基于 BCNF 的数据模型的层次分解算法*)

胡 军 夏 英

(重庆邮电大学计算机科学与技术学院 重庆 400065)

摘 要 模式分解是减小关系数据库数据冗余,排除操作异常的有效工具,同时也是关系数据库数据模型设计的难点。针对该问题,本文基于分层递阶的思想,提出了一种方便可行的基于 BCNF 的数据模型的层次分解算法,为模式分解提供了新的方法,并通过实例验证了算法的有效性。

关键词 模式分解,范式,候选码,函数依赖

1 引言

数据库技术在现代数据管理中的广泛应用,促进了数据管理的自动化,有效地利用数据库技术可以提高数据管理的效率,同时给用户带来便利。但随着数据的日积月累,数据量越来越庞大,使得数据库系统的运行变得无法忍受,甚至造成系统的崩溃和数据的丢失。因而,如何设计一个好的数据模型,提高数据库系统的运行效率和减小冗余度是数据库技术中的关键技术。

文[1]从函数依赖和多值依赖的角度给出了评价一个数据模型的标准,其中主要有第一范式(1NF),第二范式(2NF),第三范式(3NF),BC 范式(BCNF)和第四范式(4NF)。从一般的应用来说,对于数据模型的设计达到 3NF 或 BCNF 就可以了。文[1]中给出了在不同的约束条件下的模式分解算法,但这些方法操作起来比较复杂,且不易理解。文[2]基于函数依赖,给出了一个较简单的模式分解方法,并证明该分解算法具有无损连接性,但该算法操作起来可行性不大,特别是在函数依赖关系比较复杂,并且函数依赖之间存在多级级连的情况下。针对该问题,本文基于 BCNF 提出了一种数据模型的层次分解算法,并证明该分解具有无损连接性。

2 基本概念介绍

在关系数据模型中,一般一个关系数据库包含若干个关系模式。关系模式是对关系的描述,它是一个五元组:$R(U,D,dom,F)$,其中 R 是关系名,U 为组成该关系的属性名集合,D 为属性组 U 中属性所来自的域,dom 为属性向域的映象集合,F 为属性间数据的依赖关系集合。

定义 1 设 $R(U)$ 是属性集 U 上的关系模式。X,Y 是 U 的子集。若对于 $R(U)$ 的任意一个可能的关系 r,r 中不可能存在两个元组在 X 上的属性值相等,而在 Y 上的属性值不等,则称 X 函数确定 Y 或 Y 函数依赖于 X,记 $X\rightarrow Y$。

定义 2 设 K 为 $R(U,F)$ 中的属性或属性组合,若 $K\xrightarrow{F}U$ 则 K 为 R 的码,并且任意的 $SK\in U$,若 $K\subseteq SK$,则 SK 为 R 的超码。

定义 3 关系模式 $R(U,F)\in 1NF$。若 $X\rightarrow Y$ 且 $Y\not\subset X$ 时 X 为 R 的超码,则 $R(U,F)\in BCNF$。

定义 4 设 F 为属性集 U 上的一组函数依赖,$X\subseteq U$,$X_F^+=\{A\mid X\rightarrow A$ 能有 F 根据 *Armstrong* 公理导出$\}$,X_F^+ 称为属性集 X 关于函数依赖集 F 的闭包。

定义 5 在关系模式 $R(U,F)$ 中为 F 所逻辑蕴涵的函数依赖的全体叫做 F 的闭包,记为 F^+。

定义 6 设 $U_i\subseteq U$,函数依赖集合$\{X\rightarrow Y\mid X\rightarrow Y\in F^+\wedge XY\subseteq U_i\}$的一个覆盖 F_i 叫做 F 在属性 U_i 上的投影。

定义 7 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。

若关系模式 $R(U,F)$ 的一个分解 $\rho=\{R_1(U_1,F_1),R_2(U_2,F_2),\cdots,R_n(U_n,F_n)\}\in BCNF$,则 $R_i(U_i,F_i)\in$BCNF,$1\leqslant i\leqslant n$,即分解后得到的每一个关系模式为 BCNF。

