

关系数据库中基于 EPTV 的模糊查询

陈逸菲, 叶小岭, 张颖超

(南京信息工程大学信息与控制学院, 南京 210044)

摘要: 关系数据库中空值存在不同的语义, 并且会影响模糊查询结果。针对该问题, 提出用标号来区分空值的语义, 并且在 EPTV 逻辑的基础上, 对关系运算和一些复杂的嵌套查询进行扩展, 给出相关定义和计算方法。通过实例说明, 与常规模糊查询相比, 该方法能较好地反映空值对模糊查询结果的影响。

关键词: 空值; 模糊查询; 关系数据库

Fuzzy Queries Based on EPTV in Relational Databases

CHEN Yi-fei, YE Xiao-ling, ZHANG Ying-chao

(College of Information and Control, Nanjing University of Information Science & Technology, Nanjing 210044)

【Abstract】 Null values have different semantics in relational databases and influence the results of fuzzy queries. This paper distinguishes the different semantics of null values and attaches labels to these semantics. Relational operations and the complex nested queries are extended on the base of Extended Possibilistic Truth Values(EPTV) logic. Related definitions and calculation methods are proposed. Instances illuminate that the method can reflect the effects of different null values on query results compared with regular fuzzy queries.

【Key words】 null value; fuzzy query; relational databases

1 概述

关系数据库只能处理二值逻辑表示的查询, 而现实中用户的许多查询条件具有一定的模糊性、不确定性, 很难精确地表示。文献[1]利用模糊集合理论扩展关系代数以解决此问题, 并提出了 SQLf 语言。之后, 有许多研究致力于对 SQL 进一步扩展, 如文献[2-3]分别提出了 SQL2 和 SQL3 的模糊化版本 SQLf2 和 SQLf3; 文献[4]在模糊查询中引入量词的计算; 文献[5]给出了在 SQLf 中引入权重的计算方法。但这些研究均未考虑关系数据库中可能存在的空值。

文献[6]指出关系数据库中的空值有 2 种: 一是 unknown 型, 即值存在, 但是现在还不知道, 比如某员工的年龄肯定存在, 但是还没有记录到数据库中; 二是 inapplicable 型, 即值没有意义、不存在, 比如某个人还没有结婚, 那么其配偶姓名的属性值就是不存在的。目前, 关系数据库的模糊查询研究大都没有考虑空值对模糊查询结果的影响。文献[7]提出了扩展的可能性真值(Extended Possibilistic Truth Values, EPTV), 用于不确定信息的计算, 指出 EPTV 可解决带有空值的模糊查询计算。但是并没有具体给出关系数据库中不同语义的空值计算方法, 此外仅通过实例说明了选择、投影和连接操作的计算方法, 而没有给出形式化的描述。本文在文献[7]的基础上, 给出不同语义的空值在 EPTV 逻辑下的表示方法, 对关系代数的其他运算, 如集合交、并、差进行了扩展, 给出了完整的定义, 此外还扩展了复杂的嵌套查询。

2 EPTV 逻辑

EPTV 是在 PTV(Possibilistic Truth Values)基础上扩展而来的。PTV 定义在可能性分布之上, 是关于命题真值的一种认知论表示, 利用 PTV 可以表示 unknown, 但无法表示 inapplicable^[7], 因此引入 \perp 来表示未定义, 将布尔集 I 扩展为 $I^* = \{T, F, \perp\}$ 。

定义 1^[7] P 是所有命题的论域, $\tilde{\varphi}(I^*)$ 表示在论域 $I^* = \{T, F, \perp\}$ 之上所有有序模糊集的集合, 一个命题 $p \in P$ 的 EPTV 记为 $\tilde{t}^*(p)$, 就是一个映射: $\tilde{t}^*: P \rightarrow \tilde{\varphi}(I^*)$, 该映射使得每个 $p \in P$ 与一个模糊集合 $\tilde{t}^*(p)$ 相关。

一般来说一个 EPTV 就是一个析取形式表示的模糊集, $\tilde{t}^*(p) = \{(T, \mu_{\tilde{t}^*(p)}(T)), (F, \mu_{\tilde{t}^*(p)}(F)), (\perp, \mu_{\tilde{t}^*(p)}(\perp))\}$ 。表 1 给出了 EPTV 中的几个特殊值以及它们的意义。

