文章编号: 1001 - 9081(2008)S2 - 0235 - 03

非确定性数据库中空值处理

姜小华, 罗 军
(重庆大学 计算机学院,重庆 400030)
(luckylife@126.com)

摘 要:尽管关系数据库有很多优势,但它缺乏一种处理非确定性数据的能力。目前,已经提出了几种将非确定性结合到关系数据库模型的方法,它们对关系数据库模型做了诸多扩展。但空值问题依旧存在,一些模型根本就没有考虑空值因素。这违背了非确定性数据库要更加真实地反应现实世界的初衷。为此,给出了一种非确定性数据库系统中空值处理方法,改进现有非确定性数据库模型中对空值处理不完善的情况。

关键词:非确定性数据;非确定性数据库;空值;关系数据库;数据库模型
中图分类号: TP311.13 **文献标志码**: A

On handling NULL values in uncertain database

JIANG Xiao-hua, LUO Jun
(*College of Computer Science, Chongqing University, Chongqing* 400030, *China*)

Abstract: Although the relational databases have a wide range of advantages, it lacks a way to handle uncertain data. Several methods have been proposed for incorporating uncertain data into relational databases and extended the relational databases model in some aspect. However, most of them lack a way to handle NULL value. In some model, NULL value is ignored. These violate a rule that uncertain database will reflect the real world as well as possible. A method of handling NULL value in uncertain database systems was put forward, which could improve the problem of handling NULL value.

Key words: uncertain data; uncertain databases; NULL value; relational databases; database model

1 空值简介

1.1 数学中的空

在没有数字 0以前,人们以空格或者括弧等表示没有的概念,比如 123、1()2()3。在古印度数学中印度人开始用“.”表示没有,并逐渐地演变成一个圆圈,即今天的数字 0。采用 0后,上面的数字就可以简洁地表示成: 10203。

数字 0的引入不仅仅表示了没有,还起到了占位的作用,表示界限、起点、精度,有时还表示“有”的作用。严格说,在没有引入 0以前,计数法是不完整的。

1.2 关系数据库中的空

关系数据库中的空值表示未知的值,这个值即不是数字 0也不是空字符串也不是其他任何有意义的值。在早期的数据库中没有空值的概念,所有值都是确定可知的。有了空值,使得我们可以表示那些不知道的数据,也可以表示那些没有定义的数据,当然还可以表示那些由于人为因素而有误的数据。空值的引入使得关系数据库的数据表示更加完整。

1.2.1 相对空和绝对空

所谓相对空是指在关系 R中一个元组 T的某个属性 A在之前的某个时刻被赋予某个值,但在之后的某个时刻值被清除掉了的状态。

所谓绝对空是指在关系 R中一个元组 T的某个属性 A从创建开始到最后都没有赋过值,一直处于空的状态。

假设关系表 R中某个元组 T的 X属性值是空白的,在 R上分别执行查询:

Q1: SELECT * FROM R WHERE X IS NULL

Q2: SELECT * FROM R WHERE X IS NULL OR X = ''

查询 Q1和 Q2并无太大差别,但当元组 T的 X属性的值是相对空时,查询 Q1返回的结果表中不包括元组 T;当是绝对空时,Q1和 Q2均包括元组 T。

在实际数据库系统中,我们看到的绝对空和相对空都表现为空白的(在没有特别注明的情况下,下文讨论的空均为绝对空)。

1.2.2 空值的不足

由于空值的引入,使得某些查询产生了与实际不符的结果。

如表 1的关系,其中 E是雇员信息表,雇员 E1的 AGE属性值存在空值(以“NULL”标识)。

表 1 关系 E(雇员信息表)

ENO	NAME	AGE
E1	NAME1	NULL
E2	NAME2	20
E3	NAME3	30

现在,考虑下面三个查询语句:

1) SELECT COUNT(*) FROM E;

2) SELECT COUNT(NAME) FROM E;

3) SELECT COUNT(AGE) FROM E;

我们发现,查询 A、B返回的结果为 3;而查询 C返回的结果为 2。

关系 E中有三条记录,但查询 C返回的却是两条。这正是由于 AGE属性存在空值,这个未知的值让数据库系统做出了与实际不符的回答。对于以下查询:

4) SELECT * FROM E WHERE AGE <> '20';

