

GaussDB数据库实验二：数据库应用程序设计实验

姓名：陶拥鼎

学号：2021110801011

前言

一、实验任务——人力资源管理系统

二、项目设计

1. 项目结构

2. 项目源代码

三、功能展示

1. Staff（普通职员）

2. Manager（部门经理）

3. HrManager（人事经理）

总结与感悟

前言

本实验承接GaussDB数据库实验一：SQL基础实验，旨在开发一个简单的人力资源管理系统。

本报告网页版 GaussDB数据库实验二：数据库应用程序设计实验

一、实验任务——人力资源管理系统

设计人力资源管理系统

- 修改表staffs，增加一列password，并设置数据
- 设置三个角色
 - 员工 staff
 - 部门经理 manager
 - 人事经理 hr_manager

1. 员工具备以下功能

- 1) 输入staff_id 和 正确的密码，进入员工主页面；
- 2) 在员工主页面，可以选择查看员工自己基本信息；
- 3) 在员工主页面，修改员工自己的电话号码；

2. 部门经理具备以下功能

- 1) 输入staff_id 和 正确的密码，进入部门经理主页面；
- 2) 在部门经理主页面，可以查看本部门所有员工基本信息（选择按员工编号升序排列，或者按工资降序排列）；
- 3) 在部门经理主页面，可以按员工编号查询员工基本信息；
- 4) 在部门经理主页面，可以按员工姓名查询员工基本信息；
- 5) 在部门经理主页面，可以统计查询本部门员工最高工资，最低工资以及平均工资；

3. 人事经理具备以下功能

- 1) 输入特定编号hr001 和 特定密码，进入人事经理主页面；
- 2) 在人事经理主页面，可以查看所有员工基本信息（选择按员工编号升序排列，或者按工资降序排列）；
- 3) 在人事经理主页面，可以按员工编号查询员工基本信息；
- 4) 在人事经理主页面，可以按员工姓名查询员工基本信息；
- 5) 在人事经理主页面，可以统计各部门员工最高工资，最低工资以及平均工资；
- 6) 在人事经理主页面，可以查询各部门基本信息，并可以根据部门编号修改部门名称；
- 7) 在人事经理主页面，可以各工作地点基本信息，并可以增加新的工作地点；
- 8) 在人事经理主页面，可以按员工编号查询员工工作信息，包括其历史工作信息，返回员工编号，职位编号和部门编号；

二、项目设计

1. 项目结构

本项目的结构如下：

```
- .classpath
- .project
+ .settings/
  - org.eclipse.core.resources.prefs
  - org.eclipse.jdt.core.prefs
+ bin/
+ openGauss/
```

```

- HrManager.class
- Human.class
- human_resources_management.class
- Manager.class
- utils.class
+ src/
+ openGauss/
- HrManager.java
- Human.java
- human_resources_management.java
- Manager.java
- utils.java

```

- `human_resources_management.java` : 主程序
- `Human.java` : 普通职员类
- `Manager.java` : 部门经理类, 继承了Human类
- `HrManager.java` : 人事经理类
- `utils.java` : 定义了一些工具

2. 项目源代码

- `human_resources_management.java`

```

package openGauss;

import java.sql.*;
import java.util.Scanner;

public class human_resources_management {

    static String driver = "org.postgresql.Driver";
    // 服务器的ip和数据库的端口, 需要根据自己设置
    static String sourceURL = "jdbc:postgresql://120.46.176.58:26000/human_resource";

    // 数据库的用户名与密码, 需要根据自己的设置
    static final String username = "taoyongding";
    static final String passwd = "taoyongding@123";

    static Connection conn = null;
    static Scanner sc = new Scanner(System.in);

    static void menuWorker(int staff_id) {
        System.out.println("您好! 请输入您的password:");
        Human worker = new Human(staff_id);
        if(utils.checkpassword(worker.queryPassword())) {
            while (true) {
                System.out.println("请输入数字进入对应功能:\\n"
                    + "1 查看您的基本信息\\n"

