第 23 卷 第 3 期 华侨大学学报(自然科学版) Vol. 23 No. 3
2002 年 7 月 Journal of Huaqiao University (Natural Science) Jul. 2002

文章编号 1000-5013(2002)03-0317-05

一个面向对象的扩展 Select 语句的设计与实现

陈维斌 陈启泉 林 晓

(华侨大学信息科学与工程学院, 泉州 362011)

摘要 标准 SQL 是一种基于关系数据模型的数据库查询语言, 其中的 select 语句基本不具备复杂类型数据查询能力. 因此, 在设计多媒体数据支撑环境 E-Support 时, 针对其对象描述和查询方面的需求, 提出一个面向对象数据模型. 在此基础上, 给出一个基于面向对象数据模型的扩展 select 语句的文法. 介绍其解释程序的实现技术, 重点讨论复杂类型属性和对象引用属性的处理方法以及如何将查询结果封装成对象.

关键词 面向对象数据模型, 扩展 select 语句, 对象引用

中图分类号 TP 311.132.3 文献标识码 A

SQL 既是交互式的数据库语言, 又广泛地嵌入各种新一代的程序设计语言, 与各种符合 ODBC 标准的数据引擎, 以及处理数据存储的组件(例如 ADO)有着良好的接口. 尽管 SQL 在处理结构化数据方面近乎于完美, 但是在复杂类型数据处理及查询方面的缺陷是显而易见的. 解决问题的途径之一, 是采用面向对象查询语言. 目前, 虽已有一些较成功的 OODBMS 和 OSQL, 但与标准化、实用化的目标相比还有相当的距离. 此外, 由于很多应用是建立在关系数据库系统基础上的, 人们对关系数据库的依赖短期内难以改变. 这些不可忽视的因素使得这一途径并不那么有效. 另一个途径是"扩充关系数据库以支持对象"[1], 也即在关系数据库之上增加一个对象描述与管理层. 该层采用面向对象数据模型建立对象模式, 通过一个与 RDBMS 的接口实现对象模式与关系存储模式的转换. 本文提出一个面向对象数据模型[2,3]. 基于该模型对标准 SQL 的 select 语句进行扩充, 使扩展 select 语句支持复杂类型数据查询和对象引用等多种扩展的数据查询功能. 该语句已嵌入到设计的多媒体数据支撑环境 E-Support 中.

1 扩展 select 语句的文法设计

1.1 E-Support 的面向对象数据模型简介

E-Support 的核心部分是一个数据对象管理器. 它包括一个类描述器和一个与关系数据库的接口. 前者支持表层的数据对象描述, 后者负责表层和底层之间的模式转换, 即对象模式

收稿日期 2002-04-09 作者简介 陈维斌(1957-), 男, 副教授

基金项目 国家高科技应用部门发展基金资助项目; 福建省自然科学基金资助项目

和关系存储模式之间的转换. E-Support 表层的对象模式, 可定义为数据库对应于类族, 数据库表对应于类, 元组则对应于类的实例, 即对象. 相应地, 我们将面向对象数据模型定义为 $C = f(P, M)$. 其中, $P = \{P_i, t_i) \mid t_i \in T\}$ 表示类中的属性集合, T 是系统定义的属性类型集合, t_i 可以是文本、日期、数值、备注、图像、音频、视频、RTF 文本、对象引用等. $M = \{(M_i, P_i) \cup (M_i, D) \mid P_i \in P\}$, 表示类的操作集合, 且操作分为两种. 一种是属性级操作, 是作用于属性的, 例如针对一个视频属性的操作就有装载视频、播放视频、视频另存等. 另一种是类级操作, 是对该类所有的对象都起作用的操作, 例如向类库中添加一个对象、删除对象、浏览对象等. 模型中增加消息机制的目的在于支持对象表示的实现. 与类的方法相似, 类的消息也分为类级消息与属性级消息. 模型中任何一个属性都必须能够表示为相应对象中的一个变量和一对消息, 变量用来保存属性的值. 消息中的一个与属性的存取有关, 另外一个则与属性的更新有关. 我们可以读取、显示或更新这个属性值.