期望查询 D能返回记录 E1和 E3,但是,它仅仅返回了

收稿日期: 2008 - 06 - 25;**修回日期**: 2008 - 08 - 04。

作者简介:姜小华(1980 -),男,重庆人,硕士研究生,主要研究方向:数据库; 罗军(1961 -),男,重庆人,副教授,主要研究方向:网络及数据库、基于数据系统的应用平台的架构、大型 MIS建模与设计。

E3, E1再一次由于空值的存在而被忽略掉了。

2 非确定性数据库

关系数据库自诞生以来,以其数据结构简单、清晰,数据存取的灵活性和独立性,完整性,冗余少以及应用方便,很快在各个领域得到了广泛的应用。但在实际应用中也反应出关系数据库的一些不足,比如对现实世界中非确定性数据处理,现在的关系数据库还无能为力。而这样的数据在现实世界中随处可见,比如传感器的数据。随着科技和数据库应用领域的不断发展深入,很多领域都迫切需要对非确定性数据更精确更合理的处理,如数据整合、数据抽取、科学计算、多媒体应用以及传感数据领域等。

对非确定性数据的处理迫切需求促使相关的研究蓬勃发展,现已提出很多非确定性数据库模型及查询方法[1, 3-6]。它们中的大多数都是基于现有的关系数据库模型,是对现有关系数据库模型的扩展。这些非确定性数据库模型对关系数据库模型扩展基本可归纳为以下几点:

1)打破关系数据库属性值的原子性,大多数非确定性数据库都以集合的或者类集合的形式来描述属性值。

2)关系数据库元组是确定的,但是在非确定性数据库中是以一定的概率存在的。

3)在元组、属性值或者元组和属性值中加入了置信度的概念。

4)主键不再唯一标识一个元组,可能标识几个元组。我们可以把主键看做一个对象 ID,标识一组元组,其中的每一个元组为这个对象的一种状态。

非确定性数据库模型实例可以参考文献[1, 3-6]。

非确定性数据库系统中利用各种非确定性数据处理技术,在一定程度上更真实反映了现实世界。但是对空值的处理研究甚少。文献[1]中的非确定性数据模型 URM把非确定性分为概率和认知的不确定性,并在此基础上提出了一个非确定性数据模型及其语义,并为每个元组定义了置信度,为模型定义了非确定性世界,但是忽略了对空值的处理;同样在文献[3]ULDBs中利用数据沿袭和可能性集合来描述非确定性数据,提出了 ULDB数据模型,模型中没有明确提出空值的处理,只是变相地利用可能性集合理论来表示非确定性数据,对那些不可知的数据 ULDB是无法描述的。

虽然非确定性数据库中引入了置信度、可能性集合等概念,但是并不能完全解决空值问题,比如,在一个记录病人信息的数据库中,病人信息表的一个属性“是否有小孩”,这个属性值对我们来说是不可知的(除病人提供),置信度和可能性集合在这里失效了。然而从1.2.2节可知道,如果在非确定性数据库沿用关系数据库中的空值概念,又至少存在以下不足[2]:

1)在语义上模糊,而且根据空值的定义,在非确定性数据库中空值缺乏表现多个可能取值的能力;

2)在对空值进行检索中,可能会造成信息的丢失和不完整;

3)由于空值的未知性,在很多商业应用中,空值往往是被忽略的,这也造成很多决策信息的不完整和不精确。

所以需要一个更为系统的方法来处理空值,这也是本文的主要研究内容。

3 空值处理

定义 1 让 O 表示一个非确定性对象,A_i 表示对象 O 的一个属性,D_j 是一个 A_i 的域。属性 A_i 的值(表示为 $O.A_i$)表示为一个集合,这个集合包含在 $D_j \cup \{N\}$。其中 $\{N\}$ 表示一个 D_j 之外未定义的集合。

不失一般性,定义 $T.A_i$ 的值为一个信息对的集合。

定义 2 让 $I \in D_j \cup \{N\}$ 的幂集,P 为 I 的概率值(在此我们忽略 P 和 I 之间的映射关系定义,具体非确定性模型的映射关系定义有区别),信息对 $\langle I, P\rangle$ 表示 $O.A_i$ 在取值 I 时的置信度为 P。

讨论 P 的几种可能的取值:

