



## [12] 发明专利说明书

专利号 ZL 200610144061.0

[45] 授权公告日 2009 年 10 月 7 日

[11] 授权公告号 CN 100547591C

[22] 申请日 2006.11.24

[21] 申请号 200610144061.0

[73] 专利权人 北京神舟航天软件技术有限公司

地址 100036 北京市海淀区阜成路裕惠大厦 17 层

[72] 发明人 黄文左 栾森

[56] 参考文献

CN1647080A 2005.7.27

US5263167A 1993.11.16

US6587856B1 2003.7.1

面向对象数据库的查询转换成关系数据库的查询. 王学荣, 曾晓勤. 计算机工程与应用, 第 20 期. 2002

审查员 李楠

[74] 专利代理机构 北京北新智诚知识产权代理有限公司  
代理人 张卫华

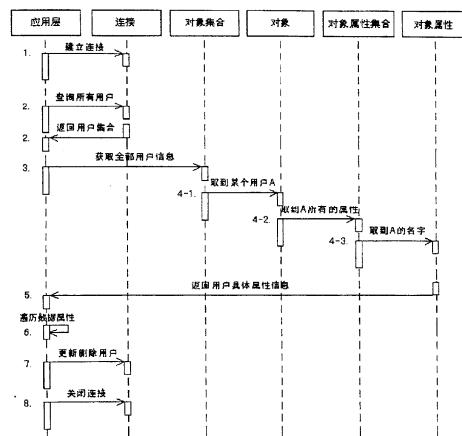
权利要求书 1 页 说明书 4 页 附图 3 页

[54] 发明名称

应用层调用基于关系数据库的数据库抽象层的方法

[57] 摘要

本发明提供了一种应用层调用基于关系数据库的数据库抽象层的方法，包括步骤：应用层与对象抽象组件进行连接；应用层对所有用户进行查询，并将结果全部返回应用层；应用层在对象抽象组件中获取全部用户的信息；在对象抽象组件中，逐一对每个用户获取该用户的属性信息；对象抽象组件返回用户的具体属性信息；应用层对查询返回的结果遍历；根据用户指令，应用层对用户信息进行更新、删除；应用层关闭与对象抽象组件的连接。数据库抽象层将底层关系数据库完全屏蔽，用户可不必考虑底层各关系数据库间存在的差异，而直接以面向对象的方式对所有应用逻辑对象进行建模与操作。



1、一种应用层调用基于关系数据库的数据库抽象层的方法，其特征在于它包括如下步骤：

步骤 1：应用层与数据库抽象层中的对象抽象组件进行连接；

步骤 2：通过与对象抽象组件的连接，应用层对一种对象进行查询，并将该种对象的信息全部返回应用层；

步骤 3：应用层在数据库抽象层的对象抽象组件中获取该种对象的全部信息；

步骤 4：逐一针对获取的每个对象，在对象抽象组件中获取该对象的属性信息；

步骤 5：对象抽象组件返回该种对象的全部具体属性信息；

步骤 6：应用层对查询返回的对象属性信息进行遍历；

步骤 7：应用层根据对象的指令设定，对该种对象信息进行更新、删除操作；

步骤 8：应用层关闭与对象抽象组件的连接；

上述的对象是指应用系统中模型化的对象。

2、如权利要求 1 所述的应用层调用基于关系数据库的数据库抽象层的方法，其特征在于：

步骤 4 包括：

步骤 4-1：在对象抽象组件中获取与该对象相关的对象信息；

步骤 4-2：根据该对象的对象信息，获取该对象所有的对象属性信息；

步骤 4-3：获取该对象的关键属性信息。

## 应用层调用基于关系数据库的数据库抽象层的方法

### 技术领域

本发明涉及关系数据库间的数据集成，特别涉及基于关系数据库的数据库抽象层的调用方法。

### 背景技术

目前的数据库管理系统可分为两种：面向对象的数据库管理系统和关系数据库管理系统。数据库管理系统用于将数据有效地存储于计算机内。

面向对象的数据库管理系统用于存储利用对象模型进行描述的数据。面向对象的数据库管理系统已提出很多年，但由于其的复杂性，人们对它的认识不一致，且开发过程中会遇到许多技术难题，所以目前市场上还没有一个通用的、成熟的产品出现，且近几年内也不会有太大的发展。

