

Oracle PL/SQL 面向对象特性 Web 应用研究

柳 丹

(华东师范大学, 上海 200062)

摘 要:随着计算机软件技术的不断发展,面向对象的程序设计思想正在逐步取代原有的面向过程的程序设计思想。面向对象程序设计可以大大降低建立复杂应用的开销时间,已经被应用开发人员广泛接受并付诸实践。在 PL/SQL 中,面向对象程序设计目前是基于对象类型来完成的。文中主要探究 PL/SQL 对象类型在 Web 上的应用,提出了应用模型,并以实例加以说明。

关键词:过程结构化查询语言;面向对象;对象类型;视图对象

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2006)11-0234-04

Research on Web Application with Object Oriented Characteristic of Oracle PL/SQL

LIU Dan

(East China Normal University, Shanghai 200062, China)

Abstract: Object oriented has been replacing process oriented along with development of computer technology in software. Time spending on software designed by OOP (object oriented programming) can be reduced consumedly, so OOP has been accepted and applied widely. OOP of PL/SQL is implemented based on object type at present. This paper mainly probes into Web application with object type of PL/SQL and brings forward application model which is explained with an example.

Key words: PL/SQL; object oriented; object type; view object

0 引 言

目前,以 B/S(Browser/Server)为模式的瘦客户端、强服务器端的 Web 系统得到广泛应用。这对服务器端提出了更高的要求。PL/SQL(Procedural Language/SQL)是 Oracle 对 SQL 扩展的过程性语言,它的引入使得 Oracle 数据库服务器担当数据库存储器和业务处理的双重角色,起到了保证数据的安全性和完整性作用。而且与一般的匿名 SQL 代码块主要的不同是:当客户端应用程序调用时,只需发送一条调用命令,数据库服务器就会执行该过程,无须在网上传送大量的源程序代码,只需传送一条调用命令,同时接收,这就大大降低了网络通信的负担^[1]。Oracle 8i 在引入对象关系特性之后,PL/SQL 更是能方便快速地实现各种复杂业务逻辑。使用对象关系特性的主要原因是以一种自然的方式扩展 PL/SQL 语言,增强其功能;并且最主要的是它实现了关系和对象之间的转换^[2];另外,数据流以封装的对象实例传递,可以避免反复访问数据库,更能减少网络传输的压力。

Oracle 10g 主要面向对象特性,包括用户定义对象类

型、对象类型继承和演化、动态方法调度、对象表、SQLJ 对象等。这些特性都是在关系结构的基础上进行的面向对象的扩展,并能无缝地与 Oracle BC4J (Business Components for Java)中间件结合实现强大的 Web 功能。

1 对象类型和嵌套表

对象类型是扩展 PL/SQL 新功能的一种很好的方法,与 C++ 或 Java 中用类结构扩展新功能一样,它具有类的特征,如封装性、抽象性、继承性和多态性^[3]。对象类型定义好后,开发人员就能像使用 Varchar2, Date 等简单类型一样方便地使用它。严格意义上,对象类型是一种 Oracle 的数据结构,其实例可以存储单条记录;并可以定义基于对象类型的嵌套表,其实例能用于存储多条记录。

1.1 定义对象类型

可以定义基本数据表的对象类型,其属性和数据表中列一一对应,对其进行操作就是对数据表操作,方便实现增、删、改、查;也可以自定义对象类型,其属性可以对应多张数据表的列;其相对灵活,在多表多列多条件的复杂查询中起到重要作用,比具有同样功能的视图更易于创建和维护。

下面是自定义类型的示例:

```
CREATE OR REPLACE TYPE rec_emp_det AS OBJECT
```

收稿日期:2006-02-21

作者简介:柳 丹(1982-),男,湖南长沙人,硕士研究生,研究方向为智能化教育信息处理、软件技术。

```
(
emp_no VARCHAR2(10),
dep_name VARCHAR2(20),
fullname VARCHAR2(20),
tel_no VARCHAR2(11),
address VARCHAR2(40),
salary NUMBER(6),
CONSTRUCTOR FUNCTION rec_emp_det RETURN SELF AS
RESULT,
MEMBER FUNCTION get_emp_det RETURN VARCHAR2
)
```

