第六章 关系数据理论

- 学习内容
  - 6.1 关系模式设计的问题
  - 6.2 规范化
  - 6.3 函数依赖的推理规则
  - 6.4 模式分解

•学习目标

- 理解数据库模式设计的数据语义问题
- 掌握函数依赖的概念
- 掌握1NF,2NF,3NF的概念及判断
- 掌握Armstrong公理及其推理规则
- 掌握模式分解的基本概念以及无损连接性的判断方法

6.1 关系模式的设计问题

- 6.1.1 关系数据模型的简单回顾
- 6.1.2 数据库设计中的数据语义问题

6.1.1 关系数据模型的简单回顾

- $R(A_1/D_1, A_2/D_2,\cdots, A_n/D_n)$

- R（U, D, DOM, F）
  - 关系名R，它是符号化的元组语义；
  - 一组属性U；
  - 属性组U中属性所来自的域D；
  - 属性到域的映射DOM；
  - 属性组U上的一组数据依赖F

- R（U, F）

6.1.2 数据库设计中的数据语义问题

- 1. 示例关系

  考虑为管理职工的工资信息而设计一个关系模式

职工	级别	工资
赵明	4	500
钱广	5	600
孙志	6	700
李开	5	600
周祥	6	700

6.1.2 数据库设计中的数据语义问题(续)

- 2. 示例关系的问题:

  - (1) 信息的不可表示问题
    - 插入异常:
      - 如果没有职工具有8级工资，则8级工资的工资数额就难以插入
    - 删除异常:
      - 如果仅有职工赵明具有4级工资，如果将赵明删除，则有关4级工资的工资数额信息也随之删除了

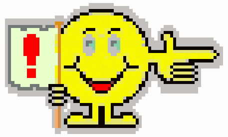
6.1.2 数据库设计中的数据语义问题(续)

- 2. 示例关系的问题:

  - (2) 信息的冗余问题
    - 数据冗余
      - 职工很多，工资级别有限，每一级别的工资数额反复存储多次
    - 更新异常
      - 如果将5级工资的工资数额调为620，则需要找到每个具有5级工资的职工，逐一修改

6.1.2 数据库设计中的数据语义问题(续)

- 3. 问题的解决方法



分解！分解！！再分解！！！

职工	级别
赵明	4
钱广	5
孙志	6
李开	5
周祥	6

级别	工资
4	500
5	600
6	700

6.1.2 数据库设计中的数据语义问题(续)

- 3. 问题的解决方法
  - **探讨： 引入空值能否解决问题**

职工	级别	工资
赵明	4	500
钱广	5	600
Null	6	700
李开	null	null
周祥	6	700

6.1.2 数据库设计中的数据语义问题(续)

- 4. 有关学生的关系模式*S(Sno , SN , SD , DEAN , Cno , G)*

S#	SN	SD	DEAN	C#	G
S01	杨明	D01	思齐	C01	90
S02	李婉	D01	思齐	C01	87
S01	杨明	D01	思齐	C02	90
S03	刘海	D02	述圣	C01	87
S04	安然	D02	述圣	C02	78
S05	乐天	D03	省身	c03	82

- **该关系模式存在哪些问题**
- **问题产生的原因**

- 数据库设计中的数据语义问题(续)
  - 补充说明
    - 数据依赖
      - 通过一个关系中属性间值的相等与否体现出来的数据间的相互关系，是现实世界属性间相互联系的抽象，是语义的体现。
      - 数据依赖的类型：函数依赖，多值依赖

数据库设计中的数据语义问题(续)

- 关系模式*S(Sno , SN , SD , DEAN , Cno , G)*在现实世界中的体现的属性之间的依赖关系
  - 一个系由若干学生，但一个学生只属于一个系（1-n）
    Sno -> SD
  - 一个系只有一名主任
    SD -> DEAN
  - 每个学生学习一个课程，都有一个成绩G
    (Sno, Cno) -> G

数据库设计中的数据语义问题(续)

- 插入异常 ：

  应该插入的数据未被插入。

- 删除异常

  不该删除的数据被删除。

- 数据冗余和更新问题

  不必要地重复存储某些属性的值;

  更新操作代价非常大。

数据库设计中的数据语义问题(续)

- 职工关系模式E(EN,R,S) / E(Ename, Rating, Salary)能够通过引用空值来解决问题
  - 不能
  - 原因:
    - 若主码为空，违背关系模式中主码不能为空

数据库设计中的数据语义问题(续)

属性间联系

- 1-1
- 1-M
- N-M

6.2 规范化

- 6.2.1 函数依赖
- 6.2.2 码
- 6.2.3 范式
- 6.3.4 小结

6.2.1 函数依赖 (续)

- 1. 定义

设R(U)是属性集U上的关系模式，$X, Y \subseteq U$， r是R(U) 上的任意一个关系，如果成立

对$\forall t, s \in r$，若$t[X] = s[X]$，则$t[Y] = s[Y]$

那么称“X**函数决定**Y”，或“Y**函数依赖**于X”，记作**X→Y**

称X为**决定因素**

如Sno → SN，（Sno，Cno）→ G

6.2.1 函数依赖 (续)

- Ex 1: 辨析下列关系模式中的函数依赖

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

解答： A -> C , AB->C…

6.2.1 函数依赖 (续)

- Ex 2: 辨析下列关系模式中的函数依赖

A	B	C
1	2	3
4	2	3
5	3	3

解答：A->BC, B->C,AB->C…

6.2.1 函数依赖 (续)

- 2. 相关说明
  - 函数依赖成立的条件
  - 平凡的函数依赖

    如果$X \rightarrow Y$，但$Y \subseteq X$，则称其为平凡的函数依赖，否则称为非平凡的函数依赖

    如（Sno，SN）$\rightarrow$ SN是平凡的函数依赖

6.2.1 函数依赖 (续)

- 2. 相关说明

  - 部分函数依赖

在R(U)中，如果X→Y，且对于任意X的真子集X',都有 $X' \not\rightarrow Y$ 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 $X \xrightarrow{P} Y$

思考：找出S中的部分函数依赖

6.2.1 函数依赖 (续)