表 1 几个 EPTV 的特殊值^[7]

$\tilde{t}^*(p)$	语义
$\{(T, 1)\}$	p 完全为真
$\{(F, 1)\}$	p 完全为假
$\{(T, 1), (F, 1)\}$	p 完全未知
$\{(\perp, 1)\}$	p 完全不存在
$\{(T, 1), (F, 1), (\perp, 1)\}$	完全无信息, 即不能确定 p 是未知还是不存在, 两者皆有可能

EPTV 上的逻辑运算包括:

(1) NOT/ \neg

$$\forall p \in P: \tilde{t}^*(\text{NOT } p) = \neg(\tilde{t}^*(p))$$

其中,

$$\neg: \tilde{\varphi}(I^*) \rightarrow \tilde{\varphi}(I^*): \tilde{V} \mapsto \neg(\tilde{V})$$

$$\mu_{\neg \tilde{V}}(T) = \mu_{\tilde{V}}(F), \mu_{\neg \tilde{V}}(F) = \mu_{\tilde{V}}(T), \mu_{\neg \tilde{V}}(\perp) = \mu_{\tilde{V}}(\perp)$$

(2) AND/ \wedge

$$\forall p, q \in P: \tilde{t}^*(p \text{ AND } q) = \tilde{t}^*(p) \tilde{\wedge} \tilde{t}^*(q)$$

其中,

基金项目: 江苏省六大人才高峰高层次人才培养对象基金资助项目(06-A-027); 南京信息工程大学校科研基金资助项目(y644)

作者简介: 陈逸菲(1981-), 女, 讲师、硕士, 主研方向: 模糊信息处理, 时空数据库; 叶小岭, 副教授; 张颖超, 教授、博士生导师

收稿日期: 2009-11-16 **E-mail:** ch_yi_f@126.com

$$\begin{aligned} \tilde{\wedge} : \tilde{\rho}(I^*) \times \tilde{\rho}(I^*) &\rightarrow \tilde{\rho}(I^*) : (\tilde{U}, \tilde{V}) \mapsto \tilde{U} \tilde{\wedge} \tilde{V} \\ \mu_{\tilde{U} \tilde{\wedge} \tilde{V}}(T) &= \min(\mu_{\tilde{U}}(T), \mu_{\tilde{V}}(T)), \mu_{\tilde{U} \tilde{\wedge} \tilde{V}}(F) = \max(\mu_{\tilde{U}}(F), \mu_{\tilde{V}}(F)) \\ \mu_{\tilde{U} \tilde{\wedge} \tilde{V}}(\perp) &= \max \left(\begin{array}{l} \max \left(\begin{array}{l} \min(\mu_{\tilde{U}}(T), \mu_{\tilde{V}}(\perp)) \\ \min(\mu_{\tilde{U}}(\perp), \mu_{\tilde{V}}(T)) \end{array} \right) \\ \min(\mu_{\tilde{U}}(\perp), \mu_{\tilde{V}}(\perp)) \end{array} \right) \end{aligned}$$

(3)OR/ $\tilde{\vee}$

$$\forall p, q \in P : \tilde{t}^*(p \text{ OR } q) = \tilde{t}^*(p) \tilde{\vee} \tilde{t}^*(q)$$

其中,

$$\begin{aligned} \tilde{\vee} : \tilde{\rho}(I^*) \times \tilde{\rho}(I^*) &\rightarrow \tilde{\rho}(I^*) : (\tilde{U}, \tilde{V}) \mapsto \tilde{U} \tilde{\vee} \tilde{V} \\ \mu_{\tilde{U} \tilde{\vee} \tilde{V}}(T) &= \max(\mu_{\tilde{U}}(T), \mu_{\tilde{V}}(T)), \mu_{\tilde{U} \tilde{\vee} \tilde{V}}(F) = \min(\mu_{\tilde{U}}(F), \mu_{\tilde{V}}(F)) \\ \mu_{\tilde{U} \tilde{\vee} \tilde{V}}(\perp) &= \max \left(\begin{array}{l} \max \left(\begin{array}{l} \min(\mu_{\tilde{U}}(F), \mu_{\tilde{V}}(\perp)) \\ \min(\mu_{\tilde{U}}(\perp), \mu_{\tilde{V}}(F)) \end{array} \right) \\ \min(\mu_{\tilde{U}}(\perp), \mu_{\tilde{V}}(\perp)) \end{array} \right) \end{aligned}$$

