数据库系统原理

陈岭

浙江大学计算机学院

9 函数依赖和关系模式分解

- 第一范式
- 关系数据库设计中易犯的错误
- Armstrong公理系统
- 函数依赖理论
- 关系模式分解

第一范式

- 如果某个域中元素被认为是不可分的，则这个域称为是原子的
  - 非原子域的例子：
    - 复合属性：名字集合
    - 多值属性：电话号码
    - 复杂数据类型：面向对象的
- 如果关系模式R的所有属性的域都是原子的，则R称为属于第一范式（1NF）
- 非原子值存储复杂并易导致数据冗余
  - 我们假定所有关系都属于第一范式

第一范式

- 如何处理非原子值
  - 对于组合属性：让每个子属性本身成为成为一个属性
  - 对于多值属性：为多值集合中的每个项创建一条元组
- 原子性实际上是由域元素在数据库中如何被使用决定的
  - 例，字符串通常会被认为是不可分割的
  - 假设学生被分配这样的标识号：CS0012或EE1127，如果前两个字母表示系，后四位数字表示学生在该系内的唯一号码，则这样的标识号不是原子的
  - 当采用这种标识号时，是不可取的。因为这需要额外的编程，而且信息是在应用程序中而不是在数据库中编码

关系数据库设计中易犯的错误

- 关系数据库设计要求我们找到一个“好的”关系模式集合。一个坏的设计可能导致
  - 数据冗余
  - 插入、删除、修改异常
  - 假设，我们用以下模式代替*instructor*模式和*department*模式：

    inst_dept(ID, name, salary, dept_name, building, budget)

关系数据库设计中易犯的错误

- 数据冗余
  - 每个系的*dept_name*, *building*, *budget*数据都要重复一次
  - 缺点：浪费空间，可能会导致不一致问题

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

关系数据库设计中易犯的错误

- 更新异常
  - 更新复杂，容易导致不一致问题。例，修改*dept_name*，很多相关元组都需要修改
- 插入/删除异常
  - 使用空值(null)：存储一个不知道所在系的教师信息，可以使用空值表示*dept_name*, *building*, *budget*数据，但是空值难以处理

模式分解

- 模式分解
  - 例，可以将关系模式(*A, B, C, D*)分解为：(*A, B*)和(*B, C, D*)，或(*A, C, D*)和(*A, B, D*)，或(*A, B, C*)和(*C, D*)，或(*A, B*)、(*B, C*)和(*C, D*)，或 (*A, D*)和(*B, C, D*)
  - 例，将关系模式*inst_dept*分解为：
    - *instructor*(*ID, name, dept_name, salary*)
    - *department*(*dept_name, building, budget*)
- 原模式(*R*)的所有属性都必须出现在分解后的(R_1, R_2)中：$R = R_1 \cup R_2$
- 无损连接分解
  - 对关系模式*R*上的所有可能的关系*r*

$$r = \prod_{R1}(r) \bowtie \prod_{R2}(r)$$

模式分解

- 例，分解$R = (A, B)$，得到$R_1 = (A)$和$R_2 = (B)$

$r =$

A	B
α	1
α	2
β	1

⇒

A
α
β

$\Pi_A(r)$

+

B
1
2

$\Pi_B(r)$

$\Pi_A(r) \bowtie \Pi_B(r) =$

A	B
α	1
α	2
β	1
β	2

$\neq$ r

不好的分解！

因为它不是无损连接分解，是非法的

模式分解

- 目标：设计一个理论
  - 以判断关系模式R 是否为“好的”形式（不冗余）
  - 当R 不是“好的”形式时，将它分解成模式集合$\{R_1, R_2, \ldots, R_n\}$使得
    - 每个关系模式都是“好的”形式
    - 分解是无损连接分解
  - 我们的理论基于：
    - 函数依赖（ functional dependencies ）
    - 多值依赖（ multivalued dependencies ）

函数依赖

- 设R是一个关系模式，且有属性集$\alpha \subseteq R$，$\beta \subseteq R$
- 函数依赖

$$\alpha \rightarrow \beta$$

借用了数学上的函数概念：
x → f(x)