```

```

        + "2 修改您的电话号码\\n"
        + "3 退出系统");
        int opt = sc.nextInt();

        if (opt == 3) {
            System.out.println("感谢您的使用!");
            break;
        }
        worker.choose(opt);
    }
}

static void menuManager(int staff_id) {
    System.out.println("经理您好！请输入您的password:");
    Manager manager = new Manager(staff_id);
    if(utills.checkpassword(manager.queryPassword())) {
        while (true) {
            System.out.println("请输入数字进入对应功能:\\n"
                + "1 查看本部门所有员工基本信息\\n"
                + "2 按员工编号查询员工基本信息\\n"
                + "3 按员工姓名查询员工基本信息\\n"
                + "4 统计查询本部门员工最高工资，最低工资以及平均工资\\n"
                + "5 退出系统");
            int opt = sc.nextInt();
            if (opt == 5) {
                System.out.println("感谢您的使用!");
                break;
            }
            manager.choose(opt);
        }
    }
}

static void menuHrManager() {
    System.out.println("人事经理您好！请输入您的password:");
    if(utills.checkpassword("hr@001")) {
        HrManager hrManager = new HrManager();
        while (true) {
            System.out.println("请输入数字进入对应功能:\\n"
                + "1 查看所有员工基本信息（选择按员工编号升序排列，或者按工资降序排列）\\n"
                + "2 按员工编号查询员工基本信息\\n"
                + "3 按员工姓名查询员工基本信息\\n"
                + "4 统计各部门员工最高工资，最低工资以及平均工资\\n"
                + "5 查询各部门基本信息\\n"
                + "6 根据部门编号修改部门名称\\n"
                + "7 查询各工作地点基本信息\\n"
                + "8 增加新的工作地点\\n"
                + "9 按员工编号查询员工工作信息，包括其历史工作信息，返回员工编号，职位编号和部门编号\\n"
                + "10 退出系统");
            int opt = sc.nextInt();
            if (opt == 10) {
                System.out.println("感谢您的使用!");
                break;
            }
            hrManager.choose(opt);
        }
    }
}

```

```

    }
}

public static void main(String[] args) {
    PreparedStatement stmt = null;
    try {
        // 注册JDBC驱动
        Class.forName(driver).newInstance();

        // 打开链接
        conn = DriverManager.getConnection(sourceURL, username, passwd);
        System.out.println("连接数据库...");

        System.out.println("请输入您的staff_id:");
        //测试样例:员工 107 经理 100 hr hr001

        //Michael Hartstein
        String staff_id = sc.nextLine();
        if(staff_id.equals("hr001") == false) {
            //用staff_id 查询对应的manager_id
            int s_id = Integer.parseInt(staff_id);
            String sql;
            sql = "SELECT * FROM staffs WHERE manager_id=?";
            stmt = conn.prepareStatement(sql);
            stmt.setInt(1, s_id);
            ResultSet rs = stmt.executeQuery();
            if(rs.next()) { //能够查到, 说明是某个部门的经理
                menuManager(s_id);
            }
            else { //为普通员工
                menuWorker(s_id);
            }
        }
        else { //为hr
            menuHrManager();
        }
        sc.close();
    } catch (SQLException se) {
        // 处理JDBC异常
        se.printStackTrace();
    } catch (Exception e) {
        // 处理其他异常
        e.printStackTrace();
    } finally {
        // 关闭资源
        try {
            if (stmt != null) {
                stmt.close();
            }
        } catch (SQLException se2) {

        }
        // 什么都不做
        try {
            if (conn != null) {
                conn.close();
            }
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
}