上面已经对一个包含员工详细信息类型进行了声明,此类型包括 5 个属性、一个公用的不带参数的构造函数和一个公用的普通成员过程,这里要注意的就是属性的类型只能是基本类型。此对象类型还不能被继承,如要被继承可以将其声明为 NOT FINAL。类型的主体为(如没函数或过程则不必实现主体):

```
CREATE OR REPLACE TYPE BODY rec_emp_det IS
CONSTRUCTOR FUNCTION rec_emp_det RETURN SELF
AS RESULT IS
BEGIN
RETURN;
END;
MEMBER FUNCTION get_emp_det RETURN VARCHAR2
IS
emp_det VARCHAR2(100);
BEGIN
emp_det := emp_no || ' ' || dep_name || ' ' || fullname || ' ' ||
tel_no || ' ' || address || ' ' || salary;
RETURN emp_det;
END;
```

定义对象类型实例:

```
p_emp_det rec_emp_det;
p_emp_det := rec_emp_det();
可以通过实例访问对象类型属性、函数或过程:
p_emp_det.emp_no := '123123';
g_details := p_emp_det.get_emp_det;
```

1.2 定义嵌套表

嵌套表是数据元素可以无限大而且无序的一个一维数组;基于对象类型的嵌套表实例存储在内存中,能像数据表一样实现多条记录的临时存储。

```
CREATE OR REPLACE TYPE nt_rec_emp_det IS TABLE OF
rec_emp_det
```

2 Java 业务组件

BC4J(Business Components for Java)是 Oracle 提出的一种面向对象的 Java 组件架构,从面向对象系统设计的角度考虑就很好理解,就是将对象的属性、关系和逻辑规则等信息用 Java 代码和 XML 文件进行定义,实现了对象

的封装^[4]。

2.1 为何引入 BC4J

为什么要使用一个新的应用程序层来强制执行业务逻辑,而不在用户界面(UI)中强制执行业务逻辑呢?原因就是与 UI 层强制执行业务逻辑相比,BC4J 的优点是它具有很好的可重用性,一个单独的 BC4J 层可以为公司的所有需求提供业务逻辑(当然核心业务可以数据库存储过程来执行),业务组件可以被 UI 一次次地重用,而不必为每个用户界面都重新编写业务逻辑;并且 BC4J 的实体对象可以与数据库表无缝连接,数据能实现实时对应;在中间层处理数据高级缓存,支持使用瘦客户端^[5]。

2.2 视图组件

VO(View Object)是 BC4J 主要业务组件之一,它是伙伴数据访问对象(Companion Data Access Object),它通过为客户提供行集(Row Set)使客户端可以查询或更新数据。它分为 4 种:EO Based VO, Query VO, Transient VO, Combox VO。其中 Transient VO 就是连接对象类型实例和 UI 的中间层,它最灵活,其对象属性与 PL/SQL 对象类型属性实现一一对应,实现了零散数据的传递转化为封装了属性数据的对象实例的传递。

3 基于对象类型 Web 应用开发

对象类型 Web 应用过程实际上就是对象数据以对象实例的形式传递的过程。这里将讲述一种新型的基于对象类型的 Web 应用技术。

3.1 应用模型

此应用模型实现了 MVC 设计模式。BC4J 充当了模型层,能和数据库对应,UIX Handler 充当了控制层,UIX (User Interface XML)是视图层。对象类型的能力体现在与模型层的交互上。

由图 1 可以看出,PL/SQL 引擎把从数据库中得到的数据对象化,也就是用对象类型的构造函数把本来从多个表中得到的零散数据“打包”,存入对象类型实例中(过程 A,以下用字母表示过程),这个过程实现了对对象和关系的转换;然后通过 SQLJ 的连接把数据传递到与对象类型同名的 Java 类的实例中(B);UIX Handler 再去捕捉这个实例,把数据塞到视图对象(C);BC4J 利用数据显示绑定机制将显示到相应的 UIX 页面上(D)。这个过程是部分可逆的:用户要想更新数据库,首先页面提交操作(E),然后 Handler 将更新的数据放在视图对象中,以后就是 A、B、C 的逆过程了,只是在 A 中可以利用对象类型来更新数据,但是必须编写相应的 PL/SQL 代码。

上述过程中对象类型充当了传递数据的载体。与之对应的 Java 对象类型类(图中的 Java Class: Nt_Rec)也是载体,但是它们之间是如何通讯的呢?