1.2 扩展 select 语句的文法

扩展 select 语句, 沿用了标准 select 语句的语法结构. 该语句文本为:

Select PropertyList (PropertyName {PropertyType} (PropertyType in Set{图像, 视频, 声音, RTF 文本}) ‖ (PropertyName (PropertyType not in {图像, 视频, 声音, RTF 文本) ‖ PropertyName1 {对象引用}. PropertyName2)

From ClassName

Where Condition;

Order by 〈OID| PropertyName〉

该文法与标准 select 语句文法的不同之处有 4 点. (1) from 子句中出现的不是关系模式的表名而是对象模式的类名. (2) 属性列表中的各个属性是数据对象的属性, 而不是数据库表的属性. 考虑到类中含有多媒体类型的属性, 而这些属性在底层的关系存储模式中一律以二进制字节流的形式存储. 为了使解释系统能够准确地区分属性的类型, 因此必须附加显式的类型标识符 PropertyName{ PropertyType}(PropertyType in Set{ 图像, 视频, 声音, RTF 文本}). (3) 在类的属性中还包含有类型为对象引用的属性. 该属性的值是被引用对象的 OID, 可以是字符串也可以是整数. 为了避免与一般的文本类型属性混淆, 当类型为对象引用的属性出现在 select 子句时, 也必须附加显式的对象引用类型标记: PropertyName1{对象引用}. PropertyName2. (4) Where 子句中的 cndition 可以是含有关键字运算的条件表达式, 支持基于关键字的查询. 它也可以是指定的对象标识符, 支持按 OID 查询. 我们设计了一个扩展 select 语句的可视化描述器. 当用它来构造扩展 select 语句时, 多媒体属性的显式类型标记和对象引用类型标记是自动附加的.

1.3 扩展 select 语句的应用实例

示例 1. 从产品类中查询产品 ID 为 B9910 的产品对象, 并显示产品名称、数量、单价、产品简介(RTF 文本)、产品图片(图像). 对应的扩展 select 语句为

select ProductName, Price, quantity, ProPicture {图像}, ProIntro {RTF 文本}

from Product where ProductNo = 'B9904' (1)

示例 2. 从课程类中查询课程号为 C002 的课程, 并显示课程名、课程简介和任课老师的有关情况(从教师类通过对象引用获得). 对应的扩展 select 语句为

select courseName, courseIntro {RTF 文本}, teacher {对象引用}. teacherResume
from course where 课程号 = ´C002´　　(2)

2　扩展 select 语句解释程序的实现方法

2.1　解释程度的处理流程

解释程序除了对语句进行常规的语法分析之外, 还须将其翻译成标准 select 语句, 提交给底层的 RDBMS 执行. 同时, 将关系查询处理所获取的关系型数据集按对象模式重新封装成若干数据对象. 如果查询涉及到对象引用, 那么就可能有来自不同类的多个对象被封装. 解释程序的处理流程, 如图 1 所示. 将扩展 select 语句翻译成标准 select 语句, 关键是要解决多媒体类型标识和对象引用标识等语法成分的识别、分离和保存, 以及如何将关系型查询结果封装成数据对象等问题. 下面结合节 1.2 所给的两条例句, 讨论其处理方法.