```

```

    }
}
System.out.println("Goodbye!");
}
}

```

- `Human.java`

```

package openGauss;

import java.sql.*;
import java.util.Scanner;

public class Human {
    // 普通员工类，实现以下功能：
    // 1 在员工主页面，可以选择查看员工自己基本信息
    // 2 在员工主页面，修改员工自己的电话号码
    protected int staff_id;
    public Connection conn = null;
    public Scanner sc = null;
    public Human(int staff_id) {
        this.sc = human_resources_management.sc;
        this.conn = human_resources_management.conn;
        this.staff_id = staff_id;
    }

    public void query() {
        //System.out.println("您的基本信息如下:");
        String sql = "SELECT * FROM staffs WHERE staff_id=" + staff_id;
        utils.queryBySql(sql);
    }

    public int querySectionId() {
        String sql;
        sql = "SELECT section_id FROM staffs WHERE staff_id=" + staff_id;
        int section_id = -1;
        try {
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);

            if(rs.next()) {
                section_id = rs.getInt("section_id");
                return section_id;
            }

            rs.close();
            stmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
            return -1;
        }
        return section_id;
    }
}

```

```

public String queryPassword() {
    String sql;
    sql = "SELECT password FROM staffs WHERE staff_id=" + staff_id;
    String password = null;
    try {
        Statement stmt = conn.createStatement();

        ResultSet rs = stmt.executeQuery(sql);

        if(rs.next()) {
            password = rs.getString("password");
            return password;
        }

        rs.close();
        stmt.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return password;
}

public void updatePhoneNumber() {
    System.out.println("请输入新的电话号码(请按照xxx.xxx.xxxx的格式):");
    String phone_number = sc.next();
    String sql = "UPDATE staffs SET phone_number = " + phone_number + "WHERE staff_id="
+ staff_id;
    utils.updateBySql(sql);
}

public void choose(int opt) {
    switch(opt) {
        case 1:
            query();
            break;
        case 2:
            updatePhoneNumber();
            break;
    }
}
}
}

```

- **Manager.java**

```

package openGauss;

import java.util.Scanner;

public class Manager extends Human {
    // 部门经理类，实现以下功能：
    // 1 查看本部门所有员工基本信息（选择按员工编号升序排列，或者按工资降序排列）；
    // 2 按员工编号查询员工基本信息；
    // 3 按员工姓名查询员工基本信息；
    // 4 统计查询本部门员工最高工资，最低工资以及平均工资；
}

```

```

Scanner sc = new Scanner(System.in);
public Manager (int staff_id) {
    super(staff_id);
}
public void queryAll() {
    // 查看本部门所有员工基本信息
    String sql = null;

    System.out.println("您好！\n输入1，可按员工编号升序查询\n输入2，可按薪资降序查询");

    int opt = sc.nextInt();
    switch(opt) {
        case 1:
            sql = "SELECT * FROM staffs WHERE section_id="+this.querySectionId()+" ORDER B
Y staff_id ASC";
        case 2:
            sql = "SELECT * FROM staffs WHERE section_id="+this.querySectionId()+" ORDER B
Y salary DESC";
        default:
            utils.queryBySql(sql);
            break;
    }
}

public void queryById() {
    // 按员工编号查询员工基本信息
    System.out.println("请输入所要查询的员工编号:");
    int id = sc.nextInt();
    Human worker = new Human(id);
    if(worker.querySectionId() != this.querySectionId()) {
        System.out.println("查询失败，请输入本部门员工的编号！\n");
    }
    else {
        worker.query();
        System.out.println("查询成功！\n");
    }
}

public void queryByName() {
    // 按员工姓名查询员工基本信息（可能重名，所以要考虑多个）
    System.out.println("请输入所要查询的员工姓名(例如Yongding Tao):");
    String name = sc.nextLine();
    utils.queryByName(name, this.querySectionId());
}

public void querySalary() {
    // 统计查询本部门员工最高工资，最低工资以及平均工资
    String sql = "SELECT MAX(salary) AS highest_salary,\n"
        + "      MIN(salary) AS lowest_salary,\n"
        + "      AVG(salary) AS average_salary\n"
        + "FROM staffs\n"
        + "WHERE section_id = " +this.querySectionId()+"\n";
    utils.queryBySql(sql);
}

```



```

@Override
public void choose(int opt) {
    switch(opt) {
        case 1:
            queryAll();
            break;
        case 2:
            queryById();
            break;
        case 3:
            queryByName();
            break;
        case 4:
            querySalary();
            break;
    }
}
}
}

