
  - Multivalued dependences ➔ Fourth Normal Form

第八章 关系数据库设计理论

- 8.1 数据依赖对关系模式的影响
- 8.2 函数依赖 (Functional Dependencies)
- 8.3 BC范式 (Boyce-Codd Normal Form)
- 8.4 多值依赖 (Multivalued Dependencies) (自学)
- 8.5 第4范式 (Fourth Normal Form) (自学)
- 8.6 BC范式和第4范式的局限
- 8.7 小结

8.2 函数依赖

- 大学申请数据库(举例)

Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)

Apply (SSN, Cname, state, date, major)

- 假设priority取决于GPA
  - GPA> 3.8 ➔ priority = 1
  - 3.3<GPA<=3.8 ➔ priority = 2
  - GPA<=3.3 ➔priority = 3
- 相同GPA的两个元组具有相同的priority

8.2 函数依赖

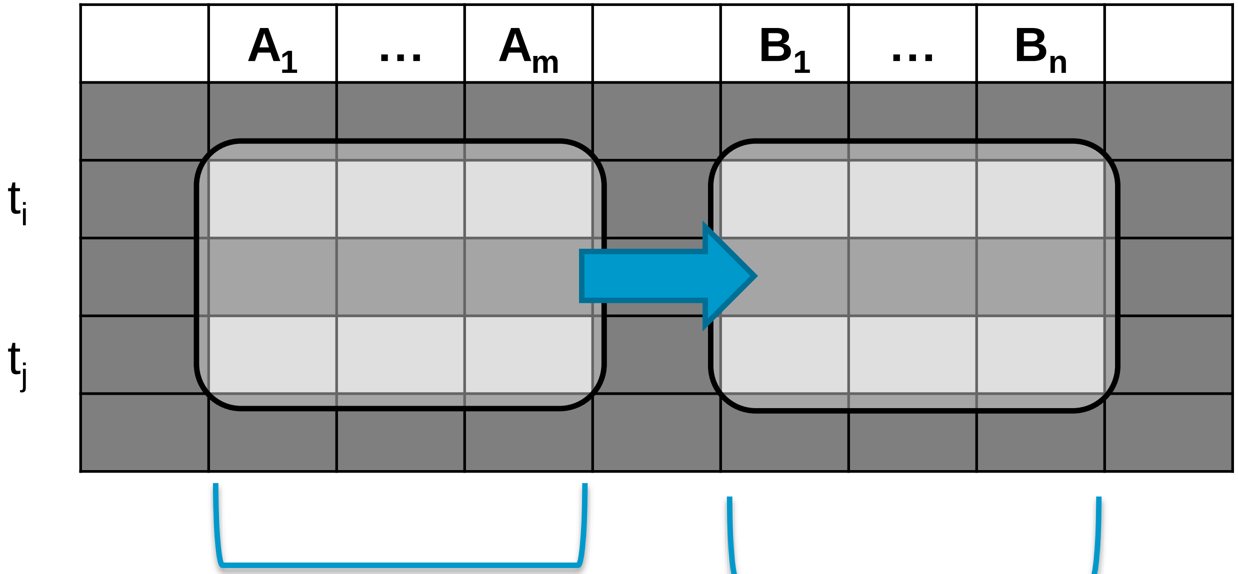
Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)

- 相同GPA的两个元组具有相同的priority
- 函数依赖 Functional Dependencies (FDs)
- 关系R中，属性集A和属性集B存在函数依赖的定义(t,u是关系R的元组)：

$$\forall t, u \in R:$$
$$t[A_1, \ldots A_n] = u[A_1, \ldots A_n] \Rightarrow t[B_1, \ldots B_m] = u[B_1, \ldots B_m]$$

8.2 函数依赖

- 函数依赖从哪儿来？
  - Based on knowledge of real word
    - 需求分析→数据流图&数据字典→E-R图→关系与依赖
  - All instances of relation must adhere
    - R($\bar{A}, \bar{B}, \bar{C}$)，$\bar{A} \rightarrow \bar{B}$



If t1,t2 agree here..   …they also agree here!

Given attribute sets **A={A_1,…,A_m}** and **B = {B_1,…B_n}** in **R,**

The ***functional dependency*** **A→ B on R** holds if for ***any*** t_i,t_j in R:

<u>if</u> $t_i[A_1] = t_j[A_1]$ AND $t_i[A_2]=t_j[A_2]$ AND … AND $t_i[A_m] = t_j[A_m]$

<u>then</u> $t_i[B_1] = t_j[B_1]$ AND $t_i[B_2]=t_j[B_2]$ AND … AND $t_i[B_n] = t_j[B_n]$

8.2 函数依赖

- 关系R(A, B, C, D, E)具有以下2个函数依赖
  - A,B → C
  - C,D → E

假设A，B，D分别最多有3个不同的取值，E最多有多少个不同的取值?

27

8.2 函数依赖

- 关系R(A, B, C)当前仅有元组(0,0,0)，其函数依赖是A→B, B→C，下列哪些元组可以插入关系R，而不违反已有的函数依赖?

  A. (0,1,0)　　B. (0,0,2)　　C. (1,1,0)

  D. (1,0,2)　　E. (0,1,2)　　F. (2,1,0)

  G. (2,0,1)　　H. (1,0,0)

CF or HF

8.2 函数依赖

- 关系Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)存在哪些函数依赖？
  - SSN → Sname
  - SSN → address
  - HScode → HSname, HScity
  - HSname, HScity → HScode
  - SSN → GPA
  - GPA → priority
  - SSN → priority

函数依赖的传递性

8.2 函数依赖

- 关系Apply (SSN, Cname, state, date, major)存在哪些函数依赖?
  - Cname → date
  - SSN, Cname → major ?
  - SSN → state

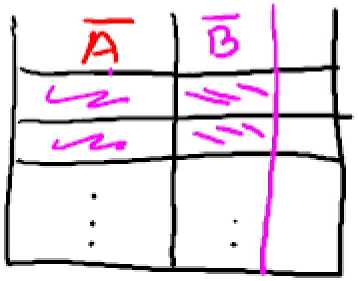
8.2 函数依赖

- 函数依赖与码(Key)的关系
  - Relations with no duplicates
  - R($\bar{A}$, $\bar{B}$), suppose $\bar{A} \rightarrow$ all attributes
    - $\bar{A}$是关系R的码