3 基于 BCNF 的数据模型的层次分解算法

在给出关系模式分解算法之前,这里先讨论如何求关系模式的候选码。求解一个关系模式所以候选码的问题被证明是一个 NP 完全问题。文[4]和

*)基金项目:重庆市精品课程《数据库原理》(2004JPKC-1);重庆邮电大学自然科学基金项目(A2006-56)。胡 军 博士研究生,讲师,主要研究领域包括知识发现、粗糙集等;夏 英 副教授,主要研究领域包括数据库系统等。

文[5]对该问题进行了分析，并给出了相应的候选码的求解方法。这两种方法都是根据函数依赖中的属性在函数依赖两边出现的情况将函数依赖中的属性进行了分类，主要有 A_L（代表仅仅出现在函数依赖左边的属性集）、A_R（代表那些仅仅出现在函数依赖右边的属性集）、A_{LR}（代表既出现函数依赖左边，也出现在函数依赖右边的属性集），并给出了各类属性的性质。但是，这两种方法都忽略了一类属性，那些属性既没有出现在函数依赖的左边，也没有出现在函数依赖的右边，这里本文用 A_{NULL} 表示。

定理 1 设有关系模式 $R(U,F)$，若 A_{NULL} 非空，则 A_{NULL} 中的任一属性必包含在关系模式的候选码中。

证明：设 K 为 R 的码，属性 $A\in A_{NULL}$，但 $A\notin K$。易知，$K\rightarrow A\in F^+$，这说明一定存在函数依赖：$X\rightarrow A$，$X\subseteq U$，这与 $A\in A_{NULL}$ 矛盾，从而定理得证。

根据以上定义，下面给出一个求解关系模式候选码的算法。

算法 1（求关系模式候选码的算法）

输入：关系模式 $R(U,F)$

输出：候选码 candidateKey

第一步：根据定义将 U 中的属性分为 A_L、A_R、A_{LR}、A_{NULL}，令 condidateKey $=A_L\cup A_{NULL}$，计算 $condidateKey_F^+$，若 $condidateKey_F^+=U$，算法结束，并且 condidateKey 为关系模式的唯一候选码，否则令 $A_{Temp}=\phi$；

第二步：若 $A_{LR}\neq\phi$，从 A_{LR} 中选择一属性 a，令 condidateKey = condidateKey $\cup\{a\}$，$A_{LR}=A_{LR}-\{a\}$，$A_{Temp}=A_{Temp}\cup\{a\}$；

第三步：计算 $condidateKey_F^+$，若 $condidateKey_F^+\neq U$，转第二步；

第四步：若 $A_{Temp}\neq\phi$，从 A_{Temp} 中选择一属性 a，令 condidateKey = condidateKey $-\{a\}$，$A_{Temp}=A_{Temp}-\{a\}$，否则算法结束并且 condidateKey 为关系模式的候选码之一；

第五步：计算 condidateKey$_F^+$，若 condidateKey$_F^+\neq U$，condidateKey = condidateKey $\cup\{a\}$。转第四步；

第六步：结束，输出 condidateKey。

对于复杂问题的求解，人们习惯于将该问题分解成小的问题，逐步细化并分别求解，从而得到原问题的解。这也是本文提出的层次分解算法的基本思想，具体地说，就是对于一个不符合 BCNF 的关系模式，首先将其分解成两个关系模式，如果其中仍然存在不符合 BCNF 的关系模式，对其重复以上过程，直到分解得到的每一个关系模式都符合 BCNF，即得到一个满足 BCNF 的原关系模式的分解。

算法 2（基于 BCNF 的数据模型的层次分解算法）

输入：关系模式 $R(U,F)$

输出：关系模式 $R(U,F)$ 的一个分解 $\rho\in BCNF$

第一步：设 $\rho=\phi$，$\rho'=\{R(U,F)\}$；

第二步：若 $\rho'\neq\phi$，从 ρ' 中顺序取 $R(U,F)$，令 $\rho'=\rho'-\{R(U,F)\}$，$F'=F=\{F_1,F_2,\cdots,F_n\}$，否则算法结束；

第三步：若 $F'\neq\phi$，从 F' 中顺序取 $F_i:X\rightarrow Y$，$1\leqslant i\leqslant n$，$F'=F'-\{F_i\}$，否则 $\rho=\rho\cup\{R(U,F)\}$，转第二步；

