

**数据库系统**

**实 验 报 告**

学 号： 2021211901

学生姓名： 吕长鸿

班 级： 2111102

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **评价** | **优** | **良** | **中** | **差** | **评语：** |
| **程序介绍** |  |  |  |  |
| **程序演示** |  |  |  |  |
| **实验报告** |  |  |  |  | **评分** |

**2023-04**

**数据库系统 实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 吕长鸿 | | 院系 | 软件学院 | | 学号 | 2021211901 |
| 任课教师 | | 王金宝 | | | 指导教师 | 王金宝 | |
| 实验地点 | | 研究院2楼机房 | | | 实验时间 | 2023年 4 月 16 日 | |
| 实验名称 | | 实验2: 使用高级语言操作MySQL数据库 | | | | | |
| 实验环境 | | Windwos10、MySQL | | | | | |
| 实验目的 | | | | | | | |
| 学会使用高级语言访问MySQL数据库，并进行查询。 | | | | | | | |
| 实验过程和结果 | | | | | | | |
| **主函数**：  public class Main {  public static void main(String[] args) throws SQLException {  CompanyQuery CompanyQuery = new CompanyQuery();  Scanner scanner = new Scanner(System.in);  while(true) {  String s = scanner.nextLine();  if(s.equals("exit")) break;  String[] argv = s.split(" ");  String number = argv[2];  switch (number) {  case "1": {  for(int i = 5; i < argv.length; i ++) {  argv[4] += " " +argv[i];  }  CompanyQuery.queryOne(argv[4]);  continue;  }  case "2": {  for(int i = 5; i < argv.length; i ++) {  argv[4] += " " +argv[i];  }  CompanyQuery.queryTwo(argv[4]);  continue;  }  case "3": {  for(int i = 5; i < argv.length; i ++) {  argv[4] += " " +argv[i];  }  CompanyQuery.queryThree(argv[4]);  continue;  }  case "4": {  for(int i = 5; i < argv.length - 1; i ++) {  argv[4] += " "+ argv[i];  }  CompanyQuery.queryFour(argv[4], Integer.parseInt(argv[argv.length - 1]));  continue;  }  case "5": {  for(int i = 5; i < argv.length; i ++) {  argv[4] += " " +argv[i];  }  CompanyQuery.queryFive(argv[4]);  continue;  }  case "6": {  for(int i = 5; i < argv.length; i ++) {  argv[4] += " " +argv[i];  }  CompanyQuery.querySix(argv[4]);  continue;  }  case "7": {  CompanyQuery.querySeven(argv[4], argv[5]);  continue;  }  case "8": {  for(int i = 5; i < argv.length; i ++) {  argv[4] += " " +argv[i];  }  CompanyQuery.queryEight(Integer.parseInt(argv[4]));  continue;  }  case "9": {  CompanyQuery.queryNine(Integer.parseInt(argv[4]), Integer.parseInt(argv[5]));  continue;  }  default: {  System.out.println("repeat !!!");  }  }  }  }  }  负责接受命令，解析请求  **CompanyQuery**类负责存储数据库信息：URL、username、password等    建立连接：  内含9个不同的成员函数，每个成员函数负责执行不同的请求，如：  public void queryOne(String argv) {  String sql = "select essn from works\_on where pno = ?";  try {  pstmt = conn.prepareStatement(sql);  pstmt.setString(1, argv);  resultSet = pstmt.executeQuery();  System.out.println("参加了项目编号为" + argv + "的员工号为：");  while(resultSet.next()) {  String essn = resultSet.getString("essn");  System.out.println(essn);  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  }  就查询并输出了参加了项目编号为 ？ 的员工号，？ 通过main函数传入。  **程序运行效果演示**  **以第6号查询为例：Messi领导的工作人员的姓名和所在部门的名字：** | | | | | | | |
| 实验结论（结果分析、遇到的困难和解决方法等） | | | | | | 备注 |  |
| 在本次JDBC实验中，我们学习了JDBC，该API提供了一种编程方式，使得Java代码能够连接和操作各个关系型数据库管理系统  通过本次实验，我们发现JDBC具有以下优点：   1. 通过JDBC,我们可以很方便地获得对数据库的访问及控制。 2. 使用JDBC，我们可以使用SQL语句在程序中访问和处理数据库的内容。 3. JDBC允许使用预编译的SQL语句，这样可以优化性能，避免SQL注入。   JDBC API是一个强大的工具，它可以简化Java程序员与关系数据库之间的交互 | | | | | | | |