- 2. 相关说明
  - 传递函数依赖

在R(U)中，如果

$X \rightarrow Y$，$Y \rightarrow Z$，$Y \not\rightarrow X$，且 $Y \not\subseteq X$

则称Z对X传递函数依赖

$Sno \rightarrow SD$，$SD \rightarrow DEAN$

思考：找出职工工资表中的传递函数依赖

6.2.2 码

- 候选码

  设K为R< U , F >的属性或属性组合，若K U F 则称K为R的候选码

- 主码

  若R(U , F)有多个候选码，则可以从中选定一个作为R的主码

- 主属性/非主属性

  包含/（不包含）在任一个候选码中的属性，称作主/(非主)属性

- 全码

示例

关系模式*S(Sno , SN , SD , DEAN , Cno , G)*

主码：（Sno，Cno）

函数依赖：

（Sno，Cno） G

Sno $\rightarrow$ SN，（Sno，Cno）$\xrightarrow{p}$ SN

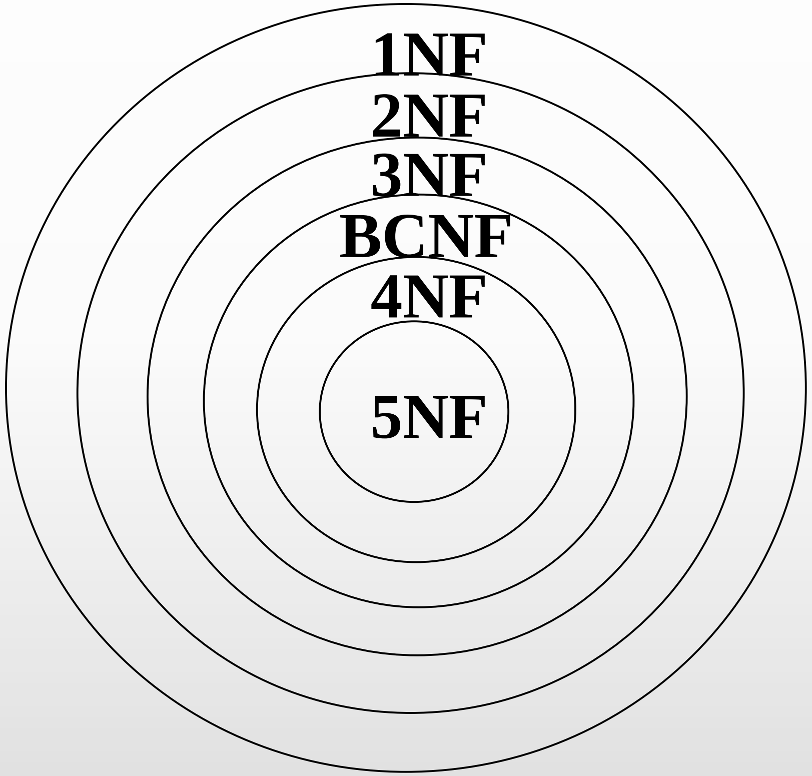
Sno $\rightarrow$ SD，（Sno，Cno）$\xrightarrow{p}$ SD

SD $\rightarrow$ DEAN

6.2.3 范式

- 1. 定义
  - 范式
    - 范式是对关系的不同数据依赖程度的要求
  - 规范化
    - 通过模式分解将一个低级范式转换为若干个高级范式的过程称作规范化

范式关系图



现实世界

6.2.3 范式(续)

- 2. 1NF
  - 关系中每一分量不可再分。即不能以集合、序列等作为属性值

Sno	Cno
S1	{C1，C2，C3}

Sno	Cno
S1	C1
S1	C2
S1	C3

6.2.3 范式(续)

- 2. 1NF
  - 分量是否需要再分，与具体应用有关。如果用到值的一部分，则需要进一步分割

姓名	生日
王军	68.7.10
张立	69.7.10
李明	80.3.28

姓名	年	月日
王军	68	7.10
张立	69	7.10
李明	80	3.28

- 如果只是查询出生日期，满足1NF?

- 如果查询两人生日是否相同，则只比较月、日，需要将生日分解，满足1NF?

1NF 练习

- Ex1: 假设部门关系Dept如下定义：

  部门（部门号，部门名，部门成员，部门总经理）

  请问这个关系模式是否满足1NF的定义？如不满足，是因为哪一个属性使它不满足1NF

  A. 部门总经理　　C. 部门成员

  B. 部门名　　　　D. 部门号

- Ex2: 一定要分解成1NF的组合属性?

6.2.3 范式(续)

- 3. 2NF

  - 关系模式
    *S(Sno , SN , SD , DEAN , Cno , G)*的问题

    - 插入异常
    - 删除异常
    - 更新异常
    - 数据冗余

S#	SN	SD	DEAN	C#	G
S01	杨明	D01	思齐	C01	90
S02	李婉	D01	思齐	C01	87
S01	杨明	D01	思齐	C02	90
S03	刘海	D02	述圣	C01	87
S04	安然	D02	述圣	C02	78
S05	乐天	D03	省身	C01	82

6.2.3 范式(续)

- 3. 2NF
  - 2NF的定义
    - 若R∈1NF，且每个非主属性完全依赖于码，则称R∈2NF
    - 消除非主属性对码的部分依赖

      如S∉2NF，因为

$$(\text{Sno}，\text{Cno}) \xrightarrow{p} \text{SN}$$

$$(\text{Sno}，\text{Cno}) \xrightarrow{p} \text{SD}$$

6.2.3 范式(续)

- 3. 2NF
  - 1NF到2NF的改造
    - 非主属性有两种，一种完全依赖于码，一种部分依赖于码。
      将S分解为：
      SC(Sno , Cno , G)
      S_SD(Sno , SN , SD , DEAN)
    - 消除非主属性对码的部分函数依赖

改造结果

Sno	SD	SD	DEAN
S01	杨明	D01	思齐
S02	李婉	D01	思齐
S03	刘海	D02	述圣
S04	安然	D02	述圣
S05	乐天	D03	省身

Sno	Cno	G
S01	C01	90
S01	C02	87
S02	C01	90
S03	C01	87
S04	C02	78
S05	C01	82

SC(Sno , Cno , G)

S_SD(Sno , SN , SD , DEAN)

2NF 练习

- Ex 1:
  关系模式R（A，B，C，D），码为AB，给出它的一个函数依赖集，使得R属于1NF而不属于2NF

解答：AB->CD, B->C

6.2.3 范式(续)

- 4. 3NF
  - 关系模式
    S_SD(Sno , SN , SD , DEAN)
    SC(Sno , Cno , G) 的问题
    - 插入异常
    - 删除异常
    - 更新异常
    - 数据冗余

6.2.3 范式(续)

- 4. 3NF
  - 3NF的定义
    - 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$R \in 3NF$

- 消除非主属性对码的传递依赖

如S_SD $\notin$3NF，因为有Sno→SD，SD→DEAN

6.2.3 范式(续)

- 4. 3NF

  - 2NF到3NF的改造
    - 将S分解为

      STUDENT(Sno , SN , SD)

      DEPT(SD , DEAN)

    R=>SC(Sno,Cno,G),STUDENT (Sno , SN , SD),DEPT(SD , DEAN)

  - Ex :

    关系模式R（A，B，C，D），码为AB，给出它的一个函数依赖集，使得R属于2NF而不属于3NF

6.2.3 范式(续)

- 6. BCNF
  - 示例

    STC(Sno , Tno , Cno),

    Tno → Cno，每位老师只教授一门课

    （Sno，Tno）→ Cno

    （Sno，Cno）→ Tno，某学生选定一门课，就对应一位老师

    （Sno，Tno），（Sno，Cno）为候选码。

  - 问题 STC ∈ 3NF？

6.2.3 范式(续)

- 6. BCNF
  - 3NF的问题　STC(Sno , Tno , Cno)
    - 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
    - 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
    - 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
  - 原因

    主属性对码的不良依赖(部分依赖)

6.2.3 范式(续)

- 6. BCNF
  - 定义
    - 关系模式R< U , F >中，对于属性组X，Y，若X→Y且Y⊄X时X必含有码，则R< U , F > ∈BCNF

      如STC ∉ BCNF，因为Tno → Cno，而Tno是候选码的一部分
  - 改造

    将S分解为（Sno，Cno），（Tno，Cno）

BCNF练习

- Ex 1:

  关系模式SCO(Sno , Cno , O)，表示学生选修课程的名次，有函数依赖(Sno，Cno)→ O， (Cno，O) → Sno，它属于BCNF吗?

