# ORCAL存储过程/函数

存储过程/函数简单来说就是存储在数据库中的类似程序片段,他和普通的数据库操作语言(DDL,DML)最大的不同在于它是一种过程性语言,也就是一段通过顺序/循环/判断三种基本过程来操作DDL,DML语句的语言.存储过程和函数基本相同,也可以说函数是存储过程的一种有显示返回值的形式(函数定义有return而存储过程不直接定义return).

Mybatis和Ibatis操作存储过程/函数有所不同

1. **Mybatis对存储过程/函数的调用**

首先,看下存储过程的定义:

CREATE OR REPLACE PROCEDURE FOR\_TAKON(NEW\_TAKON IN VARCHAR2,NEW\_IP IN VARCHAR2,NEW\_TIME IN DATE,THE\_RESLUT OUT VARCHAR2,THE\_UNAME OUT VARCHAR2,THE\_RNAME OUT VARCHAR2,THE\_USERID OUT INTEGER)

当然,函数是存储过程的一种特殊形式,这段代码也可以改为:

CREATE OR REPLACE FUNCTION FOR\_TAKON(NEW\_TAKON IN VARCHAR2,NEW\_IP IN VARCHAR2,NEW\_TIME IN DATE,THE\_RESLUT OUT VARCHAR2,THE\_UNAME OUT VARCHAR2,THE\_RNAME OUT VARCHAR2)RETURN INTEGER

题外话一点,存储过程/函数还有种特殊的参数形式,游标CURSOR,游标也可以想做是数据库给Select语句查询出的数据开辟的一块内存空间,对程序员来说,可以看做是JAVA中ResuluSet接口在数据中的实现方式,游标.Mybatis处理游标比Ibatis容易.游标的返回形式下面会介绍

定义过程中红色字体,in/out,显而易见表示了参数的传入/传出,定义过程中的蓝色字体VARCHAR2/DATE/INTEGER以及Mypackage.mycursor(我自定义的游标)表示数据库中参数的类型,和java中do(String name)的形式对应,不同的是JAVA中只能给形参传值,数据库既可以传值进来(in),也能传值出去(out).

下面是在Mybatis中XXXMAPPER.XML中调用存储过程的例子:

<select id="selectTakon" statementType="CALLABLE">

<![CDATA[

{call FOR\_TAKON(#{takon,mode=IN,jdbcType=VARCHAR},#{ip,mode=IN,jdbcType=VARCHAR},#{time,mode=IN,jdbcType=TIMESTAMP},#{result,mode=OUT,jdbcType=VARCHAR},#{uname,mode=OUT,jdbcType=VARCHAR},#{rname,mode=OUT,jdbcType=VARCHAR},#{userid,mode=OUT,jdbcType=INTEGER})}

]]>

</select>

同样,换成函数的调用:

<select id="selectTakon" statementType="CALLABLE">

<![CDATA[

{#{userid,mode=OUT,jdbcType=INTEGER} = call FOR\_TAKON(#{takon,mode=IN,jdbcType=VARCHAR},#{ip,mode=IN,jdbcType=VARCHAR},#{time,mode=IN,jdbcType=TIMESTAMP},#{result,mode=OUT,jdbcType=VARCHAR},#{uname,mode=OUT,jdbcType=VARCHAR},#{rname,mode=OUT,jdbcType=VARCHAR})}

]]>

</select>

调用时的红色字体和数据库中对应需要传入/传出的值,”按顺序”的对应, 蓝色字体和数据库中的参数类型对应,不同的是数据库的VARCHAR2这里需要写成VARCHAR,数据中的DATE这里有几种定义, DATE表示是日期(年月日)类型,TIME表示是时间类型(时分秒),TIMESTAMP表示准确时间(年月日时分秒),其他类型有疑问的可以在网上查询,注意这里游标类型,除了ORCAL需要定义为ROCALCURSOR类型,其它数据库都是CURSOR类型

使用存储过程的话XXXMAPPER.JAVA可以写成

void selectTakon(Map<String,Object> map)throws Exception

使用函数的话,这里的可以写成:

Integer selectTakon(Map<String,Object> map)throws Exception

之后和其他SQL语句流程一样,配置XXXSERVICEINTERFACE,XXXSERVICEIMP…,之后开始调用XXXSERVICEIMP,下面是使用存储过程的例子:

先将需要in进存储过程的参数放入map,注意这里Map的的Key:”takon”,”ip”,”time”和XXXMAPPER.XML中#{takon…/#{ip…/#{time一一对应,区分大小写,参数类型对应

java.util.Map<String,Object> map = **new** HashMap<String, Object>();

map.put("takon", “TUTUMAN”);

map.put("ip", “0.0.0.0”);

map.put("time", new Date());

其次,执行XXXSERVICEIMP对应的方法:

XXXSERVICEIMP.selectTakon(map);

//函数的话是Integer userid = XXXSERVICEIMP.selectTakon(map);

最后,对应XXXMAPPER.XML中#{result…/#{uname…/#{rname/#{userid

//函数的话这里就是用userid来获得,上面已获得了

String result = (String)map.get("result");

String uname = (String)map.get("uname");

String rname = (String)map.get("rname");

Integer userId = (Integer)map.get("userid");

题外话,是游标的话ResultSet rs = (ResultSet)map.get(“youBiaoName”), ResultSet都获得了,剩下的…你懂的..

1. **Ibatis对存储过程/函数的调用**

Ibatis的区别主要在于XXXMAPPER.XML中,Mybatis使用select的形式,而Ibatis使用:

<parameterMap class="map" id="itIsMe">

    <parameter property="takon"  javaType="java.lang.String" jdbcType="varchar" mode="IN"/>

    <parameter property="uname"  javaType="java.lang.String" jdbcType="int" mode="OUT”/>

//下面略

</parameterMap>

  <produerce id="selectTakon" parameterMap=" itIsMe " >

<![CDATA[

{call FOR\_TAKON(?,?,?,?,?,?,?}

]]>

</produerce>

其他基本和Mybatis相同