

# 面向对象的关系数据库设计

陈文宇\*

(电子科技大学计算机科学与工程学院 成都 610054)

**【摘要】**针对面向对象技术和关系数据库的特点,将两者相结合,建立了对象关系数据库管理系统,既支持已经被广泛使用的SQL,不仅具有良好的通用性,又具有面向对象特性,能支持复杂对象和复杂对象的复杂行为,是对象技术和传统关系数据库技术的最佳融合。该技术同时拥有关系型数据库的强大功能和可靠性以及对象的灵活性和模型化功能。文中还提出了对象关系数据库的设计方法和对象映射成关系数据库的方式,其方式简便,实用性强。

**关键词** 面向对象; 关系数据库; 映射; 对象设计

**中图分类号** TP311.132.4

## 1 面向对象数据库技术

数据库技术已成为现代信息技术的重要组成部分,是现代计算机信息系统和计算机应用系统的基础和核心。传统的经典数据库有网状数据库、层次数据库和关系数据库。针对不同领域的应用提出了许多新的数据管理需求,其中的一个重要特点是将面向对象的思想、方法和技术引入数据库,即采用面向对象数据模型的面向对象数据库,这是新一代数据库的主流。

在面向对象技术和数据库技术相结合的过程中,基本上沿着以下两种途径发展:1) 建立纯粹的面向对象数据库管理系统(OODBMS),以面向对象语言为基础,增加数据库的功能,支持持久对象和实现数据共享。利用类的设施来描述复杂对象,利用类中封装的方法来模拟对象的复杂行为,利用继承性来实现对象的结构和方法的重用,但这种纯粹的面向对象数据库系统并不支持SQL语言;2) 从传统的关系数据库加以扩展,增加面向对象的特性,把面向对象技术与关系数据库相结合,建立对象关系数据库管理系统(ORDBMS),这种方式既不是纯粹的面向对象的数据库也不是纯粹的关系数据库,它代表两者的结合,系统既支持已经被广泛使用的SQL,不仅具有良好的通用性,又具有面向对象特性,支持复杂对象和复杂对象的复杂行为,是对象技术和传统关系数据库技术的最佳融合。面向对象的数据库对用户当前的关系数据库是向下兼容的,能够提供一种支持对象而又不必抛弃现有应用系统的机制。面向对象与关系表达的结合在面向对象的数据库中语义清晰,比单独的关系或面向对象的表示更有力,使设计紧凑有效的数据库更为容易<sup>[1]</sup>。

允许用户以处理关系数据的同样方式来处理对象数据,而且为处理对象数据专门设计了新功能,也可用来处理关系数据。面向对象数据库的产生主要是为了解决“阻抗失配”,它强调高级程序设计语言与数据库的无缝连接。由于实现了无缝连接,使面向对象数据库能够支持非常复杂的数据模型,数据的无缝操作体现了对象技术渗透到数据库服务器之中,而不是在现有的关系数据库上附加一个采用对象技术的外壳,或者在关系数据库和客户端应用软件之间提供一个对象服务器网关。

对象关系数据库管理系统具有永久性能力,即可以使用安全、一致的方式存储和检索对象的能力,这正是对象关系数据库可以发展的地方。对象关系数据库管理系统能够象处理关系型数据那样

2001年8月27日收稿

\* 男 32岁 硕士 讲师

存储和检索对象数据。将对象和关系型模型合二为一,可以同时拥有关系型数据库的强大功能和可靠性以及对象的灵活性和模型化功能。

## 2 对象关系数据库设计的重要性

关系数据库方法是在一个最低级的层次上,用一系列的表列和行处理数据。面向对象的方法是在更高的层次上处理数据,处理包括数据的对象。在面向对象的数据库中处理客户时,是处理一个称为“客户”的对象。当处理一个序时,引用一个称为序的对象。因为对象数据库理解对象客户和所有关系,能容易地处理对象客户和所有与它一起工作的对象。

在关系模型中,序实际是许多不同的表的组合,并装有支持和维护一个序所需属性的交表。对象模型中,数据库对相互间的关系具有智能,在关系模型中则没有。当对关系模型作出更改时,通常翻译成一系列的新表,若模型继续使用则必须建立,且由一个数据库设计者来重新设计。

一般数据库设计方法有属性主导型和实体主导型两种。属性主导型从归纳数据库应用的属性出发,在归并属性集合(实体)时维持属性间的函数依赖关系。实体主导型则先从寻找对数据库应用有意义的实体入手,然后通过定义属性来定义实体。面向对象的数据库设计是从对象模型出发的,属于实体主导型设计<sup>[2]</sup>。

一般数据库应用系统都遵循以下相关开发步骤:

- 1) 设计应用系统结构;
- 2) 选择便于将应用程序与DBMS结合的DBMS体系结构,如RDBMS;
- 3) 根据应用程序使用的环境平台,选择适宜的DBMS和开发工具;
- 4) 设计数据库,编写定义数据库模式的SQL程序;
- 5) 编写确保数据正确录入数据库的用户接口应用程序;
- 6) 录入数据库数据;
- 7) 运行各种与数据库相关的应用程序,以确认和修正数据库的内容。