  解答:

  - (Sno,Cno)或者(Cno,O)都可以作为候选码
  - 不存在属性对码传递依赖或部分依赖，SCO∈3NF
  - 没有其他决定因素，SCO ∈BCNF

练习

- Ex 1: 关系模式中，满足2NF的模式，——。

  A. 可能是1NF　　C. 必定是1NF

  B. 必定是3NF　　D. 必定是BCNF

  解答:

- Ex 2: 设有如图所示的关系R，它是——

  A. 1NF

  B. 2NF

  C. 3NF

  解答：

材料号	材料名	生产厂
M1	线材	武汉
M2	型材	武汉
M3	板材	广东
M4	型材	武汉

6.2.3 范式(续)

BCNF性质

1）所有非主属性都完全fd于候选码;

2）所有非主属性都不传递fd于候选码;

3）所有主属性都完全fd于不包含它的候选码;

4）所有主属性都不传递fd于候选码。

定理：如果R∈BCNF，则R∈3NF

证明：（反证法）设R∈BCNF，但R∉3NF,则总可找到属性集X，Y，Z，其中X为候选玛，Y为某一个属性组，Z为非主属性（R ∉ 3NF，则存在非主属性Z，它们存在传递fd），使得X $\rightarrow$ Y, Y $\rightarrow$Z，Y$\not\rightarrow$X成立，即X$\xrightarrow{t}$Z，而Y不包含R的候选码X，但Y→Z成立（Y是非主属性，这样决定因素不含候选码）。

根据BCNF定义，R ∉ BCNF，与假设矛盾。

定理得证。

6.2.3 范式(续)

6　关系规范化小结

6.1关系规范化过程

非规范关系

去掉嵌套属性或重复组(使每个属性及其值都成为不可再分的数据项)

1NF

消去非主属性对候选KEY的部分fd

2NF

消去非主属性对候选KEY的传递fd

3NF

消去主属性对候选KEY的部分和传递fd

BCNF

6.2.3 范式(续)

6. 2、结论

1）3NF必定为2NF和1NF，反之不一定；
2）BCNF必为3NF，反之不一定；
3）3NF已在很大程度上控制了数据冗余；
4）3NF已在很大程度上消去了插入和删除操作异常；
5）3NF分解仍不够彻底（可能存在主属性对候选码的部分fd和传递fd）；
6）在fd范围内，BCNF下已完全消去了插入删除异常；
7）范式并非越高越好；范式越高，异常越少，但查询操作越麻烦；
8）适可而止：
理论上：一般到3NF
应用：存取垃圾；连接运算。
9）分解不唯一。

6.2.3 范式(续)

7 多值依赖（MVD：multivalued dependency）

先看一个例子：

例R（其中KM：课程名，JSM:教师名，SM:参考书名）

非规范化的关系：

K M	JSM	SM
数学	邓军	数学分析
	陈斯	高等代数
		微分方程
物理	李平	普通物理学
	王强	光学原理
	刘明	

K M	JSM	SM
数学	邓军	数学分析
数学	邓军	高等代数
数学	邓军	微分方程
数学	陈斯	数学分析
数学	陈斯	高等代数
数学	陈斯	微分方程
物理	邓军	普通物理学
物理	邓军	光学原理
物理	王强	普通物理学
物理	王强	光学原理
物理	刘明	普通物理学
物理	刘明	光学原理

6.2.3 范式(续)

4、多值依赖的形式定义

任给R（U），x、y、z为U中子集，Z=U-x-y，γ是R（U）中任意一个关系集，t、s是γ的任意两个元组。若t[x]=s[x]，必有γ的两个元组u、v存在（u、v可与t、s相同），使得:

1）u[x]=v[x]=t[x]=s[x];

2）u[y]=t[y]且u[z]=s[z];

3）v[y]=s[y]且v[z]=t[z]。

则称y多值依赖于x。

换句话说:

任给R（U），γ是其任意关系集，若γ中有两个元组在x属性上的值相等，则交换这两个元组在y上的属性值，所得两个新元组仍为γ中的元组。

6.2.3 范式(续)

例R（其中KM：课程名，JSM：教师名，SM：参考书名）

非规范化的关系：

K M	JSM	SM
数学	邓军	数学分析
	陈斯	高等代数
		微分方程
物理	李平	普通物理学
	王强	光学原理
	刘明	

K M	JSM	SM
数学	邓军	数学分析
数学	邓军	高等代数
数学	邓军	微分方程
数学	陈斯	数学分析
数学	陈斯	高等代数
数学	陈斯	微分方程
物理	邓军	普通物理学
物理	邓军	光学原理
物理	王强	普通物理学
物理	王强	光学原理
物理	刘明	普通物理学
物理	刘明	光学原理
……	……	……

6.2.3 范式(续)

1）非平凡多值依赖:

∵ （KM， SM）→→JSM

且JSM与SM无关（无论哪一组为参考书，其对应JSM不变）。

如：{数学，数学分析}→→{邓军，陈斯}

{数学，高等代数}→→{邓军，陈斯}

∴根据MVD定义：KM→→JSM

同理可得：KM→→SM

2）R∈BCNF

码：（KM, JSM， SM）

6.2.3 范式(续)

7.2　MVD性质

1、对称性规则

若x→→y，则x→→z，其中z=U-x-y

如R中：KM→→JSM

根据对称性：KM→→SM　　z=U-x-y=U-KM-JSM=SM

2、传递性规则

若x→→y，y→→z，则x→→z

3、复制规则

若x→y，则x→→y

（x→y是x→→y的一子类，前者一对一，后者一对多）

4、并规则

若x→→y，x→→z，则x→→yz

5、交规则

若x→→y，x→→z，则x→→y∩z

6.2.3 范式(续)

6、差规则

若 x→→y， x→→z， 则 x→→（y−z），x→→（z−y）

7、伪传递规则

若x→→y， wy→z， 则wx→→（z−w−y）

6.2.3 范式(续)

7.3　存在问题

1、数据冗余

有多少个教师上课，就有多少套参考书重复存放。

2、修改麻烦

同一门课程参考书修改，须修改很多元组（有多少个教师任课，就必须修改多少套）。

3、插入麻烦

同一门课增加一任课教师，则插入多个元组（多本参考书）。

如物理课增加一名讲课教员田野，必须插入两个元组：

(物理，田野，普通物理学)

(物理，田野，光学原理)

6.2.3 范式(续)

4、删除麻烦

同一门课程减少一本参考书，有多少个教师任课，则须删除多少个元组。

如删除微分方程，则必须删除两个元组：

（数学，邓军，微分方程）

（数学，陈斯，微分方程）

6.2.3 范式(续)

7.4　原因

SM和JSM的值相互独立，只与KM相关，即存在KM→→JSM或KM→→SM。

4.4.7.5　投影分解

——消去非平凡多值依赖KM→JSM

R1

KM	JSM
数学	邓军
数学	陈斯
物理	邓军
物理	王强
物理	刘明

R2

KM	SM
数学	数学分析
数学	高等代数
数学	微分方程
物理	普通物理学
物理	光学原理

分解后，尽管还存在KM→→JSM，KM→→SM，但这是平凡多值依赖。

6.2.3 范式(续)

