# 6.Mybatis

## 6.1.原生的JDBC的缺点

1.原生的JDBC操作数据库时,需要频繁的开关链接.

2.JDBC操作数据库时需要手写sql语句.学习需要成本

3.查询数据库的结果集,需要人为的进行封装.

4.JDBC中没有缓存处理

5.JDBC的sql语句写到java文件

## 6.2.Hibernate框架

面向对象的sql语句---Hql.好处操作数据库时可以使用对象进行操作.能够将查询的结果集自动的封装为对象.(ORM)自动的对象关系映射.

缺点:产生了很多冗余的sql语句

## 6.3.Mybatis框架

Mybatis是一个半自动的对象关系映射(ORM),实现结果集的自动封装,但是sql需要自己手写.

Mybatis中可以将sql写到配置文件中

## 6.4.Mybatis的入门案例

1.导jar包

计算机生成了可选文字:
绝asm一3.3.ljar
绝cglib一2.2.2.jar
绝CommonS一logging一1.1.i.j
绝java,,i，一3.17.1一GA.jar
冠log4）一i.2.i7.j。r
绝mybatis一3.2·2.jar
绝mysql一connector一ava一5.1
绝引叫一api一1.7.5.jar
退引闷一log4j12一1.7.s.jar

2.配置数据源

计算机生成了可选文字:
一配置数据源一＞
<enVironmentS
默认连接
<enVironment
<tr日ns日C
<dataSOU
10二”川夕sqL"1不能省略
鱿口DBC"/>
rCe
type二r,P00LED">
>>
了了了了
<property
<property
<property
<property
</dataSOurce>
</environment>
namb二书介呛节
name二“U尸乙“
池化的
,value二’'co阴．Inysq乙．jdbc
value二”dbc:Inysq乙：/／乙oc'
name二”Use户门Q切e"V日IUe二”尸OOt"
name二’'pass闪。尸d"value二’‘尸oot"

3.编辑映射文件

计算机生成了可选文字:
1＜尹XmlVerSIOn二’'1．日”encodlng二”UTF一8'’尹＞
2<IDOCTYPEmapper
3PUBLIC”一／/mybatis.org//DTDMapper3.0//EN"
4
S-
6
7-
8
9
.e
.1。
.2
.3
.4-
.5
.6
.7-
.8
.,
""ttp'／懊些垫丝钾哄孕吵华匕契黔per·dtd">
《ma"""r}n"mes""C”二”p刃“'"s“阴app“尸”》｝唯一标识映射文件
尸歇甲，一一一一一一一一一一一，
根据不同的操作选择不同的标签
<insertid二”"></insert>
<deleteid二”"></delete>
<updateid二”"></update>
<selectid二”"></select>
<！一查询全部的用户信息
id唯一标识冬q工语句
二＞
<selectid二‘,f乞ndA乙乙
自动完成结果集映射
”卜esultType二“;0,-0·Use。”卜
SeleCt*fromUSer
</select>

4.管理映射文件

计算机生成了可选文字:
>
／了
<！一映射文件一＞
<mappers>
<mapperresource二”pOJ'o/Use厂何appe尸．姗乙”
</mappers>

5.测试:

计算机生成了可选文字:
@Test
publicvoidtestel()throwsIOException{
/／通过流读取呵h御大如核心配置文件
Inputstreaminputstream二
Resources.getResou尸ceAsst尸ea切（"sq1Mapconfig.xml");
/／获取sq1SessionFactory
Sq1SessionFactoryfactory二
new
Sq1SessionFactoryBuilder().build(inputstream)j
/／数据库连接
Sq1Sessionsession二factory.opensession();
LISt<USer>
for(User
System
}
userList二session.selectList(
pojO.UserMapper.findAI
USer
:userList){
.out.println(user);
ndmesp己Ce·IC
}

## 6.5.Mybaits入库操作

计算机生成了可选文字:
/／新增用户
@Test
publicvoidtesteZ()throwsIOException{
Useruser二newUser()j
user.setName(”阿科”);
user.setAge(Ze);
user.setsex(”女”)j
Sq1Sessionsession二factory·opensession(）三
session.insert("pojo.UserMapper.addUser',,user);
/／更新操作需要提交事务
session.commit()j
}

计算机生成了可选文字:
<！一新增用户一＞
<insertid二“QddUSe尸“>
insertintouser(id,name,age,sex)
ValueS(null,#{name},#{age},#{sex})
</insert>

## 6.6.#{}用法

使用#{对象的属性}

计算机生成了可选文字:
<！一新增用户
<insertid二
二＞
"QddUSe尸’'>
insertinto
values(null
</insert>
name,age,SeX)
#{age},#{sex})

作用:

1.含有预编译的效果,能够防止sql注入攻击

2.为参数添加了一对""号

注意事项:

如果sql语句中以列名为参数时,切记使用${Map中的key},必须配置Map一起联用.

## 6.7.${}的用法:

计算机生成了可选文字:
Test
Jblicvoidteste4()throwsIOException{
HashMap<string,String>map二newHashMap<String,String>();
准备需要的mop
Sq1Sessionsession二factory·opensession()j
LISt<USer>USerLISt二
卜essiO”·
for(Useruser
System.Out
}
selectList("pojo.UserMapper.findUserByAge",map）三
userList){
.println(user)j

计算机生成了可选文字:
<SeleCtid
SeleCt
</select>
二”艺ndU5e户即八ge''
reSUltT
.Use尸‘,>
y
』口
*fromuserorder
Map中的key

总结:

以列名为参数时使用${},其他的使用#{}

**能用#{}取值,决不用${}**

## 6.8.Mybatis中的多值传递问题

需求:

要求查询年龄在18-22之间的人

问题:在Mybatis中只支持单值的传递.多值传递时没有现成的API

解决方法:

可以将多个值转化为**单个对象**或**Map**

**建议:**

**如果是插入操作/更新操作,使用对象.其他的使用MAP**