**Java调用Oracle存储过程**

作者：   来源：中文JAVA技术网   发布者：admin   
时间：2009-04-20 14:41:40   点击：1788

好久不用java调用存储过程的系统了，今天被别人给问住了。。。晕的很。。。

下面是自己的一些总结:

    创建需要的测试表:create table Test(tid varchar2(10),tname varchar2(10));

**第一种情况：无返回值.**

      create or replace procedure test\_a(param1 in varchar2,param2 in varchar2) as

       begin

            insert into test value(param1,param2);

     end;

    Java调用代码:

package com.test;

import java.sql.\*;

import java.io.\*;

import java.sql.\*;

public class TestProcA  
{  
   public TestProcA(){  
    
   }  
     
   public static void main(String []args)  
   {  
          
        ResultSet rs = null;  
        Connection conn = null;   
        CallableStatement proc = null;   
           
        try{  
          Class.forName("oracle.jdbc.driver.OracleDriver");   
          conn =  DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:test", "test", "test");   
          proc = conn.prepareCall("{ call test\_a(?,?) }");   
          proc.setString(1, "1001");   
          proc.setString(2, "TestA");   
          proc.execute();   
        }catch(Exception e){  
     e.printStackTrace();  
 }finally{  
           try{  
       if(null!=rs){  
                 rs.close();

          if(null!=proc){  
                    proc.close();  
          }

          if(null!=conn){  
                    conn.close();  
          }   
       }    
           }catch(Exception ex){

           }  
        }  
   }

}

**第二种情况：有返回值的存储过程(返回值非列表).**

存储过程为:  
create or replace procedure test\_b(param1 in varchar2,param2 out varchar2)  
as   
 begin   
    select tname into param2 from test where tid=param1;  
 end;

Java调用代码：

 package com.test;

import java.sql.\*;

import java.io.\*;

import java.sql.\*;

public class TestProcB  
{  
   public TestProcB(){  
    
   }  
     
   public static void main(String []args)  
   {  
          
        Connection conn = null;   
        CallableStatement proc = null;   
         
        try{  
          Class.forName("oracle.jdbc.driver.OracleDriver");   
          conn =  DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:test", "test", "test");   
          proc = conn.prepareCall("{ call test\_b(?,?) }");   
          proc.setString(1, "1001");   
          proc.registerOutParameter(2, Types.VARCHAR);  
          proc.execute();   
   System.out.println("Output is:"+proc.getString(2));  
        }catch(Exception e){  
     e.printStackTrace();  
 }finally{  
           try{

          if(null!=proc){  
                    proc.close();  
          }

          if(null!=conn){  
                    conn.close();  
          }   
   
           }catch(Exception ex){

           }  
        }  
   }

}

**第三种情况：返回列表.**

由于oracle存储过程没有返回值，它的所有返回值都是通过out参数来替代的，列表同样也不例外，但由于是集合，所以不能用一般的参数，必须要用pagkage了.要分两部分来写：  
create or replace package tpackage as  
type t\_cursor is ref cursor;  
procedure test\_c(c\_ref out t\_cursor);  
end ;

create or replace package body tpackage as  
procedure test\_c(c\_ref out t\_cursor) is  
   begin  
      open c\_ref for select \* from test;  
  end test\_c;  
end tpackage;

Java调用代码：

package com.test;

import java.sql.\*;

import java.io.\*;

import java.sql.\*;

public class TestProcB  
{  
   public TestProcB(){  
    
   }  
     
   public static void main(String []args)  
   {  
          
        Connection conn = null;   
        CallableStatement proc = null;   
        ResultSet rs =  null;  
        try{  
          Class.forName("oracle.jdbc.driver.OracleDriver");   
          conn =  DriverManager.getConnection("jdbc:oracle:thin:@127.0.0.1:1521:test", "test", "test");   
          proc = conn.prepareCall("{? = call tpackage.test\_b(?) }");   
             
          proc.registerOutParameter(1,OracleTypes.CURSOR);  
          proc.execute();   
          while(rs.next()){  
              System.out.println(rs.getObject(1)+"\t"+rs.getObject(2));  
          }  
        }catch(Exception e){  
     e.printStackTrace();  
 }finally{  
           try{  
          if(null!=rs){  
              rs.close();  
             if(null!=proc){  
                    proc.close();  
             }

             if(null!=conn){  
                    conn.close();  
             }   
          }  
          }catch(Exception ex){

           }  
        }  
   }

}