设 $\tilde{t}^*(p) = \{(T, x_1), (F, y_1), (\perp, z_1)\}$, $\tilde{t}^*(q) = \{(T, x_2), (F, y_2), (\perp, z_2)\}$, 如果只有 x_1 和 x_2 不为 0, 则 $\tilde{\wedge}, \tilde{\vee}$ 退化为常规的模糊逻辑运算符; 如果只有 z_1 和 z_2 为 0, 则为 PTV 上的逻辑运算。显然它们都是 EPTV 逻辑的特殊情况。

3 EPTV 上的关系运算

为关系 $R(a_1, a_2, \dots, a_n)$ 增加 *contained* 列, 以表示元组属于关系的程度, 得到的新关系记作 $\tilde{R}(a_1, a_2, \dots, a_n, \text{contained})$ 。对于传统关系数据库中的元组而言, $\text{contained} = \{(T, 1)\}$ ^[7]。有 $\mathcal{O}(\tilde{R}) = R$, \mathcal{O} 表示不包含 *contained* 列。

(1)选择运算

$$\tilde{R} = \tilde{\sigma}_f(\tilde{S}) : \tilde{S}(a_1, a_2, \dots, a_k, \text{contained}), f \text{ 为选择条件}, \tilde{R}(a_1,$$

$a_2, \dots, a_k, \text{contained}$) 为结果集。如果选择条件 f 是合取的, 即 $f = f_1 \wedge f_2 \wedge \dots \wedge f_n$, 对于 $\forall \tilde{r} \in \tilde{R}$, 有

$$\tilde{r}(\text{contained}) = \tilde{s}(\text{contained}) \tilde{\wedge} \tilde{\mu}_{f_1}(\tilde{s}) \tilde{\wedge} \tilde{\mu}_{f_2}(\tilde{s}) \tilde{\wedge} \dots \tilde{\wedge} \tilde{\mu}_{f_n}(\tilde{s}) \quad (1)$$

若 f 为析取的, 即 $f = f_1 \vee f_2 \vee \dots \vee f_n$, 则

$$\tilde{r}(\text{contained}) = \tilde{s}(\text{contained}) \tilde{\wedge} (\tilde{\mu}_{f_1}(\tilde{s}) \tilde{\vee} \tilde{\mu}_{f_2}(\tilde{s}) \tilde{\vee} \dots \tilde{\vee} \tilde{\mu}_{f_n}(\tilde{s})) \quad (2)$$

其中, $\tilde{s} \in \tilde{S}$, $\mathcal{O}(\tilde{r}) = \mathcal{O}(\tilde{s})$, $\tilde{\mu}_{f_i}(\tilde{s})$ 表示元组 \tilde{s} 关于第 i 个模糊查询条件 f_i 的 EPTV 值。

(2)投影运算

$$\tilde{R} = \tilde{\Pi}_{a_1, \dots, a_k}(\tilde{S}) : \tilde{S}(a_1, a_2, \dots, a_k, a_{k+1}, \dots, a_n, \text{contained}), \text{ 结果集}$$

为 $\tilde{R}(a_1, a_2, \dots, a_k, \text{contained})$, 对于 $\forall \tilde{r} \in \tilde{R}$, 可能存在 $\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_m \in \tilde{S}$ 与之对应, 而投影运算会去除重复元组, 因此

$$\tilde{r}(\text{contained}) = \tilde{\vee}_{i=1}^m \tilde{s}_i(\text{contained}) \quad (3)$$

其中, $\tilde{r}.a_t = \tilde{s}_i.a_t, t = 1, 2, \dots, k, i = 1, 2, \dots, m, \tilde{s}_i \in \tilde{S}$ 。

