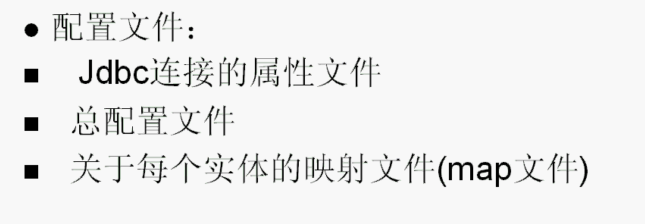
1.

## 1.创建一个数据库 然后创建一个表进行操作

create table user (id int primary key auto\_increment ,name varchar(20),age int);

## 2.建立相应数据库的domain user

**public** **class** User **implements** Serializable {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** **int** id;

**private** String name;

**private** **int** age;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "User [id=" + id + ", name=" + name + ", age=" + age + "]";

}

## 3.建立配置文件与数据库对应（相当于hibernate中的那个）

<transactionManager type=*"JDBC"*>

<dataSource type=*"SIMPLE"*>

<property name=*"JDBC.Driver"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"JDBC.ConnectionURL"* value=*"jdbc:mysql://localhost:3306/ibatis"* />

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

<property name=*"JDBC.Password"* value=*"147094"* />

<property name=*"Pool.MaximumActiveConnections"* value=*"10"* />

<property name=*"Pool.MaximumIdleConnections"* value=*"5"* />

<property name=*"Pool.MaximumCheckoutTime"* value=*"120000"* />

<property name=*"Pool.TimeToWait"* value=*"500"* />

<property name=*"Pool.PingQuery"* value=*"select 1 from ACCOUNT"* />

<property name=*"Pool.PingEnabled"* value=*"false"* />

<property name=*"Pool.PingConnectionsOlderThan"* value=*"1"* />

<property name=*"Pool.PingConnectionsNotUsedFor"* value=*"1"* />

</dataSource>

</transactionManager>

<!-- 映射文件 --> 相当于hibernate中的那个与数据库对应的文件，但是这里的区别就出来了，这里的区别是它是sql语句的集合

<sqlMap resource=*"com/itmyhome/User.xml"* />

## 4.编写User.xml文件，这里用来编写数据库操作语句

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

<!DOCTYPE sqlMap

PUBLIC "-//iBATIS.com//DTD SQL Map 2.0//EN"

"http://www.ibatis.com/dtd/sql-map-2.dtd">

<sqlMap namespace=*"User"*>

下面这个user是别名，就是代表后面的type的java文件 ，下文用它来对数据进行草操作

<typeAlias alias=*"user"* type=*"com.itmyhome.User"* />

</select>

<!-- 查询所有的 --> 这里的id在java文件中会调用的

<select id=*"getAllUser"* resultClass=*"user"*>

select \* from user

</select>

</sqlMap>

## 5.建立第一个查询语句

**private** **static** **void** getAlluser() {

// **TODO** Auto-generated method stub

System.*err*.println("获得所有的user");

**try** {

List<User> userList = *sqlMap*.queryForList("getAllUser");

System.*err*.println("得到list的大小为"+userList.size());

**for**(User user:userList){

System.*out*.println(user.toString());

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

## 6．注意事项

<!-- 查询 -->

<!—int = java.lang.Integer

parameterClass 传入参数的类型

resultClass 返回结果的类型 -->

<select id=*"getUser"* parameterClass=*"int"* resultClass=*"user"*>

select \* from user where id = #id#

</select>