7.6　效果

1、冗余避免

同一套参考书保存一次。

2、修改麻烦避免了

参考书修改，在R2中修改一次。

3、插入麻烦避免了

某一门课增加一任课教师，只需在R1中插入一个元组。

4、删除麻烦避免了

某一门课减少一本参考书，仅需在R2中删除一个元组。

6.2.3 范式(续)

7.7　4NF

定义：任给关系模式R（U，F），若R∈1NF，且对于F中的任给非平凡多值依赖x→→y（yx），x都含有候选码，则R∈4NF。

R：码：（JSM，SM）

∵KM→→JSM

KM→→SM

左部均不含候选码

∴R4NF

R1：KM→→JSM，但z=φ，平凡多值依赖；

R2：KM→→SM，但z=φ，平凡多值依赖，

∴R1，R2中不存在非平凡多值依赖。

即：R1∈4NF

R2∈4NF

6.2.3 范式(续)

7.8　定理：4NF必定为BCNF

证明（反证法）

设R(U),R∈4NF，但R $\notin$ BCNF，则R中必存在某个非平凡函数依赖x→y，但x不是R的候选码。据非平凡fd定义，若xy=U，则显然x是R的候选码，这与假设矛盾；若xy≠U，从x→y，可据复制规则（x→y，则x→→y），有x→→y成立且为非凡平多值依赖，此时x不是R的候选码，则违反4NF条件，也与假设矛盾。

∴R不是BCNF的假设不成立，R必为BCNF。

规范化小结

- 关系模式规范化的思想
  - 逐步消除数据依赖中不合适的部分，使模式中的各关系模式达到某种程度的分离
  - “一事一地”，概念的单一化，采用投影的方法
  - 关系规范化在现实应用中可在任一步终止

规范化的步骤

1NF

2NF

3NF

BCNF

4NF

消除非主属性对码的部分函数依赖

消除非主属性对码的传递函数依赖

消除主属性对码的传递、部分函数依赖

消除非平凡的多值依赖

6.3 函数依赖的公理系统

6.3.1函数依赖的逻辑蕴涵

1．逻辑蕴涵的定义

研究的问题：对于给定的一组函数依赖，要判断另外的一些函数依赖是否成立?

例如:对关系模式R(A，B，C)。已知A→B，B→C，判断是否有A→C?

定义:

设F是关系模式R的一个函数依赖集，X，Y是R的属性子集，如果从F中的函数依赖能够推导出X→Y（对R的任意一个r都成立），则称F逻辑蕴涵X→Y，或X→Y是F的逻辑蕴涵。

6.3 函数依赖的公理系统

2．F的闭包F+

定义:

所有被F逻辑蕴含的函数依赖的集合称为F的闭包（Closure），记为F+。即：F+是所有能从F中推导出来的函数依赖的集合。

通常F⊂F+, 若F= F+, 则称F是函数依赖的完备集。

3．说明

F+的计算相当麻烦;

F不大，但F+可能很大（NP难度问题）。

例：若有F={A→B1, A→B2, …, A→Bn}

则F+:

所有形如X→Y的fd，其中Y={B1, B2, …, Bn}, 2^n个fd;

6.3 函数依赖的公理系统

6.3.2公理的作用

1.从已知的F出发，推出F+中的所有函数依赖。

2.已知F和X，Y，判断X→Y是否在F+中。

要求有一套正确和完备的推理规则：

Armstrong推理规则系统——阿氏公理；

1974年，W.W.Armstrong总结了各种规则，将其中最主要、最基本的作为公理，形成了最著名的Armstrong公理系统——简称阿氏公理。

公理系统是模式分解算法的理论基础。

6.3 函数依赖的公理系统

6.3.3公理的内容

设有关系模式R（U，F），U={A1,A2,…,An}为属性全集，F是R的函数依赖集，X，Y，Z ⊆ U。则有：

- 自反律（Reflexivity）

  若Y ⊆ X，则X→Y；

  即若Y ⊆ X ⊆ U则X→Y为F所蕴涵。

- 增广律（Augmentation）

  若X→Y，Z ⊆ U，则XZ→YZ。

  即若X→Y为F的逻辑蕴涵，Z ⊆ U，则XZ→YZ为F的逻辑蕴涵。

- 传递律（Transtivity）

  若X→Y，Y→Z，则X→Z

  即若X→Y，Y→Z为F的逻辑蕴涵，则X→Z为F 的逻辑蕴涵。

6.3 函数依赖的公理系统

6.3.4公理的正确性

公理系统具备以下性质：

1. 正确性：从F出发，用公理推出的每一个X→Y ⊆F+（一定在F的逻辑蕴涵中）。
2. 完备性：F+中的所有函数依赖都能用阿氏公理推导出来。即不能从F使用阿氏公理推导出来的函数依赖不在F+中。
3. 独立性：每一条公理所推导出来的函数依赖均不能由其他公理推导出来。
4. 相容性：每一条公理所推导出的函数依赖，不会有矛盾的结果。

6.3 函数依赖的公理系统

定理6.1 Armstrong公理是正确的

1） 自反律：若Y $\subseteq$ X，则X→Y

设r是R的任意一个关系，s，t是r的任意两个元组。

若s[X]=t[X]（全体相等）

∵Y $\subseteq$ X，即Y是X的逻辑子集(已知条件)

∴s[Y]=t[Y]（部分也相等）

则在s和t中的X的任何子集也必相等。

根据函数依赖的定义，有X→Y

6.3 函数依赖的公理系统

2）增广律：若X→Y，Z ⊆ U，则XZ→YZ。

设s，t，r的含义同上，X→Y，Z ⊆ U

又设s[XZ]=t[XZ]

则有s[X]=t[X]，且s[Z]=t[Z]

∵X→Y，根据函数依赖定义，有s[Y]=t[Y]

∴s[Y]s[Z]=t[Y]t[Z] 即s[YZ]=t[YZ]

故在s[XZ]=t[XZ]的条件下，推出了s[YZ]=t[YZ]

由函数依赖的定义，有XZ→YZ。

6.3 函数依赖的公理系统

3）传递律：X→Y，Y→Z，则X→Z

设s，t，r的含义同上

若s[X]=t[X],由X→Y，有s[Y]=t[Y]

再由Y→Z，有s[Z]=t[Z]

∴由s[X]=t[X]能导出s[Z]=t[Z]

根据函数依赖的定义有：X→Z

证毕

6.3 函数依赖的公理系统

4．3．4公理的推论

1．推论规则:

- 合成规则：若X→Y，X→Z，则X→YZ
- 分解规则：若X→YZ，则X→Y，X→Z
- 伪传递规则：若X→Y，YW→Z，则XW→Z

2．推论的正确性

定理4.2：Armstrong公理的三个推论是正确的。

证明:

1)合成规则（若X→Y，X→Z，则X→YZ)

∵X→Y，∵有增广律XX→XY，即X→XY

又∵X→Z∵XY→ZY

∴X→YZ(由传递律)

6.3 函数依赖的公理系统

2)分解规则 (若X→YZ, 则X→Y, X→Z)

∵Y ⊆ YZ ⊆ U, ∴YZ→Y(自反律)

同理YZ→Z(自反律)

∵X→YZ, ∴X→Y(传递律)

同理X→Z(传递律)

3) 伪传递规则 (若X→Y, YW→Z, 则XW→Z)

∵X→Y, ∴XW→YW(增广律, 两边扩充W)

∵YW→Z, ∴XW→Z(传递律)

重要结论 (由合成规则和分解规则可得)

如果Ai (i=1,2,3,…,n) 是关系模式R的属性

则X→A1A2…An的充要条件是

X→Ai (i=1,2,…,n) 均成立

证明提示: 根据合成规则: X→Y,X→Z,则X→YZ (充分性)

根据分解规则: X→YZ,则X→Y,X→Z (必要性)

6.3 函数依赖的公理系统

- 示例

  $R\langle U, F\rangle$, $U = (A, B, C, G, H, I)$, $F = \{A\to B, A\to C, CG\to H, CG\to I, B\to H\}$,

  - $A\to H$?
  - $CG\to HI$?
  - $AG\to I$?