(3)连接运算

$$\tilde{R} = \tilde{S}_1 \tilde{\bowtie}_{\tilde{s}_1, b, \theta, \tilde{s}_2, b} \tilde{S}_2 : \tilde{S}_1(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m, \text{contained}),$$

$\tilde{S}_2(b_1, b_2, \dots, b_m, c_1, c_2, \dots, c_k, \text{contained})$ 是参与连接运算的 2 个关系, \tilde{R} 为结果集。其中, $\tilde{s}_1, b, \theta, \tilde{s}_2, b, t = 1, 2, \dots, m$ 为连接条件, θ_t 可以是常规的比较运算符, 如 $=, >, <, \neq$ 等; 也可以是模糊比较运算符, 如 $\approx, >>, <<$ 等, 这样的连接称为模糊连接^[7]。

在计算 \tilde{R} 中元组的 *contained* 值时, 除了要考虑参与连接运算的 \tilde{S}_1 和 \tilde{S}_2 中元组各自的 *contained* 值外, 还要计算元组关于比较运算符 θ_t 的 EPTV 值。

$$\begin{aligned} \tilde{r}(\text{contained}) &= \tilde{s}_1(\text{contained}) \tilde{\wedge} \tilde{s}_2(\text{contained}) \tilde{\wedge} \\ &(\tilde{\wedge}_{t=1}^m \tilde{\mu}_{\theta_t}(\tilde{s}_1, b_t, \tilde{s}_2, b_t)) \end{aligned} \quad (4)$$

(4)集合并运算

$$\tilde{R} = \tilde{S}_1 \tilde{\cup} \tilde{S}_2 : \tilde{S}_1(a_1, a_2, \dots, a_n, \text{contained}), \tilde{S}_2(a_1, a_2, \dots, a_n,$$

contained), 对于 $\forall \tilde{r} \in \tilde{R}(a_1, a_2, \dots, a_n, \text{contained})$, 有

$$\tilde{r}(\text{contained}) = \tilde{s}_1(\text{contained}) \tilde{\vee} \tilde{s}_2(\text{contained}) \quad (5)$$

其中, $\mathcal{O}(\tilde{r}) = \mathcal{O}(\tilde{s}_1) = \mathcal{O}(\tilde{s}_2)$ 。如果 $\mathcal{O}(\tilde{s}_k) \notin \mathcal{O}(\tilde{S}_k) (k=1, 2)$, 则 $\tilde{s}_k(\text{contained}) = \{(F, 1)\}$ 。

(5)集合交运算

$$\tilde{R} = \tilde{S}_1 \tilde{\cap} \tilde{S}_2 : \tilde{R}, \tilde{S}_1, \tilde{S}_2 \text{ 的定义同集合并。对 } \forall \tilde{r} \in \tilde{R}, \text{ 有}$$

$$\tilde{r}(\text{contained}) = \tilde{s}_1(\text{contained}) \tilde{\wedge} \tilde{s}_2(\text{contained}) \quad (6)$$

其中, $\mathcal{O}(\tilde{r}) = \mathcal{O}(\tilde{s}_1) = \mathcal{O}(\tilde{s}_2), \tilde{s}_1 \in \tilde{S}_1, \tilde{s}_2 \in \tilde{S}_2$ 。

(6)集合差运算

$$\tilde{R} = \tilde{S}_1 \tilde{-} \tilde{S}_2 : \tilde{R}, \tilde{S}_1, \tilde{S}_2 \text{ 的定义同集合并。对在 } \tilde{S}_1 \text{ 内而不}$$

在 \tilde{S}_2 内元组 \tilde{r} 的 *contained* 值进行 “ $\tilde{\wedge}$ ” 运算, 即

$$\tilde{r}(\text{contained}) = \tilde{s}_1(\text{contained}) \tilde{\wedge} (\neg \tilde{s}_2(\text{contained})) \quad (7)$$

