维普资讯 http://www.cqvip.com

计算机科学1997Vol.24№.6

继承性实现嵌套关系的方法*)

An Approach to Implementing Nested Relation Using Inheritance

秦小麟 林钧海 皮德常 徐笛子

(南京航空航天大学计算机科学与工程系 南京210016)

摘 要 This paper presents an approach to implementing nested relation using object-oriented inheritance in relational database. Its principle is that nested relations are constructed by relational multi-inheritance and transitive inheritance. It discusses emphatically the definition and construction of nested relations as well as strategies of implementing various SQL statements. The method has features of good interface and easy implementation.

关键词 Nested relation, Inheritance, Relational database

1 引言

所谓嵌套关系(即非第一范式),即是允许关系中的属性又可是一个关系,打破了关系中属性必须是原子属性的约束,可以出现表中有表的情况,用以表示具有层次结构的复杂对象,提高了关系模型表达现实世界的能力。

本文不准备讨论嵌套关系的理论,仅探讨一下实现嵌套关系的方法和技术。分析一些嵌套关系的原型系统,多数的实现方法是在关系数据库上加一层能够处理嵌套关系的机制,通过基本关系来构成嵌套关系。其实现手段不尽相同,有的采用引入关系数据类型[1],有的采用扩充SQL语言[2,3],还有的采用过程类型[4]等。本文介绍一种用面向对象的继承性技术实现嵌套关系的方法。

2 嵌套关系的表示

嵌套关系通常可表示为一棵模式树,如图1。其

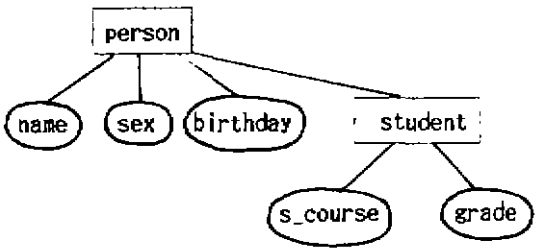

图1

中矩形表示关系,椭圆形表示属性。对应的嵌套表格为图2所示。

person				
name	sex	birthday	student	
			s_course	grade

图2

继承性概念来源于面向对象的思想,我们在关系中引入继承性质,以此来构造嵌套关系。

定义1 未构成继承性质的关系称为基关系,即是平关系。

定义2 关系可以按ISA联系构成层次。对两个关系R1和R2,R1 ISA R2表示R1是R2的直接子关系,R2是R1的直接超关系。

定义3 关系间存在继承性质,子关系可以全部或部分继承其超关系的属性。

定义4 关系继承存在传递性。若关系B继承关系A的全部属性,关系C继承关系B的全部属性,则关系C同时继承关系A和关系B的全部属性。

关系可以单继承(如图1),也可以多继承(如图3)。

图3中关系person存储基本信息,关系teacher存储教师授课的课程名与教室,关系student存储学生选修的课程名与成绩,关系ts存储同时具有教师与学生身份的人(如在职研究生)的信息。可以看出

*)本课题系航空科学基金资助项目

维普资讯 http://www.cqvip.com

teacher 继承了 person 的全部属性，student 继承了关系 person 的 sex 与 birthday 属性，ts 继承了 teacher 和 student 的全部属性。

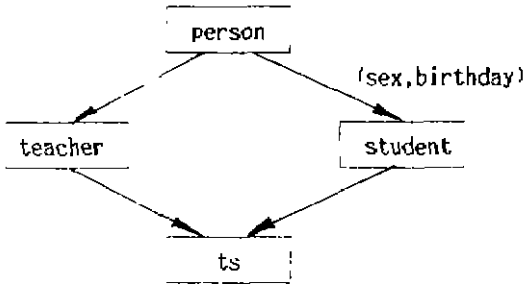

图3

继承性提供了共享相同结构与值的途径。实现嵌套关系的关键是如何表示嵌套模式及其中的值。我们用带 INHERITS 短语的 CREATE 语句来定义嵌套关系的模式。

例1 依次用 CREATE 语句创建图3所示的四个关系。

```
CREATE TABLE person(name char 6,sex char,dept char
    20,birthday date,address char 30,zip int)
                KEY(name)
CREATE TABLE teacher (t_course char 20,classroom
                char 10)
                INHERITS person
CREATE TABLE student(s_course char 20,grade float)
                INHERITS person(sex,birthday)
CREATE TABLE ts(type char)
                INHERITS teacher,student
```

3 实现中的几个关键问题

3.1 嵌套关系的存储结构

通过平关系的继承性来实现嵌套关系，数据库中实际存储的仍是一个个平关系。关系的继承性使得关系间的联系为一有向图(有向树是其特例)，顶点为关系，有向边表示继承。嵌套关系的数据结构在数据字典上实现，嵌套关系的形成通过具体的数据操纵语言来实现。