$\bar{A}$	$\bar{B}$
$\bar{a}$	$\bar{b}$
$\bar{a}$	$\bar{b}$

8.2 函数依赖

- 函数依赖分类
  - 平凡函数依赖 (Trivial FD)
    - If $\bar{A} \rightarrow \bar{B}$, $\bar{B} \subseteq \bar{A}$
  - 非平凡函数依赖 (Nontrivial FD)
    - If $\bar{A} \rightarrow \bar{B}$, $\bar{B} \subseteq \bar{A}$
  - 完全非平凡函数依赖 (Completely nontrivial FD)
    - If $\bar{A} \rightarrow \bar{B}$, $\bar{B} \cap \bar{A} = \emptyset$



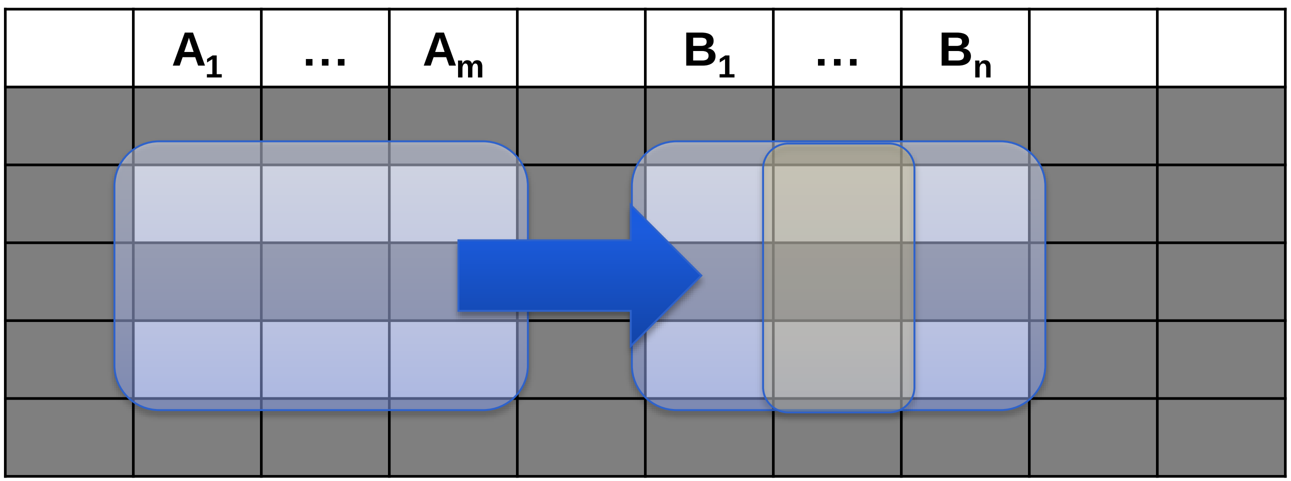
8.2 函数依赖

- High-level idea: why do we care about FDs?
  - Start with some relational schema
  - Find out its functional dependencies (FDs)
  - Use these to design a better schema
    - One which minimizes possibility of anomalies
      - Redundancy
      - Update anomaly
      - Delete anomaly
      - Insert anomaly

8.2 函数依赖

函数依赖规则(Armstrong’s Rules)

- 分解规则(Splitting rule)
  - $\bar{A} \rightarrow B_1, \dots B_m$
  - $\bar{A} \rightarrow B_1, \bar{A} \rightarrow B_2, \dots, \bar{A} \rightarrow B_m$



8.2 函数依赖

函数依赖规则(Armstrong's Rules)

- 分解规则(Splitting rule)
  - $\bar{A} \rightarrow B_1, \ldots B_m$
  - $\bar{A} \rightarrow B_1, \bar{A} \rightarrow B_2, \ldots, \bar{A} \rightarrow B_m$

  - $\cancel{A_1, \ldots A_n \rightarrow \bar{B}}$
  - $\cancel{A_1 \rightarrow \bar{B}, A_2 \rightarrow \bar{B}, \ldots, A_n \rightarrow \bar{B}}$