## 3 应用对象模型与RDBMS模型的映射

数据库设计(模式)是否支持应用系统的对象模型,这是判断面向对象数据库系统的基本出发点。由于应用系统设计在前,数据库设计随后,所以应用系统对象模型向数据库模式的映射是面向对象数据库设计的关键。

### 3.1 3层数据库模式面向对象模型的扩展

一般数据库设计参照ANSI/SPARC关于数据库模式的3层标准结构提案,最接近物理数据库的内部模式由DBMS提供的SQL来描述。概念模式可以由若干个内部模式聚集而成,是由数据库用户规范的一些表的集合。一般的概念模式是数据库物理模式作用域的边界,能实现数据库的物理意义、特定DBMS的特殊操作对外部应用程序的信息隐蔽。外部模式是从特定用户应用角度看待的数据库模式,从不同的应用出发对同一概念模式可以给出多种不同的外部模式。当外部应用系统以对象模型进行抽象时,从各个应用出发抽象出的对象模型可以映射到外部模型上,称为外部对象模型。但是,外部模型只是概念模式的子集,所以面向对象的数据库设计核心在于系统对象模型(概念对象模型)向数据库概念模型的映射。

### 3.2 对象模型向数据库表的映射规则

在对象范例和关系范例之间“阻抗不匹配”。对象范例基于软件工程的一些原理,例如耦合、聚合和封装,而关系范例则基于数学原理,特别是集合论的原理。两种不同的理论基础导致各自不同的优缺点。而且,对象范例侧重于从包含数据和行为的对象中构建应用程序,而关系范例则主要

针对数据的存储。当为访问而寻找一种合适的方法时,“阻抗不匹配”就成了主要矛盾:使用对象范例,是通过它们的关系来访问对象,而使用关系范例,则通过复制数据来联接表中的行,这种基本的差异导致两种范例的结合并不理想。

由于RDBMS以二维表为基本管理单元,所以对象模型最终是由二维表及表间关系来描述,即对象模型向数据库概念模型的映射就是向数据库表的变换过程<sup>[3]</sup>。

### 3.3 对象映射成关系数据库

对象映射成关系数据库需要将属性映射成列,即类属性将映射成关系数据库中的零或几列。另外,将类映射成表和在关系数据库中实现继承。

整个类层次结构使用一个数据实体:将一个完整类层次结构映射成一个数据实体,而层次结构中所有类的属性都存储在实体中。缺点是每次在类层次结构的任何地方添加一个新属性时都必须将一个新属性添加到表中。这增加了类层次结构中的耦合,如果在添加一个属性时有错误,除获得新属性类的子类外,还可能影响到层次结构中的所有类,可能浪费数据库中的许多空间。

每个具体类使用一个数据实体:每个数据实体既包含属性又包含所表示的类继承的属性。但有不足之处,如修改类时,必须修改表和所有子类的表。

每个类使用一个数据实体:为每个类创建一张表,其属性是OID和特定于该类的属性,能够最好地适应面向对象的概念,很好地支持多态性,对于对象可能有的每个角色,只需在相应的表中保存记录。修改超类和添加新的子类也非常容易,因为只需要修改或添加一张表。缺点是数据库中有大量的表,读取和写入数据的时间比较长。

对象映射成数据库不仅将对象映射到数据库中,还必须将对象之间的关系进行映射,这样才能进行恢复。对象之间有继承、关联、聚合和组成四种关系。从数据库的角度看,关联和聚合/组合关系之间的唯一不同是对象相互之间的绑定程度。对于聚合和组合,在数据库中对整体所做的操作通常需要同时对部分进行操作,而关联则不同。从数据库的观点看,聚合/组合和关联是不同的,在聚合情况下,在整体中读取时,通常希望在部分中读取,而在关联情况下,需要执行什么操作则不明显。在将对象保存到数据库中或从数据库中删除对象也存在相同的情况。

关系数据库中的关系是通过使用外键来维护。外键是在一张表中出现的一个或多个数据属性;可以是另一张表的键的一部分,或者是另一张表的键。外键将一张表中的一行与另一张表中的一行相关起来,要实现一对一和一对多的关系,只需将一张表的键包括在另一张表中。要实现多对多关系,需要关联表的概念,它是一种数据实体,唯一目标是在关系数据库中维护两个或多个表之间的关联<sup>[4, 5]</sup>,如图1所示。

有关变换规则归纳如下<sup>[6]</sup>:

1) 一个对象类可以映射为一个以上的库表,当类间有一对多的关系时,一个表也可以对应多个类;

2) 关系(一对一、一对多、多对多以及三项关系)的映射可能有多种情况,但一般映射为一个表,也可以在对象类表间定义相应的外键;

3) 单一继承的泛化关系可以对超类、子类分别映射表,可以不定义父类表而让子类表拥有父类属性,也可以不定义子类表而让父类表拥有全部子类属性;

4) 对多重继承的超类和子类分别映射表,对多次多重继承的泛化关系也映射一个表;