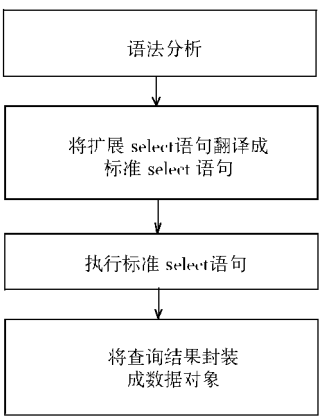

图 1　解释程序的处理流程

2.2　复杂类型属性的识别与处理方法

我们从语法分析阶段建立的中间语言中获取 select 子句的各个属性名单词, 通过判断属性名是否后随 "{ 多媒体类型标识符}", 可以识别出该属性的类型. 对于识别和分离出来的每一个属性, 必须保存其基本要素, 即名称、类型标识符和值. 名称对应于属性名单词; 属性值必须等到执行完标准 select 语句才能写入(参见 2.4 节). 类型标识符的获得分为 3 种情况. (1) 对于各种多媒体类型, 直接保存其类型标识符. (2) 对于各种文本类型, 其类型标识符一律设置成 "字符", 这是因为所有的文本类型最终都可以转换成字符串. 当再现对象时, 都可以通过一个文本框来显示. (3) 对于对象引用类型的处理, 参见 2.3 节. 此外, 还需建立一个属性要素表, 该表用来保存这些要素. 表类型定义为

```
Struct PropertyFactor{
        Char PropertyName[31]          // 属性名
        Char PropertyType[12]          // 属性名
        Char * PropertyValue           // 属性值
        PropertyFactor * link1;
        PropertyFactor * link2;
        } * pft
```

因为解释程序无法预先确定需要保存的属性要素个数, 所以用链表来实现属性要素表. 解释程序处理语句(1) 时所建立的属性结构表, 如表 1 所示. 将一条含有多媒体属性的扩展 select 语句翻译成标准 select 语句的处理流程(解释程序处理流程的步骤 2) 图, 如图 2 所示.

2.3　对象引用的处理方法

分析对象引用的语法 PropertyName {对象引用}. 由 PropertyName2(参见语句(2)) 可以

看出,PropertyName1实际上是被引用的类名,对应的值是该类的某个对象的OID,PropertyName2则给出被引用类的指定属性.因此,当我们保存类型为对象引用的属性之要素时,必须同时保存被引用对象的属性要素.虽然二者结构相同,可以保存在同一张表中(只需在属性要素表中增加一行).但为了能够单独封装被引用对象,我们采用增设附加表的办法来保存Property2的属性要素.附加表的结构与属性要素表的结构完全相同.在建立附加表后,还需将指向附加表的指针置入属性要素表的link2中,以建立两个表的连接.对扩展select语句的每一个对象引用属性,还要生成一条对被引用类所对应的数据库表进行查询的标准Select词句,以便从底层的RDBMS中获得该类的查询结果.这一类语句的where子句包含一个关系运算式,即〈OID〉=〈值〉.该式必须在所有的OID的值都确定之后,才最后形成.综上所述,如果属性类型是对象引用,则图2的步骤4改为图3所示的处理流程.

表1 属性要素

PropertyName	PropertyType	PropertyValue
PropertyName	字符	-
Quantity	字符	-
Price	字符	-
ProIntro	RTF文本	-
ProPicture	图像	-