  - HSname, HScity → HScode

8.2 函数依赖

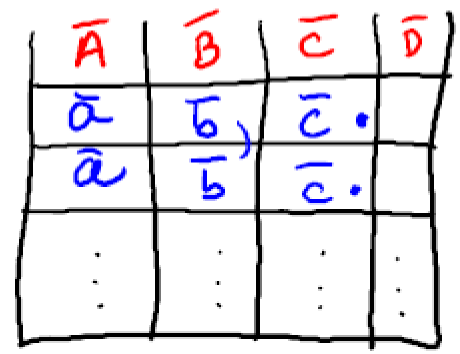
函数依赖规则

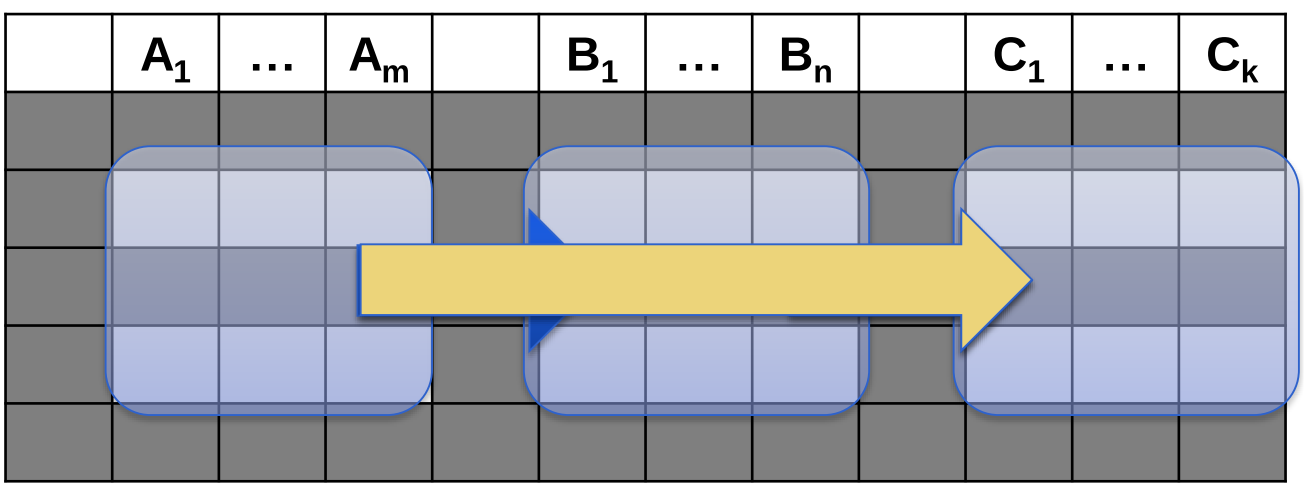
- 分解规则(Splitting rule)
- 合并规则 (Combining rule)
  - $\bar{A} \to B_1, \bar{A} \to B_2, \ldots, \bar{A} \to B_m$
  - $\bar{A} \to B_1, \ldots B_m$
- 平凡依赖规则 (Trivial-dependency rules)
  - If $\bar{A} \to \bar{B}$, $\bar{B} \subseteq \bar{A}$
  - If $\bar{A} \to \bar{B}$, then $\bar{A} \to \bar{A} \cup \bar{B}$
  - If $\bar{A} \to \bar{B}$, then $\bar{A} \to \bar{A} \cap \bar{B}$

8.2 函数依赖

函数依赖规则

- 分解规则(Splitting rule)
- 合并规则 (Combining rule)
- 平凡依赖规则 (Trivial-dependency rules)
- 传递规则 (Transitive rule)
  - If $\bar{A} \rightarrow \bar{B}$, $\bar{B} \rightarrow \bar{C}$, then $\bar{A} \rightarrow \bar{C}$





8.2 函数依赖

- 属性的闭包 (Closure of Attributes)
  - Given relation, FDs, set of attributes $\bar{A}$
  - Find all B such that $\bar{A} \rightarrow B$
  - $\bar{A}^{+}$ $\{A_1, \ldots, A_n\}^{+}$

  - 闭包计算算法

    Start with $\{A_1, \ldots, A_n\}$

    Repeat until no change:

        if $\bar{A} \rightarrow \bar{B}$, and $\bar{A}$ in set

            add $\bar{B}$ to set

8.2 函数依赖

- Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)
  - SSN → Sname, address, GPA
  - GPA → priority
  - HScode → HSname, HScity
- 属性闭包计算 $\{SSN, HScode\}^+ =$

{SSN, HScode, Sname, address, GPA, priority, HSname, HScity}

思考： {SSN, HScode}是关系Student的码？

8.2 函数依赖

- 关系Apply(SSN,cName,state,date,major), what real-world constraint is captured by SSN,cName $\rightarrow$ date?   D

A. A student can only apply to one college.

B. A student can apply to each college only once.

C. A student must apply to all colleges on the same date.

D. Every application from a student to a specific college must be on the same date.

8.2 函数依赖

- 关系R(A, B, C, D, E)的函数依赖：D
  - AB → C
  - C → D
  - BD → E

  E不函数依赖于下列哪个属性集？

  A. ABC　　B. AB　　C. BC　　D.AD

8.2 函数依赖

闭包和码

- Is $\bar{A}$ a key for R? (Given R and FDs)
  - Compute $\bar{A}^+$ if $\bar{A}^+$ = all attributes, then $\bar{A}$ is a key

- How can we find all keys given a set of FDs
  - Consider every subset of attributes as $\bar{A}$
  - If $\bar{A}^+$ = all attributes, then $\bar{A}$ is a key
  - Minimum set → increasing size of the subset

8.2 函数依赖

- 关系R(A, B, C, D, E)的码是？ D
  - A,B → C
  - A,E → D
  - D → B

A. AB　　B. AC　　C. AD　　D. AE

8.2 函数依赖

如何为关系指定函数依赖?

- S1 and S2 sets of FDs
- S2 “follows from” S1 if every relation instance satisfying S1 also satisfies S2
  - S1 = {SSN → GPA, GPA → priority}
  - S2 = {SSN → priority}
- How to test? Does $\bar{A} \rightarrow \bar{B}$ follow from S?
  - Compute $\bar{A}^{+}$ base S check if $\bar{B}$ in set
  - Armstrong’s Axioms

思考：数据库设计时，只需要保留函数依赖S1或S2，还是S1和S2?

8.2 函数依赖

- 关系R(A, B, C, D, E), S1 = {AB → C, AE → D, D → B}, which of the following sets S2 does not follow from S1? C

A. S2 = {AD → C}

B. S2 = {AD → C, AE → B}

C. S2 = {ABC → D, D → B}

D. S2 = {ADE → BC}

8.2 函数依赖

- 如何为关系指定函数依赖？
  - Minimal set of completely nontrivial FDs such that all FDs that hold on the relation follow from the dependencies in the set

8.2 函数依赖

- 需求分析→…→关系和函数依赖
- 函数依赖定义
  - 来源于真实世界(应用需求), 所有元组都要满足
- 函数依赖、闭包与码的关系
  - 计算属性的闭包判断是否为码(Key)
- 函数依赖分类（三类）
- 函数依赖规则
  - splitting, combining, trivial-dependency, transitive
- 如何指定函数依赖(E-R图→关系, 那么FDs？)

第八章 关系数据库设计理论

- 8.1 数据依赖对关系模式的影响
- 8.2 函数依赖 (Functional Dependencies)
- 8.3 BC范式 (Boyce-Codd Normal Form)
- 8.4 多值依赖 (Multivalued Dependencies) (自学)
- 8.5 第4范式 (Fourth Normal Form) (自学)
- 8.6 BC范式和第4范式的局限
- 8.7 小结