```

- `HrManager.java`

```

package openGauss;

import java.sql.*;
import java.util.Scanner;

public class HrManager{
    // 人事经理类，实现以下功能：
    // 1 查看所有员工基本信息（选择按员工编号升序排列，或者按工资降序排列）；
    // 2 按员工编号查询员工基本信息；
    // 3 按员工姓名查询员工基本信息；
    // 4 统计各部门员工最高工资，最低工资以及平均工资；
    // 5 查询各部门基本信息；
    // 6 根据部门编号修改部门名称；
    // 7 查询各工作地点基本信息；
    // 8 增加新的工作地点；
    // 9 按员工编号查询员工工作信息，包括其历史工作信息，返回员工编号，职位编号和部门编号；
    Scanner sc = null;
    Connection conn = null;

    public HrManager() {
        this.sc = human_resources_management.sc;
        this.conn = human_resources_management.conn;
    }

    public void queryAll() {
        // 查看所有员工基本信息
        String sql = null;

        System.out.println("您好！\n输入1，可按员工编号升序查询\n输入2，可按薪资降序查询");

        int opt = sc.nextInt();
        switch(opt) {
            case 1:
                sql = "SELECT * FROM staffs ORDER BY staff_id ASC";
                break;

```

```

        case 2:
            sql = "SELECT * FROM staffs ORDER BY salary DESC";
            break;
    }
    utils.queryBySql(sql);

}

public void queryById() {
    // 按员工编号查询员工基本信息
    System.out.println("请输入所要查询的员工编号:");
    int id = sc.nextInt();
    Human worker = new Human(id);
    worker.query();
    System.out.println("查询成功!\n");
}

public void queryByName() {
    // 按员工姓名查询员工基本信息 (可能重名, 所以要考虑多个)
    System.out.println("请输入所要查询的员工姓名(例如Yongding Tao):");
    sc.nextLine();
    String name = sc.nextLine();
    utils.queryByName(name);
}

public void querySalary() {
    // 统计各部门员工最高工资, 最低工资以及平均工资;
    String sql = "SELECT section_id,"
        + "MAX(salary) AS highest_salary,\n"
        + "MIN(salary) AS lowest_salary,\n"
        + "AVG(salary) AS average_salary\n"
        + "FROM staffs\n"
        + "GROUP BY section_id\n"
        + "ORDER BY section_id ASC";
    utils.queryBySql(sql);
}

public void querySection() {
    // 查询各部门基本信息;
    String sql = "SELECT * FROM sections";
    utils.queryBySql(sql);
}

public void updateSection() {
    // 根据部门编号修改部门名称;
    System.out.println("请输入部门编号:");
    int section_id = sc.nextInt();
    System.out.println("当前部门名称为:");
    String sql = "SELECT section_name FROM sections WHERE section_id = " + section_id;
    utils.queryBySql(sql);
    System.out.println("请输入新的部门名称:");
    sc.nextLine();
    String section_name = sc.nextLine();
    sql = "UPDATE sections SET section_name='" + section_name + "' WHERE section_id=" + section_id;
    utils.updateBySql(sql);
}

```

```

public void queryPlace() {
    // 查询各工作地点基本信息；
    String sql = "SELECT * FROM places";
    utils.queryBySql(sql);
}

public void addNewPlace() {
    System.out.println("您想要插入一个新地点，请根据引导完成数据读入");
    System.out.println("请输入部门编号:");
    int place_id = sc.nextInt();
    System.out.println("请输入部门所在地街道地址:");
    String street_address = sc.nextLine();
    System.out.println("请输入部门所在地邮政编码:");
    String postal_code = sc.nextLine();
    System.out.println("请输入部门所在城市:");
    String city = sc.nextLine();
    System.out.println("请输入部门所在省:");
    String state_province = sc.nextLine();
    System.out.println("请输入国家简写（如CN）");
    String state_id = sc.next();
    String sql = "INSERT INTO places (place_id, street_address, postal_code, city, state_province, state_id)"
        + String.format("VALUES (%d, '%s', '%s', '%s', '%s', '%s')", place_id, street_
address, postal_code, city, state_province, state_id);
    utils.updateBySql(sql);
    //
    //    VALUES (1001, '123 Main St', '12345', 'New York', 'NY', 'US');

}

public void queryHistory() {
    // 查询工作历史信息 //test 101
    System.out.println("请输入所要查询的员工编号:");
    sc.nextLine();
    int staff_id = sc.nextInt();
    String sql = "SELECT * FROM employment_history WHERE staff_id="+staff_id;
    utils.queryBySql(sql);
    System.out.println("查询成功！\n");
}

public void choose(int opt) {
    switch(opt) {
        case 1:
            queryAll();
            break;
        case 2:
            queryById();
            break;
        case 3:
            queryByName();
            break;
        case 4:
            querySalary();
            break;
        case 5:
            querySection();
            break;
    }
}

