

武昌理工学院人工智能学院

实验报告

课程名称	Java EE		指导教师	方洁
实验名称	实验七 MyBatis 的一对多内嵌查询		实验地点	Home
专业班级	计科 1702		实验日期	2020/5/19
学 号	20174221054			
姓 名	郑雷明			
成 绩				

一、实验目的和实验内容

实验目的：

- 1.练习使用 MySql 数据库；
- 2.练习使用持久层框架；
- 3.练习使用 MyBatis 的一对多内嵌查询

实验内容：

用一对多内嵌方式查询所有部门下的所有员工的所有信息操作

二、实验结果（含程序、数据记录及分析和实验总结等，可附页）

数据库表：

username	password	deptid
张龙	zlm1	3
王朝	zlm2	1
赵虎	zlm3	2
马汉	zlm4	2

id	deptname
1	开发部
2	宣传部
3	测试部

Client.java

```

1 package client;
2
3 import java.io.*;
10
11 public class Client {
12     public static void main(String args[]) throws IOException {
13         InputStream in = Resources.getResourceAsStream("mybatis-
14         config.xml");
15         SqlSessionFactory factory = new SqlSessionFactoryBuilder().build
16         (in);
17         SqlSession sqlSession = factory.openSession();
18         List<Department> list = sqlSession.selectList("findAllEmployees");
19         System.out.println("查询所有部门下的所有员工的所有信息（内嵌子查询方式）：");
20         for (Department dept : list) {
21             System.out.println("部门: "+dept.getDeptname());
22             for (Employee emp : dept.getEmp())
23                 System.out.println("员工部门号: "+emp.getDeptid()+" 员工姓
24                 名: "+emp.getUsername()+" 员工密码: "+emp.getPassword());
25         }
26     }
27 }

```

Department.java

```

1 package domain;
2
3 import java.util.List;
4
5 public class Department {
6     private int id;
7     private String deptname;
8     private List<Employee> emp;
9     public int getId() {
10         return id;
11     }
12     public void setId(int id) {
13         this.id = id;
14     }
15     public String getDeptname() {
16         return deptname;
17     }
18     public void setDeptname(String deptname) {
19         this.deptname = deptname;
20     }
21     public List<Employee> getEmp() {
22         return emp;
23     }
24     public void setEmp(List<Employee> emp) {
25         this.emp = emp;
26     }
27 }

```

Employee.java

```
1 package domain;
2 public class Employee {
3     private String username;
4     private String password;
5     private int deptid;
6     public String getUsername() {
7         return username;
8     }
9     public void setUsername(String username) {
10         this.username = username;
11     }
12     public String getPassword() {
13         return password;
14     }
15     public void setPassword(String password) {
16         this.password = password;
17     }
18     public int getDeptid() {
19         return deptid;
20     }
21     public void setDeptid(int deptid) {
22         this.deptid = deptid;
23     }
24 }
```

UserMapper.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="/">
6     <resultMap type="domain.Department" id="department">
7         <id property="id" column="id" />
8         <result property="deptname" column="deptname" />
9         <collection property="emp" javaType="ArrayList"
10             column="id" select="findEmpById"></collection>
11     </resultMap>
12     <resultMap type="domain.Employee" id="employee">
13         <id property="username" column="username" />
14         <result property="password" column="password" />
15         <result property="deptid" column="deptid" />
16     </resultMap>
17     <select id="findAllEmployees" resultMap="department">
18         select * from department
19     </select>
20     <select id="findEmpById" resultMap="employee"
21         parameterType="int">
22         select * from employee
23         where deptid=#{id}
24     </select>
25 </mapper>

```

运行结果:

```

[UMS]DEBUG [main] - Logging initialized using 'class org.apache.ibatis.logging.log4j.Log4jImpl' adapter.
[UMS]DEBUG [main] - PooledDataSource forcefully closed/removed all connections.
[UMS]DEBUG [main] - PooledDataSource forcefully closed/removed all connections.
[UMS]DEBUG [main] - PooledDataSource forcefully closed/removed all connections.
[UMS]DEBUG [main] - PooledDataSource forcefully closed/removed all connections.
[UMS]DEBUG [main] - Opening JDBC Connection
[UMS]DEBUG [main] - Created connection 876563773.
[UMS]DEBUG [main] - Setting autocommit to false on JDBC Connection [com.mysql.jdbc.JDBC4Connection@343f4d3d]
[UMS]DEBUG [main] - ==> Preparing: select * from department
[UMS]DEBUG [main] - ==> Parameters:
[UMS]DEBUG [main] - =====> Preparing: select * from employee where deptid=?
[UMS]DEBUG [main] - =====> Parameters: 1(Integer)
[UMS]DEBUG [main] - <===== Total: 1
[UMS]DEBUG [main] - =====> Preparing: select * from employee where deptid=?
[UMS]DEBUG [main] - =====> Parameters: 2(Integer)
[UMS]DEBUG [main] - <===== Total: 2
[UMS]DEBUG [main] - =====> Preparing: select * from employee where deptid=?
[UMS]DEBUG [main] - =====> Parameters: 3(Integer)
[UMS]DEBUG [main] - <===== Total: 1
[UMS]DEBUG [main] - <===== Total: 3
[UMS]DEBUG [main] - <===== Total: 3
查询所有部门下的所有员工的所有信息(内嵌子查询方式):
部门: 开发部
员工部门号: 1 员工姓名: 王朝 员工密码: z1m2
部门: 宣传部
员工部门号: 2 员工姓名: 赵虎 员工密码: z1m3
员工部门号: 2 员工姓名: 马汉 员工密码: z1m4
部门: 测试部
员工部门号: 3 员工姓名: 张龙 员工密码: z1m1
[UMS]DEBUG [main] - Resetting autocommit to true on JDBC Connection [com.mysql.jdbc.JDBC4Connection@343f4d3d]
[UMS]DEBUG [main] - Closing JDBC Connection [com.mysql.jdbc.JDBC4Connection@343f4d3d]
[UMS]DEBUG [main] - Returned connection 876563773 to pool.

```

实验总结：（实验结果和存在的问题及解决方法）

实验结果：完成。这次实验按进度应该用 `association` 做多对一内嵌查询方式，可是是 `collection`，但差不多的。用 `column` 作为参数，`select` 作为子查询。

存在问题：子查询和普通的 SQL 联合查询还是有区别的，看了半天才发现用错了 `Type`

解决方法：不停地测试

评阅意见：

指导教师签名：

实验结果（附页）

实验结果（附页）