8.3 BC范式

- 关系模式的分解

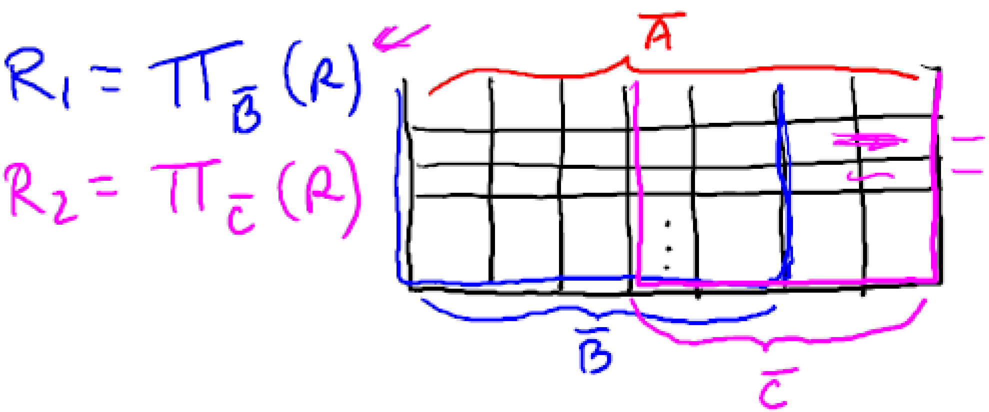
  R(A$_1$, …, A$_n$)　$\bar{A}$

  R1(B$_1$, …, B$_k$)　$\bar{B}$

  R2(C$_1$, …, C$_m$)　$\bar{C}$

$\bar{B} \cup \bar{C} = \bar{A}$

R1 ⋈ R2 = R



8.3 BC范式

$\bar{B} \cup \bar{C} = \bar{A}$

$R1 \bowtie R2 = R$

例1

Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)

以下分解正确(即没有信息丢失)?

S1 (SSN, Sname, address, HScode, GPA, priority)

S2 (HScode, HSname, HScity)

YES!

8.3 BC范式

$\bar{B} \cup \bar{C} = \bar{A}$

$R1 \bowtie R2 = R$

例2

Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)

以下分解正确(即没有信息丢失)?

S1 (SSN, Sname, address, HScode, HSname, HScity)

S2 (Sname, HSname, GPA, priority)

NO!

8.3 BC范式

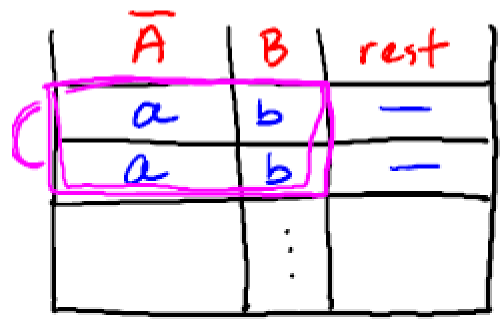
- 基于分解的关系模式设计
  - “Mega” relations + properties of the data
  - System decomposes based on properties
    - “Good” decompositions only (Lossless join property)
    - Into “good” relations (即BCNF)

- A decomposition R to (R1, R2) is lossless if R = R1 Join R2

8.3 BC范式

- Boyce-Codd Normal Form 定义
  - Relation R with FDs is in BCNF if:

    For each $\bar{A} \rightarrow \bar{B}$, $\bar{A}$ is a key
  - BCNF violation



  - $\bar{A}$ contains a key, $\bar{A}$ is a superkey

8.3 BC范式

关系R(A, B, C, D, E)是BC范式，假设ABC是关系R唯一的码，下列哪个函数依赖肯定成立？ D

A. BCE → A　　B. BCDE → A

C. ACD → E　　D. ABCE → D

8.3 BC范式

- 关系Student是否属于BCNF？

Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)

  – SSN → Sname, address, GPA
  – GPA → priority
  – HScode → HSname, HScity

NO!

- 如何判断？
  – 每个函数依赖的左端是否都包含码？
  – Keys: {SSN, HScode}

8.3 BC范式

- 关系Apply是否属于BCNF？

Apply (SSN, Cname, state, date, major)

  – SSN, Cname, state → date, major

YES!

  – Cname → state
  – SSN, Cname → date, major

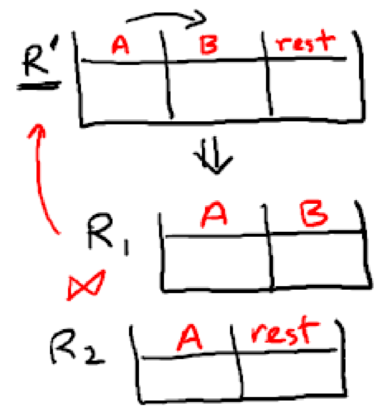
NO!

8.3 BC范式

BCNF decomposition algorithm

Input: relation R + FDs for R

Output: decomposition of R into BCNF relations with "lossless join"



Compute keys for R (using FDs)

Repeat until all relations are in BCNF:

- Pick any R' with $\bar{A} \rightarrow \bar{B}$ that violates BCNF
- Decompose R' into $R_1(\bar{A}, \bar{B})$ and $R_2(\bar{A}, rest)$
- Compute FDs for R_1 and R_2
- Compute keys for R_1 and R_2

8.3 BC范式

例1

Student (SSN, Sname, address, HScode, HSname, HScity, GPA, priority)

- SSN → Sname, address, GPA
- GPA → priority
- HScode → HSname, HScity

S1 (HScode, HSname, HScity)

S2 (GPA, priority)

S3 (SSN, Sname, address, GPA)

S4 (SSN, HScode)

8.3 BC范式

例2

Apply (SSN, Cname, state, date, major) B

- Cname → state
- SSN, Cname → date, major

A. Apply(SSN,cName,state,date,major)

B. A1(cName,state), A2(SSN,cName,date,major)

C. A1(cName,state), A2(SSN,date,major)

D. A1(cName,state), A2(SSN,cName,date), A3(SSN,cName,major)

8.3 BC范式

Compute keys for R (using FDs)

Repeat until all relations are in BCNF:

$\boxed{\text{Pick any R' with } \bar{A} \to \bar{B}}$ that violates BCNF

Decompose R' into $\boxed{R_1(\bar{A}, \bar{B}) \text{ and } R_2(\bar{A}, rest)}$

$\boxed{\text{Compute FDs for } R_1 \text{ and } R_2}$

Compute keys for R_1 and R_2

- 选择不同的R'，最终分解结果是否相同?
- Extend: A → B，A → BA^+
- Implied FDs closure

8.3 BC范式

关系R(A, B, C, D)的函数依赖如下：

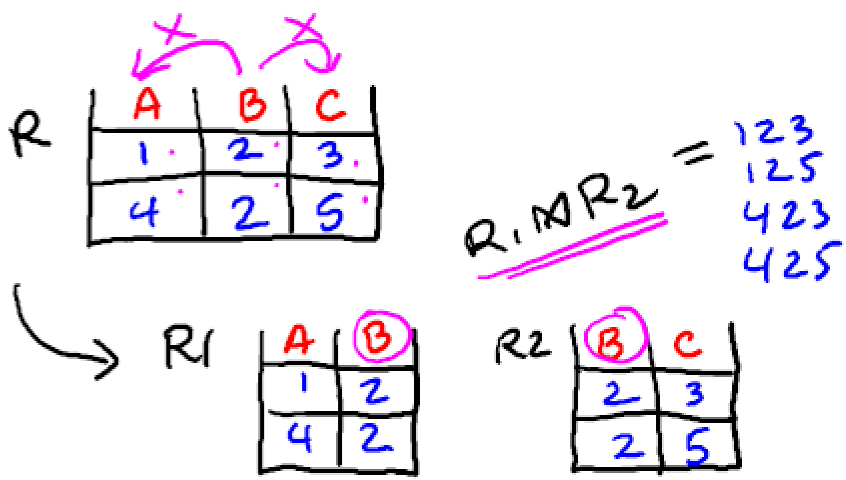
$A \rightarrow B,\ C \rightarrow D,\ AD \rightarrow C,\ BC \rightarrow A$

将R分解成BC范式，下列关系不可能在分解后的结果中？ A

A. ABC　　B. AB　　C. AC　　D. CD

8.3 BC范式

- Does BCNF guarantee a good decomposition?
  - Removes anomalies? YES!
  - Can logically reconstruct original relation?
    - Too few or too many tuples?



第八章 关系数据库设计理论

- 8.1 数据依赖对关系模式的影响
- 8.2 函数依赖 (Functional Dependencies)
- 8.3 BC范式 (Boyce-Codd Normal Form)
- 8.4 多值依赖 (Multivalued Dependencies) (自学)
- 8.5 第4范式 (Fourth Normal Form) (自学)
- 8.6 BC范式和第4范式的局限
- 8.7 小结

8.4 多值依赖

- Apply (SSN, Cname, hobby)
  - FDs? NO!
  - Keys? All attributes!
  - BCNF? YES!
  - Good design? NO!
    - 5 Colleges, 6 hobbies → 30个元组
    - 期望是11个元组
- **多值依赖 (Multivalued Dependencies)**
  - MVDs

8.4 多值依赖

- 多值依赖$\bar{A} \to\to \bar{B}, \bar{A} = A_1, \dots A_n, \bar{B} = B_1, \dots B_m$
  – Based on knowledge of real world
  – All instances of relation must adhere
- 关系R中，属性集A和属性集B存在多值依赖的定义(t, u, v是关系R的元组)：

$$\forall t, u \in R: t[\bar{A}] = u[\bar{A}] \; then$$
$$\exists v \in R: v[\bar{A}] = t[\bar{A}] \; and$$
$$v[\bar{B}] = t[\bar{B}] \; and$$
$$v[rest] = u[rest]$$

	$\bar{A}$	$\bar{B}$	rest
t	$\bar{a}$	$\bar{b}_1$	$\bar{r}_1$
u	$\bar{a}$	$\bar{b}_2$	$\bar{r}_2$
v	$\bar{a}$	$\bar{b}_1$	$\bar{r}_2$
w	$\bar{a}$	$\bar{b}_2$	$\bar{r}_1$

Tuple-generating dependencies

8.4 多值依赖

- 关系R(A, B, C)具有多值依赖 A→→B，假设A至少有3个不同取值，对于每个A，B和C分别至少有4个和5个不同的取值。关系R至少有多少个元组？

60

8.4 多值依赖

- 关系R(A, B, C, D, E)的多值依赖如下：
  - A →→ B
  - B →→ D

  假设R包含元组(0, 1, 2, 3, 4)和(0, 5, 6, 7, 8)，下列哪些元组肯定在关系R中？ C

  A. (0, 5, 6, 3, 4) B. (0, 5, 6, 7, 4)

  C. (0, 1, 6, 3, 8) D. (0, 5, 2, 7, 8)

  (0, 5, 2, 3, 4), (0, 1, 6, 7, 8)

  (0, 1, 2, 7, 4), (0, 1, 6, 3, 8)

  (0, 5, 6, 3, 8), (0, 5, 2, 7, 4)

8.4 多值依赖

- 关系R (A, B, C, D)的实例如下图，对于每个多值依赖，使R满足多值依赖需要增加多少个元组？
  - AB →→ C　　0
  - CD →→ A　　0
  - D →→ C　　4

A	B	C	D
1	2	3	7
1	2	3	8
4	2	5	7
4	2	5	8

8.4 多值依赖

例1

- Apply (SSN, Cname, hobby)
  - SSN →→ Cname
  - SSN →→ hobby

	SSN	CName	hobby
t	123	Stanford	trumpet
u	123	Berkeley	tennis
v	123	Stanford	tennis
w	123	Berkeley	trumpet
	⋮	⋮	⋮

8.4 多值依赖

例2

- Apply (SSN, Cname, hobby)
  - Reveal hobbies to colleges selectively

  - MVDs? NO!
  - Good design? YES!