```

```

        case 6:
            updateSection();
            break;
        case 7:
            queryPlace();
            break;
        case 8:
            addNewPlace();
            break;
        case 9:
            queryHistory();
            break;
    }
}
}

```

- `utils.java`

```

package openGauss;

import java.sql.*;
import java.util.Scanner;

public class utils {
    static public Scanner sc = human_resources_management.sc;
    static public Connection conn = human_resources_management.conn;

    public static void printCurrentRowData(ResultSet resultSet) throws SQLException {
        ResultSetMetaData metaData = resultSet.getMetaData();
        int columnCount = metaData.getColumnCount();

        for (int i = 1; i <= columnCount; i++) {
            String columnName = metaData.getColumnName(i);
            Object columnValue = resultSet.getObject(i);

            System.out.println(columnName + ": " + columnValue);
        }
    }

    public static void printQueryResults(ResultSet resultSet) throws SQLException {
        ResultSetMetaData metaData = resultSet.getMetaData();
        int columnCount = metaData.getColumnCount();

        // 打印表头
        for (int i = 1; i <= columnCount; i++) {
            //System.out.print(metaData.getColumnName(i) + "\\t");
            System.out.format("%-20s", metaData.getColumnName(i));
        }
        System.out.println();

        // 打印分隔线
        for (int i = 1; i <= columnCount; i++) {
            System.out.print("-----");
        }
        System.out.println();
    }
}

```

```

        // 打印数据行
        for (int i = 1; i <= columnCount; i++) {
            //System.out.print(resultSet.getString(i) + "\\t");
            System.out.format("%-20s", resultSet.getString(i));
        }
        System.out.println();
        while (resultSet.next()) {
            for (int i = 1; i <= columnCount; i++) {
                //System.out.print(resultSet.getString(i) + "\\t");
                System.out.format("%-20s", resultSet.getString(i));
            }
            System.out.println();
        }
    }

    public static void queryByName(String fullName) {
        String[] names = fullName.split(" ");
        if(names.length != 2) {
            System.out.println("输入错误!");
            return;
        }
        String sql = "SELECT staff_id FROM staffs WHERE first_name=\\'" + names[0] + "\\''AND
last_name=\\'" + names[1] + "\\''";
        try {
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);

            while(rs.next()) {
                int staff_id = rs.getInt("staff_id");
                Human worker = new Human(staff_id);
                worker.query();
            }

            rs.close();
            stmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void queryByName(String fullName, int section_id) {
        String[] names = fullName.split(" ");
        if(names.length != 2) {
            System.out.println("输入错误!");
            return;
        }
        String sql = "SELECT staff_id FROM staffs WHERE first_name=\\'" + names[0] + "\\''AND
last_name=\\'" + names[1] + "\\''";
        try {
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);

            while(rs.next()) {
                int staff_id = rs.getInt("staff_id");
                Human worker = new Human(staff_id);
                if(worker.querySectionId() == section_id) {//是自己部门的人
                    worker.query();
                }
            }
        }
    }

