<https://www.yiibai.com/mybatis/mybatis-many2many.html>

在前面的章节中，我们学习了一对多，多对一的关系，现在我们来看看 Mybatis 中的多对多应用。

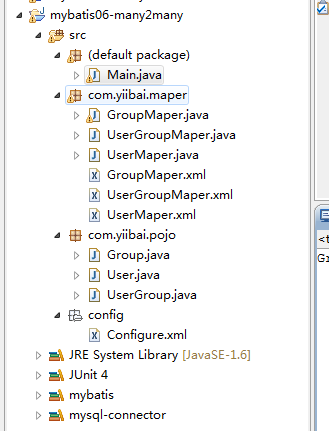
mybatis3.0 添加了association和collection标签专门用于对多个相关实体类数据进行级联查询，但仍不支持多个相关实体类数据的级联保存和级联删除操作。因此在进行实体类多对多映射表设计时，需要专门建立一个关联对象类对相关实体类的关联关系进行描述。下文将以“User”和“Group"两个实体类之间的多对多关联映射为例进行CRUD操作。

**1、应用场景**

假设项目中存在用户和用户组，从一个用户读取出它所在的用户组，从一个用户组也知道这个组内的所有用户信息。

**2、先做一些准备工作**

我们首先在创建一个 java 工程，工程名称为：mybatis06-many2many(下载)，还需要创建三张表，它们分别是用户表 user，用户组表 group 和 用户组映射表 user\_group ，一个户用户可以在多个用户组中，一个用户组中有多个用户。项目工程结构如下：



**user表的结构和数据：**

CREATE TABLE `user` (

`id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`username` varchar(64) NOT NULL DEFAULT '',

`mobile` varchar(16) NOT NULL DEFAULT '',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of user

-- ----------------------------

INSERT INTO `user` VALUES ('1', 'yiibai', '13838009988');

INSERT INTO `user` VALUES ('2', 'User-name-1', '13838009988');

**用户组 group 表的结构和数据：**

CREATE TABLE `group` (

`group\_id` int(10) unsigned NOT NULL AUTO\_INCREMENT,

`group\_name` varchar(254) NOT NULL DEFAULT '',

PRIMARY KEY (`group\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of group

-- ----------------------------

INSERT INTO `group` VALUES ('1', 'Group-1');

INSERT INTO `group` VALUES ('2', 'Group-2');

**用户组映射表 user\_group 的结构和数据：**

CREATE TABLE `user\_group` (

`user\_id` int(10) unsigned NOT NULL DEFAULT '0',

`group\_id` int(10) unsigned NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of user\_group

-- ----------------------------

INSERT INTO `user\_group` VALUES ('1', '1');

INSERT INTO `user\_group` VALUES ('2', '1');

INSERT INTO `user\_group` VALUES ('1', '2');

从上面应该看出，用户ID为1同时在用户组ID为 1 和 2 中，而用户ID为 2 仅在一个用户组ID为1中。

**2、创建表对应的 JavaBean 对象**

这个例子中，我们需要在包 com.yiibai.pojo 下创建三个类，它们分别是： User.java 、Group.java 和 UserGroup.java，让我们一个一个地来看它们的代码，User.java 类的代码如下：

package com.yiibai.pojo;

import java.util.List;

/\*\*

\* @describe: User

\* @author: Yiibai

\* @version: V1.0

\* @copyright http://www.yiibai.com

\*/

public class User {

private int id;

private String username;

private String mobile;

private List<Group> groups;

public List<Group> getGroups() {

return groups;

}

public void setGroups(List<Group> groups) {

this.groups = groups;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getMobile() {

return mobile;

}

public void setMobile(String mobile) {

this.mobile = mobile;

}

}

Group.java 类的代码如下：

package com.yiibai.pojo;

import java.util.List;

/\*\*

\* @describe: Group

\* @author: Yiibai

\* @version: V1.0

\* @copyright http://www.yiibai.com

\*/

public class Group {

private int groupId;

private String groupName;

private List<User> users;

public List<User> getUsers() {

return users;

}

public void setUsers(List<User> users) {

this.users = users;