我们用关系字典和属性字典来存放所有关系的定义信息。当用 CREATE 语句创建关系时，把所有的定义信息存入关系字典和属性字典中。实际上并非是真正构造一个物理的嵌套关系，嵌套关系只相当于在关系的定义层实现的一个虚关系。创建一个关系时，属性字典中先依次存放该关系的各属性(个数由关系字典中的分量 fld-num 确定)；然后再依次存放继承的属性名(仅存储属性名，其它各分量为空)。其个数由关系字典中的分量 inherits-attr-num 确定。若关系字典中的分量 inherits[0]非空且 inherits-attr-num[0]等于零，则表示继承第一个关系的全部属性。所继承的各属性不需填入属性字典中。

3.2 关系间继承性的实现

INHERITS 短语只确定了关系间的继承性质，子关系与其超关系中的元组值通过关键字来实现。用户必须先创建超关系，然后再创建子关系。在创建超关系(如 person)时必须确定关键字属性(如 name)，这样用 INHERITS 短语创建子关系(如 student)时将自动继承其超关系的关键字属性(name)作为子关系的关键字属性。

3.3 多继承处理策略

当一个子关系从多个直接超关系继承其属性时，就可能导致所继承的属性名发生冲突。我们采用如下准则进行处理关系的多继承性：

1)若属性类型相同，则认为命名相同的属性所表达的语义相同，选择第一个继承关系中的属性进行继承。

2)若两个关系的属性表示不同的语义，一定要用不同的属性名，以防进行多继承时丢失信息。例如，关系 teacher 中的课程为教师讲授的课程名，关系 student 中的课程为学生选修的课程名，两个属性的语义不同，所以分别用 t-course 和 s-course 来表示。

3)当子关系的两个直接超关系的关键字相同(名与类型)时，子关系继承第一个超关系的关键字。若所继承的两个直接超关系的关键字不相同时，子关系同时继承这两个超关系的关键字，构成一个复合关键字。

4)理论上，一个子关系可同时继承任意个直接超关系。但为了利于实现，现约定一个子关系最多可继承三个直接超关系。

5)在关系继承图中，对任何一个连通分量必须存在一个无超关系的顶点，相当于嵌套关系树状表示时的根。

6)如果关系 A 有两个直接超关系 B 和 C，则在嵌套关系模式中认为 A 和 B、C 处于同一层上。

3.4 继承性的层次

关系可以多层地继承其超关系的属性，各层关系间的元组值仍通过关键字联系。通常，日常生活中的嵌套表格不会超过三～四层，故约定一个子关系传递继承超关系的层次为四层。任何一个系统的 SELECT 语句所包含的关系个数都有一个上限值，我们实现的 SELECT 语句可同时处理12个关系。所

维普资讯 http://www.cqvip.com

以对任一个子关系，它所继承(包括多继承和传递继承)的超关系不能超过11个。

因为 SELECT 语句实现中采用递归算法，只要机器的空间与速度能满足要求，很容易扩充子关系传递继承的层次和多继承的超关系个数。

4 各种操作的处理

下面讨论如何对建立在继承性基础上的嵌套关系用 SQL 语句进行查询、增、删、更新等操作。如果是嵌套关系，往往需要从当前关系出发沿着继承性的正、反方向进行操作。在对数据库做任何操作时，都要先根据语句中的信息查找关系字典和属性字典。为提高执行效率，首先把关系字典中的各关系根据继承性构造关系继承图的邻接表和逆邻接表。若关系 A 继承关系 B，则从 B 到 A 有一条有向边(与图3保持一致)。

4.1 SELECT 语句

1)若查询的关系无超关系(如图3中的关系 person)，则和传统的 SELECT 语句完全一样。

2)若查询的关系存在超关系，首先根据逆邻接表找出以该关系为始点的各层超关系，然后再从关系字典和属性字典中继承其超关系的指定属性。此时，子关系作为其超关系的一个复合属性，从而构成嵌套关系。为使嵌套关系的属性名更符合用户要求，对 SELECT 语句的 FROM 短语进行扩充：

FROM 关系名(输出属性名)

在此格式中，关系名必须是子关系名。

嵌套关系是一个虚关系，只有在执行 SELECT 语句时才生成真正的嵌套关系。具体的实现就相当于子关系与超关系执行连接操作。

例2 对存储学生所学课程、成绩的关系 student，查询出所有的信息。

```
SELECT * FROM student(Grade_list)
```

在系统内部，实际上做了如下的操作：

```
SELECT person.name,sex,birthday,
        s_course,grade
  FROM person,student
  WHERE person.name=student.name
```

但与传统的连接操作有两点区别：一是查询结果的表头为嵌套表头(见图4)，二是查询结果的属性无重复值。

4.2 UPDATE 语句

1)若更新任一关系的非关键字属性值，则与传统的 UPDATE 语句完全一样。

2)若要更新任一子关系的关键字属性值，认为是非法操作，因为子关系的关键字是从其超关系继承来的。

3)若更新无超关系的关系的关键字属性值，需同时更新其所有子关系的关键字属性值。

例3 把关系 person 中的 name 为"丁一"的元组改为 name 等于"王二"。

```
UPDATE person SET name="王二" WHERE
        name="丁一"
```

