

数据库系统2-6：关系运算的安全性 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_\\_E6\\_95\\_B0\\_E6\\_8D\\_AE\\_E5\\_BA\\_93\\_E7\\_c98\\_133948.htm](https://www.100test.com/kao_ti2020/133/2021_2022__E6_95_B0_E6_8D_AE_E5_BA_93_E7_c98_133948.htm)

计算机等级考试训练软件《百宝箱》为了使计算机能有效地处理关系运算，必须对关系运算进行限制，使关系运算不致产生无限关系和导致无穷验证。这就是关系运算的安全性问题。

1. 产生无限关系和无穷验证的原因：(1)在关系代数中规定的集合运算只包含并、交、差和广义笛卡儿积，而不引入补运算，究其原因就是为了防止产生无限关系。从关系代数操作的定义看，对任一个有限关系的关系代数操作结果都不会产生无限关系和无穷验证，所以，关系代数系统是安全的。(2)在关系演算中，表达式 $\{t \mid R(t)\}$ 、 $\{t_1, t_2, \dots, t_k \mid R(t) \wedge t_2 > 9\}$ 可能表示无限关系，也就是说，关系演算，不管是元组关系演算，还是域关系演算，系统都有可能产生无限关系。(3)在关系演算中，当变量 $t$ 的定义域无穷时，要判断 $(\exists t)(F(t))$ 为假或 $(\forall t)(F(t))$ 为真，必须对变量的所有可能取值都进行验证，这就导致无穷验证。

2. 实现关系运算安全性的设想 (1)倘若对任何表达式 $\{t \mid j(t)\}$ 都规定一个有限符号集 $DOM(j)$ ，使 $j$ 的演算结果、中间结果所产生的关系，及其元组的各个分量都必须属于集合 $DOM(j)$ ，这样便保证运算结果的有限性。(2)在解决 $(\exists t)(F(t))$ 和 $(\forall t)(F(t))$ 无穷验证中，可设想规定一个有限集合 $DOM(w)$ ，并限制 $t$ 只能在 $DOM(w)$ 中取值。定义1：设 $j$ 是一个元组关系演算公式， $j$ 的符号集合（记为 $DOM(j)$ ）是由下面两类符号构成的集合：  
①  $j$ 中的常量。  
②  $j$ 中出现的关系的所有元组的所有分量值。显然，集合 $DOM(j)$ 是一个有限集合

。定义2：一个元组关系演算表达式 $\{t|j(t)\}$ 是安全的，如果如果 $j(t)$ 为真，则元组 $t$ 的每个分量都属于 $DOM(j)$ 。对于 $j$ 中的每个形如 $(\exists u)(F(u))$ 的子表达式，如果元组 $u$ 使 $F(u)$ 为真，则元组 $u$ 的每个分量都属于 $DOM(F)$ ，对于 $j$ 中的每个形如 $(\forall u)(F(u))$ 的子表达式，如果元组 $u$ 使 $F(u)$ 为假，则元组 $u$ 的每个分量都属于 $DOM(F)$ 。定理：安全元组表达式的运算是安全的。证明略。例如有关系 $R$ 、 $S$ 如下图：(1) 设有元组演算表达式 $\{t|R(t)\}$ ，因为 $R$ 是一个有限关系，所以该表达式可能为无限关系。如果定义该表达式的安全约束有限集合为： $DOM(j) = \{1(R), 2(R), 3(R)\} = [b, c, 9, 2, dh, eg]$  其中将一元关系  $i(R)$  看作一个集合，则原表达式 $\{t|R(t)\}$ 经过安全约束后，它的安全表达式 $\{t|R(t)\}$ 所示的关系 $R_1$ 如下图，这是一个有限关系，但与原关系不等价。若定义原表达式的 $DOM(j) = [b, c, 2, dh]$ ，则经过安全约束后，其安全表达式 $\{t|R(t)\}$ 所表示的关系 $R_2$ 如下图，这也是一个有限关系，但也与原关系不等价。(2) 设有元组演算表达式 $\{t|S(t) \wedge t[2]=b\}$ ，若定义安全约束有限集合为： $DOM(j) = \{1(S), 2(S), 3(S)\} = [2, 3, 4, d, b, f, c, e]$  则经过安全约束后，它的安全表达式 $\{t|S(t) \wedge t[2]=b\}$ 所表示的关系 $R_3$ ，与原表达式结果一致；若定义安全约束有限集合为 $DOM(j) = [2, 3, 4, d, b, f, c]$ ，则它的安全表达式 $\{t|S(t) \wedge t[2]=b\}$ 所表示的关系 $R_4$ ，与原表达式不等价。原表达式的最终运算结果为一有限关系时，对该表达式能够找到一个合理的有限集合 $DOM(j)$ ，使得它在安全约束前后的结果关系一致且等价；反之亦然。今后当我们提及安全表达式时，除声明外，总是指安全表达式等价于原表达式，即  $(DOM(j)) \{t|j(t)\} = \{t|j(t)\}$

试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)