第四步：计算 X_F^+，若 $X_F^+=U$，转第三步，否则令 $U_1=X_F^+$，F_1 为 F 在 U_1 上的投影，得到新的关系模式 $R_1(U_1,F_1)$，根据算法 1 求关系模式 R_1 的码 K_1，又令 $U_2=(U-U_1)\cup K_1$，F_2 为 F 在 U_2 上的投影，得到另一个新的关系模式 $R_2(U_2,F_2)$，$\rho'=\rho'\cup\{R_1(U_1,F_1),R_2(U_2,F_2)\}$，转第二步；

第五步：算法结束，输出 ρ。

由以上算法可知，若要证明该算法得到的关系模式的分解具有无损连接性，则根据文[1]的引理 5.5 只需要证明 $R(U,F)$ 到 $R_1(U_1,F_1)$ 和 $R_2(U_2,F_2)$ 的分解具有无损连接性。

证明：设 $R_1(U_1,F_1)$ 和 $R_2(U_2,F_2)$ 是 $R(U,F)$ 按算法 2 在一次循环得到的分解，K_1 是 $R_1(U_1,F_1)$ 的码，由算法 2 可知 $U_1\cup U_2=K_1$。由码的涵义，$R_1(U_1,F_1)$ 中存在函数依赖 $K_1\rightarrow U_1-K_1$，即 $U_1\cap U_2\rightarrow U_1-U_2$，根据文[1]的定理 5.5 可知 $R(U,F)$ 到 $R_1(U_1,F_1)$ 和 $R_2(U_2,F_2)$ 的分解具有无损连接性。

4 实例分析

有关系模式 $R(U,F)$，其中 $U=$ {Sno, Sname, Sdept, Mname, Sloc, Cno, Cname, Grade }，F = {(sno, Con) → Sname, (Sno, Cno) → Sdept, (Sno, Cno) → Mname, (Sno, Cno) → Sloc, (Sno, Cno) → , Cname, (Sno, Cno) → Grade, Sno → Sname, Sno → Sdept, Sno → Sloc, Sno → Manme, Sdept → Sloc, Sdept → Mname, Cno → Cname}。

(1) 取 R(U, F)，由于 Sno$_F^+$ = {Sno, Sname, Sdept, Sloc, Mname} ≠ U，得 $R_1(U_1,F_1)$，其中 U_1 = {Sno, Sname, Sdept, Sloc, Mname}，F_1 = {Sno → Sname, Sno → Sdept, Sno → Sloc, Sno → Mname, Sdept → Sloc, Sdept → Mname }，且 R_1 的码为 Sno。因此得到 $R_2(U_2,F_2)$，其中 U_2 = {Sno, Cno, Cname, Grade}，F_2 = {(Sno, Cno) → Cname, (Sno, Cno) → Grade, Cno → Cname}。

(2) 取 $R_1(U_1,F_1)$，由于 Sdept$_{F_1}^+$ = {Sdept, Sloc, Mname} ≠ U_1，得 $R_3(U_3,F_3)$，其中 U_3 = {Sdept, Sloc, Mname}，F_3 = {Sdept → Sloc, Sdept →

（下转第 91 页）

lld,Statement)"+")";

该表用来存储各个 RDF 模型。

4.2 视图生成实例:产生视图包含数据库中所有的 statement

sql="CREATE OR REPLACE VIEW RootModel" +" AS SELECT UNIQUE Id, Subject, Predicate, ObjResource, ObjLiteral, Res, Asserted, Reified"+"FROM RDFStatement,RDFModel" +"WHERE RDFModel。Statement = RDFStatement. Id";

