

## association

```
<resultMap type="com.bean.MyUser" id="myUserResultMap">
  <id column="id" property="id"/>
  <result column="name" property="name"/>
  <result column="age" property="age"/>
  <!-- association 指定联合的 javaBean 对象
    property="dept": 指定哪个属性是联合的对象
    javaType: 指定这个属性对象的类型[不能省略] -->
  <association property="dept" javaType="com.bean.Dept">
    <id column="did" property="id"/>
    <result column="dname" property="name"/>
  </association>
</resultMap>
```

```
<select id="selectMyUserById" resultMap="myUserResultMap"
parameterType="integer">
  SELECT m.id, m.name, m.age, m.did, d.name AS dname FROM myuser m,dept d
  WHERE m.did = d.id AND m.id = #{id}
</select>
```

使用 association 二次查询，即有两条 SQL

```
<?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">
<!-- namespace 对应接口文件的全路径 -->
<mapper namespace="com.dao.DeptMapper">
  <select id="selectDeptById" resultType="dept" parameterType="string">
    select * from dept where id = #{id}
  </select>
```

```
  <resultMap id="myUserResultMap" type="com.bean.MyUser">
    <id column="id" property="id"/>
    <result column="name" property="name"/>
    <result column="age" property="age"/>
    <!-- association 定义关联对象的封装规则
      select: 当前属性是调用 select 指定的方法查出的结果
      column: 将哪一列的值传给这个方法 -->
    <association property="dept" select="com.dao.DeptMapper.selectDeptById"
column="did"/>
  </resultMap>
<select id="selectMyUserById" resultMap="myUserResultMap"
parameterType="integer">
  SELECT * FROM myuser WHERE id = #{id}
</select>
</mapper>
```

开启懒加载：在没有使用 Dept 的属性时，则只会加载 MyUser 的属性。即只会发送一条 SQL 语句，要使用 Dept 属性时才会发送第二条 SQL。不会一次性发送两条 SQL

<!-- 延迟加载的全局开关。当开启时，所有关联对象都会延迟加载。特定关联关系中可通过设置fetchType属性来覆盖该项的开关状态。默认false -->

<setting name="lazyLoadingEnabled" value="true"/>

<!-- 当开启时，任何方法的调用都会加载该对象的所有属性。否则，每个属性会按需加载。默认false (true in ≤3.4.1) -->

<setting name="aggressiveLazyLoading" value="false"/>