关系数据库管理系统是目前主流的数据库产品。关系数据库几乎统治着所有的操作系统平台上的数据管理领域。许多通用数据库操作标准及相关产品（如 ODBC、ADO 等）的出现使得关系数据库具有良好的通用性和互操作性。随着关系数据库应用领域的不断扩充，关系数据库产品也在不断地增强。

目前，市场上有许多不同类型的数据库在被广泛应用，如主流数据库 ORACLE、SQLSERVER、DB2 等，还有开源数据库，如 MYSQL 等。这些数据库产品除了遵循共同的基础标准，如 SQL92、SQL99 等，各自也在标准之上进行扩展。例如，在数据类型方面，SQLSERVER 用 BIGINT 来支持长整型数据的存储，而 ORACLE 用 NUMBER 来支持长整型数据的存储。在语言方面，SQLSERVER 使用 TSQL 语法，而 ORACLE 使用 PLSQL 语法。在功能方面，ORACLE 有自己的语法来支持递归数据获取，而 SQLSERVER 不直接支持。

PDM 系统的重要任务是帮助产品开发人员更加有效地管理大量的产品数据，而产品数据的一个非常突出的特点就是数据项之间的关系比较复杂，如继承关系、装配关系等。这种复杂的关系用常规数据模型（主要指关系数据模型）很难描述清楚。但由于目前许多主流关系数据库产品，如 Oracle、DB2 等，通过在关系内核的基础上引入对复杂对象模型的有限支持，使得关系数据库能够打入 CAD 等新应用领域，所以许多流行的 PDM 产品的底层数据管理模块几乎都基于关系数据库系统（如 MetaPhase、PM、SmarTeam 等）。

但是，关系数据模型具有一定的局限性，如逻辑对象间的复杂关系难以直接用关系数据模型表达得形象、直观。且由于应用逻辑的任意性所导致的关系复杂性，很难保证数据的一致性和完整性。所以，面对不同的客户，一套管理信息系统如何

支持不同的数据库是目前需要解决的问题。

### 发明内容

本发明的目的在于提供一种应用层调用基于关系数据库的数据库抽象层的方法。数据库抽象层是应用层进行数据库操作的唯一通道，该调用方法屏蔽了底层各关系数据库，使操作用户可直接对底层各关系数据库进行操作。

为达到上述目的，本发明采用以下技术方案：

一种应用层调用基于关系数据库的数据库抽象层的方法，其特征在于它包括如下步骤：

步骤 1：应用层与数据库抽象层中的对象抽象组件进行连接；

步骤 2：通过与对象抽象组件的连接，应用层对一种对象进行查询，并将该种对象的信息全部返回应用层；

步骤 3：应用层在数据库抽象层的对象抽象组件中获取该种对象的全部信息；

步骤 4：逐一针对获取的每个对象，在对象抽象组件中获取该对象的属性信息；

步骤 5：对象抽象组件返回该种对象的全部具体属性信息；

步骤 6：应用层对查询返回的对象属性信息进行遍历；

步骤 7：应用层根据对象的指令设定，对该种对象信息进行更新、删除操作；

步骤 8：应用层关闭与对象抽象组件的连接。

其中，步骤 4 又包括：

步骤 4-1：在对象抽象组件中获取与该对象相关的对象信息；

步骤 4-2：根据该对象的对象信息，获取该对象所有的对象属性信息；

步骤 4-3：获取该对象的关键属性信息。

在本发明的方法中，所述的对象是指应用系统中模型化的对象，如用户、部门、角色、文档、邮件等。

本发明具有以下优点：

1、通过调用数据库抽象层，应用层可直接以面向对象的方式对所有应用逻辑对象进行建模、操作，而不必考虑对象所依赖的数据库表以及表之间的关系维护，避免了复杂的数据库操作，使用户以所认知的面向对象语义对数据进行操作。

2、数据库抽象层将底层关系数据库完全屏蔽，应用层不必考虑底层各关系数据库间存在的差异，具有自适应关系数据库的作用。

### 附图说明

图 1 是数据库抽象层中的一对象抽象组件的组成示意图。

图 2 是调用数据库抽象层的流程示意图。

图 3 是实施本发明的一实施例示意图。

## 具体实施方式

数据库抽象层由多个对象抽象组件（Object Abstract Component，缩写为 OAC）组成，每一对象抽象组件向外提供相同的服务。参见图 1，图中所示为本发明所涉及的数据库抽象层中的一对象抽象组件的组成示意图。根据底层关系数据库的数量，数据库抽象层包括若干对象抽象组件，每一对象抽象组件对应于一底层关系数据库。对象抽象组件主要由连接模块、对象集合模块、对象模块、对象属性模块、对象属性集合模块、类模块、类集合模块、类属性模块和类属性集合模块组成。

