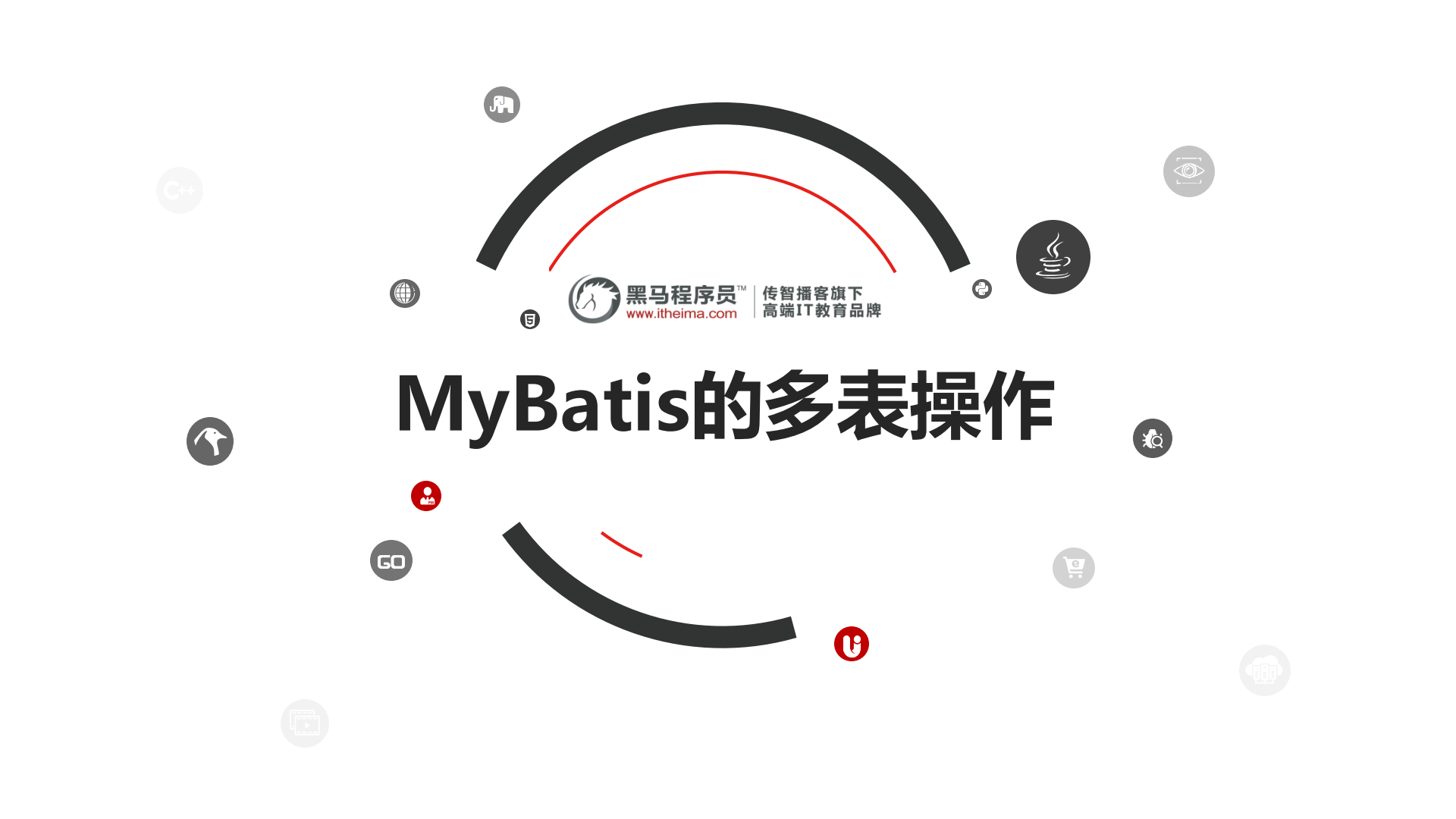


MyBatis的多表操作

目录 Contents

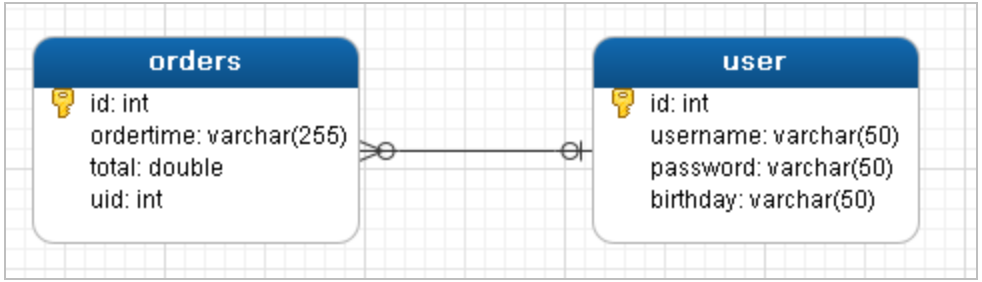
1.Mybatis多表查询

1.1 一对一查询

1. 一对一查询的模型

用户表和订单表的关系为，一个用户有多个订单，一个订单只从属于一个用户

一对一查询的需求：查询一个订单，与此同时查询出该订单所属的用户



1.Mybatis多表查询

1.1 一对一查询

2. 一对一查询的语句

对应的sql语句：select * from orders o,user u where o.uid=u.id;