8.4 多值依赖

例3

- Apply (SSN, Cname, date, major, hobby)
  - Reveal hobbies to colleges selectively
  - Apply once to each college [one day]
  - May apply to multiple majors

  - SSN, Cname → date
  - SSN, Cname, date →→major

8.4 多值依赖

例4

- Apply (SSN, Cname, date, major)
  - SSN, Cname → date

  What real-world constraint is captured by

  SSN →→ Cname, date C

A. A student can only apply to one major at each college.

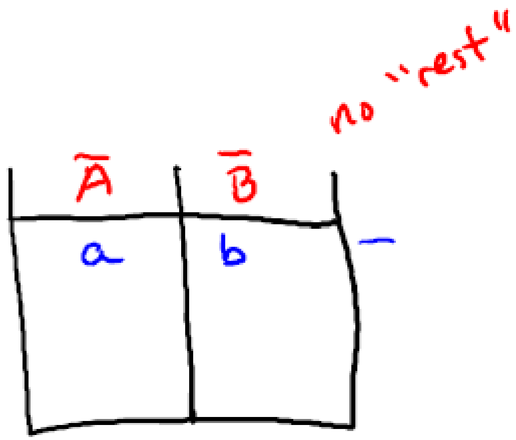
B. A student can apply to different majors at each college, but each major must be applied for on a different date.

C. A student must apply to the same set of majors at all colleges.

D. A student must apply to a different major at each college.

8.4 多值依赖

- 多值依赖分类
  - 平凡多值依赖
    - If $\bar{A} \rightarrow\rightarrow \bar{B}$, $\bar{B} \subseteq \bar{A}$ or $\bar{B} \cup \bar{A}$ = all attributes
  - 非平凡多值依赖
    - otherwise



8.4 多值依赖

- 关系R(A, B, C)的实例如右图

  关系R不具有哪个多值依赖？ D

  A. BC →→ A　　B. BC →→ C

  C. C →→ A　　D. A →→ B

A	B	C
1	2	3
1	3	2
1	2	2
3	2	1
3	2	3

8.4 多值依赖

- 关系R(A, B, C, D)的实例如右图

  关系R具有哪个多值依赖？ A

  A. AB →→ C　　B. B →→ AD

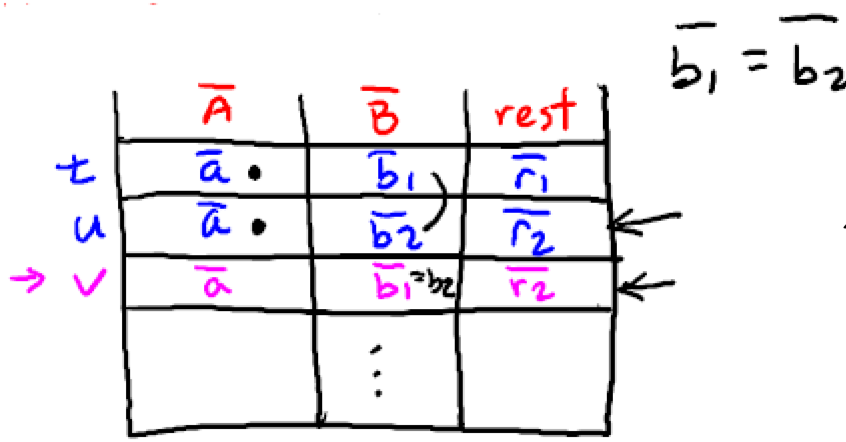
  C. D →→ BC　　D. BD →→ C

A	B	C	D
1	2	3	4
1	3	3	3
1	3	3	4
1	2	3	3
2	2	4	4
2	4	2	4
2	4	4	4
2	2	2	4

8.4 多值依赖

多值依赖规则

- FD is an MVD rule
  - If $\bar{A} \rightarrow \bar{B}$, then $\bar{A} \rightarrow\rightarrow \bar{B}$



思考1：MVD is a FD rule?

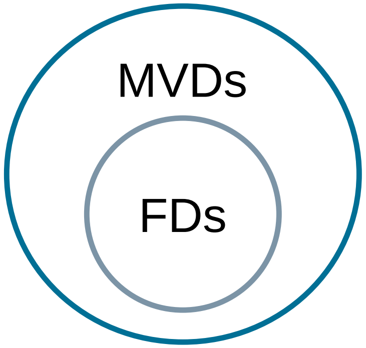
思考2：Union rule?

- Intersection rule
  - If $\bar{A} \rightarrow\rightarrow \bar{B}$, $\bar{A} \rightarrow\rightarrow \bar{C}$, then $\bar{A} \rightarrow\rightarrow \bar{B} \cap \bar{C}$

8.4 多值依赖

多值依赖规则

- FD is an MVD rule
- Intersection rule
- Transitive rule
  - If $\bar{A} \rightarrow\rightarrow \bar{B}$, $\bar{B} \rightarrow\rightarrow \bar{C}$, then $\bar{A} \rightarrow\rightarrow \bar{C} - \bar{B}$



8.4 多值依赖

- 多值依赖定义
  - 来源于真实世界(应用需求), 所有元组都要满足
  - 语义不同时，相同关系可能存在或不存在MVDs
- 多值依赖分类（两类）
- 多值依赖规则
  - FD is an MVD rule
  - Intersection rule
  - Transitive rule
- 多值依赖与函数依赖的关系

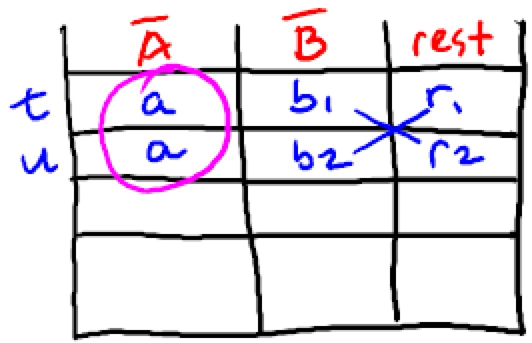
第八章 关系数据库设计理论

- 8.1 数据依赖对关系模式的影响
- 8.2 函数依赖 (Functional Dependencies)
- 8.3 BC范式 (Boyce-Codd Normal Form)
- 8.4 多值依赖 (Multivalued Dependencies) (自学)
- 8.5 第4范式 (Fourth Normal Form) (自学)
- 8.6 BC范式和第4范式的局限
- 8.7 小结

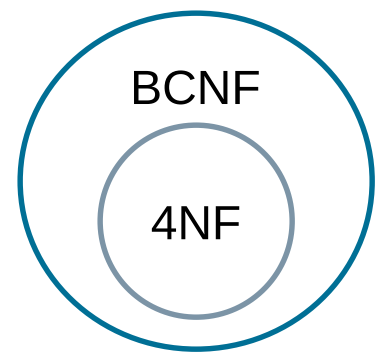
8.5 第4范式

- 第4范式 (Fourth Normal Form) 定义
  - Relation R with MVDs is in 4NF if:

    For each nontrival A→→B, A is a key.