连接模块的功能为实现应用层与数据库的连接，连接时不用区分数据库的类型。对象模块以数据库表中的每行数据为单位进行获取。对象集合模块将对象模块获取的各行数据进行集合。对象属性模块对数据库表中的每列对象属性进行获取，列对象属性如列名、数据类型、列值等。对象属性集合模块是将所有对象属性进行集合。这些模块用于应用层调用数据库抽象层时使用。

类模块主要是表示数据库表，而类集合模块是将数据库中的多个表进行集合。类属性模块用来表示数据库表的列在应用层的含义。而类属性集合模块是将数据库表的多个列的含义进行集合。这些模块在操作用户进行建模时使用。

应用系统中建立有许多模型化的对象，如用户、部门、角色、文档、邮件等。图 2 所示为应用层调用数据库抽象层来查询、设定用户情况的方法流程，具体步骤如下：

步骤 1：应用层与数据库抽象层中的对象抽象组件进行连接；

步骤 2：通过与对象抽象组件的连接，应用层对一种对象进行查询，并将该种对象的信息全部返回应用层；

步骤 3：应用层在数据库抽象层的对象抽象组件中获取该种对象的全部信息；

步骤 4：逐一针对获取的每个对象，在对象抽象组件中获取该对象的属性信息；

步骤 5：对象抽象组件返回该种对象的全部具体属性信息；

步骤 6：应用层对查询返回的对象属性信息进行遍历；

步骤 7：应用层根据对象的指令设定，对该种对象信息进行更新、删除操作；

步骤 8：应用层关闭与对象抽象组件的连接。

其中，步骤 4 又包括：

步骤 4-1：在对象抽象组件中获取与该对象相关的对象信息；

步骤 4-2：根据该对象的对象信息，获取该对象所有的对象属性信息；

步骤 4-3：获取该对象的关键属性信息。

对于应用系统中的其它对象，如部门、角色、文档、邮件等，应用层调用数据库抽象层来查询或设定这些对象的流程与图 2 所示的流程同理。

如图 3 所示为实施本发明所述方法的一实施例示意图，该实施例已在 AVIDM 专业版 3.0 版本中得到实现与应用。

如图，应用层包括一数据库抽象层和 PDM 系统。PDM 系统包括各种应用组件，如用户管理组件、权限管理组件等。数据库抽象层中包含三个对象抽象组件，每一个对象抽象组件分别对应于底层的神舟 OSCAR 数据库、SQL Server 数据库和 Oracle 数据库。虽然每种类型的数据库具有各自的连接方式，如神舟 OSCAR 数据库使用 OSCAR ODBC 连接方式进行连接，SQL Server 数据库和 Oracle 数据库使用 OLEDB 连接方式进行连接，但在该实施例中，各对象抽象组件与底层数据库间通过共同的 ADO 接口进行访问。

通过调用数据库抽象层中的对象抽象组件，PDM 系统可直接通过对象抽象组件进行数据库访问，可直接使用户以所认知的面向对象语义对所有应用逻辑对象进行建模与操作，而不必考虑对象所依赖的数据库表以及表之间的关系维护，避免了复杂的数据库操作，增强了 PDM 系统的适应性和可扩展性。

数据库抽象层对 PDM 系统提供的支持主要包括：

- 1) 实现对象、类的语义（包括继承、多态、对象 ID 等）以及对象间三种关系（组成、关联、引用）；
- 2) 支持复杂数据类型，如一维数组、二维表；
- 3) 丰富的对象查询功能；
- 4) 支持对象权限控制，主要是对象级的 ACL 的支持；
- 5) 支持对象版本化；
- 6) 支持大对象数据存取，大对象数据主要为多媒体文件。

如图 3，对于底层不同的关系数据库，应用层建立有不同的对象抽象组件。对象抽象组件针对不同的关系数据库实现各自的数据库访问操作，所有对象抽象组件对外提供相同的 ADO 接口，使得数据库抽象层将底层关系数据库完全屏蔽，开发使用 PDM 系统时不必考虑底层各关系数据库间存在的差异，具有自适应关系数据库的作用。

