

collection

关联查询的封装，一对多，JavaBean 属性包含 JavaBean 的集合

使用 collection

```
public Dept getDeptById(Integer id);
```

```
<resultMap type="com.bean.Dept" id="MyDept">
  <id column="did" property="id"/>
  <result column="dname" property="name"/>
  <!-- collection 定义关联集合类型的属性封装规则
    ofType 指定集合里面元素的类型 -->
  <collection property="myUsers" ofType="com.bean.MyUser">
    <!-- 定义集合中元素的封装规则 -->
    <id column="id" property="id"/>
    <result column="name" property="name"/>
    <result column="age" property="age"/>
  </collection>
</resultMap>
<select id="getDeptById" resultMap="MyDept">
  SELECT m.id,m.name,m.age,m.did,d.name AS dname FROM myuser m,dept d
  WHERE m.did = d.id AND d.id = #{id}
</select>
```

关闭懒加载，使用二次查询

```
public Dept getDeptByIdStep(Integer did);
```

```
<!-- Collection 分段查询 -->
<resultMap type="com.bean.Dept" id="MyDeptStep">
  <id column="id" property="id"/>
  <result column="name" property="name"/>
  <collection property="myUsers"
select="com.dao.MyUserMapper.selectMyUserByDid"
    column="{did=id}" fetchType="eager"/>
  <!-- column 要处理复合主键或传递多个值过去：可以将多列的值封装 Map 传递，指定
多个列名通过 column="{prop1=col1,prop2=col2}"
    语法来传递给嵌套查询语句。这会引入 prop1 和 prop2 以参数对象形式来设置给目标
嵌套查询语句
    fetchType="lazy": 是否延迟加载，优先级高于全局配置，lazy: 延迟，eager: 立即
-->
</resultMap>
<select id="getDeptByIdStep" resultMap="MyDeptStep">
  select * from dept where id = #{id}
</select>
```

```
public List<MyUser> selectMyUserByDid(Integer did);
```

```
<select id="selectMyUserByDid" resultType="myUser">  
  select * from myuser where dId = #{did}  
</select>
```