```

```

        }
        else continue;
    }

    rs.close();
    stmt.close();
} catch (SQLException e) {
    e.printStackTrace();
}
}

public static void queryBySql(String sql) {
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql);
        if(rs.next()) {
            printQueryResults(rs);
        }
        rs.close();
        stmt.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public static void updateBySql(String sql) {
    try {
        Statement stmt = conn.createStatement();
        stmt.executeUpdate(sql);

        System.out.println("更新成功!");
        stmt.close();
    } catch (SQLException e) {
        System.out.println("更新失败!");
        e.printStackTrace();
    }
}

public static boolean checkpassword(String correct) {
    String password = sc.next();
    int cnt = 0;
    while(password.equals(correct) == false) {
        System.out.println("登入失败, 请重新输入!");
        cnt++;
        if(cnt == 5) {
            System.out.println("连续5次错误, 系统退出!");
            return false;
        }
        password = sc.next();
    }
    System.out.println("登入成功!");
    return true;
}
}

```

三、功能展示

这部分将会对人力资源管理系统三种角色的功能分别进行展示。

1. Staff（普通职员）

- 输入staff_id，系统自动识别是 staff、Manager 还是 HrManager

```
5月 31, 2023 1:42:04 上午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl
INFO: [ba5fb197-7380-4e19-b657-7de8355d231b] Try to connect. IP: 120.46.176.58:26000
5月 31, 2023 1:42:04 上午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl
INFO: [192.168.31.67:14262/120.46.176.58:26000] Connection is established. ID: ba5fb197-7380-4e19-b657-7de8355d231b
5月 31, 2023 1:42:05 上午 org.postgresql.core.v3.ConnectionFactoryImpl openConnectionImpl
INFO: Connect complete. ID: ba5fb197-7380-4e19-b657-7de8355d231b
连接数据库...
请输入您的staff_id:
107
您好! 请输入您的password:
password@123
登入失败, 请重新输入!
password123
登入成功!
请输入数字进入对应功能:
1 查看您的基本信息
2 修改您的电话号码
3 退出系统
```

CSDN @Asker_CXQ

- 输入失败，将会重新输入，连续失败5次则退出系统

```
1
staff_id      first_name  last_name    email        phone_number  hire_date    employment_id  salary    commission_pct
-----
107           Diana      Lorentz      DLORENTZ     590.423.5569  1999-02-07 00:00:00 IT_PROG      4200.00    null

请输入数字进入对应功能:
1 查看您的基本信息
2 修改您的电话号码
3 退出系统
2
请输入新的电话号码(请按照xxx.xxx.xxxx的格式):
111.111.1111
更新成功!
请输入数字进入对应功能:
1 查看您的基本信息
2 修改您的电话号码
3 退出系统
1
```

staff_id	first_name	last_name	email	phone_number	hire_date	employment_id	salary	commission_pct
107	Diana	Lorentz	DLORENTZ	111.111.1111	1999-02-07 00:00:00	IT_PROG	4200.00	null

2. Manager（部门经理）

- 功能1

```
请输入您的staff_id:
100
经理您好! 请输入您的password:
password123
登入成功!
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
1
您好!
输入1, 可按员工编号升序查询
输入2, 可按薪资降序查询
1
```

staff_id	first_name	last_name	email	phone_number	hire_date	employment_id	salary	commission_pct
100	Steven	King	SKING	515.123.4567	1987-06-17 00:00:00	AD_PRES	24000.00	null
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21 00:00:00	AD_VP	17000.00	null
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13 00:00:00	AD_VP	17000.00	null

@Asker_CXQ

```
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
1
您好!
输入1, 可按员工编号升序查询
输入2, 可按薪资降序查询
2
staff_id      first_name    last_name     email         phone_number  hire_date     employment_id salary      commission_pct
-----
100           Steven        King          SKING         515.123.4567  1987-06-17 00:00:00 AD_PRE      24000.00    null
101           Neena        Kochhar       NKOCHHAR     515.123.4568  1989-09-21 00:00:00 AD_VP       17000.00    null
102           Lex          De Haan       LDEHAAN      515.123.4569  1993-01-13 00:00:00 AD_VP       17000.00    null
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
```