查询的结果如下：

信息 | 结果1 | 概况 | 状态

id	ordertime	total	uid	id1	username	password	birthday
1	2018-12-12	3000	1	1	lucy	123	1539751863457
2	2019-12-12	4000	1	1	lucy	123	1539751863457
3	2020-12-12	5000	2	2	tom	123	1539751863457

1.Mybatis多表查询

1.1 一对一查询

3. 创建Order和User实体

```
public class Order {

    private int id;
    private Date ordertime;
    private double total;

    //代表当前订单从属于哪一个客户
    private User user;
}
```

```
public class User {

    private int id;
    private String username;
    private String password;
    private Date birthday;

}
```

1.Mybatis多表查询

1.1 一对一查询

4. 创建OrderMapper接口

```
public interface OrderMapper {
    List<Order> findAll();
}
```

1.Mybatis多表查询

1.1 一对一查询

5. 配置OrderMapper.xml

```
<mapper namespace="com.itheima.mapper.OrderMapper">
    <resultMap id="orderMap" type="com.itheima.domain.Order">
        <result column="uid" property="user.id"></result>
        <result column="username" property="user.username"></result>
        <result column="password" property="user.password"></result>
        <result column="birthday" property="user.birthday"></result>
    </resultMap>
    <select id="findAll" resultMap="orderMap">
        select * from orders o,user u where o.uid=u.id
    </select>
</mapper>
```

1.Mybatis多表查询

1.1 一对一查询

5. 配置OrderMapper.xml

其中<resultMap>还可以配置如下：

```
<resultMap id="orderMap" type="com.itheima.domain.Order">
    <result property="id" column="id"></result>
    <result property="ordertime" column="ordertime"></result>
    <result property="total" column="total"></result>
    <association property="user" javaType="com.itheima.domain.User">
        <result column="uid" property="id"></result>
        <result column="username" property="username"></result>
        <result column="password" property="password"></result>
        <result column="birthday" property="birthday"></result>
    </association>
</resultMap>
```

<!-- 手动指定字段和属性字段的对应关系-->
<!-- column:数据库的字段名-->
<!-- property:属性名称-->

<!-- 将属性user单独抽取出来-->
<!-- private User user-->
<!-- property是orders中的属性名称(user)，javaType是数据类型(User)-->

1.Mybatis多表查询

1.1 一对一查询

6. 测试结果

```
OrderMapper mapper = sqlSession.getMapper(OrderMapper.class);
List<Order> all = mapper.findAll();
for(Order order : all){
    System.out.println(order);
}
```

```
09:12:24,650 DEBUG findAll:54 - ==>  Preparing: select * from orders o,user u where o.uid=u.id
09:12:24,672 DEBUG findAll:54 - ==> Parameters:
09:12:24,699 DEBUG findAll:54 - <==      Total: 3
Order{id=1, ordertime=Wed Dec 12 00:00:00 GMT+08:00 2018, total=3000.0, user=User{id=1, username='lucy',
Order{id=2, ordertime=Thu Dec 12 00:00:00 GMT+08:00 2019, total=4000.0, user=User{id=1, username='lucy',
Order{id=3, ordertime=Sat Dec 12 00:00:00 GMT+08:00 2020, total=5000.0, user=User{id=2, username='tom',
09:12:24,706 DEBUG JdbcTransaction:54 - Resetting autocommit to true on JDBC Connection [com.mysql.jdbc.
09:12:24,706 DEBUG JdbcTransaction:54 - Closing JDBC Connection [com.mysql.jdbc.JDBC4Connection@28ac3dc3
09:12:24,706 DEBUG PooledDataSource:54 - Returned connection 682376643 to pool.
```

