Ibatis入门

习惯了hibernate太化自动了，学习下ibatis的半自动化， 各有各的优势，各有各的长处。

## 数据库表（oracle）：

create table student(

sid int ,

sname varchar2(50),

major varchar2(50)，

birth Date

)

create sequence studentPKSequence start with 1 increment by 1

select studentPKSequence.nextval from dual

insert into student values(1, 'deng1', 'major01', TO\_DATE('2014-12-22','yyyy-mm-dd'));

## 配置文件（SqlMapConfig.xml）：

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE sqlMapConfig

PUBLIC "-//ibatis.apache.org//DTD SQL Map Config 2.0//EN"

"http://ibatis.apache.org/dtd/sql-map-config-2.dtd">

<sqlMapConfig>

<transactionManager type=*"JDBC"* commitRequired=*"false"*>

<dataSource type=*"SIMPLE"*>

<property name=*"JDBC.Driver"* value=*"oracle.jdbc.driver.OracleDriver"*/>

<property name=*"JDBC.ConnectionURL"* value=*"jdbc:oracle:thin:@127.0.0.1:1521:orcl"*/>

<property name=*"JDBC.Username"* value=*"deng"*/>

<property name=*"JDBC.Password"* value=*"xxxx"*/>

</dataSource>

</transactionManager>

<sqlMap resource=*"com/dyh/ibatis/test01/Student.xml"*/>

</sqlMapConfig>

# 配置文件（Student.xml）：

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!DOCTYPE sqlMap

PUBLIC "-//ibatis.apache.org//DTD SQL Map 2.0//EN"

"http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMap namespace=*"Student"*>

<typeAlias alias=*"Student"* type=*"com.dyh.ibatis.test01.Student"*/>

<select id=*"selectAllSudent"* resultClass=*"Student"*>

select \* from student

</select>

<select id=*"selectSudentById"* parameterClass=*"int"* resultClass=*"Student"*>

select \* from student where sid=#sid#

</select>

<insert id=*"insertStudent"* parameterClass=*"Student"*>

insert into Student(

sid,

sname,

major,

birth

)

values(

#sid#,#sname#,#major#,#birth#

)

</insert>

<delete id=*"deleteStudentById"* parameterClass=*"int"*>

delete from Student where sid =#sid#

</delete>

<update id=*"updateStudentById"* parameterClass=*"Student"*>

update Student set major=#major# where sid = #sid#

</update>

<select id=*"selectStudentByName"* parameterClass=*"String"* resultClass=*"Student"*>

select \* from Student where sname like '%$sname$%'

</select>

<insert id=*"insertStudentBySequence"* parameterClass=*"Student"*>

<selectKey resultClass=*"int"* keyProperty=*"sid"*>

select studentPKSequence.nextVal from dual

</selectKey>

insert into Student(

sid,

sname,

major,

birth

)

values(

#sid#,#sname#,#major#,#birth#

)

</insert>

</sqlMap>

# Student的实体类

**package** com.dyh.ibatis.test01;

**import** java.util.Date;

**public** **class** Student {

**private** **int** sid;

**private** String sname;

**private** String major;

**private** Date birth;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** String getMajor() {

**return** major;

}

**public** **void** setMajor(String major) {

**this**.major = major;

}

**public** Date getBirth() {

**return** birth;

}

**public** **void** setBirth(Date birth) {

**this**.birth = birth;

}

@Override

**public** String toString() {

**return** "Student [sid=" + sid + ", sname=" + sname + ", major=" + major

+ ", birth=" + birth + "]";

}

}

# DAO接口（IStudentDAO）：

**package** com.dyh.ibatis.test01;

**import** java.util.List;

**public** **interface** IStudentDAO {

**public** **void** addStudent(Student student);

**public** **void** addStudentBySequence(Student student);

**public** **void** delectStudentById(**int** id);

**public** **void** updateStudentById(Student student);

**public** List<Student> querAllStudent();

**public** List<Student> querStudentByName(String name);

**public** Student queryStudentById(**int** id);

}

# DAO实现（IStudentDAOimpl）：

package com.dyh.ibatis.test01;

import java.io.IOException;

import java.io.Reader;

import java.sql.SQLException;

import java.util.List;

import com.ibatis.common.resources.Resources;

import com.ibatis.sqlmap.client.SqlMapClient;

import com.ibatis.sqlmap.client.SqlMapClientBuilder;

public class IStudentDAOimpl implements IStudentDAO {

private static SqlMapClient sqlMapper = null;

static {

try {

Reader reader = Resources.getResourceAsReader("com/dyh/ibatis/test01/SqlMapConfig.xml");

sqlMapper = SqlMapClientBuilder.buildSqlMapClient(reader);

reader.close();

} catch (IOException e) {

}

}

@Override

public void addStudent(Student student) {

// TODO Auto-generated method stub

try {

sqlMapper.insert("insertStudent", student);

} catch (SQLException e) {

e.printStackTrace();

}

}

@Override

public void addStudentBySequence(Student student) {

// TODO Auto-generated method stub

try {

sqlMapper.insert("insertStudentBySequence", student);

} catch (SQLException e) {

e.printStackTrace();

}

}

@Override

public void delectStudentById(int id) {

// TODO Auto-generated method stub

try {

sqlMapper.delete("deleteStudentById", id);

} catch (SQLException e) {

e.printStackTrace();

}

}

@Override

public void updateStudentById(Student student) {

// TODO Auto-generated method stub

try {

sqlMapper.update("updateStudentById", student);

} catch (SQLException e) {

e.printStackTrace();

}

}

@Override

public List<Student> querAllStudent() {

// TODO Auto-generated method stub

List<Student> studentList = null;

try {

studentList = sqlMapper.queryForList("selectAllSudent");

} catch (SQLException e) {

e.printStackTrace();

}

return studentList;

}

@Override

public List<Student> querStudentByName(String name) {

// TODO Auto-generated method stub

List<Student> studentlist = null;

try {

studentlist = sqlMapper.queryForList("selectStudentByName", name);

} catch (SQLException e) {

e.printStackTrace();

}

return studentlist;

}

@Override

public Student queryStudentById(int id) {

// TODO Auto-generated method stub

Student student = null;

try {

student = (Student) sqlMapper.queryForObject("selectSudentById", id);

} catch (SQLException e) {

e.printStackTrace();

}

return student;

}

}

# 测试类：

**package** com.dyh.ibatis.test01;

**import** java.sql.Date;

**public** **class** test {

**public** **static** **void** main(String[] args) {

//查找所有

IStudentDAO dao = **new** IStudentDAOimpl();

// for(Student student : dao.querAllStudent()){

// System.out.println(student);

// }

// //通过id查找

// System.out.println(dao.queryStudentById(3));

// //插入实体

// Student student = new Student();

// student.setSid(4);

// student.setSname("deng4");

// student.setMajor("major4");

// student.setBirth(Date.valueOf("2014-12-23"));

// dao.addStudent(student);

//通过id删除

// dao.delectStudentById(4);

//通过id来更新

// Student student = new Student();

// student.setSid(3);

// student.setMajor("major33");

// dao.updateStudentById(student);

//模糊查询

// for(Student student : dao.querStudentByName("g3")){

// System.out.println(student);

// }

//通过序列插入数据

Student student = **new** Student();

student.setSname("deng444");

student.setMajor("major444");

student.setBirth(Date.*valueOf*("2014-12-23"));

dao.addStudent(student);

}

}