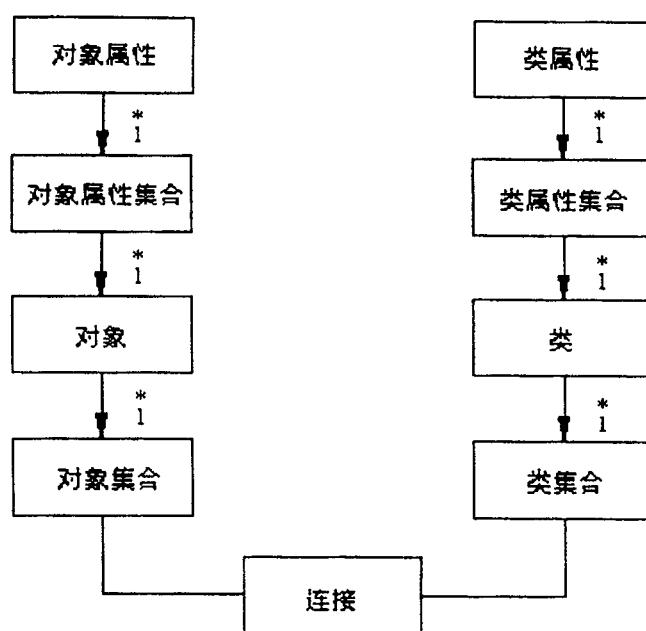


图 1

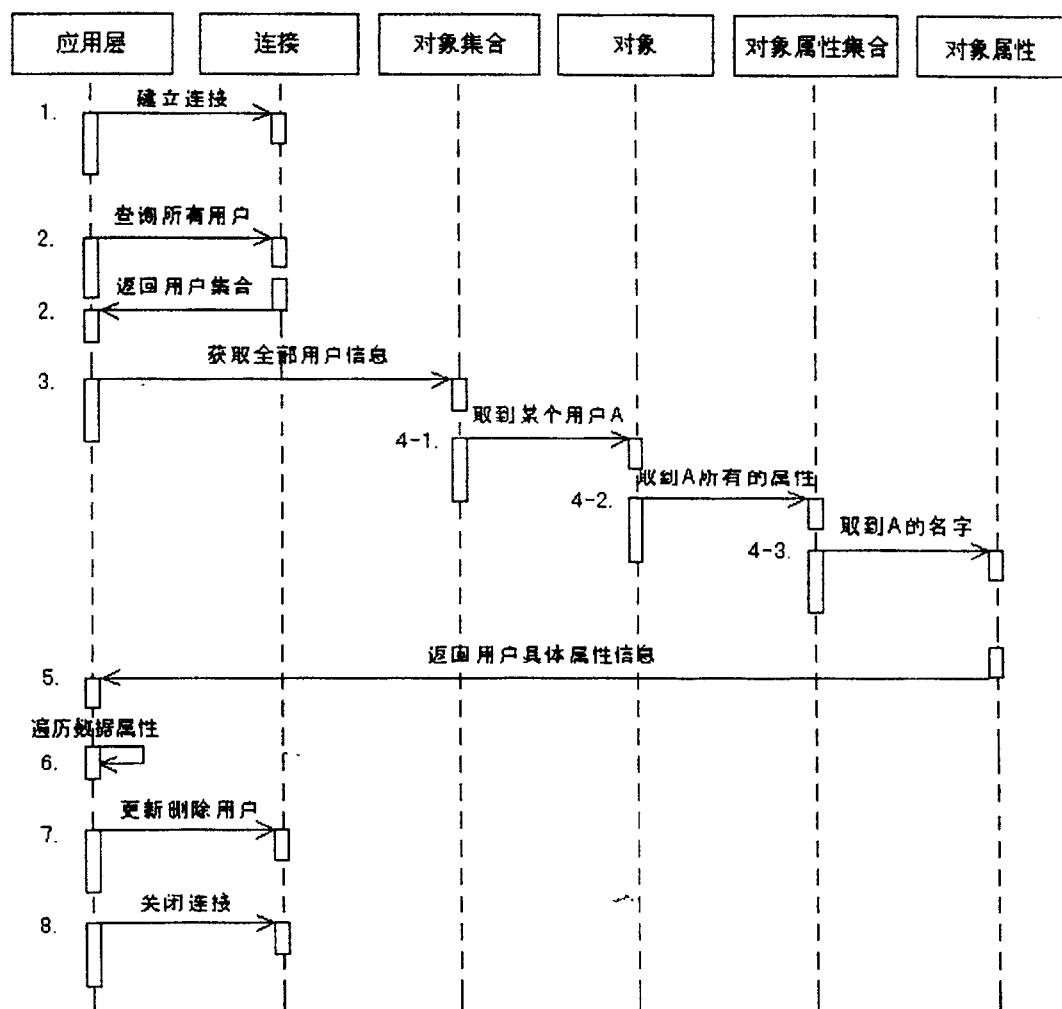