CSDN @Asker_CXQ

• 功能2

```
2
请输入所要查询的员工编号:
101
staff_id      first_name    last_name     email         phone_number  hire_date     employment_id salary      commission_pct
-----
101           Neena        Kochhar       NKOCHHAR     515.123.4568  1989-09-21 00:00:00 AD_VP       17000.00    null
查询成功!
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
2
请输入所要查询的员工编号:
103
查询失败, 请输入本部门的编号!
```

CSDN @Asker_CXQ

• 功能3

```
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
3
请输入所要查询的员工姓名(例如Yongding Tao):
Steven King
staff_id      first_name    last_name     email         phone_number  hire_date     employment_id salary      commission_pct
-----
100           Steven        King          SKING         515.123.4567  1987-06-17 00:00:00 AD_PRE      24000.00    null
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
3
请输入所要查询的员工姓名(例如Yongding Tao):
sdfsfsdf
输入错误!
```

CSDN @Asker_CXQ

• 功能4

```
请输入数字进入对应功能:
1 查看本部门所有员工基本信息
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计查询本部门员工最高工资, 最低工资以及平均工资
5 退出系统
4
highest_salary  lowest_salary  average_salary
-----
24000.00        17000.00      19333.333333333333
```

CSDN @Asker_CXQ

3. HrManager（人事经理）

• 功能1


```
1
您好!
输入1, 可按员工编号升序查询
输入2, 可按薪资降序查询

1
staff_id      first_name    last_name     email          phone_number  hire_date      employment_id  salary      commission_pct
-----
100           Steven       King          SKING          515.123.4567  1987-06-17 00:00:00 AD_PRES       24000.00      null
101           Neena       Kochhar       HKOCHHAR       515.123.4568  1989-09-21 00:00:00 AD_VP         17000.00      null
102           Lex         De Haan       LDEHAAN        515.123.4569  1993-01-13 00:00:00 AD_VP         17000.00      null
103           Alexander   Hartstein     AHUNOLD        590.423.4567  1990-01-03 00:00:00 IT_PROG       9000.00       null
104           Bruce       Ernst         BERNST         590.423.4568  1991-05-21 00:00:00 IT_PROG       6000.00       null
105           David       Austin        DAUSTIN        590.423.4569  1997-06-25 00:00:00 IT_PROG       4800.00       null
106           Valli       Pataballa     VPATABAL       590.423.4560  1998-02-05 00:00:00 IT_PROG       4800.00       null
107           Diana       Lorentz       DLORENTZ       111.111.1111  1999-02-07 00:00:00 IT_PROG       4200.00       null
108           Nancy       Greenberg     NGREENBE       515.124.4569  1994-08-17 00:00:00 FI_MGR        12000.00      null
109           Daniel      Faviet        DFAVIET        515.124.4169  1994-08-16 00:00:00 FI_ACCOUNT    9000.00       null
110           John        Chen          JCHEN          515.124.4269  1997-09-28 00:00:00 FI_ACCOUNT    8200.00       null
111           Yongding    Tao           MYEMAIL        123.456.7890  2002-10-19 00:00:00 IT_PROG       88888.88      null
198           Donald      OConnell      DOCONNEL       650.507.9833  1999-06-21 00:00:00 SH_CLERK      3600.00       null
199           Douglas     Grant         DGRANT         650.507.9844  2000-01-13 00:00:00 SH_CLERK      3600.00       null
200           Jennifer    Whalen        JWHALEN        515.123.4444  1987-09-17 00:00:00 AD_ASST       5400.00       null
201           Michael     Hartstein     MHARTSTE       515.123.5555  1996-02-17 00:00:00 MK_MAN        13000.00      null
202           Pat         Fay           PFAY           603.123.6666  1997-08-17 00:00:00 MK_REP        6000.00       null
203           Susan       Mavris        SMAVRIS        515.123.7777  1994-06-07 00:00:00 HR_REP        6500.00       null
204           Hermann     Baer          HBAER          515.123.8888  1994-06-07 00:00:00 PR_REP        10000.00      null
205           Shelley     Higgins        SHIGGINS       515.123.8080  1994-06-07 00:00:00 AC_MGR        12000.00      null
206           William     Gietz         WGIEZT         515.123.8181  1994-06-07 00:00:00 AC_ACCOUNT    8300.00       null

请输入数字进入对应功能:
1 查看所有员工基本信息 (选择按员工编号升序排列, 或者按工资降序排列)
2 按员工编号查询员工基本信息
```