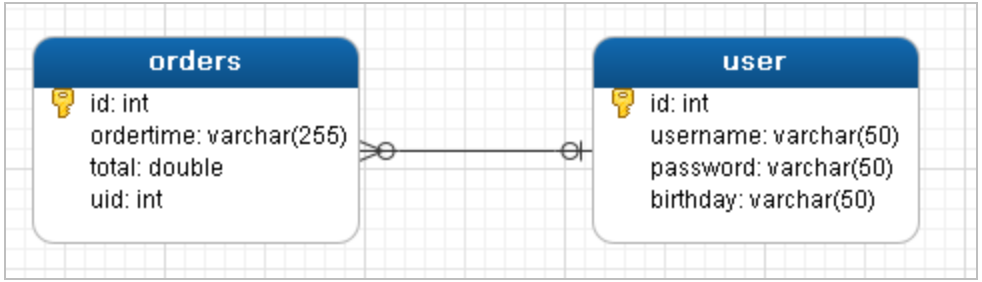
1.Mybatis多表查询

1.2 一对多查询

1. 一对多查询的模型

用户表和订单表的关系为，一个用户有多个订单，一个订单只从属于一个用户

一对多查询的需求：查询一个用户，与此同时查询出该用户具有的订单



1.Mybatis多表查询

1.2 一对多查询

2. 一对多查询的语句

对应的sql语句：select *,o.id oid from user u left join orders o on u.id=o.uid;

查询的结果如下：

信息 | 结果1 | 概况 | 状态

id	username	password	birthday	id1	ordertime	total	uid	oid
1	lucy	123	2018-12-12	1	2018-12-12	3000	1	1
1	lucy	123	2018-12-12	2	2019-12-12	4000	1	2
2	tom	123	2018-12-12	3	2020-12-12	5000	2	3
5	haohao	123	2018-12-12	(Null)	(Null)	(Null)	(Null)	(Null)

1.Mybatis多表查询

1.2 一对多查询

3. 修改User实体

```
public class Order {

    private int id;
    private Date ordertime;
    private double total;

    //代表当前订单从属于哪一个客户
    private User user;
}
```

```
public class User {

    private int id;
    private String username;
    private String password;
    private Date birthday;
    //代表当前用户具备哪些订单
    private List<Order> orderList;
}
```

1.Mybatis多表查询

1.2 一对多查询

4. 创建UserMapper接口

```
public interface UserMapper {
    List<User> findAll();
}
```

1.Mybatis多表查询

1.2 一对多查询

5. 配置UserMapper.xml

```
<mapper namespace="com.itheima.mapper.UserMapper">
    <resultMap id="userMap" type="com.itheima.domain.User">
        <result column="id" property="id"></result>
        <result column="username" property="username"></result>
        <result column="password" property="password"></result>
        <result column="birthday" property="birthday"></result>
        <collection property="orderList" ofType="com.itheima.domain.Order">
            <result column="oid" property="id"></result>
            <result column="ordertime" property="ordertime"></result>
            <result column="total" property="total"></result>
        </collection>
    </resultMap>
    <select id="findAll" resultMap="userMap">
        select *,o.id oid from user u left join orders o on u.id=o.uid
    </select>
</mapper>
```

1.Mybatis多表查询

1.2 一对多查询

6. 测试结果

```
UserMapper mapper = sqlSession.getMapper(UserMapper.class);
List<User> all = mapper.findAll();
for(User user : all){
    System.out.println(user.getUsername());
    List<Order> orderList = user.getOrderList();
    for(Order order : orderList){
        System.out.println(order);
    }
    System.out.println("----------------------------------");
}
```

1.Mybatis多表查询

1.2 一对多查询

6. 测试结果

```
10:02:27,817 DEBUG findAll:54 - ==>  Preparing: select *,o.id oid from user u left join orders o on u.id=o.uid
10:02:27,843 DEBUG findAll:54 - ==> Parameters:
10:02:27,865 DEBUG findAll:54 - <==      Total: 4
lucy
Order{id=1, ordertime=Wed Dec 12 00:00:00 GMT+08:00 2018, total=3000.0, user=null}
Order{id=2, ordertime=Thu Dec 12 00:00:00 GMT+08:00 2019, total=4000.0, user=null}
----------------------------------
tom
Order{id=3, ordertime=Sat Dec 12 00:00:00 GMT+08:00 2020, total=5000.0, user=null}
----------------------------------
haohao
----------------------------------
10:02:27,868 DEBUG JdbcTransaction:54 - Resetting autocommit to true on JDBC Connection [com.mysql.jdbc.JDBC4Co
10:02:27,869 DEBUG JdbcTransaction:54 - Closing JDBC Connection [com.mysql.jdbc.JDBC4Connection@289d1c02]
10:02:27,869 DEBUG PooledDataSource:54 - Returned connection 681384962 to pool.
```