}

public int getGroupId() {

return groupId;

}

public void setGroupId(int groupId) {

this.groupId = groupId;

}

public String getGroupName() {

return groupName;

}

public void setGroupName(String groupName) {

this.groupName = groupName;

}

}

UserGroup.java 类(用户和用户组的关系映射)的代码如下：

package com.yiibai.pojo;

public class UserGroup {

private int userId;

private int groupId;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public int getGroupId() {

return groupId;

}

public void setGroupId(int groupId) {

this.groupId = groupId;

}

}

**3、配置文件**

在这一章节中，要用到的配置文件有四个，一个是 mybatis 的主配置文件：src/config/Configure.xml ，另外就是上面三个Bean类对应的配置文件，如，User.java 对应的配置文件 User.xml，等，我们先来看看 src/config/Configure.xml，其详细配置信息如下：

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

<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

<typeAliases>

<typeAlias alias="User" type="com.yiibai.pojo.User" />

<typeAlias alias="UserGroup" type="com.yiibai.pojo.UserGroup" />

<typeAlias alias="Group" type="com.yiibai.pojo.Group" />

</typeAliases>

<environments default="development">

<environment id="development">

<transactionManager type="JDBC" />

<dataSource type="POOLED">

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

<property name="url" value="jdbc:mysql://127.0.0.1:3306/yiibai" />

<property name="username" value="root" />

<property name="password" value="" />

</dataSource>

</environment>

</environments>

<mappers>

<!-- // power by http://www.yiibai.com -->

<mapper resource="com/yiibai/maper/UserMaper.xml" />

<mapper resource="com/yiibai/maper/GroupMaper.xml" />

<mapper resource="com/yiibai/maper/UserGroupMaper.xml" />

</mappers>

</configuration>

Group.java 对应的配置文件 src/com/yiibai/maper/Group.xml 的内容如下：

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

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.yiibai.maper.GroupMaper">

<parameterMap type="Group" id="parameterGroupMap">

<parameter property="groupId"/>

<parameter property="groupName"/>

</parameterMap>

<insert id="insertGroup" parameterMap="parameterGroupMap">

INSERT INTO `group` (group\_name)

VALUES(#{groupName});

</insert>

<resultMap type="Group" id="resultGroupMap\_1">

<result property="id" column="id" />

<result property="groupName" column="group\_name" />

<collection property="users" column="group\_id"

select="com.yiibai.maper.UserGroupMaper.getUsersByGroupId" />

</resultMap>

<select id="getGroup" resultMap="resultGroupMap\_1"

parameterType="int">

SELECT \*

FROM `group`

WHERE group\_id=#{id}

</select>

</mapper>

User.java 对应的配置文件 src/com/yiibai/maper/User.xml 的内容如下：

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

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.yiibai.maper.UserMaper">

<parameterMap type="User" id="parameterUserMap">

<parameter property="id"/>

<parameter property="username"/>

<parameter property="mobile"/>

</parameterMap>

<insert id="insertUser" parameterMap="parameterUserMap">

INSERT INTO user(username,mobile)

VALUES(#{username},#{mobile});

</insert>

<resultMap type="User" id="resultUser">

<result property="id" column="group\_id"/>

<result property="name" column="name"/>

<collection property="groups" column="id" select="com.yiibai.maper.UserGroupMaper.getGroupsByUserId"/>

</resultMap>

<select id="getUser" resultMap="resultUser" parameterType="int">

SELECT \*

FROM user

WHERE id=#{id}

</select>

</mapper>

UserGroup.java 对应的配置文件 src/com/yiibai/maper/UserGroup.xml 的内容如下：

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

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.yiibai.maper.UserGroupMaper">

<parameterMap type="UserGroup" id="parameterUserGroupMap">

<parameter property="userId"/>

<parameter property="groupId"/>

</parameterMap>

<insert id="insertUserGroup" parameterMap="parameterUserGroupMap">

INSERT INTO user\_group(user\_id, group\_id)

VALUES(#{userId},#{groupId})