导出规则

根据合并规则和分解规则，可得引理6.1

引理6.1 $X \rightarrow A_1 A_2 \ldots A_k$成立的充分必要条件是$X \rightarrow A_i$成立（$i$=1，2，…，$k$）。

6.3 函数依赖的公理系统

函数依赖闭包

定义6.12　在关系模式$R<U，F>$中为F所逻辑蕴含的函数依赖的全体叫作 F的闭包，记为F^{+}。

6.3 函数依赖的公理系统

在进行数据库设计时, 规定了一些明显的函数依赖, 但还能够推导出一些其他的依赖.例如对于关系供应商表S.

S（SNO, SNAME, CITY, STATUS)

我们规定了依赖:

SNO→{SNAME, CITY, STATUS}, CITY →STATUS.

我们能够推导出:

SNO→SNAME, SNO →CITY, …,SNO →{SNAME,CITY}, {SNO, SNAME} →CITY, …

一般说来, 对于一个关系R, 规定了一个函数依赖集合F, 能够推导出其他一些函数依赖来. 从F 所推导出的所有的函数依赖的集合称为 F 的闭包, 用F^+表示.

F的闭包

F={X→Y,Y→Z}, F^+计算是NP完全问题， X→A1A2...An

F^+={

X→φ, Y→φ, Z→φ, XY→φ, XZ→φ, YZ→φ, XYZ→φ, →

X→X, Y→Y, Z→Z, XY→X, XZ→X, YZ→Y, XYZ→X,

X→Y, Y→Z , XY→Y, XZ→Y, YZ→Z, XYZ→Y,

X→Z, Y→YZ, XY→Z, XZ→Z, YZ→YZ, XYZ→Z,

X→XY, XY→XY, XZ→XY, XYZ→XY,

X→XZ, XY→YZ, XZ→XZ, XYZ→YZ

X→YZ, XY→XZ, XZ→XY, XYZ→XZ,

X→ZYZ, XY→XYZ, XZ→XYZ, XYZ→XYZ }

Armstrong公理系统

- Armstrong公理是有效的、完备的。

  有效性：由F出发，根据Armstrong公理推导出来的每一个函数依赖一定在F^+中。

  完备性：F^+中的每一个函数依赖，必定可以由F出发，根据Armstrong公理推导出来。

  公理的正确性（有效性）保证导出的所有函数依赖都为真；公理的完备性保证可以推出所有的函数依赖。

6.3 函数依赖的公理系统

- 属性集的闭包

  - 设F为属性集U上的一组函数依赖，$X \subseteq U$，

  $X_F^+ = \{A \mid X \rightarrow A$ 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

关于闭包的引理

- 引理6.2

设F为属性集U上的一组函数依赖，X，$Y \subseteq U$，$X \to Y$能由F 根据Armstrong公理导出的充分必要条件是$Y \subseteq X_F^+$

- 用途

将判定$X \to Y$是否能由F根据Armstrong公理导出的问题，

就转化为求出X_F^+，判定Y是否为X_F^+的子集的问题

闭包的计算

问题：有没有一般性的算法判定X→Y是否能由F根据Armstrong公理导出？

如果计算出F^+，再判断X→Y是否属于F^+，则由于F^+的计算非常复杂，实际上是不可行的。

由引理2，判定X→Y是否能由F根据Armstrong公理导出，可转化为求X_F^+，判定Y⊆ X_F^+ 是否成立。

6.3 函数依赖的公理系统

闭包的计算

- 算法（求属性集的闭包）

Input：X，F　　　　　　Output：X_F^+

方法：计算$X^{(i)}$ (i = 1,2,…n)

(1) $X^{(0)} = X$, $i = 0$

(2) $X^{(i+1)} = X^{(i)}A$　其中，A是这样的属性：

在F中寻找尚未用过的左边是$X^{(i)}$的子集的函数依赖 $Y_j \to Z_j$ (j= 1,2,…,k)，其中$Y_j \subseteq X^{(i)}$，

在Z中寻找$X^{(i)}$中未出现过的属性集合A

(3) 判断是否有$X^{(i+1)} = U$或者$X^{(i+1)} = X^{(i)}$，若是则转（4），否则转（2）

(4) 输出$X^{(i)}$，即为X_F^+

函数依赖闭包

[例1] 已知关系模式$R<U, F>$，其中

$U=\{A, B, C, D, E\}$；

$F=\{AB\rightarrow C, B\rightarrow D, C\rightarrow E, EC\rightarrow B, AC\rightarrow B\}$。

求（AB）$_F^+$ 。

解 设$X^{(0)}=AB$；

(1)计算$X^{(1)}$：逐一的扫描F集合中各个函数依赖，

找左部为A，B或AB的函数依赖。得到两个：

$AB\rightarrow C$，$B\rightarrow D$。

于是$X^{(1)}=AB\cup CD=ABCD$。

函数依赖闭包

(2)因为$X^{(0)} \neq X^{(1)}$，所以再找出左部为$ABCD$子集的那些函数依赖，又得到$AB \rightarrow C$，$B \rightarrow D$，$C \rightarrow E$，$AC \rightarrow B$，

于是$X^{(2)}=X^{(1)} \cup BCDE=ABCDE$。

(3)因为$X^{(2)}=U$，算法终止

所以$(AB)_F^+=ABCDE$。

函数依赖闭包

算法的正确性:

1）$X^{(j)} \subseteq X^{+}$ （$j=0,1,\cdots$），即用算法计算出的每一个属性A，必有 $A \in X^{+}$，说明算出的结果不会多出来某些不属于X^{+}的属性。

2）$X^{+} \subseteq X^{(k)}$ （k为某个j，即X^{+}中的任一属性A必属于计算中的某个$X^{(j)}$ （$j=0,1,\cdots$），说明计算的结果不会遗漏某些属于X^{+}的属性。