5) 对映射后的库表进行冗余控制调整,使其达到合理的关系范式。

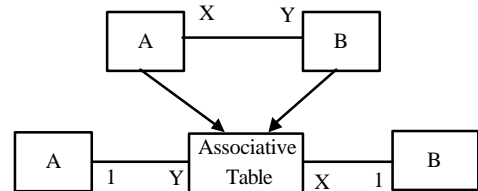


图 1 关联表简介

## 4 面向对象关系数据库设计优势

数据库设计的面向对象特征奠定了整个系统的面向对象性,使面向对象方法在程序开发阶段极具优势,其效果归纳如下<sup>[7]</sup>:

1) 数据库结构清晰,便于实现 OOP。

由于实现了应用模块对象对数据库对象的完全映射,数据库逻辑模型可以自然且直接地模拟现实世界的实体关系。用户所处的物理世界、系统开发者所抽象的系统外部功能,与支持系统功能的内部数据库(数据结构)一一对应,所以用户、开发者和数据库维护人员可以用一致的语言进行沟通。特别是对多数不了解业务的程序开发人员,这种将应用对象与相应的数据对象封装在对象统一体中的设计方法,大大减轻了程序实现的难度,只要知道加工的数据及所需的操作即可,而且应用程序大多相似,可以多处继承由设计人员抽象出来、预先开发好的各种物理级超类。

2) 数据库对象具有独立性,便于维护。

除了数据库表对象与应用模块对象一一对应外,在逻辑对象模型中没有设计多重继承的泛化关系,所以得到的数据库结构基本上是由父表类和子表类构成的树型层次结构,表类间很少有继承以外的复杂关系,是一个符合局部化原则的结构,从而使数据库表数据破坏的影响控制在局部范围且便于修复,给系统开通后的数据库日常维护工作带来便利。

3) 需求变更时程序与数据库重用率高,修改少。

在映射应用对象时,除关系映射规范化后可能出现一对多的表映射外,大多数应用对象与表对象是一一对应的。可以把规范化处理后由一个应用对象映射出来的多个表看成一个数据库对象,因此,当部分应用需求变更时,系统修改可以不涉及需求不变更的部分,而变更部分的修改可以基本上只限于追加或删除程序模块或追加新库表,基本上不必修改原有程序代码或原有库表定义,从而大大减少了工作量,降低了工作难度。此外,面向对象数据库还吸收了面向对象程序设计语言的思想,如支持类、方法、继承等概念。

## 5 结束语

综上所述,数据库系统发展的趋势是面向对象数据库和关系数据库不断融合,而对象关系数据库由于继承了上述两者的优点,已经成为目前数据库发展的主流。

### 参 考 文 献

- 1 Abraham Silberschatz著. 数据库系统概念. 扬冬青译. 北京: 机械工业出版社, 1999
- 2 Cardenas. A. F. Data Base Management System. 1998
- 3 Gio. Wiederhold. Data Base Design. 1999
- 4 Scott W. Ambler. The Object Primer 2nd Edition. 2000
- 5 Cameron Hughes Thomas Hamilton, Tracey Hughes著. 面向对象程序设计. 尤晓东译. 北京: 电子工业出版社, 1999
- 6 苏 森, 唐雪飞, 刘锦德. 面向对象的互操作技术. 电子科技大学学报, 1998, 27(1): 90-95
- 7 丁志强. 基于对象数据流图的可复用方案. 电子科技大学学报, 1999, 28(3): 306-311

(下转第75页)

## 参 考 文 献

- 1 Carlton R, Davis. IPSec: VPN的安全实施. 北京: 清华大学出版社, 2002
- 2 Naganand Doraswamy, Dan Harkins. IPSEC新一代因特网安全标准. 北京: 机械工业出版社, 2000
- 3 王茂忠, 方志聪, 周明天. 基于IPv6的安全机制. 电脑技术信息, 2000, (11): 30-31
- 4 秦忠林, 黄本雄. IPSEC研究及实现. 计算机应用, 2001, 21(4): 25-27

## The Research About Secure Protocol of Ipv6

Shen Li

(Sichuan Normal University Institute of Computer Science Chengdu 610066)

**Abstract** The research about secure protocol of IPv6 discusses the IPv6's secure protocol in aspects of IPSec's secure systems, the working principles, the working patterns, the methods of solving the secure problem and the application of it in VPN. IPSec's AH and ESP protocols provide authentication and encoding for IP layer's transmission. This character can solve the problem that today's secure system which can not protect the data in the IP layer but can protect the data in the application layer.

**Key words** security; protocol; IPSec; authentication header; encapsulating security payload

---

(上接第56页)

## Design Oriented-object Relative Data Base

Chen Wenyu

(College of Computer Science and Engineering, UEST of China Chengdu 610054)

**Abstract** Based on the character of oriented-object and relative data base management system, we can join the oriented-object technology to relative data base .The system supports the SQL, complex object and the complex action of complex objects. It is the rejoin of oriented-object technology and trade relative data base technology. It has the strong function of relative data base and the model function of object. This paper supply the method of the relative data base design and map object to relative data base. The method is easy and useful.

**Key words** oriented-object; relative data base; map; object oriented