1)当 $0 < P < 1$ 时,I 为非确定性数据,在不同的非确定性数据库模型中已经分别讨论,而且当 $0 < P < 1$ 时不存在空值问题,所以本文不再讨论这种情况。

2)当 $P = 0$ 时,即概率为0的事件为不可能事件,可忽略。

3)重点讨论当 $P = 1$ 时 I 的各种取值。假设一个雇员关系模式 Employee = (ENO, NAME, AGE, FIRSTCHILD, HASBANDNAME, BIRTHDAY, ADDR)。用 E_i 表示一个 Employee对象。E_i 的属性 A_j 的取值表示为 $E_i.A_j$,它的取值域为 $D \cup \{N\}$。

①当 $|I| = 1$,精确数据,即关系数据库中的确定数据。

$E_i.$ BIRTHDAY = <1/1/1980, 1.0>

这就意味着 E_i 的生日是 1980年 1月 1日。

②当 $1 < |I| \leq |D|$,表示我们知道一个确定的值在域 D 的子集中。这是非确定性信息的一种,它的处理见文献[1, 3-6],本文不再表述。

③当 $I = D$,这时的属性值是为我们所不知的,因为它的取值可能是域 D 中的任何一个。关系数据库中用空值表示。

定义 3 当 $I = D$ 时,定义这个属性值是不可知的。用标识符 NK表示。

假设雇员关系中 AGE的域 $D = \{15, 16, 17, 18, \cdots, 148, 149, 150\}$。

$E_i.AGE = \langle$NK, 1.0$\rangle$

表示雇员 i 的年龄可能是域 D 中的任何一个值。

④当 $I = \{N\}$,这时的属性值不可用的,因为它的取值不在域中。关系数据库中用空值表示。

定义 4 当 $I = \{N\}$,定义这个属性值是不可用的。用标识符 NV表示。

在雇员关系中,如果知道某个女雇员还没有结婚。那么她的 HASBANDNAME就是不可用的。表示为:

$E_i.$ HASBANDNAME = $\langle$NV, 1.0$\rangle$

⑤当 $I = D \cup \{N\}$,这时的属性值既是不可用的也是不可知的,即完全不能知道这个属性的值。

定义 5 当 $I = D \cup \{N\}$,定义这个属性值是不可用也不可知的。用标识符 NVK表示。

在雇员关系中如果完全不知道某个雇员是否有小孩。表示为:

$E_i.$ FIRSTCHILD = $\langle$NVK, 1.0$\rangle$

以上的 NV、NK、NVK分别表示了绝对空的三种不同情况。

⑥当 $I = \varnothing$,这时的属性值为集合空,即没有值。这就是我们第一节定义的相对空的情况。

定义 6 当 $I = \varnothing$ 时,定义这个属性值是相对空的。用标识符 NE表示。

至此,我们把空值根据不同情况分别表示成了 NK、NV、NVK、NE。在非确定性数据库系统中完全祛除掉了空值。

用 NK、NV、NVK、NE代替空值主要有以下好处:

1)语义上非常清晰,NK是不可知的,NV表示不可用的,NVK既不可用也不可知,NK、NV、NVK都表示绝对空,NE表示相对空。也非常清晰明了地表现了非确定性数据库系统中多个取值的情况。

2)利用 NK、NV、NVK和 NE检索出的数据将更加准确。

3)在应用中可以忽略空值,但不能忽略 NK、NV、NVK、NE,因为 NK、NV、NVK、NE代表了不同意义的集合。

4 NK、NV、NVK、NE运算

在大多数非确定性数据库中,属性 (个别属性除外,比如主键)都以集合形式表示,所以我们也以集合形式定义 NK、NV、NVK、NE在某个属性下的运算 (见表 2,"交 "表示交集,"并 "表示并集,由于对称性,表中只标出下半部分),本文只讨论交集和并集的情况,其他集合操作将另文讨论。

表 2 NK、NV、NVK、NE运算

	NK	NV	NVK	NE
NK	交:NK并:NK			
NV	交:NE并:NVK	交:NV并:NV		
NVK	交:NK并:NVK	交:NV并:NVK	交:NVK并:NVK	
NE	交:NE并:NK	交:NV并:NV	交:NE并:NVK	交:NE并:NE