闭包的计算

- 示例 2

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

闭包的计算

- 示例 2

  R< U, F >, U = (A, B, C, G, H, I), F = {A→B, A→C, CG→H, CG→I, B→H},计算

  $$(AG)_F^+$$

$X^0=AG$

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

$(AG)_F^+$ AGBCHI

闭包的计算

- 示例3

  R< U, F >, U = (A, B, C, D, E, G), F = {A→E, BE→AG, CE→A, G→D},计算

  $$(AB)_F^+$$

6.3 函数依赖的公理系统

闭包的计算

- 示例3

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

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

$(AB)_F^+$ ABEGD

6.3 函数依赖的公理系统

公理有效性证明

设F是关系模式R(A1, A2, …An)的函数依赖集，U是属性全集，X，Y是U的子集，则X→Y是用阿氏公理从F推出的充要条件是$Y \subseteq X^+$

充分性：设$Y \subseteq X^+$，并设$Y=B_1B_2\ldots B_k$,其中$B_i(i=1,\ldots,k)$为某个$A_j(j=1,\ldots,n)$,即$B_i \in U$。根据属性闭包的定义可知，$X \rightarrow B_1,\ldots X \rightarrow B_k$是用阿氏公理从F推出的，并用合成规则可得X→Y。所以，当$Y \subseteq X^+$时，X→Y是用阿氏公理从F推出

6.3 函数依赖的公理系统

必要性： 设$X\rightarrow Y$是用阿氏公理从F推出的，

$Y=B_1B_2\cdots B_k$,B_i的含义同上，根据分解规则有$X\rightarrow B_1,\cdots X\rightarrow B_k$，则由$X^+$的定义可知$B_i\in X^+(i=1,\cdots,k)$,所以$Y\subseteq X^+$

6.3 函数依赖的公理系统

公理完备性的另一种理解是所有不能用公理推出的函数依赖都不为真，即它不能由F逻辑蕴涵。或者说存在一个具体关系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却不满足r，即F不能逻辑蕴涵X→Y。

6.3 函数依赖的公理系统

阿氏公理完备性的证明

定理：Armstrong公理是完备的。

证明：设F是关系模式R（$A_1,A_2,\cdots,A_n$）的函数依赖集，U是属性全集，X，Y是U的子集，并假定用阿氏公理推不出$X \rightarrow Y$。

构造一个关系r，包含两个元组。

	X^+的属性	其它属性
t_1	1 1 1 1 1 1 …1	1 1 1 1 1 1 …1
t_2	1 1 1 1 1 1 …1	0 0 0 0 0 0 …0

6.3 函数依赖的公理系统

证明两点:

1.在关系r中，F的所有函数依赖都成立;

2.在关系r中， $X \rightarrow Y$不能成立。

1 设$V \rightarrow W$是F中任一函数依赖，则有下列两种情况;

1） $V \subseteq X^+$，则显然$X \rightarrow V$能用阿氏公理推出，根据假设$V \rightarrow W$，所以， $X \rightarrow W$，即$W \subseteq X^+$，从上表中可以看出两个元组在V和W上均一致，于是$V \rightarrow W$在r中成立

6.3 函数依赖的公理系统

2） $V \not\subseteq X^{+}$，则由上表可以看出，如果一个属性集不完全属于X^{+}，则该属性集在两个元组上的属性值必不相等。根据函数依赖定义可知，$V \rightarrow W$在关系r中成立。

因此，在关系r中，F的任一函数依赖都成立。

2 因为$X \rightarrow Y$不能用阿氏公理从F推出，所以$Y \not\subseteq X^{+}$，而$X \not\subseteq X^{+}$，于是在关系r的两个元组中X的属性值一致，而Y的属性值不一致，所以$X \rightarrow Y$在r中不成立。

函数依赖的等价和覆盖

- **函数依赖集的等价性** 如果$G^+=F^+$，就说函数依赖集F覆盖G（F是G的覆盖，或G是F的覆盖），或F与G等价。

最小依赖集

定义6.15　如果函数依赖集F满足下列条件，则称F为一个极小函数依赖集。亦称为最小依赖集或最小覆盖。

(1) F中任一函数依赖的右部仅含有一个属性。

(2) F中不存在这样的函数依赖X→A，使得F与F-{X→A}等价。

(3) F中不存在这样的函数依赖X→A，X有真子集Z使得F-{X→A}∪{Z→A}与F等价。

最小依赖集

[例2] 关系模式*S*<*U*，*F*>，其中：

U={ SNO，SDEPT，MN，CNAME，G }，

F={ SNO→SDEPT，SDEPT→MN，(SNO，CNAME）→G }

设F ’={SNO→SDEPT，SNO→MN，SDEPT→MN，

(SNO，CNAME)→G， (SNO，SDEPT)→SDEPT}

*F*是最小覆盖，而*F* ’不是。

因为：*F* ’-{SNO→MN}与*F* ’等价

F ’-{(SNO，SDEPT)→SDEPT}也与*F* ’等价

定理6.3 每一个函数依赖集F均等价于一个极小函数依赖集F_m。此F_m称为F的最小依赖集

分三步对F进行“极小化处理”，找出F的一个最小依赖集。

(1)逐一检查F中各函数依赖FD_i：$X \rightarrow Y$,

若$Y=A_1A_2 \ldots A_k$，$k > 2$,

则用 $\{ X \rightarrow A_j | j=1，2，\ldots，k\}$ 来取代$X \rightarrow Y$。

(2)逐一检查F中各函数依赖FD_i：$X \to A$，

令$G=F-\{X \to A\}$，

若$A \in X_G^+$，　则从F中去掉此函数依赖。

(3)逐一取出F中各函数依赖FD_i：$X \to A$，

设$X=B_1B_2 \ldots B_m$，

逐一考查B_i（i=l，2，…，m），

若$A \in (X-B_i)_F^+$，

则以$X-B_i$取代X。

最小依赖集

- 示例一

  F = {A→B， B→A， A→C， B→C}， 求F_{min}

  - 检查A→B， G=F−{A→B}={B→A， A→C， B→C}

    $(A)_G^+$={A， C}， B∉{A， C}

  - 检查A→C， G=F−{A→C}={A→B， B→A， B→C}

    $(A)_G^+$={A， B， C}， C *in* {A， B， C}

  所以从F中删除A→C，

  F_{min} = {A→B， B→A， B→C}

  或者

  F_{min} = {A→B， B→A， A→C}

注意：

- F的最小依赖集F_m不一定是唯一的，它与对各函数依赖FD_i及$X \rightarrow A$中X各属性的处置顺序有关。

最小依赖集

- 示例二

F = {C→A， A→G， CG→B， B→A}， 求F_{min}

F是无冗余的

判断CG→B，

$(CG-C)_F^+$ = $(G)_F^+$ = {G}

B$\notin$ $(CG-C)_F^+$

$(CG-G)_F^+$ = $(C)_F^+$ = {C， A， G， B}

B $\in (CG-G)_F^+$ ， 以C代替CG

最后， F_{min} = {C→A， A→G， C→B， B→A}

6.3 函数依赖的公理系统

关于正则覆盖

F的一个覆盖Fc，如果满足以下性质，则称其为F的正则覆盖

- Fc中的任何函数依赖都不包含无关属性；
- Fc中函数依赖的左半部是唯一的，即不存在两个函数依赖的左半部完全相同。

Fc=F

Repeat

　　使用合并律合并左侧相同的函数依赖

　　找出每个函数依赖中的无关属性，并去之

Until Fc 不再改变

6.4 模式分解

- 6.4.1 模式分解的定义
- 6.4.2 模式分解中的问题
- 6.4.3 无损连接分解
- 6.4.4 保持函数依赖的分解
- 6.4.5关系模式的分解算法