3.2 数据传输

过程 B 中 SQLJ 是一个与 Java 编程语言紧密集成的

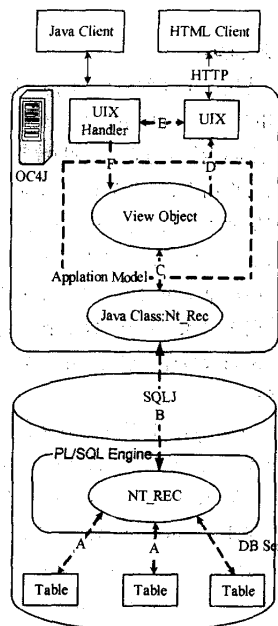


图 1 基于对象类型 Web 应用模型

嵌入式 SQL 的版本,Java 通过它将 SQL 语句传给数据库,就如同 JDBC 一样。但是不同的是 JDBC 不能识别对象类型,SQLJ 可以。经过仔细查看,SQLJ 文件经过编译后成为 Java 类型,里面的代码符合 JDBC 的标准。原来,SQLJ 只是对 JDBC 进行了扩展和封装,使得开发更加简单,开发者只需编写简单的 SQLJ 代码,不必硬编码实现 JDBC。那 SQLJ 识别对象类型的呢?这个简单,只要新类型注册了即可。那如何实现自定义类型注册呢?其实 oracle.jdbc.OracleCallableStatement 中的 registerInParameter()和 registerOutParameter()方法解决了这个问题。当然,这个过程由 OC4J 服务器来自动完成。

3.3 实例分析

下面以实例说明整个开发过程。

3.3.1 简单需求

从部门表 dep 和员工表 emp 中找出所有员工的详细资料,称为 EmpDet 应用。

3.3.2 分析设计

采用对象类型来实现 Web 开发。图 2、图 3 中描述了功能实现所需的类和系统中各个类实例的执行顺序。容易看出此架构符合 MVC 模式。BrEmpDetView 是 View-er,BrEmpDetViewHdlr 是 Controller,其它都是 Model。下面只就 Model 部分的关键代码进行说明。

3.3.3 实作部分

(1)在 Pk_emp 包中,利用前面定义的对象类型封装数据。

```
FOR c IN (SELECT rec_emp_det(a.emp_no, b.dep_name, a.
fullname, a.tel_no, a.address, a.salary)rec
FROM emp a, dep b WHERE a.dep_no = b.dep_no)
LOOP
```

```
i:=i+1;
p_emp_det:=nt_rec_emp_det(); p_emp_det.EXTEND;
p_emp_det(i):=rec_emp_det(); p_emp_det(i):=c.rec;
END LOOP;
```

对象类型构造函数 rec_emp_det()对数据对象化,然后一条条塞给嵌套对象类型实例 p_emp_det,这个实例是作为 OUT 参数传出的。

(2)在 SQLJ 文件中,方法 get_emp_det()负责传送 SQL 到数据库。

```
#sql [sqlCtx] {
begin
pk_emp.get_emp_det(
:OUT (p_emp_det[0]));
end;
```

(3)在 EmpDetServiceImpl 类中,p_get_emp_det()方法调用 SQLJ 中 get_emp_det(),此方法返回一个 HashMap 类型 hm 实例,将 PL/SQL OUT 出来的值给