其中, $\mathcal{O}(\tilde{r}) = \mathcal{O}(\tilde{s}_1) = \mathcal{O}(\tilde{s}_2), \tilde{s}_1 \in \tilde{S}_1$ 。如果 $\mathcal{O}(\tilde{s}_2) \notin \mathcal{O}(\tilde{S}_2)$, 则 $\tilde{s}_2(\text{contained}) = \{(F, 1)\}$ 。

4 子查询的计算

第 3 节讨论了在 EPTV 基础上的关系运算。SQL 还提供了子查询机制来处理更复杂的查询。本节在第 3 节的基础上对子查询进行扩展。为了方便下文不再区分 R/r 和 \tilde{R}/\tilde{r} 。

4.1 组成员资格测试 IN

若 $fc_1(S_1), fc_2(S_2)$ 表示施加于关系 S_1, S_2 上的模糊条件, 则下面两语句等价:

$$\text{select } S_1.* \text{ from } S_1, S_2 \text{ where } fc_1(S_1) \text{ and } fc_2(S_2) \text{ and } S_1.A=S_2.B \quad (Q1)$$

$$\text{select } * \text{ from } S_1 \text{ where } fc_1(S_1) \text{ and } A \text{ IN } (\text{select } B \text{ from } S_2 \text{ where } fc_2(S_2)) \quad (Q2)$$

IN 后面 select 子句的结果是一个基于 EPTV 的扩展模糊集 S_2' 。对子句先进行选择运算, 得到 S_2 中各元组在 S_2' 中的 *contained* 值记作 $r_2'(\text{contained})$ 。主句进行选择运算后的结果同样是一个模糊集 S_1' , 每个元组关于 S_1' 的 *contained* 值为 $r_1'(\text{contained})$, 然后按下式计算:

$$\begin{aligned} r(\text{contained}) &= r_1'(\text{contained}) \tilde{\wedge} \\ &(\tilde{\vee}_{r_2' \in S_2', r_2'.B=r_1'.A} r_2'(\text{contained})) \end{aligned} \quad (8)$$

其中,

$$r_1'(\text{contained}) = r_1(\text{contained}) \tilde{\wedge} \tilde{\mu}_{fc_1}(r_1)$$

$$r_2'(\text{contained}) = r_2(\text{contained}) \tilde{\wedge} \tilde{\mu}_{fc_2}(r_2)$$

$$r_1' \in S_1', r_1 \in S_1, r_2' \in S_2'$$

对 IN 本身也可以作模糊扩展, IN_f 表示一个元素属于一个集合的程度。此时 *contained* 值的计算为

$$\begin{aligned} r(\text{contained}) &= r_1'(\text{contained}) \tilde{\wedge} \\ &(\tilde{\vee}_{r_2' \in S_2'} (r_2'(\text{contained}) \tilde{\wedge} \tilde{\mu}_{\text{IN}_f}(r_1'.A, r_2'.B))) \end{aligned} \quad (9)$$

4.2 存在测试 EXISTS

select * from S_1 where fc_1 and EXISTS (select * from S_2 where fc_2 and $S_1.A=S_2.B$)

(Q3)

把嵌套查询分离为主关系 S_1' : select * from S_1 where fc_1 和 exists 关系 S_2' : select * from S_2 where fc_2 。分别计算 2 个关系中相关元组的 *contained* 值: $r_1'(\text{contained}), r_2'(\text{contained})$, 然后对两者求 $\tilde{\wedge}$ 。

$$\begin{aligned} r(\text{contained}) &= r_1'(\text{contained}) \tilde{\wedge} \\ &(\tilde{\vee}_{r_2' \in S_2'} (r_2'(\text{contained}) \tilde{\wedge} \mu_{(r_2'.B, r_1'.A)})) \end{aligned} \quad (10)$$

其中, $r_1'(\text{contained}) = r_1(\text{contained}) \tilde{\wedge} \tilde{\mu}_{fc_1}(r_1)$, $r_2'(\text{contained}) = r_2(\text{contained}) \tilde{\wedge} \tilde{\mu}_{fc_2}(r_2)$ 。

如果(Q3)中的 “=” 改成模糊比较符 $f\theta$, 如 $\approx, <<, >>$ 等。

则匹配度的计算将式(10)中的“=”改成“ $f\theta$ ”即可。

4.3 限定测试 ALL, ANY

select * from S_1 where $A f\theta$ ANY (select B from S_2 where fc) (Q4)