6.4.1 模式分解的定义

- 定义1(定义6.17):
  - 函数依赖集合$F_i = \{X\rightarrow Y \mid X\rightarrow Y\in F^+ \wedge X,Y \subseteq U_i\}$称为F在$U_i$上的投影

- 定义2 (定义6.16)：
  - 关系模式R<U , F>的一个分解是指

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

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

6.4.1 模式分解的定义

- 2. 分解的基本代数运算
  - 投影
  - 自然连接
- 3. “等价”分解的定义
  - 无损连接分解
  - 保持函数依赖
  - 既是无损连接分解，又要保持函数依赖

6.4.2 模式分解中的问题 Ex1

R(A, B, C)

A	B	C
1	1	2
2	2	1

$\prod_{AB}(R)$

A	B
1	1
2	2

$\prod_{BC}(R)$

B	C
1	2
2	1

$\prod_{AB}(R) \bowtie \prod_{BC}(R)$

A	B	C
1	1	2
2	2	1

无损分解

R(A, B, C)

A	B	C
1	1	1
2	1	2

$\prod_{AB}(R)$

A	B
1	1
2	1

$\prod_{BC}(R)$

B	C
1	1
1	2

$\prod_{AB}(R) \bowtie \prod_{BC}(R)$

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

有损分解

6.4.2 模式分解中的问题 Ex2

$\{A \rightarrow B, B \rightarrow C\}$

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

A
a1
a2
a3
a4

B
b1
b2
b3

C
c1
c2

插入

A	B
a1	b1
a2	b1
a3	b2
a4	b3
a5	b3

$\bowtie$

A	C
a1	c1
a2	c1
a3	c2
a4	c1
a5	c3

=

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1
a5	b3	c3

违反
$B \rightarrow C$

6.4.2 模式分解中的问题 Ex2

A	C
a1	c1
a2	c1
a3	c2
a4	c1

$\bowtie$

B	C
b1	c1
b2	c2
b3	c1

=

A	B	C
a1	b1	c1
a1	b3	c1
a2	b1	c1
a2	b3	c1
a3	b2	c2
a4	b1	c1
a4	b3	c1

A	B
a1	b1
a2	b1
a3	b2
a4	b3

$\bowtie$

B	C
b1	c1
b2	c2
b3	c1

=

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c2
a4	b3	c1

6.4.3 无损连接分解

- 1. $m_{\rho}(r)$
  - 设$\rho=\{R_1,R_2,\cdots R_n\}$是关系模式R的一个分解，r是R的一个关系，定义

$$m_{\rho}(r) = \Pi_{R1}(r) \bowtie \Pi_{R2}(r) \bowtie \ldots \bowtie \Pi_{Rn}(r)$$

即$m_{\rho}(r)$ 是r在ρ中各关系模式上投影的连接

6.4.3 无损连接分解（续）

- 2. 无损连接分解
  - $\rho=\{R_1,R_2,\cdots R_n\}$是关系模式R的一个分解，若对R的任何一个关系r均有$r=m_\rho(r)$，则称分解ρ具有无损连接性，简称ρ为无损连接分解。
- 3. 无损连接分解的判别算法
  - 通用算法
  - 简单算法

6.4.3 无损连接分解（续）

- 4. 无损连接分解的判别算法

  - 通用算法:

  输入: $R(A_1,A_2,\cdots A_n)$,R的函数依赖集F,R的分解 $\rho=\{R_1,R_2,\cdots R_k\}$

  输出： 分解ρ是否具有无损连接性

方法:

- (1) 建立矩阵S,列j 对应属性Aj,行i对应Ri;
- (2) FOR i = 1 TO k DO

FOR j =1 TO n DO

IF R_i 包含属性A_j THEN

$S[i,j] := a_j$;

ELSE

$S[i,j] : = b_{ij}$;

END FOR;

END FOR;

$S = \{S_{ij} \mid$ 若$A_j \in R_i$, $S_{ij} = a_j$, 否则$S_{ij} = b_{ij}\}$

构造的初始S记为T^0

示例 第1,2步

- $U=\{A,B,C,D,E\}$, $F=\{AB\rightarrow C, C\rightarrow D, D\rightarrow E\}$
  $\rho=\{(A, B, C), (C, D), (D, E)\}$

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

无损连接分解算法描述

（3）对F中每一个函数依赖X→Y，若S中存在元组t_1，t_2，使得$t_1[X]=t_2[X]$，$t_1[Y]\neq t_2[Y]$，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_j，则另一个也改 为a_j；

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

若在某次更改后，有一行成为$a_1, a_2, \cdots, a_n$，则算法中止。对F中FD逐一处理，称为一次扫描，扫描的结果关系S记为T^k。

无损连接分解

（4）比较扫描前后，T^{k-1}和T^{k}，有无变化，若有，则转（3），否则算法中止。

示例 第3步

- $U=\{A,B,C,D,E\}$, $F=\{AB\to C, C\to D, D\to E\}$
  $\rho=\{(A, B, C), (C, D), (D, E)\}$

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

$AB\to C$

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

$C\to D$

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

- 如果S中存在一行全为“a”类符号，则ρ具有无损连接性，否则ρ不具有无损连接性

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

$D \rightarrow E$

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

无损连接分解 Ex

- U={A,B,C,D,E},
  F={A→C, B→C, C→D,DE→C ,CE→A}
  ρ ={(A, D), (A, B), (B, E), (C, D, E), (A, E)}

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{23}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{33}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	b_{33}	b_{54}	a_5

A→C

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{33}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	b_{13}	b_{54}	a_5

无损连接分解 Ex续

B→C

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	b_{24}	b_{25}
BE	b_{31}	a_2	b_{13}	b_{34}	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	b_{13}	b_{54}	a_5

C→D

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	a_4	b_{25}
BE	b_{31}	a_2	b_{13}	a_4	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	b_{13}	a_4	a_5

无损连接分解 Ex续

$DE \to C$

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	a_4	b_{25}
BE	b_{31}	a_2	a_3	a_4	a_5
CDE	b_{41}	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	a_3	a_4	a_5

$CE \to A$

	A	B	C	D	E
AD	a_1	b_{12}	b_{13}	a_4	b_{15}
AB	a_1	a_2	b_{13}	a_4	b_{25}
BE	a_1	a_2	a_3	a_4	a_5
CDE	a_1	b_{42}	a_3	a_4	a_5
AE	a_1	b_{32}	a_3	a_4	a_5

6.4.3 无损连接分解（续）

- 4. 无损连接分解的判别算法
  - 简单算法：定理6.5
    - R<U,F>的一个分解ρ = {R_1<U_1,F_1>,R_2<U_2,F_2>}具有无损连接性的充要条件是：

      $U_1 \cap U_2 \to U_1 - U_2 \in F^+$

      或 $U_1 \cap U_2 \to U_2 - U_1 \in F^+$

  - 定义（分解为两个关系模式）

    关系模式R(U)，U_1, $U_2 \subseteq U$，$U_1 \cup U_2 = U$，r是R上的一个关系，$r_1 = \prod_{U_1}(r)$，$r_2 = \prod_{U_2}(r)$，若$r = r_1 |\times| r_2$，则称(r_1，r_2)是r的一个无损连接分解。