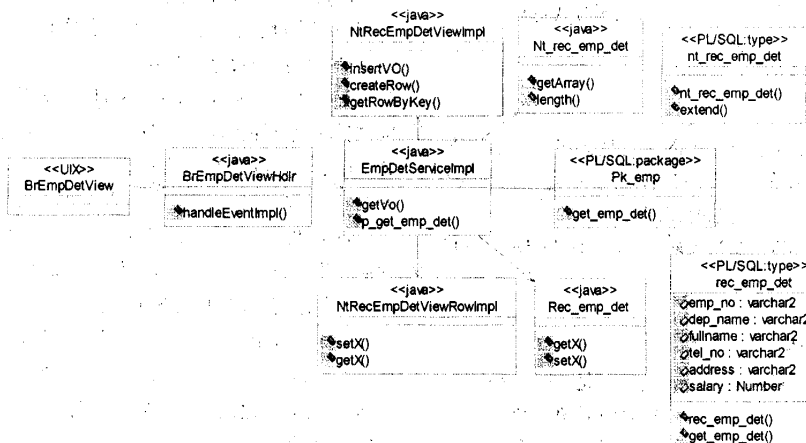


图 2 EmpDet 应用的 UML 类图

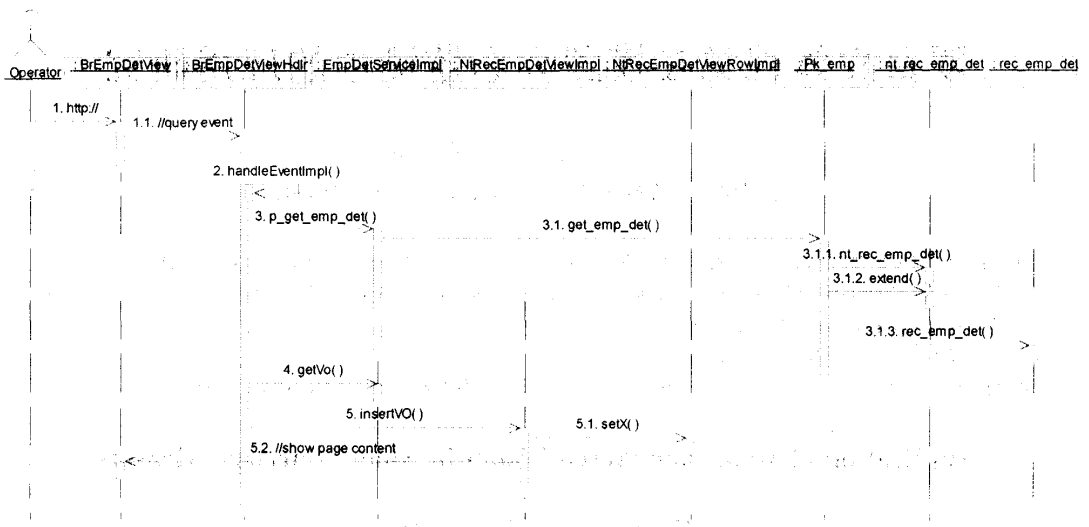


图 3 EmpDet 应用的 UML 时序图

Nt_rec_emp_det 类型 p_emp_det, 并装入 hm 中, 便于存取。

```
HashMap hm = new HashMap();
```

```
Nt_rec_emp_det p_emp_det = null;
```

```
Pk_emp.get_emp_det(sqlCtx, p_contract_no, p_emp_det);
```

```
hm.put(g_giz38mla, "g_giz38mla_key");
```

(4) 在其他类, 如 NtRecEmpDetViewRowImpl 和 Rec_emp_det 符合 JavaBean 规范, 包含对单列数据的存取的 getX 和 setX 方法。而 NtRecEmpDetViewImpl 中主要负责从 Nt_rec_emp_det 中取数据然后绑定到页面上。

4 总结

Oracle PL/SQL 面向对象特征是在关系结构的基础

上进行的面向对象的扩展, 它在对海量数据进行处理的过程中发挥着重要的作用, 有利于提高 Web 系统的性能, 必将得到广泛的应用。

参考文献:

- [1] 王秋生. 基于 PL/SQL 的 Oracle 数据库性能优化[J]. 微机发展, 2003, 13(12): 46-49.
- [2] 海 亮, 林立新. Oracle 10g PL/SQL 编程[M]. 北京: 中国水利出版社, 2004.
- [3] Kyte T. Oracle 专家高级编程[M]. 袁勤勇, 张玉魁, 等译. 北京: 清华大学出版社, 2002.
- [4] Koletzke P, Dorsey P. Oracle 9i JDeveloper 开发手册[M]. 冯锐, 由渊霞译. 北京: 清华大学出版社, 2004.
- [5] Oracle 公司. Oracle ADF White Paper[Z]. 2004.

(上接第 150 页)

的, 其它应用程序可以通过 Bean 的调用接口重用已部署的 Bean。

(3) 具有较高构建应用程序的能力。业务逻辑与表示逻辑相分离, 组件设计与集成, 简化了构建复杂的企业应用程序的过程。

(4) 容易开发 Web 服务。EJB 框架为用户提供了一种简单的开发和访问 Web 服务的方法。

(5) 分布式部署。EJB 框架支持把应用程序部署到一个网络的多个服务器上, 便于分布式系统的开发。

5 结论

文中简要讨论了 EJB 体系结构模型。与传统的体系结构模型相比, EJB 体系结构有明显的优势, 尤其适合于构造功能复杂的企业级 Web 应用系统。特别是 EJB 技术的使用实现了业务逻辑的封装, 使多层体系结构模型在分

布式应用中发挥极大作用。

参考文献:

- [1] 张友生. 软件体系结构[M]. 北京: 清华大学出版社, 2004: 33-120.
- [2] 卫红春. 信息系统分析与设计[M]. 西安: 西安电子科技大学出版社, 2003: 52-59.
- [3] Buest C. J2EE 编程指南[M]. 北京: 电子工业出版社, 2002: 649-867.
- [4] 田雪松. J2EE 网络编程标准教程[M]. 上海: 科学普及出版社, 2003.
- [5] Matcna V. EJB 应用指南[M]. 北京: 清华大学出版社, 2004.
- [6] 沈建南. Enterprise Java Bean 程序设计实例详解[M]. 北京: 中国铁道出版社, 2004: 219-324.