  - 4NF ➔ BCNF
    - A→B ➔ A→→B ➔ A is a key



8.5 第4范式

4NF decomposition algorithm

Input: relation R + FDs for R + MVDs for R

Output: decomposition of R into 4NF relations with "lossless join"

Compute Keys for R

Repeat until all relations are in 4NF:

- Pick any R' with nontrivial A→→B that violates 4NF
- Decompose R' int $R_1(A, B)$ and $R_2(A, rest)$
- Compute FDs and MVDs for R_1 and R_2
- Compute keys for R_1 and R_2

8.5 第4范式

例1

Apply (SSN, Cname, hobby)

- SSN →→ Cname

R1(SSN, Cname)

R2(SSN, hobby)

8.5 第4范式

例2

Apply (SSN, Cname, date, major, hobby)

- SSN, Cname → date
- SSN, Cname, date →→ major

R1(SSN, Cname, date)

R2(SSN, Cname, major)

R3(SSN, Cname, hobby)

8.5 第4范式

例3

StudentInfo (SID, name, dorm, major)   D

- SID → name
- SID →→ dorm

A. StudentInfo(sID, name, dorm, major)

B. S1(sID, name), S2(sID, dorm, major)

C. S1(sID, name, dorm), S2(sID, major)

D. S1(sID, name), S2(sID, dorm), S3(sID, major)

8.5 第4范式

例4

关系R(A, B, C, D, E)的多值依赖如下：

- A →→ B
- B →→ D

选择不同顺序的多值依赖将产生不同的分解结果，关系R分解为第4范式的结果是？

{AB,AD,ACE}

{BD,AB,ACE}

8.5 第4范式

例5

关系R(A, B, C, D)的数据依赖如下：

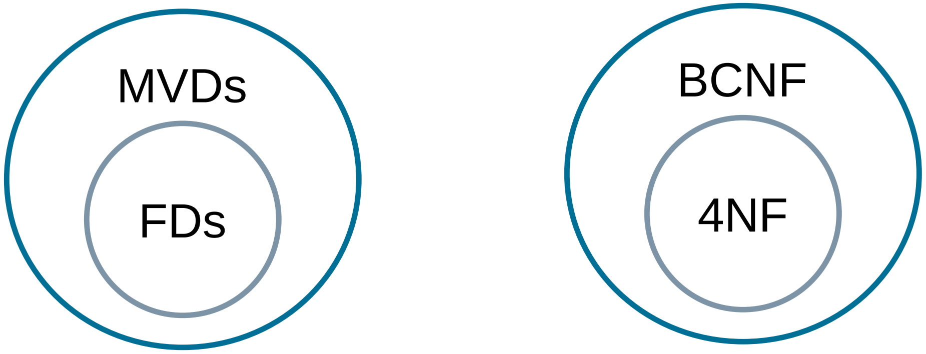
- A → B
- C → D
- B →→ C

将R分解成第4范式，选择不同顺序的多值依赖将产生不同的分解结果，下列哪个关系能够在最终分解的结果中？C

A. ABCD　　B. ABC　　C. CD　　D. ACD

8.5 第4范式

- 关系模式设计理论
  - 函数依赖 & BC范式
    - R(A, B, C) A→B
  - 多值依赖 & 4NF
    - R(A, B, C, D) A→→B



第八章 关系数据库设计理论

- 8.1 数据依赖对关系模式的影响
- 8.2 函数依赖 (Functional Dependencies)
- 8.3 BC范式 (Boyce-Codd Normal Form)
- 8.4 多值依赖 (Multivalued Dependencies) (自学)
- 8.5 第4范式 (Fourth Normal Form) (自学)
- 8.6 BC范式和第4范式的局限
- 8.7 小结

8.6 BC范式和第4范式的局限

- BC范式
  - Relation R with FDs is in BCNF if:

    For each $A \rightarrow B$, A is a key
- 4范式
  - Relation R with MVDs is in 4NF if:

    For each nontrivial $A \rightarrow\rightarrow B$, A is a key

8.6 BC范式和第4范式的局限

例1

- Apply (SSN, Cname, date, major)
  - Can apply to each college once for one major
  - Colleges have non-overlapping application dates

函数依赖在分解后的关系中丢失

- FDs: SSN, Cname → date, major date → Cname
- Keys: SSN, Cname
- BCNF: NO. R1(date, Cname), R2(SSN, date, major)
- Good design? Not necessarily. 3rd Normal Form

8.6 BC范式和第4范式的局限

例2

- Student (SSN, HSname, GPA, priority)
  - Multiple HS okay, priority determined from GPA

函数依赖在分解后的关系中丢失

- FDs: SSN → GPA $\boxed{\text{GPA → priority}}$ SSN → priority
- Keys: SSN, HSname

选择不同函数依赖顺序会得到不同分解结果

- BCNF: NO. R1(SSN, priority), R2(SSN, GPA), R3(SSN, HSname)
- Good design? Not necessarily. 3rd Normal Form

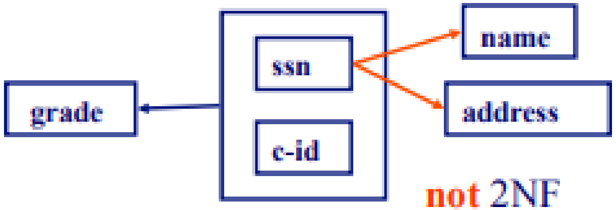
8.6 BC范式和第4范式的局限

- 1^{st} Normal Form (1NF)
  - Attributes are atomic (no set-valued attr.)

