

哈尔滨工业大学

<<数据库系统>>

实验报告之二

(2019 年度春季学期)

姓名：	1160300814
学号：	姜思琪
学院：	计算机学院
教师：	高宏

实验二 使用高级语言操作 MySQL 数据库

一、实验目的

学会使用高级语言访问 MySQL 数据库，并进行查询。

二、实验环境

MySQL 关系数据库管理系统、eclipse 编译器。

本次实验主要利用 JAVA 语言访问 MySQL 数据库。

三、实验过程及结果

在上次上机实验课建立的 COMPANY 数据库上，用 JAVA 语言编写程序，完成 9 个查询为如下：

本次实验用 Contact.java 类实现连接数据库，在 Query.java 中实现查询。输入格式：数字和查询参数用空格隔开；查询参数均在同一个双引号内，用逗号隔开。符号均是英文字符。例如，7 “P1,P2” 。具体输入输出示例和查询代码如下如下：

1: 参加了项目编号为%PNO%的项目的员工号，其中%PNO%为 C 语言编写的程序的输入参数；

程序代码：

```

// 参加了项目编号为%PNO%的员工的员工号
public static void EmployeeNum(String[] params) {
    int num = 0;
    sql = "select ESSN from " + wor + " where PNO = \"" + params[0] + "\"";
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        // System.out.println("+-----+");
        System.out.println("参加了项目编号为" + params[0] + "的员工的员工号");
        System.out.println("|    ESSN    |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println("| " + ret.getString("ESSN") + " |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

查询结果:

```

查询开始: 例如 n "x,y"
MengSQL>1 "P1"
参加了项目编号为P1的员工的员工号
|    ESSN    |
-----
| 12301 |
| 12302 |
| 12303 |
| 12304 |
| 12305 |
| 12306 |
| 12324 |
| 12344 |
| 12350 |
-----
查询出9行数据!

```

2: 参加了项目名为%PNAME%的员工名字, 其中%PNAME%为 C 语言编写的程序的输入参数;

程序代码:

```
// 参加了项目名为PNAME%的员工名字
public static void EmployeeName(String[] params) {
    int num = 0;
    sql = "select ENAME from " + emp + " natural join WORKS_ON natural join PROJECT where PNAME = \"" + params[0]
        + "\""; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        // System.out.println("-----");
        System.out.println("参加了项目名为" + params[0] + "的员工名字");
        System.out.println(" | ENAME |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println(" | " + ret.getString("ENAME") + " |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:

```
查询开始: 例如 n "x,y"
MengSQL>2 "SQLProject"
参加了项目名为SQLProject的员工名字
 | ENAME |
-----
 | 张红 |
 | 张二 |
 | 张三 |
 | 张四 |
 | 张五 |
 | 张六 |
-----
查询出6行数据!
```

3: 在%DNAME%工作的所有工作人员的名字和地址, 其中%DNAME%为 C 语言编写的程序的输入参数;

程序代码:

```
// 在%DNAME%工作的所有工作人员的名字和地址
public static void EmployeeWorkAT(String[] params) {
    int num = 0;
    sql = "select ENAME,ADDRESS from " + emp + ", " + dep + " where DNAME = \"" + params[0] + "\"
        + " and EMPLOYEE.DNO = DEPARTMENT.DNO;"; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        System.out.println("在" + params[0] + "工作的所有工作人员的名字和地址");
        System.out.println(" | ENAME | ADDRESS |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println(" | " + ret.getString("ENAME") + " | " + ret.getString("ADDRESS") + " |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:

查询开始: 例如 n "x,y"

MengSQL>3 "财务部"

在财务部工作的所有工作人员的名字和地址

ENAME		ADDRESS
宋一	B01	
宋二	B02	
宋三	B03	
宋四	B04	
宋五	B05	
宋六	B06	
宋七	B07	
宋八	B08	
宋九	B09	
宋十	B10	

查询出10行数据!

4: 在%DNAME%工作且工资低于%SALARY%元的员工名字和地址, 其中%DNAME%和%SALARY%为C语言编写的程序的输入参数;

程序代码:

```
// 在%DNAME%工作且工资低于%SALARY%元的员工名字和地址
public static void EmployeeWorkATandLowSalary(String[] params) {
    int num = 0;
    sql = "select ENAME,ADDRESS from " + emp + "," + dep + " where SALARY < " + Integer.parseInt(params[1])
        + " and DNAME = \"" + params[0] + "\" and EMPLOYEE.DNO = DEPARTMENT.DNO;"; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句, 得到结果集
        System.out.println("在" + params[0] + "工作且工资低于" + params[1] + "元的员工名字和地址");
        System.out.println("  ENAME      ADDRESS  ");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println("      " + ret.getString("ENAME") + "      | " + ret.getString("ADDRESS") + "      |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:

查询开始: 例如 n "x,y"

MengSQL>4 "研发部,3000"

在研发部工作且工资低于3000元的员工名字和地址

ENAME		ADDRESS
张九	A09	
张十	A10	

查询出2行数据!

