多对多关系映射

第一步：创建pojo—->User.java,Orders.java,Orderdetail.java,Items.java

映射思路: 将用户信息映射到user中

在user类中添加订单列表属性private List<Orders>orderList;，将用户创建的订单映射到ordersList

在Orders中添加订单明细列表属性

private List<Orderdetail>orderdetails;，将订单的明细映射到orderdetails

在Orderdetail中添加Items属性private Items items，将订单明细所对应的商品映射到Items

第二步：配置OrdersMapperCustom.xml

<mapper namespace="com.yb.mapper.OrderCustomMapper">

<resultMap type="com.yb.entity.User" id="UserAndItemsResultMap">

<!-- (1) 配置用户信息 -->

<id column="user\_id" property="id"/>

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

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

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

<!-- (2) 配置订单信息

一个用户对应多个订单，使用collection映射 -->

<collection property="orderList" ofType="com.yb.entity.Orders">

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

<result column="user\_id" property="userId"/>

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

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

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

<!-- (3) 配置订单明细信息

一个订单包括 多个订单明细 -->

<collection property="orderdetails" ofType="com.yb.entity.Orderdetail">

<id column="orderdetail\_id" property="id"/>

<result column="items\_id" property="itemsId"/>

<result column="items\_num" property="itemsNum"/>

<result column="orders\_id" property="ordersId"/>

<!-- (4) 配置商品信息

一个订单明细对应一个商品-->

<association property="items" javaType="com.yb.entity.Items">

<id column="items\_id" property="id"/>

<result column="items\_name" property="name"/>

<result column="items\_detail" property="detail"/>

<result column="items\_price" property="price"/>

</association>

</collection>

</collection>

</resultMap>

<select id="findUserAndItemsResultMap" resultMap="UserAndItemsResultMap">

SELECT

orders.\*,

USER.username,

USER.sex,

USER.address,

orderdetail.id orderdetail\_id,

orderdetail.items\_id,

orderdetail.items\_num,

orderdetail.orders\_id,

items.name items\_name,

items.detail items\_detail,

items.price items\_price

FROM

orders,

USER,

orderdetail,

items

WHERE

orders.user\_id = user.id AND orderdetail.orders\_id=orders.id

AND orderdetail.items\_id=items.id

</select>

</mapper>

第三步：在接口OrdersMapperCustom.java中添加

public List<User>findUserAndItemsResultMap()；

第四步：测试

@Test

public void testFindUserAndItemsResultMap() throws Exception {

InputStream inputStream = Resources.getResourceAsStream("mybatis-config.xml");

SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);

//得到会话sqlSession

SqlSession sqlSession=sqlSessionFactory.openSession();

//创建代理对象

OrderCustomMapper mapper = sqlSession.getMapper(OrderCustomMapper.class);

//调用mapper的方法,返回映射在pojo 中Orders(里面补充了Orderdetail orderdetails)

List<User>list=mapper.findUserAndItemsResultMap();

System.out.println(list);

sqlSession.close();

}