

# Mybatis\_day3

## 1. Mybatis处理一对一关联关系

### 1.1. 数据库

```
--用户表
create table t_user(
    id varchar2(40) primary key,
    name varchar2(60),
    age number(2),
);
--身份证表
create table t_card(
    id varchar2(40) primary key,
    address varchar2(100),
    userid varchar2(40) references t_user(id)
);
```

### 1.2. 实体类

```
/**
 * 用户
 */
public class Person {
    private String id;
    private String name;
    private Date bir;
    private Card card;
}

/**
 * 身份证实体
 */
public class Card {
    private String id;
    private String name;
```

注意：实体类中使用关系属性体现关系

### 1.3. 根据用户查找用户的身份信息

```
<resultMap type="zpark.entity.Person" id="personMap">
    <id column="id" property="id"/>
    <!-- association:标签用来处理一对一 -->
    <association property="card" javaType="zpark.entity.Card" >
        <id column="cid" property="id"/>
        <result column="cname" property="name"/>
    </association>
</resultMap>
```

注意：根据用户的身份信息，查找用户的身份证，sql语句需要使用表连接查询，返回值必须使用resultMap处理，不能再使用resultType处理

## 2. Mybatis处理一对多关联关系

### 2.1. 数据库模型

```
--部门
create table dept(
  id integer primary key,
  dcode varchar2(12),
  dname varchar2(12)
);
--员工
create table employee(
  id integer primary key,
  name varchar2(12),
  salary number(12,2),
  birthday date,
  did integer references dept(id)
);
```

### 2.2. 实体类实现

```
--部门
create table dept(
  id integer primary key,
  dcode varchar2(12),
  dname varchar2(12)
);
--员工
create table employee(
  id integer primary key,
  name varchar2(12),
  salary number(12,2),
  birthday date,
  did integer references dept(id)
);
```

### 2.3. 通过部门找员工

```
public Dept query(String id);

<!-- 使用resultMap -->
<resultMap type="com.baizhi.entity.Dept" id="deptMap">
  <id column="id" property="id"/>
  ....//维护其他普通属性
  <collection property="emps" ofType="com.baizhi.entity.Emp" javaType="list">
    <id column="id" property="id"/>
    <result column="name" property="name"/>
    ....//维护其他普通属性
  </collection>
</resultMap>

<select id="query" parameterType="String"
  resultMap="deptMap">
  select d.id,d.dcode,d.dname,e.id,e.name,e.salary,e.birthday
  from dept d
  left join employee e
  on d.id = e.did
  where d.id = #{id}
</select>
```

关系属性名

关系属性类型

关系属性类型

必须两张表的数据

### 2.4. 通过员工找部门

```
public Emp query(String id);

<resultMap type="com.baizhi.entity.Emp" id="empMap">
  <id column="id" property="id"/>
  <association property="dept" javaType="com.baizhi.enetity.Dept">
    <id column="did" property="id"/>
  </association>
</resultMap>

<select id="query" parameterType="string" resultMap="empMap">
  select d.id,d.dcode,d.dname,e.id,e.name,e.salary,e.birthday
  from dept d
  left join employee e
  on d.id = e.did
  where e.id = #{id}
</select>
```

注意：在配置一对一维护关系 使用association属性，维护一对多使用collection属性

### 3. Mybatis处理多对多关联关系

- 3.1. 多对多在数据库的设计里也是使用两个一对多实现的
- 3.2. 实体类在开发时两边都是集合类型的关系属性
- 3.3. mapper文件在维护时都使用 <collection 标签>