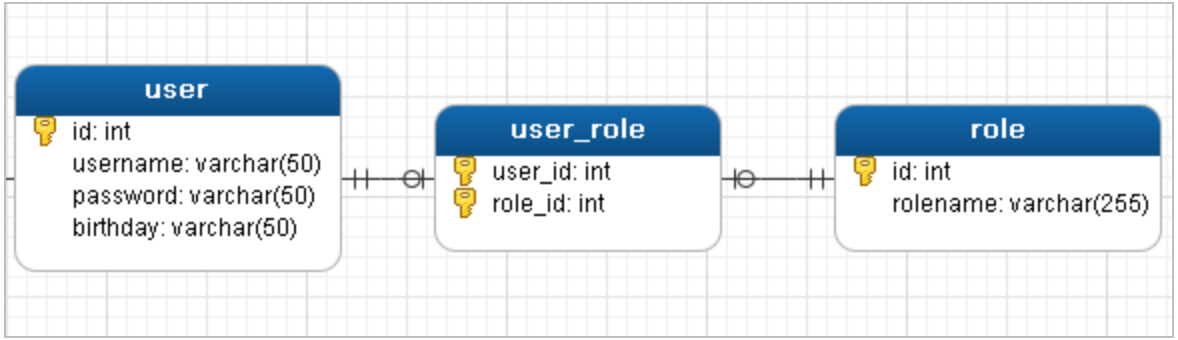
1.Mybatis多表查询

1.3 多对多查询

1. 多对多查询的模型

用户表和角色表的关系为，一个用户有多个角色，一个角色被多个用户使用

多对多查询的需求：查询用户同时查询出该用户的所有角色



1.Mybatis多表查询

1.3 多对多查询

2. 多对多查询的语句

对应的sql语句：select u.*,r.*,r.id rid from user u left join user_role ur on u.id=ur.user_id inner join role r on ur.role_id=r.id;

查询的结果如下：

信息 | 结果1 | 概况 | 状态

id	username	password	birthday	id1	rolename
1	lucy	123	2018-12-12	1	CEO
1	lucy	123	2018-12-12	2	CFO
2	tom	123	2018-12-12	2	CFO
2	tom	123	2018-12-12	3	COO

1.Mybatis多表查询

1.3 多对多查询

3. 创建Role实体，修改User实体

```
public class User {
    private int id;
    private String username;
    private String password;
    private Date birthday;
    //代表当前用户具备哪些订单
    private List<Order> orderList;
    //代表当前用户具备哪些角色
    private List<Role> roleList;
}
```

```
public class Role {

    private int id;
    private String rolename;

}
```

1.Mybatis多表查询

1.3 多对多查询

4. 添加UserMapper接口方法

```
List<User> findAllUserAndRole();
```

1.Mybatis多表查询

1.3 多对多查询

5. 配置UserMapper.xml

```
<resultMap id="userRoleMap" type="com.itheima.domain.User">
    <result column="id" property="id"></result>
    <result column="username" property="username"></result>
    <result column="password" property="password"></result>
    <result column="birthday" property="birthday"></result>
    <collection property="roleList" ofType="com.itheima.domain.Role">
        <result column="rid" property="id"></result>
        <result column="rolename" property="rolename"></result>
    </collection>
</resultMap>
<select id="findAllUserAndRole" resultMap="userRoleMap">
    select u.*,r.*,r.id rid from user u left join user_role ur on
u.id=ur.user_id
    inner join role r on ur.role_id=r.id
</select>
```

1.Mybatis多表查询

1.3 多对多查询

6. 测试结果

```
UserMapper mapper = sqlSession.getMapper(UserMapper.class);
List<User> all = mapper.findAllUserAndRole();
for(User user : all){
    System.out.println(user.getUsername());
    List<Role> roleList = user.getRoleList();
    for(Role role : roleList){
        System.out.println(role);
    }
    System.out.println("----------------------------------");
}
```

1.Mybatis多表查询

1.3 多对多查询

6. 测试结果

```
10:34:36,884 DEBUG findAllUserAndRole:54 - ==>  Preparing: select u.*,r.*,r.id rid from user u left
10:34:36,903 DEBUG findAllUserAndRole:54 - ==> Parameters:
lucy
Role{id=1, rolename='CEO'}
Role{id=2, rolename='CFO'}
----------------------------------
tom
Role{id=2, rolename='CFO'}
Role{id=3, rolename='COO'}
----------------------------------
10:34:36,937 DEBUG findAllUserAndRole:54 - <==      Total: 4
10:34:36,939 DEBUG JdbcTransaction:54 - Resetting autocommit to true on JDBC Connection [com.mysql.
```

1.Mybatis多表查询

1.4 知识小结

MyBatis多表配置方式：

一对一配置：使用<resultMap>做配置

一对多配置：使用<resultMap>+<collection>做配置

多对多配置：使用<resultMap>+<collection>做配置

传智播客旗下高端IT教育品牌