    注：$r \subseteq \prod_{U_1}(r) |\times| \prod_{U_2}(r)$

6.4.3 无损连接分解 Ex

Ex 1： 已知 R=ABC, F={A → B}, 判断$\rho_{1,}$ ρ_2是否为R的无损连接分解

- ρ_1={R_1(AB), R_2(AC)}
  - 解答： $R_1 \cap R_2$ = A, $R_1 - R_2$ = B

    由A → B， 得到ρ_1是无损连接分解
- ρ_2={R_1(AB), R_2(BC)}
  - 解答： $R_1 \cap R_2$ = B, $R_1 - R_2$ = A, $R_2 - R_1$ = C

    B→A, B→C均不成立， 所以ρ_2不是

6.4.4 保持函数依赖的分解

- 1. 定义
  - Z是U的子集， 函数依赖集合F在Z上的投影定义为

$$\prod_Z(F) = \{X \to Y \mid X \to Y \in F^+ \land XY \subseteq Z\}$$

  - 设$\rho = \{R_1, R_2, \cdots, R_n\}$是关系模式R<U , F>的一个分解， 如果$F^+ = (\prod R_i(F))^+$， 则称$\rho$是保持函数依赖的分解

6.4.4 保持函数依赖的分解(示例)

- Ex:

  关系模式R<U, F>

  U = {CITY, ST, ZIP},

  F = {(CITY, ST) → ZIP, ZIP → CITY}

  $\rho = \{R_1 = \{ST, ZIP\}, R_2 = \{CITY, ZIP\}\}$

  判断： $R_1 \cap R_2 = \{ZIP\}$, $R_2 - R_1 = \{CITY\}$

  $\because\ R_1 \cap R_2 \rightarrow R_2 - R_1$

  ∴分解是无损的

  $\prod_{R_1}(F) = \{\}$, $\prod_{R_2}(F) = \{ZIP \rightarrow CITY\}$

  $\prod_{R_1}(F) \cup \prod_{R_2}(F) = \{ZIP \rightarrow CITY\}$

  丢失了函数依赖(CITY, ST) → ZIP

- 实例:
  表（职工，级别，工资),
  其中函数依赖为{职工->级别，级别->工资}
  可以有两种分解途径,
  分解一：（职工，工资），（工资，级别）
  分解二：（职工，级别），（工资，级别）

姓名	级别	工资
ZHAO	4	500
QIAN	5	600
SUN	6	700
LI	7	600

→

姓名	工资
ZHAO	500
QIAN	600
SUN	700
LI	600

级别	工资
4	500
5	600
6	700
7	500

丢失函数依赖

不同行业机构的不同工资级别会有相同工资数额，因此按分解一，有可能导致同一职工对应不同的工资级别，从而丢失了有关职工工资级别的信息（丢失了函数依赖：职工→级别）

6.4.4关系模式的分解

- 如果一个分解具有无损连接性，则它能够保证不丢失信息。
- 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。
- 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。具有无损连接性的分解不一定能够保持函数依赖。同样，保持函数依赖的分解也不一定具有无损连接性。

6.4.5关系模式的分解算法

- 三个重要事实：
  - （1）若要求分解保持函数依赖，那么模式分离总可以达到3NF，但不一定能达到BCNF；
  - （2）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；
  - （3）若要求分解具有无损连接性，则一定可以达到4NF。

  - 相关算法参见教材P_{192}

6.4.5关系模式的分解算法

3NF的分解算法1（保持函数依赖分解）

输入：关系模式R(1NF)、其属性集U和最小函数依赖集G.

输出：具有函数依赖保持性的分解ρ,

ρ中的所有子关系模式都满足3NF.

1.找出不在F中出现的属性，把这样的属性从U中去掉，构成一个关系模式。

2.若有$X \rightarrow A \in G$，且$XA = U$，则$\rho = R$，算法中止。

3.对G按照相同左部进行分组（假定有k组），每一组G'构成一个子关系，若$U_i \subseteq U_j$,则去掉U_i。

输出$\rho=\{R_1,R_2 \cdots R_k\}$

6.4.5关系模式的分解算法

例子：

关系模式R（CTHRSG），最小依赖集为：

$$F=\{C\rightarrow T, CS\rightarrow G, HT\rightarrow R, HR\rightarrow C,$$
$$HS\rightarrow R\}$$

则$\rho=\{CT,CHR,HRT,CSG,HSR\}$为一个具有依赖保持性的3NF分解。

6.4.5关系模式的分解算法

3NF分解算法2（保持函数依赖，且无损连接分解）

输入：关系模式R(1NF)、其属性集U和最小函数依赖集G.
输出：具有函数依赖保持性和无损连接性的分解ρ,
ρ中的所有子关系模式都满足3NF.
方法：
(1)调用3NF的分解算法1得R的分解$\rho=\{R_1,R_2,R_3,\ldots,R_n\}$;
(2)返回分解$\rho=\{R_1, R_2, R_3, \ldots, R_n, R_k\}$;
其中R_k是R的一个候选键X构成的关系。
(3)若有某个R_i，$X \subseteq R_k$，则在ρ中去掉R_k。

6.4.5关系模式的分解算法

BCNF的分解算法

输入：关系模式R(1NF)、其属性集U和函数依赖集F.
输出：具有无损连接性的分解ρ,
　　ρ中的所有子关系模式都满足BCNF.
方法：
ρ:={R};
while ρ中存在非BCNF的关系模式 do
　选择一个非BCNF的关系模式S∈ρ;
　选择S的一个违反BCNF要求的函数依赖X→Y
　用两个关系模式S－Y和X∪Y取代ρ中的S;
endwhile.

6.4.6候选关键字求解

关系模式R(U,F)中，若$K^+ = U$，则K为R的一个超码，若$\forall A \in K$，$(K-A)^+ = U$，则K为R的一个候选码。

考虑关系模式$R=\{A_1,A_2,\cdots,A_n\}$和函数依赖集F，可将其属性分为四类：

(1)仅出现在F中函数依赖左部的属性L类；

(2)仅出现在F中函数依赖右部的属性R类；

(3) 在F中函数依赖左右两边均未出现的属性N类；

(4) 在F中函数依赖左右两边均出现的属性LR类；

6.4.6候选关键字求解

几个重要结论:

对于R(U,F)（F中不包括平凡依赖）

1 若X是L或N类属性，则X必为R的任一候选关键字的成员;

2 若X是R类属性，则X不在任何候选关键字中;

3 若X是L和N类组成的属性集，且$X^{+}=U$，则X是R的唯一候选关键字。

6.4.6候选关键字求解

一个候选码求解算法

输入：关系模式R (U,F)

输出：R的所有候选码

1.将R的所有属性分为L、R、N 和LR 四类，令X代表L、N类，Y代表LR类；

2.求X^+，若$X^+=U$，则X为唯一候选码，转5，否则转3；

3.在Y中取一个属性A，求XA^+，若$XA^+=U$，则转4，否则取另一个属性，直到取完。

4.若找到所有候选码，则转5，否则在Y中依次取两个、三个…，依次求它们的属性闭包，看是否等于U

6.停止，输出结果。