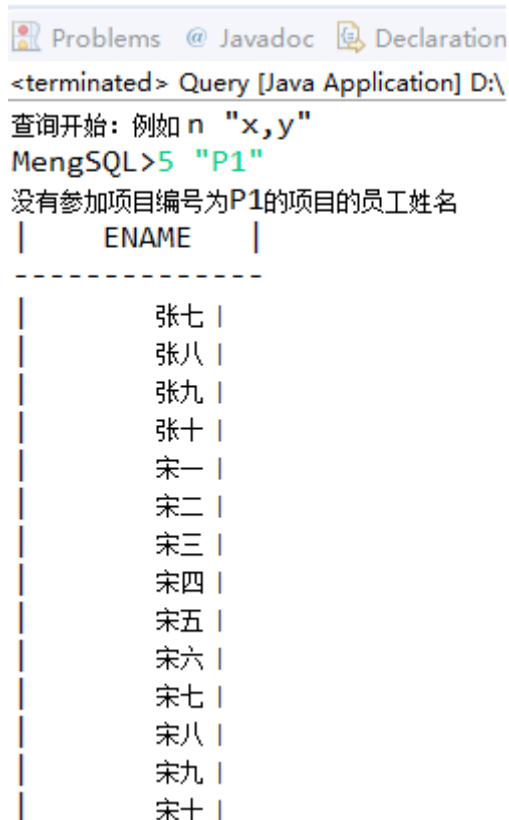
5: 没有参加项目编号为%PNO%的项目的员工姓名, 其中%PNO%为C语言编写的程序的

输入参数:

程序代码:

```
// 没有参加项目编号为%PNO%的项目的员工姓名
public static void EmployeeNotIn(String[] params) {
    int num = 0;
    sql = "select ENAME from " + emp + " where EMPLOYEE.ESSN not in ( select ESSN from WORKS_ON where PNO = \" "
        + params[0] + "\" );"; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句, 得到结果集
        System.out.println("没有参加项目编号为 " + params[0] + " 的项目的员工姓名");
        System.out.println("|      ENAME      |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println("|      " + ret.getString("ENAME") + "      |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出 " + num + " 行数据! ");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:



Problems @ Javadoc Declaration

<terminated> Query [Java Application] D:\

查询开始: 例如 n "x,y"

MengSQL>5 "P1"

没有参加项目编号为P1的项目的员工姓名

ENAME
张七
张八
张九
张十
宋一
宋二
宋三
宋四
宋五
宋六
宋七
宋八
宋九
宋十

	李一
	李二
	李三
	李五
	李六
	李七
	李八
	李九
	李十
	周一
	周二
	周三
	周四
	周五
	周六
	周日
	周八
	周九
	周十
	王一
	王二
	王三
	王五
	王六
	王七
	王八
	王九

查询出41行数据！

6：由%ENAME%领导的工作人员的姓名和所在部门的名称，其中%ENAME%为 C 语言编写的程序的输入参数；

程序代码：

```
// 由%ENAME%领导的工作人员的姓名和所在部门的名字
public static void EmployeeOf(String[] params) {
    int num = 0;
    sql = "select b.ENAME,DEPARTMENT.DNAME from " + emp + " as b," + emp + " as a," + dep
        + " where b.DNO = DEPARTMENT.DNO and a.ENAME = \" + params[0]
        + \" and b.SUPERSSN = a.ESSN and b.ENAME <> \" + params[0] + \"\""; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        System.out.println("由" + params[0] + "领导的工作人员的姓名和所在部门的名字");
        System.out.println("    ENAME      |      DNAME      |");
        System.out.println("-----+-----");
        while (ret.next()) {
            System.out.println("    " + ret.getString("ENAME") + "    |    " + ret.getString("DNAME") + "    |");
            num++;
        } // 显示数据
        System.out.println("-----+-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:

<terminated> Query [Java Application] D:\Program Files\Java\jdk1.8.0_...

查询开始: 例如 n "x,y"

MengSQL>6 "张红"

由张红领导的工作人员的姓名和所在部门的名字

ENAME	DNAME
张二	研发部
张三	研发部
张四	研发部
张五	研发部
张六	研发部
张七	研发部
张八	研发部
张九	研发部
张十	研发部

查询出9行数据!