</insert>

<!-- 根据一个用户组ID,查看这个用户组下的所有用户 -->

<resultMap type="User" id="resultUserMap\_2">

<result property="id" column="id"/>

<result property="username" column="username"/>

<result property="mobile" column="mobile"/>

</resultMap>

<select id="getUsersByGroupId" resultMap="resultUserMap\_2" parameterType="int">

SELECT u.\*, ug.group\_id

FROM user u, user\_group ug

WHERE u.id=ug.user\_id AND ug.group\_id=#{group\_id}

</select>

<!-- 根据一个用户ID,查看这个用户所对应的组-->

<resultMap type="Group" id="resultGroupMap\_2">

<result property="groupId" column="group\_id"/>

<result property="groupName" column="group\_name"/>

</resultMap>

<select id="getGroupsByUserId" resultMap="resultGroupMap\_2" parameterType="int">

SELECT g.\*, u.user\_id

FROM group g, user\_group u

WHERE g.group\_id=u.group\_id AND u.user\_id=#{user\_id}

</select>

</mapper>

**注：**在上面的配置文件中，使用到了 <association>和  <clollection>标签，关联对应的 User 类和 Group类。

**4、测试程序运行**

到这里，整个工作准备得已经差不多了，我们创建一个主类来测试上面程序，在 src 下创建一个 Main.java，代码如下：

import java.io.Reader;

import java.text.MessageFormat;

import java.util.List;

import org.apache.ibatis.io.Resources;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.ibatis.session.SqlSessionFactoryBuilder;

import com.yiibai.maper.GroupMaper;

import com.yiibai.maper.UserGroupMaper;

import com.yiibai.maper.UserMaper;

import com.yiibai.pojo.Group;

import com.yiibai.pojo.User;

import com.yiibai.pojo.UserGroup;

public class Main {

private static SqlSessionFactory sqlSessionFactory;

private static Reader reader;

static {

try {

reader = Resources.getResourceAsReader("config/Configure.xml");

sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);

} catch (Exception e) {

e.printStackTrace();

}

}

public static SqlSessionFactory getSession() {

return sqlSessionFactory;

}

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

// testAddGroup();

// testAddUser();

// testAddUserGroup();

testGetGroupAndUsers();

}

public static void testGetGroupAndUsers() {

UserGroup userGroup = new UserGroup();

SqlSession session = sqlSessionFactory.openSession();

try {

GroupMaper groupMaper = session.getMapper(GroupMaper.class);

Group group = groupMaper.getGroup(1);

System.out.println("Group => " + group.getGroupName());

List<User> users = group.getUsers();

for (User user : users) {

System.out.println("\t:" + user.getId() + "\t"

+ user.getUsername());

}

} finally {

session.close();

}

}

public static void testAddUserGroup() {

UserGroup userGroup = new UserGroup();

userGroup.setGroupId(1);

userGroup.setUserId(2);

SqlSession session = sqlSessionFactory.openSession();

try {

UserGroupMaper userGroupMaper = session

.getMapper(UserGroupMaper.class);

userGroupMaper.insertUserGroup(userGroup);

session.commit();

} finally {

session.close();

}

}

public static void testAddUser() {

// TODO Auto-generated method stub

SqlSession session = sqlSessionFactory.openSession();

try {

User user = new User();

user.setUsername("User-name-1");

user.setMobile("13838009988");

UserMaper userMaper = session.getMapper(UserMaper.class);

userMaper.insertUser(user);

session.commit();

// System.out.println(user.getGroupId());

} finally {

session.close();

}

}

public static void testAddGroup() {

// TODO Auto-generated method stub

SqlSession session = sqlSessionFactory.openSession();

try {

Group group = new Group();

group.setGroupName("用户组-1");

GroupMaper groupMapper = session.getMapper(GroupMaper.class);

groupMapper.insertGroup(group);

session.commit();

System.out.println(group.getGroupId());

} finally {

session.close();

}

}

}

运行上述程序，得出结果：

Group => Group-1

:1 yiibai

:2 User-name-1

**易百教程移动端**：请扫描本页面底部(右侧)二维码并关注微信公众号，回复："**教程**" 选择相关教程阅读或直接访问：http://m.yiibai.com 。