执行该语句后，首先从关系邻接表中找出 person 的所有子关系，然后执行如下的传统 UPDATE 语句：

```
UPDATE person SET name="王二" WHERE
        name="丁一"
UPDATE teacher SET name="王二"
        WHERE name="丁一"
UPDATE student SET name="王二"
        WHERE name="丁一"
UPDATE ts SET name="王二" WHERE
        name="丁一"
```

4.3 DELETE 语句

1)若处理的关系不是任何关系的超关系，则与传统的 DELETE 语句相同。

2)若处理的关系是其它关系的超关系，则删除该关系元组时，也应删去其所有子关系中(需考虑传递继承)关键字属性值相同的元组。处理策略与 UPDATE 语句类似。

4.4 INSERT 语句

与传统 INSERT 语句处理完全一样。

4.5 ALTER 语句

1)修改任一关系的非关键字属性，与传统 ALTER 语句处理一样。

2)修改任一子关系的关键字属性，认为是非法操作。

3)修改无超关系的关系的关键字属性，同时将修改其所有子关系的关键字。

4.6 DROP 语句

1)删除无子关系的关系操作与删除平关系的操作一样。

2)删除任一有子关系的关系，被认为是非法操作。若一定要删除该关系，则只有将其所有子关系均用 DROP 语句删除后，才能删除该关系。

结束语 本文提出的用继承性实现嵌套关系的方法已在我们研制的可扩充DBMS-NHEDB中加以

维普资讯 http://www.cqvip.com

文档库 Notes 数据模型 数据管理

计算机科学1997Vol.21№.6

80-84 G276

Lotus Notes 文档库分析

Analysis of Lotus Notes' Document Database

陈海广 李 磊

(中山大学岭南(大学)学院软件所 广州510275)

摘 要 This paper analyzes the groupware -Lotus Notes--from perspective of DBMS. Focus is set on the capability of entity-relationship description and manipulation of both data and relation in Lotus Notes' document data model. A comparison between Lotus Notes' data model and the traditional data models is also presented. Finally, the principle of applying Lotus Notes is proposed based on the analysis above.

关键词 Database management system, Document database, Groupware

近年来,飞速发展的信息科学技术,尤其是Internet/Intranet信息技术,在现代企事业和工商行政机构管理中逐渐扮演越来越重要的角色。由美国Lotus公司推出的Lotus Notes群件是这其中一个特色鲜明,具有代表性的应用软件。它最早系统地引入了工作流概念,以共享式文档库为基础,以电子消息通信为流程实施的基本手段,是目前最完善地实现实际工作流程的应用软件。Lotus Notes在我国正引起越来越广泛的关注,被越来越多的大型企事业单位和行政机关所采用。但是,由于缺乏对Lotus Notes这一软件产品的全面深入了解,可能导致在利用Lotus Notes开发不同的应用系统过程中,出现截然不同的开发效率,产生截然不同的应用效果。本文将论述Lotus Notes文档库特性,对文档库数据模式和数据模型进行分析,对数据模型中描述实体联系的能力和对联系的操作进行横向分析和纵向比较,力图从文档库数据模型角度提出什么情况下适宜利用Lotus Notes进行开发,而什么情况下不适合的一般原则。

Lotus Notes文档库数据模式

Lotus Notes文档库基本上符合数据库的三级模式结构——即用户模式、逻辑模式、存储模式。但它与传统数据库相比,又有很多不同点。下面着重讨论其逻辑模式。

一、Lotus Notes文档库逻辑模式

1. 数据模型

一个Lotus Notes文档库中不仅包含文档数据,而且包含文档结构定义(数据字典)、用户视图、存取控制表、程序代理等,这是目前许多流行数据库

实施。我们认为具有用户界面友好、系统易实现的特点。整个系统用Turbo C在486微机上实现,源程序达50000余行,已通过鉴定。目前,正在进一步研究关系的多继承与传递继承以及嵌套关系的索引机制等。

参考文献

[1]Schek H. J. and Scholl M. H., The Relational Model with Relation-valued Attributes, Information Systems, 11(2), 1986

[2]Dadam P. et al., A DBMS Prototype to Support Extended NF^2 Relations: An Integrated View on Flat Table and Hierarchies, In: ACM SIGMOD, 1986

[3]Linnemann V., Non First Normal Form Relations and Recursive Queries: An SQL-Based Approach, In: 3th Intl. Conf. on Data Engineering, 1987

[4]Stonebrake M. et al., Extending a Database System with Procedures, ACM Trans. on Database Systems, 12(3)1987

[5]Lawrence A. R. and Stonebrake M., The POSTGRES Data Model, In: Proc. of the 13th VLDB Conf., 1987