7: 至少参加了项目编号为%PNO1%和%PNO2%的项目的员工号, 其中%PNO1%和%PNO2%为C语言编写的程序的输入参数;

程序代码:


```
// 至少参加了项目编号为%PNO1%和%PNO2%的员工的员工号
public static void EmployeeIN(String[] params) {
    int num = 0;
    sql = "select ESSN from " + wor + " where PNO = \"" + params[0]
        + "\" and ESSN in (select ESSN from WORKS_ON where PNO = \"" + params[1] + "\"));";
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        System.out.println("至少参加了项目编号为" + params[0] + "和" + params[1] + "的员工的员工号");
        System.out.println("|    ESSN    |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println("| " + ret.getString("ESSN") + " |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:

```
查询开始: 例如 n "x,y"
MengSQL>7 "P1,P2"
至少参加了项目编号为P1和P2的员工的员工号
|    ESSN    |
-----
| 12304 |
| 12305 |
| 12306 |
| 12324 |
| 12344 |
-----
查询出5行数据!
```

8: 员工平均工资低于%SALARY%元的部门名称, 其中%SALARY%为 C 语言编写的程序的输入参数;

程序代码:

```
// 员工平均工资低于%SALARY%元的部门名称
public static void EmployeeSalaryLow(String[] params) {
    int num = 0;
    sql = "select DNAME from " + dep + "," + emp
        + " where EMPLOYEE.DNO = DEPARTMENT.DNO group by EMPLOYEE.DNO having avg(SALARY) < "
        + Integer.parseInt(params[0]) + " "; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        System.out.println("员工平均工资低于" + params[0] + "元的部门名称");
        System.out.println("|          DNAME          |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println("| " + ret.getString("DNAME") + " |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:

Problems @ Javadoc Declaration Cor

Query [Java Application] D:\Program Files\Java\jdk\

查询开始: 例如 n "x,y"

MengSQL>8 "3000"

员工平均工资低于3000元的部门名称

DNAME
财务部

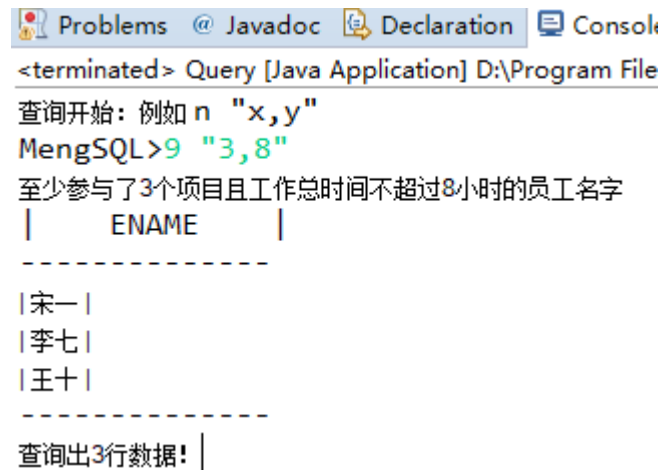
查询出1行数据!

9: 至少参与了%N%个项目且工作总时间不超过%HOURS%小时的员工名字, 其中%N%和%SALARY%为 C 语言编写的程序的输入参数;

程序代码:

```
// 至少参与了 %N%个项目且工作总时间不超过%HOURS%小时的员工名字, 其中%N%和%SALARY%为C语言编写的程序的输入参数
private static void Atlast(String[] params) {
    int num = 0;
    sql = "select ENAME from " + emp + " natural join WORKS_ON group by ESSN having count(PNO) >= " + params[0]
        + " and sum(HOURS) <= " + params[1] + " "; // SQL语句
    pre = new Contact(sql); // 创建DBHelper对象
    try {
        ret = pre.pst.executeQuery(); // 执行语句，得到结果集
        // System.out.println("-----");
        System.out.println("至少参与了" + params[0] + "个项目且工作总时间不超过" + params[1] + "小时的员工名字");
        System.out.println("|          ENAME          |");
        System.out.println("-----");
        while (ret.next()) {
            System.out.println("| " + ret.getString("ENAME") + " |");
            num++;
        } // 显示数据
        System.out.println("-----");
        System.out.println("查询出" + num + "行数据!");
        ret.close();
        pre.close(); // 关闭连接
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

查询结果:



The screenshot shows a Java IDE console with tabs for Problems, Javadoc, Declaration, and Console. The console output is as follows:

```
<terminated> Query [Java Application] D:\Program File
查询开始: 例如 n "x,y"
MengSQL>9 "3,8"
至少参与了3个项目且工作总时间不超过8小时的员工名字
|      ENAME      |
-----
|宋一|
|李七|
|王十|
-----
查询出3行数据: |
```

四、实验心得

通过本次实验，学习了如何使用高级语言访问 MySQL 数据库，并进行查询。

首先，要为编译器添加 MySQL 的依赖。我先后尝试了 C、python、Java，给 Code blocks 添加时注意：如果 codeblocks 是 32 位的，那么添加的库也要是 32 位的而不能是 64 位的；给 pycharm 添加库，注意 python2 和 python3 的差别，我用 python3 添加 MySQLdb 时，总是版本不正确，最终无法解决；最后，我选用的是 eclipse，需要向里面添加 jar 包即可，在连接数据库时，需要添加 jdbc，就可以解决。

实验主要的核心就是连接数据库以及进行查询，进行查询可将查询语句传过去即可。

此次实验进一步加深了我对数据库的查询语句的掌握程度，掌握了 Java 语言的数据库编程方法。