设子查询结果为集合 S_2' , $r_2' \in S_2'$, 则 $r_2'(contained) = r_2(contained) \tilde{\wedge} \mu_{fc}(r_2)$ 。查询结果集为 R , 对于 $\forall r \in R$, 有 $r(contained) = r_1'(contained) \tilde{\wedge}$

$$(\bigvee_{r_2' \in S_2'} (r_2'(contained) \tilde{\wedge} \mu_{f\theta}(r_2'.B, r_1'.A))) \quad (11)$$

含 ALL 的嵌套查询可以用 NOT EXISTS 的形式表示, 因此(Q5)和(Q6)是等价的。

select * from S_1 where $A f\theta$ ALL(select B from S_2 where fc) (Q5)

select * from S_1 where NOT EXISTS(select B from S_2 where fc and not ($A f\theta B$)) (Q6)

$$r(contained) = \neg(r_1'(contained) \tilde{\wedge} (\bigvee_{r_2' \in S_2'} (r_2'(contained) \tilde{\wedge} (\neg \mu_{f\theta}(r_2'.B, r_1'.A)))))) \quad (12)$$

5 举例

引入 3 个标号表示关系数据库中空值的 3 种语义^[8]:

UNK: 表示未知但一定存在的, 其可能分布函数为

$$\pi_{UNK}(x) = \begin{cases} 1 & x \in dom(A) \setminus \{\perp_A\} \\ 0 & x = \perp_A \end{cases}$$

UND: 表示无定义, 即不存在, 其可能分布函数为

$$\pi_{UND}(x) = \begin{cases} 0 & x \in dom(A) \setminus \{\perp_A\} \\ 1 & x = \perp_A \end{cases}$$

NIO: 表示无信息, 即以上 2 种可能皆有, 其可能分布函数为 $\pi_{NIO}(x) = 1, \forall x \in dom(A)$ 。

其中, $dom(A)$ 表示属性 A 的定义域, 在这里对常规的定义域进行了扩充, 使其包含一个特殊的未定义值 \perp_A 。

下面以表 2 中的关系 EMP 和表 3 中的关系 DEP 为例说明 EPTV 上空值的不同语义对模糊查询的影响。

表 2 关系 EMP 及相关 EPTV 值

雇员	部门	年龄	工资	contained	EPTV 年轻
E1	D1	20	2 300	{(T, 1)}	{(T, 1)}
E2	D3	31	NOI	{(T, 1)}	{(T, 0.4), (F, 0.6)}
E3	D5	34	5 300	{(T, 1)}	{(T, 0.1), (F, 0.9)}
E4	D3	26	7 200	{(T, 1)}	{(T, 0.9), (F, 0.1)}
E5	D4	UNK	UND	{(T, 1)}	{(T, 1), (F, 1)}
E6	D1	UNK	UNK	{(T, 1)}	{(T, 1), (F, 1)}
E7	D5	41	6 800	{(T, 1)}	{(F, 1)}
E8	D2	44	8 100	{(T, 1)}	{(F, 1)}
E9	D3	53	UND	{(T, 1)}	{(F, 1)}
E10	D5	UNK	4 500	{(T, 1)}	{(T, 1), (F, 1)}

表 3 关系 DEP 及相关 EPTV 值

部门	预算	人数	城市	contained	EPTV 预算高
D1	350 000	12	上海	{(T, 1)}	{(F, 1)}
D2	UNK	30	南京	{(T, 1)}	{(T, 1), (F, 1)}
D3	500 000	32	北京	{(T, 1)}	{(T, 0.5), (F, 0.5)}
D4	800 000	42	上海	{(T, 1)}	{(T, 1)}
D5	UND	28	广州	{(T, 1)}	{(, 1)}

Example 查找一个年轻员工, 其所在工作部门有高预算。

SELECT 雇员, 部门, 年龄 FROM EMP WHERE 年龄 is 年轻 AND 部门 IN (SELECT 部门 FROM DEP WHERE 预算 is 高)

