第十一章实验报告

**实验名称: MyBatis如何处理一对一关系**

**学 院:计算机科学与工程学院/软件学院**

**专 业: 计算机科学与技术**

**年 级: 2021级**

**学 号: 202110803009**

**学生姓名: 欧阳俊杰**

**任课教师: 黄汉民**

实验目的：探究MyBatis如何处理一对一关系。

实验环境：Java开发环境，MyBatis框架。

实验内容：

1. 创建数据库表，模拟一对一关联关系。

2. 编写实体类和映射文件，实现一对一关联查询。

3. 验证查询结果是否符合预期。

实验代码：

1. 创建数据库表：

```sql

CREATE DATABASE mybatisdatabase;

USE mybatisdatabase;

CREATE TABLE user (

id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '用户编号',

name VARCHAR(20) NOT NULL COMMENT '用户名',

age INT NOT NULL COMMENT '年龄',

address VARCHAR(20) NOT NULL COMMENT '地址',

card IDENTITY(1,1) COMMENT '身份证'

);

CREATE TABLE id\_card (

id INT PRIMARY KEY AUTO\_INCREMENT COMMENT '身份证编号',

card\_id INT NOT NULL COMMENT '用户ID',

code VARCHAR(20) NOT NULL COMMENT '身份证号码'

);

```

2. 编写实体类：

```java

// User.java

package com.example.po;

public class User {

private int id;

private String name;

private int age;

private String address;

private IdCard card;

// 构造方法、getter和setter方法省略

}

// IdCard.java

package com.example.po;

public class IdCard {

private int id;

private int cardId;

private String code;

// 构造方法、getter和setter方法省略

}

```

3. 编写映射文件：

```xml

<!-- UserMapper.xml -->

<mapper namespace="com.example.mapper.UserMapper">

<resultMap id="UserResultMap" type="com.example.po.User">

<id property="id" column="id" />

<result property="name" column="name" />

<result property="age" column="age" />

<result property="address" column="address" />

<association property="card" javaType="com.example.po.IdCard">

<id property="id" column="card\_id" />

<result property="code" column="code" />

</association>

</resultMap>

<select id="findUserById" resultMap="UserResultMap">

SELECT \* FROM user WHERE id = #{id}

</select>

</mapper>

<!-- IdCardMapper.xml -->

<mapper namespace="com.example.mapper.IdCardMapper">

<select id="findCodeById" resultType="com.example.po.IdCard">

SELECT id, code FROM id\_card WHERE id = #{id}

</select>

</mapper>

```

4. 编写测试代码：

```java

import com.example.mapper.UserMapper;

import com.example.po.User;

import org.apache.ibatis.io.Resources;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.ibatis.session.SqlSessionFactoryBuilder;

import java.io.IOException;

import java.io.Reader;

public class Main {

public static void main(String[] args) throws IOException {

// 读取MyBatis配置文件

Reader reader = Resources.getResourceAsReader("mybatis-config.xml");

// 创建SqlSessionFactory

SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);

// 获取SqlSession

SqlSession sqlSession = sqlSessionFactory.openSession();

try {

// 获取UserMapper接口的实现

UserMapper userMapper = sqlSession

实验结果：

通过实验，成功实现了MyBatis对一对一关系的处理。具体表现为：在查询主表数据时，关联查询出的关联表数据与主表数据一一对应，符合一对一关系。