Ssn	Name	Dependents
123	Smith	Peter Mary John
234	Jones	Ann Michael

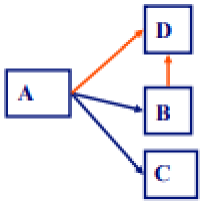
not 1NF

- 2^{nd} Normal Form (2NF)
  - 1NF + non-key attributes fully depend on the key
  - Example: Takes(ssn, cid, grade, name, address)
    - ssn → name, address
    - ssn, cid → grade



not 2NF

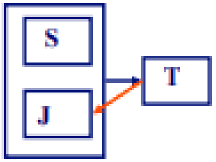
- 3^{rd} Normal Form (3NF)
  - 2NF + no transitive depdencies



in 2NF, but not in 3NF

8.6 BC范式和第4范式的局限

- A relation R is in 3NF if for every FD A → B
  - It is trivial (If $\bar{A} \rightarrow \bar{B}$, $\bar{B} \subseteq \bar{A}$)
  - A is a superkey (good FDs)
  - B is part of a candiate key (A may not be a superkey)
- STJ(Student, Teacher, subJect)



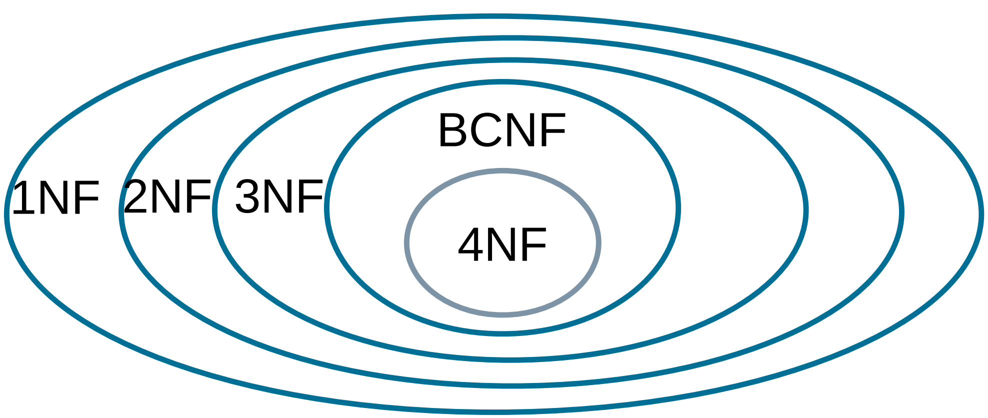
  - S, J → T, T → J
  - BCNF: R1(T, J), R2(S, T) [dependency perserving?]
  - 3NF forgives the red arrow
- In practice, aim for
  - BCNF: lossless join, dependency preservation
  - If impossible, 3NF: lossless join, dependency preservation

8.6 BC范式和第4范式的局限

- BC范式
  - Relation R with FDs is in BCNF if:

    For each A→B, A is a key
- 4范式
  - Relation R with MVDs is in 4NF if:

    For each nontrivial A→→B, A is a key
- After decomposition, no guarantee dependencies can be checked on decomposed relations



8.6 BC范式和第4范式的局限

例3

“Denormalized” relation

- Scores (SSN, Sname, SAT, ACT)
  - Multiple SATs and ACTs allowed
  - All queries return name + composite score for SSN (此时更倾向于原始关系，包含所有属性)

- FDs + keys: SSN → Sname. Key(SSN, SAT, ACT)
- MVDs: SSN, Sname →→ SAT
- 4NF: NO. R1(SSN, Sname), R2(SSN, SAT), R3(SSN, ACT)

8.6 BC范式和第4范式的局限

例4

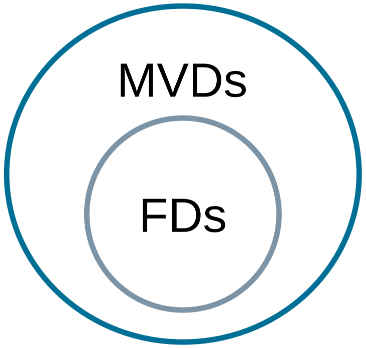
- College (Cname, state)
- CollegeSize (Cname, enrollment)
- CollegeScores (Cname, avgSAT)
- CollegeGrades (Cname, avgGPA)
- …

Too decomposed

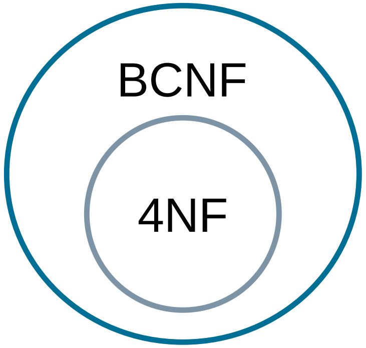
- BCNF/4NF? Yes.
- Good design? Not necessarily.
  - 可用其他4范式关系表达，且包含信息相同

8.7 关系数据库设计理论小结

- Designing a database schema
  - Usually many designs possible
  - Some are (much) better than others!
  - How do we choose?



- Very nice theory for relational DB design
  - Normal forms – “good” relations
  - Design by decomposition
  - Usually intuitive and works well
  - Some shortcomings (Dependency enforcement, Query workload, Over-decomposition)



练习

- 假设关系R(A, B, C, D, E)有如下函数依赖：
  - AB → C
  - BC → D
  - CD → E
  - DE → A
- 关系R所有的keys
- 将关系R规范化为BCNF
- 是否存在不同的分解？

练习

- 将具有如下约束的下列模式规范化为4NF

books(accessionno, isbn, title, author, publisher)

users(userid, name, deptid, deptname)

accessionno → isbn

isbn → title

isbn → publisher

isbn →→ author

userid → name

userid → deptid

deptid → deptname