

## Mapper.xml 详解

- resultType: 结果类型
  - 基本数据类型
  - 包装类
  - String类型, 根据 id 查询 User 的 username

UserRepository

```
//根据 id 查询 User 的 username  
public String findNameById(int id);
```

UserRepository.xml

```
<select id="findNameById" parameterType="int"  
resultType="java.lang.String">  
    select username from t_user where id = #{id}  
</select>
```

- JavaBean, 通过 id 查询 User

UserRepository

```
//根据 id 查询 User  
public User findById(int id);
```

UserRepository.xml

```
<select id="findById" parameterType="int"  
resultType="com.southwind.entity.User">  
    select * from t_user where id = #{id}  
</select>
```

## 及联查询

- 一对多

创建实体类

Student

```
package com.southwind.entity;  
  
public class Student {
```

```

private int id;
private String name;
private Classes classes;

public Classes getClasses() {
    return classes;
}

public void setClasses(Classes classes) {
    this.classes = classes;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}
}

```

## Classes

```

package com.southwind.entity;

public class Classes {
    private int id;
    private String name;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }
}

```

```

    public void setName(String name) {
        this.name = name;
    }
}

```

## SQL

```

select * from student,classes where student.cid = classes.id and student.id
= 1;

```

MyBatis 框架在解析结果集的时候，将结果集与某个实体类进行映射，结果集中的字段与实体类的成员变量进行映射，根据字段名与成员变量相等的规则来进行映射，能映射起来的成员变量就会完成赋值，无法映射的成员变量不会赋值。

## StudentRepository

```

public interface StudentRepository {
    public Student findById(int id);
}

```

## StudentRepository.xml

```

<?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">
<mapper namespace="com.southwind.repository.StudentRepository">

    <resultMap id="studentMap" type="com.southwind.entity.Student">
        <id property="id" column="sid"></id>
        <result property="name" column="sname"></result>
        <!-- 映射 classes -->
        <association property="classes"
javaType="com.southwind.entity.Classes">
            <id property="id" column="cid"></id>
            <result property="name" column="cname"></result>
        </association>
    </resultMap>

    <select id="findById" parameterType="int" resultMap="studentMap">
        select student.id sid,student.name sname,classes.id cid,classes.name
cname from student,classes where student.cid = classes.id and student.id = #
{id}
    </select>

</mapper>

```

## ClassesRepository

```
package com.southwind.repository;

import com.southwind.entity.Classes;

public interface ClassesRepository {
    public Classes findById(int id);
}
```

## ClassesRepository.xml

```
<?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">
<mapper namespace="com.southwind.repository.ClassesRepository">

    <resultMap id="classesMap" type="com.southwind.entity.Classes">
        <id property="id" column="cid"></id>
        <result property="name" column="cname"></result>
        <!-- 映射 students -->
        <collection property="students"
ofType="com.southwind.entity.Student">
            <id property="id" column="sid"></id>
            <result property="name" column="sname"></result>
        </collection>
    </resultMap>

    <select id="findById" parameterType="int" resultMap="classesMap">
        select classes.id cid,classes.name cname,student.id sid,student.name
sname from classes,student where classes.id = student.cid and classes.id = #
{id}
    </select>

</mapper>
```

需要注意的是：association 标签，通过设置 javaType 属性，映射实体类，collection 标签，通过设置 ofType 属性，映射实体类。

- 多对多

多对多其实是双向的一对多关系，借助于一张中间表，中间表和两张目标表都是一对多关系，间接完成多对多的关系映射。

一个 Customer 可以对应多个 Goods，一个 Goods 也可以对应多个 Customer，所以双方都是用 collection 标签进行映射。

## Customer

```

package com.southwind.entity;

import java.util.List;

public class Customer {
    private int id;
    private String name;
    private List<Goods> goods;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public List<Goods> getGoods() {
        return goods;
    }

    public void setGoods(List<Goods> goods) {
        this.goods = goods;
    }
}

```

## Goods

```

package com.southwind.entity;

import java.util.List;

public class Goods {
    private int id;
    private String name;
    private List<Customer> customers;

    public int getId() {
        return id;
    }
}

```

```

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public List<Customer> getCustomers() {
        return customers;
    }

    public void setCustomers(List<Customer> customers) {
        this.customers = customers;
    }
}

```

## CustomerRepository

```

package com.southwind.repository;

import com.southwind.entity.Customer;

public interface CustomerRepository {
    public Customer findById(int id);
}

```

## CustomerRepository.xml

```

<?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">
<mapper namespace="com.southwind.repository.CustomerRepository">

    <resultMap id="customerMap" type="com.southwind.entity.Customer">
        <id property="id" column="cid"></id>
        <result property="name" column="cname"></result>
        <!-- 映射 goods -->
        <collection property="goods" ofType="com.southwind.entity.Goods">
            <id property="id" column="gid"></id>
            <result property="name" column="gname"></result>
        </collection>
    </resultMap>

```

```

        <select id="findById" parameterType="int" resultMap="customerMap">
            select c.id cid,c.name cname,g.id gid,g.name gname
            from customer c,goods g,customer_goods cg where c.id = cg.cid and g.id =
            cg.gid and c.id = #{id}
        </select>

</mapper>

```

## GoodsRepository

```

package com.southwind.repository;

import com.southwind.entity.Goods;

public interface GoodsRepository {
    public Goods findById(int id);
}

```

## GoodsRepository.xml

```

<?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">
<mapper namespace="com.southwind.repository.GoodsRepository">

    <resultMap id="goodsMap" type="com.southwind.entity.Goods">
        <id property="id" column="gid"></id>
        <result property="name" column="gname"></result>
        <!-- 映射 customers -->
        <collection property="customers"
ofType="com.southwind.entity.Customer">
            <id property="id" column="cid"></id>
            <result property="name" column="cname"></result>
        </collection>
    </resultMap>

    <select id="findById" parameterType="int" resultMap="goodsMap">
        select c.id cid,c.name cname,g.id gid,g.name gname
        from customer c,goods g,customer_goods cg where c.id = cg.cid and g.id =
        cg.gid and g.id = #{id}
    </select>

</mapper>

```

## Student 的 CRUD

## StudentRepository

```
package com.southwind.repository;

import com.southwind.entity.Student;

public interface StudentRepository {
    public Student findById(int id);
    public void save(Student student);
    public void update(Student student);
    public void delete(Student student);
}
```

## StudentRepository.xml

```
<?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">
<mapper namespace="com.southwind.repository.StudentRepository">

    <resultMap id="studentMap" type="com.southwind.entity.Student">
        <id property="id" column="sid"></id>
        <result property="name" column="sname"></result>
        <!-- 映射 classes -->
        <association property="classes"
javaType="com.southwind.entity.Classes">
            <id property="id" column="cid"></id>
            <result property="name" column="cname"></result>
        </association>
    </resultMap>

    <select id="findById" parameterType="int" resultMap="studentMap">
        select student.id sid,student.name sname,classes.id cid,classes.name
cname from student,classes where student.cid = classes.id and student.id = #
{id}
    </select>

    <insert id="save" parameterType="com.southwind.entity.Student">
        insert into student(name,cid) values({name},{classes.id})
    </insert>

    <update id="update" parameterType="com.southwind.entity.Student">
        update student set name = {name},cid = {classes.id} where id = #
{id}
    </update>

    <delete id="delete" parameterType="com.southwind.entity.Student">
        delete from student where id = #{id}
```



```
</delete>
```

```
</mapper>
```