在R上成立当且仅当对任意合法关系$r(R)$，若r的任意两条元组t_1和t_2在属性集α上的值相同，则他们在属性集β上的值也相同。即，

$$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$

- β函数依赖于α，α函数决定β

α	β	γ
a	f	1
b	h	2
a	f	3
c	f	4

函数依赖

- 函数依赖：一种完整性约束，表示特定的属性值之间的关系，可以用来判断模式规范化和建议改进
- 例，考虑$r(A, B)$及其下列实例r

A	B
1	4
1	5
3	7

  - 对此实例，$A \rightarrow B$不成立，但$B \rightarrow A$成立

∵若B属性值确定了，则A属性值也唯一确定了。于是有 $B \rightarrow A$

函数依赖

- 函数依赖是码概念的推广
- K是关系模式R的超码当且仅当$K \rightarrow R$
- K是R的候选码当且仅当
  - $K \rightarrow R$，并且
  - 没有$\alpha \subset K$，使$\alpha \rightarrow R$（不存在K的真子集α，使之满足$\alpha \rightarrow R$）

函数依赖

- 函数依赖使我们可以表达用超码无法表达的约束，考虑模式

  $inst\_dept(\underline{ID}, name, salary, \underline{dept\_name}, building, budget)$

  - 我们期望下列函数依赖成立：

    $$dept\_name \rightarrow building$$
    $$ID \rightarrow building$$

  - 而不期望下列函数依赖成立：

    $$dept\_name \rightarrow salary$$

函数依赖的使用

- 我们用函数依赖来：
  - 检查关系在给定函数依赖之下是否合法
    - 若关系r在函数依赖集F下是合法的，则称r满足F

r =

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

F＝{

A→C

AB →D ⇒ {A, B} →D

ABC →D}

但，

A ↛ B, A ↛ D, B ↛ A, C ↛ A, C ↛ D,
B ↛ C, C ↛ B, B ↛ D, …

函数依赖的使用

- 对合法关系集合指定约束
  - 如果R上的所有合法关系都满足F, 则称F在R上成立

$r_1(R)=$

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

F＝{

A→C,

AB →D,

ABC →D

}

$r_2(R)$=…

$r_3(R)$=…

……

注：容易判别一个r是否满足给定的F。不易判别F是否在R上成立。不能仅由某个r推断出F。R上的函数依赖F, 通常由定义R的语义决定

函数依赖

- 被所有关系实例都满足的函数依赖称为平凡的
  - 例，$A \rightarrow A$，$AB \rightarrow A$

    $(ID, name) \rightarrow ID$

    $ID \rightarrow ID$
  - 一般地，若$\beta \subseteq \alpha$，则$\alpha \rightarrow \beta$是平凡的。即，

    平凡的函数依赖：若$\beta \subseteq \alpha$，$\alpha \rightarrow \beta$

    非平凡的函数依赖：若$\beta \nsubseteq \alpha$，$\alpha \rightarrow \beta$

函数依赖集的闭包

- 给定函数依赖集F，存在其他函数依赖被F逻辑蕴含
  - 例，如果$A \rightarrow B$且$B \rightarrow C$，则可推出$A \rightarrow C$
- 被F逻辑蕴含的全体函数依赖的集合称为F的闭包，用F^+表示F的闭包
  - 例，$F = \{ A \rightarrow B,\ B \rightarrow C \}$，$F^+ = \{ A \rightarrow B,\ B \rightarrow C,\ A \rightarrow C,\ A \rightarrow A,\ AB \rightarrow A,\ AB \rightarrow B,\ AC \rightarrow C,\ A \rightarrow BC,\ \cdots \}$
- 如何计算出F^+
  - 例，$R = (A,\ B,\ C,\ G,\ H,\ I)$

    $F = \{ A \rightarrow B$

    $A \rightarrow C$

    $CG \rightarrow H$

    $CG \rightarrow I$

    $B \rightarrow H \}$，那么 $F^+ = ?$

Armstrong公理

- 可利用Armstrong公理找出F^+：
  - 若$\beta \subseteq \alpha$，则$\alpha \rightarrow \beta$(自反律)
  - 若$\alpha \rightarrow \beta$，则$\gamma\alpha \rightarrow \gamma\beta$(增补律)
  - 若$\alpha \rightarrow \beta$且$\beta \rightarrow \gamma$，则$\alpha \rightarrow \gamma$(传递律)
- Armstrong公理是
  - 正确有效的(不会产生错误的函数依赖)
  - 完备的(产生所有成立的函数依赖即F^+)