主句、子句的结果分别见表 4、表 5, 最终结果见表 6。如果按照常规的基于 fuzzy 集合的模糊查询^[1,7], 结果为 {(E2,D3,31,0.4),(E4,D3,26,0.5)}, 凡是出现空值的元组都被排除在结果之外。而本文的方法则能较好地地区分空值的不同语义, 并反映在查询结果中。如 E5 的年龄是未知的, 虽然其所在的部门为高预算部门是全真的, 但根据 $\tilde{\wedge}$ 运算规则, 其

在表 6 中的 contained 肯定还是未知。E10 的年龄也是未知的, 然而其所在部门 D5 的预算是不存在的, 所以关于查询的 contained 值与 E5 不相同, 而是“完全为假”或者“完全不存在”。由此可以看出不同语义的空值对查询结果有不同的影响, 这些与人们的认知是相符的。

表 4 主句的结果

雇员	部门	年龄	contained
E1	D1	20	{(T, 1)}
E2	D3	31	{(T, 0.4), (F, 0.6)}
E3	D5	34	{(T, 0.1), (F, 0.9)}
E4	D3	26	{(T, 0.9), (F, 0.1)}
E5	D4	UNK	{(T, 1), (F, 1)}
E6	D1	UNK	{(T, 1), (F, 1)}
E10	D5	UNK	{(T, 1), (F, 1)}

表 5 子句结果

部门	contained
D2	{(T, 1), (F, 1)}
D3	{(T, 0.5), (F, 0.5)}
D4	{(T, 1)}
D5	{(, 1)}

表 6 Example 的结果

雇员	部门	年龄	contained
E2	D3	31	{(T, 0.4), (F, 0.6)}
E3	D5	34	{(F, 0.9), (, 0.1)}
E4	D3	27	{(T, 0.5), (F, 0.5)}
E5	D4	UNK	{(T, 1), (F, 1)}
E10	D5	UNK	{(F, 1), (, 1)}

6 结束语

本文用不同的标号表示关系数据库中空值的不同语义, 并且在 EPTV 逻辑的基础上给出了部分关系运算和子查询的计算方法。对关系数据库空值在模糊查询中的作用进行了探讨, 实例说明该方法相对于常规模糊查询而言, 能较好地反映出不同语义的空值对模糊查询结果的影响。本文只是给出了计算的方法, 今后的工作将考虑提高计算的效率, 以保证在大数据量情况下查询的可行性。

参考文献

- [1] Bosc P, Pivert O. SQLf: A Relational Database Language for Fuzzy Querying[J]. IEEE Transactions on Fuzzy Systems, 1995, 3(1): 1-17.
- [2] Goncalves M, Tineo L. SQLf Flexible Querying Language Extension by Means of the Norm SQL2[C]//Proc. of IEEE International Fuzzy Systems Conference. [S. l.]: IEEE Press, 2001: 473-476.
- [3] Goncalves M, Tineo L. SQLf3: An Extension of SQLf with SQL3 Features[C]//Proc. of IEEE International Fuzzy Systems Conference. [S. l.]: IEEE Press, 2001: 477-480.
- [4] Bosc P, Lietard L, Pivert O. Quantified Statements in a Flexible Relational Query Language[C]//Proc. of ACM Symposium on Applied Computing. New York, USA: ACM Press, 1995: 488-492.
- [5] Zhang Yingchao, Chen Yifei, Ye Xiaoling, et al. Weighted Fuzzy Queries in Relational Database[C]//Proc. of the 2nd International Conference on Fuzzy System and Knowledge Discovery. Berlin, Germany: Springer Berlin Press, 2005: 430-440.
- [6] Codd E F. Missing Information (Applicable and Inapplicable) in Relational Databases[J]. ACM SIGMOD, 1986, 15(4): 53-78.
- [7] Matthé T, Tré G D. Flexible Querying in a Relational Framework Supported by Possibility Logic[J]. Fuzzy Sets and Systems, 2008, 159(12): 1468-1484.
- [8] Tré G D, Caluwe R D, Prade H. Null Values in Fuzzy Databases[J]. Intelligent Information Systems, 2008, 30(2): 93-114.

编辑 任吉慧