图 2

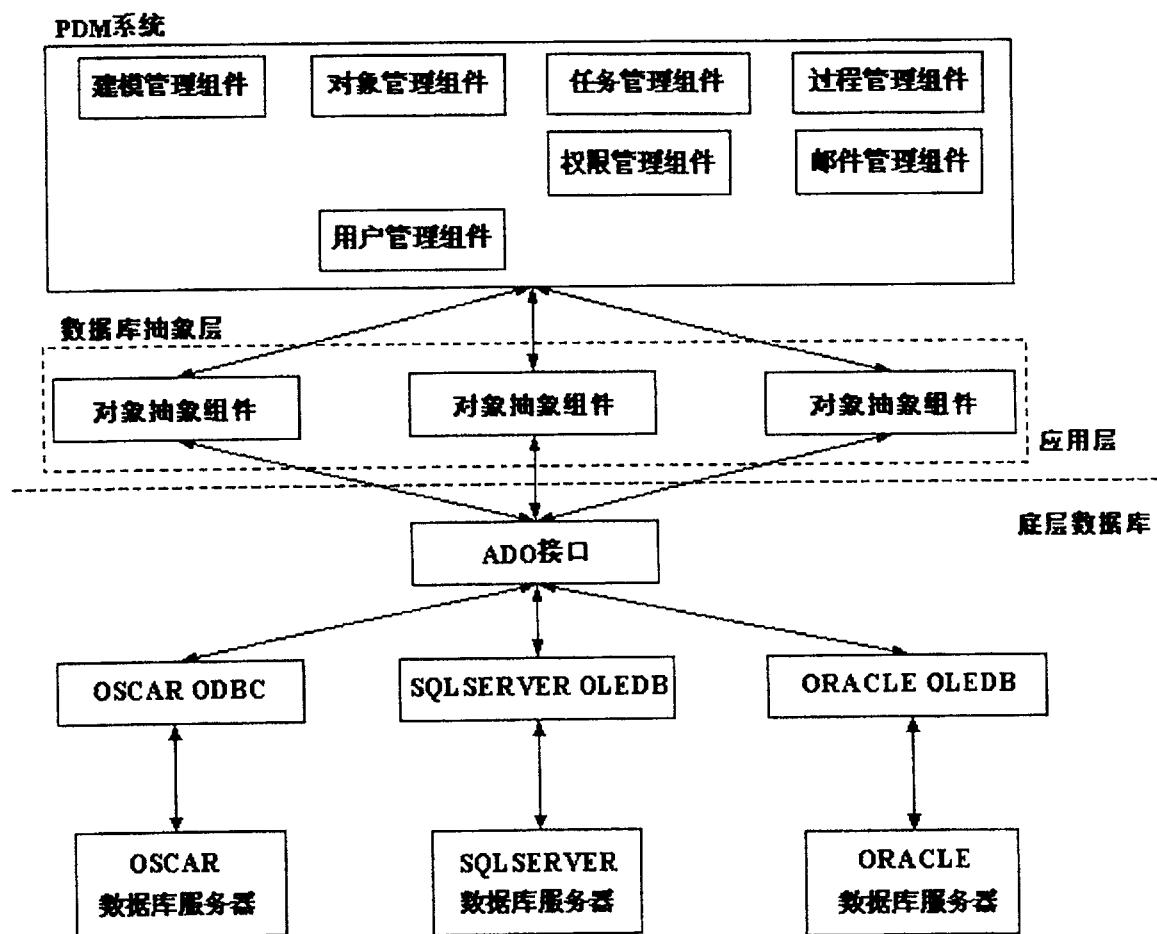


图 3