5 NK、NV、NVK、NE应用

因为本文主要讨论空值的处理,为了简便起见,忽略非确定性数据的表示和处理。以先前定义的雇员关系模式为例,简要叙述 NK、NV、NVK、NE的应用,如表 3。

表 3 NK、NV、NVK、NE应用 (忽略掉了非确定性数据)

ENO	NAME	AGE	FIRSTCHILD	HASBANDNAME	BIRTHDAY	ADDR
01	〈张三,1.0〉	〈28,1.0〉	NVK	NV	〈1/1/1980,1.0〉	NK
02	〈李四,1.0〉	NK	NK	〈王五,1.0〉	NK	NE

表 3完全摆脱了空值,取而代之的是 NV、NK、NVK和 NE,这样在语义上非常清晰。从表 3知道张三肯定没有结婚,因为其 HASBANDNAME属性值为 NV的。也很容易知道李四的年龄和出生日期是不可知的。

如果要检索所有未结婚的女员工,只需检索 HASBANDNAME属性值为 NV就行了,因为只有未婚女性的 HASBANDNAME为不可用的。在用空值表示的时候是无法表达这么明确的意义而且无法检索出符合实际的结果。

如果从表 3检索所有年龄小于 30的员工返回的是员工 01和 02,因为员工 02的 AGE是 NK的,不能被忽略掉。

6 结语

本文提出一种适用于非确定性数据模型的空值表示法,并把空值分为绝对空和相对空,利用 NK、NV、NVK和 NE分别定义和表示。该方法使得:1)概念上更加清晰;2查询和检索更加精确;3)更真实地反应了现实世界的情况。

空值的处理只是非确定性模型中需要解决的问题之一。还有很多其他问题,比如非确定性数据的表示,非确定性关系的定义,以及非确定性查询等。有关这方面的工作将是我们下一步的研究内容。

参考文献:

[1] 蒋运承,张师超. 一个不确定性数据库模型及其语义 [J]. 计算机科学,1999(26):78 - 81.

[2] 洪玫,沈琳. 关系数据库中不确定值的处理 [J]. 四川联合大学学报:工程科学版,1998,2(1),95 - 96.

[3] BENJELLOUN O, SARMA A D, HALEVY A, *et al*. ULDBs: Databases with uncertainty and lineage[C]// Proceedings of the 32nd International Conference on Very Large Data Bases. New York: ACM, 2006: 953 - 956.

[4] ANTOVA L, KOCH C, OLTEANU D. From complete to incomplete information and back[C]// Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2007: 713 - 724.

[5] BENJELLOUN O, SARMA A D, HALEVY A. Working model for uncertain data[C]// Proceedings of the 22nd International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2006: 7 - 7.

[6] AGRAWAL P, BENJELLOUN O, SARMA A D, *et al*. Trio: A system for data, uncertainty, and lineage[C]// Proceedings of the 32nd International Conference on Very Large Data Bases. New York: ACM, 2006: 1151 - 1154.

(上接第 234页)

表 1 多连接属性划分算法和同一连接属性划分算法响应时间比较

关系的元组数	多连接属性划分算法响应时间/s	同一连接属性划分算法响应时间/s
1000	2.085	2.665
2000	4.220	5.680
3000	6.460	8.745
4000	7.720	12.250
5000	11.535	15.345

4 结语

本文分析和研究了传统的直接连接查询优化算法和同一属性划分连接查询优化算法,提出了一种新的查询优化算法——多连接属性划分的查询优化算法,减少了查询的响应时间,在处理分布式数据库中海量信息查询和复杂查询方面具有实用价值。当然,由于分布式数据库的建立环境复杂,技术内容丰富,对于查询优化技术还有许多问题有待进一步研究和解决。随着计算机网络技术的飞速发展,相信分布式数据库也必将得到迅速发展,并日趋完善。

参考文献:

[1] 邵佩英. 分布式数据库系统及其应用 [M]. 北京:科学出版社,2000.

[2] 王能斌. 数据库系统原理 [M]. 北京:电子工业出版社,2001.

[3] SIEGEL J, SHIM J. 数据库管理系统 [M]. 尹买华,译. 北京:清华大学出版社,2003.

[4] ROB P, CORONET C. 数据库系统设计、实现与管理 [M]. 陈立军,译. 5版. 北京:电子工业出版社,2004.

[5] 毛国君. 高级数据库原理与技术 [M]. 北京:人民邮电出版社,2004.