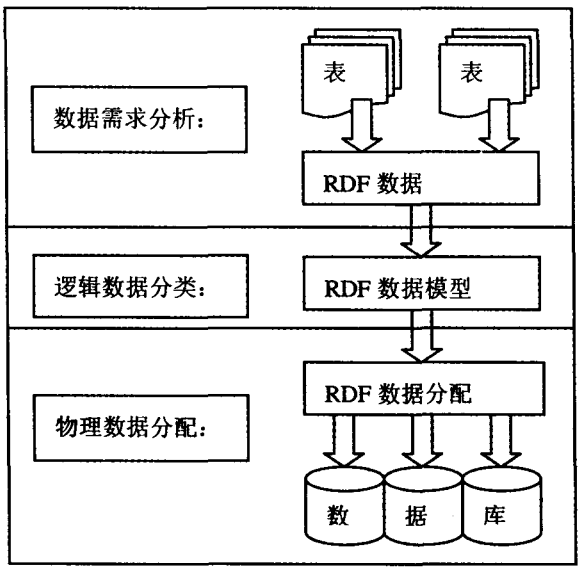

图 2 数据库设计过程

5 关系数据库的设计过程

在设计数据库时,可以从数据库表中提取和日常操作相关的字段,根据字段特点对表中的字段进行分类,在分类的过程中将具有相似特点的字段自下而上分组构造抽象类。通过字段的分类和抽象存储 RDF 数据,抽象类就是数据库中的表,基类就是表的属性。数据库处理设计分为三个阶段:数据需求分析、逻辑数据分类和物理数据分配。在数据需求分析阶段通过应用需求分析存储了 RDF 数据,逻辑数据分类阶段进一步将结构和 RDF 数据集成构造了 RDF 数据模型,物理数据分配阶段获得了满足应用需求的最小数据处理成本,并将数据分配合适。设计过程见图 2。

结束语 随着 RDF 在 Web 上的应用,对 RDF 的操纵需要对 RDF 数据进行存储,本文在介绍 RDF 的基本概念和语法后比较了 RDF 数据存储时水平行表示与垂直三元表示的优缺点,给出了存储 RDF 数据的数据库的表和视图,研究了数据库的设计过程。分析表明关系数据库是一种有效存储 RDF 数据的方法,我们将在以后对 RDF 存储技术作进一步探讨以推进 RDF 在 Web 上应用的发展。

参 考 文 献

1 Lassila, Swick R R. Resource description framework (RDF) model and syntax specification. W3C. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/,1999-02-22

2 Lassila, Swick R R. Resource Description Framework (RDF) schema specification1.0. [EB/OL]. http://www.w3.org/TR/1999/REC-rdf-syntax

3 Brickley D,Guha R V. Resource Description Framework(RDF) schema specification1.0. [EB/OL]. http://www.w3.org/TR/2000/CR-rdf-schema-2000-03-27

4 Jensen C J, Jeffery K G, Pokorny J. Storage and querying of E-commerce data[A]. Advances in Database Technology-Edbt, 2002.409

(上接第 85 页)

Mname},且 R_3 的码为 Sdept。从而得到 $R_4(U_4, F_4)$,其中 U_4={Sno, Sname, Sdept},F_4={Sno→Sname,Sno→Sdept}。

(3)取 $R_2(U_2,F_2)$,由于 $Cno_{F_2}^{+}$={Cno,Cname}≠U_2,得 $R_5(U_5,F_5)$,其中 U_5={Cno,Cname},F_5={Cno→Cname},且 R_5 的码为 Cno。从而得到 $R_6(U_6, F_6)$,其中 U_6={Sno, Cno, Grade},F_6={(Sno,Cno)→Grade}

(4)由于 R_3,R_4,R_5,R_6 中都不存在函数依赖不包含码,因此它们都属于 BCNF,也即为 R 满足 BCNF 的一个分解。

结束语 模式分解是减小关系数据库数据冗余,排除操作异常的有效工具,同时也是关系数据库数据模型设计的难点。针对该问题,本文基于分层递阶的思想,提出了一种基于 BCNF 的数据模型的层次分解算法。通过实例分析,该算法简化了模式分解的步骤,并解决了文[2]提出的算法不能处理属性之间存在多级函数依赖的情况,为模式分解提供了新的方法。

参 考 文 献

1 萨师煊,王珊. 数据库系统概论(第三版)[M]. 北京:高等教育出版社,2003

2 马雪英,冯睿. 基于函数依赖的模式分解方法[J]. 计算机应用与软件,2004,21(4):31~33

3 Ullman J D, Widom J, Widom J D. A First Course in Database Systems(second edition)[M]. 2001

4 覃遵跃,徐洪智,冯峻松,蔡国民. 利用函数依赖图寻找关系模式的候选码[J]. 安庆师范学院学报(自然科学版),2004,10(1):3~5

5 张永,顾国庆. 关系模式中候选码的求解[J]. 上海电力学院学报,2002,18(1):33~35