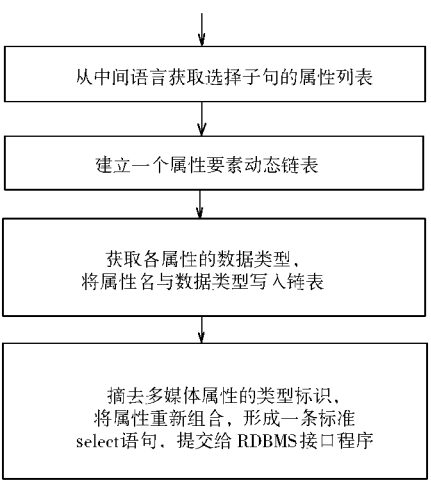

图2 多媒体属性的处理流程

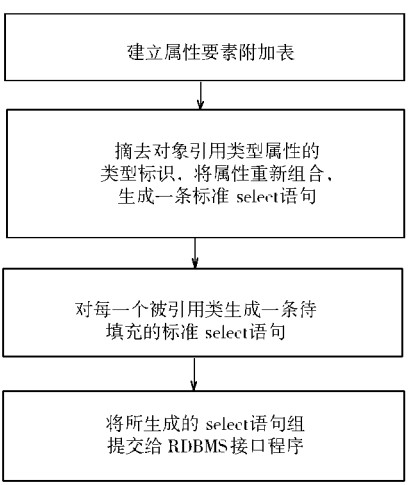

图3 对象引用的处理流程

2.4 将查询结果封装成数据对象

当完成标准select语句的组装后,将其连同属性要素表一并提交给E-Support的RDBMS接口程序.由接口程序与底层RDBMS建立连接,让RDBMS执行所提交的标准select语句,获得一个关系型查询结果数据集,并完成属性要素表的最后设置.即写入各个属性的值,如果是对象引用,写入的是被引用对象的OID.以此作为数据对象的原型,构造数据对象.对含有对象引用的查询,则要进行多个标准select语句的最后组装和执行,相应地获取多个关系型查询结果数据集.封装数据对象.确切地说是借助于属性要素表和保存在E-Support的数据对象管理器中的相应的类描述来创建查询对象的实例.需要设置对象的属性值和封装属于该对象的各个操作的方法代码.下面谈两种处理方法.(1) 设置对象属性.就属性而言,因为元组的属性值和对象的属性值是一一对应的,所以我们可以逐个取出保存在属性要素表中的属性值(PropertyValue),赋予被创建的对象的相应属性.如果是多媒体类型,则不写入,直到用户浏

览该对象, 需要显示多媒体属性的内容时, 才调用相应的操作从数据集中返回该属性的二进制字节流, 并且根据 PropertyType 的值调用相应的媒体浏览工具予以再现 . (2) 封装操作的方法代码. 由于我们将类转换成一个关系时, 对一个属性级操作至少设置两个属性. 其中, 一个用来作为操作句柄, 另一个则用来存放操作所对应的方法组件信息. 因此, 在封装数据对象的属性级方法代码时, 根据属于该对象的每个操作的句柄和组件中的方法名即可构造一个函数调用, 而该函数的动态绑定和执行则交由类浏览器完成 . 例如, 当用户在浏览器中双击当前对象的一个视频属性时, 浏览器就根据已构造的函数调用, 调用装载视频函数读入视频流, 并调用相应的播放器进行视频播放 .

3　结束语

查询技术是数据库的核心技术之一. 应用面向对象的方法来改进传统的结构化查询技术, 使之能处理复杂类型数据的查询, 具有十分重要的意义, 而且技术上也可行 . 我们已用 VC 6. 0 实现扩展 select 语句的解释程序和一个扩展 select 语句的可视化描述器, 连同嵌入在 E-Support 中的类定义和浏览工具以及一些数据操纵组件, 已初具 OQL 的基本特征和基本功能. 使用该语句对一个具有上千条记录的产品数据库(SQL Server) 进行查询测试, 从底层的关系查询到顶层的对象模式转换(利用查询结果创建主数据对象和多个被引用类的数据对象), 其响应时间在秒级以内 . 今后, 我们的工作包括改进对象引用的语法和处理程序, 实现多重引用. 应用一些比较成功的基于内容的查询技术, 以增加该语句的基于内容的查询能力. 设计类描述语句和类操纵语句的文本, 并完成标准化工作, 最终形成一个独立于 E-Support 的 OQL.

参　考　文　献

1　冯玉琳. 对象技术导论[M]. 北京: 科学出版社, 1998. 98～106
2　田增平, 党华锐, 周傲英等, 多媒体对象查询语言及其查询处理[J]. 软件学报, 1999, (10) : 694～700
3　Paul J 著. 数据库技术大全[M]. 北京: 电子工业出版社, 1999. 173～185

Design and Implementation of Object-Oriented Extended Select Statement

Chan Weibin　Chen Qiquan　Lin Xiao
(College of Info. Sci. & Eng., Huaqiao Univ., 362011, Quanzhou)

Abstract　The standsrd SQL is an query language of data base based on relational data model, in which the select statement is not provided with the capability of quering data of complicated type in the main. In designing E-support, namely, the supporting environment of multi-media data, an object-oriented data model is advanced to meet the need of object description and query; and then, the grammar of extended select statement based on object-oriented data model is given; and finally, the technology for implementing its interpreter is presented. The discussion lays emphasis on the attribute of complicated type and the attribute of object reference as well as on how to pack the result of query into object.

Keywords　object-oriented data model, extended select statement, object reference.