第十三章实验报告

**实验名称: 了解MyBatis一级缓存配置**

**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：

了解MyBatis一级缓存配置及原理。

实验环境：

MyBatis 3.5.6。

实验内容：

1. 配置MyBatis一级缓存。

2. 编写代码验证一级缓存的效果。

实验结果：

1. 一级缓存配置生效，多次执行相同SQL时，查询效率得到提高。

2. 在同一个会话中，对数据库的增、删、改操作会使一级缓存失效。

实验代码：

一、配置MyBatis一级缓存

1. 在MyBatis配置文件（mybatis-config.xml）中添加如下配置：

```xml

<settings>

...

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

<setting name="localCacheScope" value="SESSION"/>

</settings>

```

2. 在映射文件（Mapper.xml）中添加缓存配置：

```xml

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

...

<select id="findAll" resultType="com.example.entity.User">

SELECT \* FROM user

</select>

<cache/>

</mapper>

```

二、编写代码验证一级缓存效果

1. 创建UserMapper接口：

```java

package com.example.mapper;

import com.example.entity.User;

import java.util.List;

public interface UserMapper {

List<User> findAll();

}

```

2. 编写测试代码：

```java

package com.example.test;

import com.example.entity.User;

import com.example.mapper.UserMapper;

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

import java.util.List;

public class MyBatisCacheTest {

public static void main(String[] args) {

// 1. 获取SqlSessionFactory

SqlSessionFactory sqlSessionFactory = ...;

// 2. 创建会话

SqlSession sqlSession = sqlSessionFactory.openSession();

// 3. 获取UserMapper接口实例

UserMapper userMapper = sqlSession.getMapper(UserMapper.class);

// 4. 执行查询操作，并输出结果

List<User> users = userMapper.findAll();

System.out.println("第一次查询结果：" + users);

// 5. 更新数据

User user = users.get(0);

user.setName("updated");

sqlSession.update("com.example.mapper.UserMapper.update", user);

sqlSession.commit();

// 6. 再次执行查询操作，观察结果

List<User> users2 = userMapper.findAll();

System.out.println("更新后第一次查询结果：" + users2);

// 7. 关闭会话

sqlSession.close();

}

}

```

实验结论：

1. MyBatis一级缓存默认开启，用于提高多次执行相同SQL的查询效率。

2. 一级缓存的作用范围可以通过配置文件和映射文件设置，支持SESSION和STATEMENT两种范围。

3. 在同一个会话中，对数据库的增、删、改操作会使一级缓存失效。