CSDN @Asker_CXQ

```
8 增加新的工作地点
9 按员工编号查询员工工作信息, 包括其历史工作信息, 返回员工编号, 职位编号和部门编号
10 退出系统
```

```
1
您好!
输入1, 可按员工编号升序查询
输入2, 可按薪资降序查询

2
staff_id      first_name    last_name     email          phone_number  hire_date      employment_id  salary      commission_pct
-----
111           Yongding    Tao           MYEMAIL        123.456.7890  2002-10-19 00:00:00 IT_PROG       88888.88      null
100           Steven       King          SKING          515.123.4567  1987-06-17 00:00:00 AD_PRES       24000.00      null
101           Neena       Kochhar       HKOCHHAR       515.123.4568  1989-09-21 00:00:00 AD_VP         17000.00      null
102           Lex         De Haan       LDEHAAN        515.123.4569  1993-01-13 00:00:00 AD_VP         17000.00      null
201           Michael     Hartstein     MHARTSTE       515.123.5555  1996-02-17 00:00:00 MK_MAN        13000.00      null
108           Nancy       Greenberg     NGREENBE       515.124.4569  1994-08-17 00:00:00 FI_MGR        12000.00      null
205           Shelley     Higgins        SHIGGINS       515.123.8080  1994-06-07 00:00:00 AC_MGR        12000.00      null
204           Hermann     Baer          HBAER          515.123.8888  1994-06-07 00:00:00 PR_REP        10000.00      null
103           Alexander   Hunold        AHUNOLD        590.423.4567  1990-01-03 00:00:00 IT_PROG       9000.00       null
109           Daniel      Faviet        DFAVIET        515.124.4169  1994-08-16 00:00:00 FI_ACCOUNT    9000.00       null
206           William     Gietz         WGIEZT         515.123.8181  1994-06-07 00:00:00 AC_ACCOUNT    8300.00       null
110           John        Chen          JCHEN          515.124.4269  1997-09-28 00:00:00 FI_ACCOUNT    8200.00       null
203           Susan       Mavris        SMAVRIS        515.123.7777  1994-06-07 00:00:00 HR_REP        6500.00       null
104           Bruce       Ernst         BERNST         590.423.4568  1991-05-21 00:00:00 IT_PROG       6000.00       null
202           Pat         Fay           PFAY           603.123.6666  1997-08-17 00:00:00 MK_REP        6000.00       null
200           Jennifer    Whalen        JWHALEN        515.123.4444  1987-09-17 00:00:00 AD_ASST       5400.00       null
106           Valli       Pataballa     VPATABAL       590.423.4560  1998-02-05 00:00:00 IT_PROG       4800.00       null
105           David       Austin        DAUSTIN        590.423.4569  1997-06-25 00:00:00 IT_PROG       4800.00       null
107           Diana       Lorentz       DLORENTZ       111.111.1111  1999-02-07 00:00:00 IT_PROG       4200.00       null
198           Donald      OConnell      DOCONNEL       650.507.9833  1999-06-21 00:00:00 SH_CLERK      3600.00       null
199           Douglas     Grant         DGRANT         650.507.9844  2000-01-13 00:00:00 SH_CLERK      3600.00       null
```

@Asker_CXQ

• 功能2

```
1 查看所有员工基本信息 (选择按员工编号升序排列, 或者按工资降序排列)
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计各部门员工最高工资, 最低工资以及平均工资
5 查询各部门基本信息
6 根据部门编号修改部门名称
7 查询各工作地点基本信息
8 增加新的工作地点
9 按员工编号查询员工工作信息, 包括其历史工作信息, 返回员工编号, 职位编号和部门编号
10 退出系统
```

```
2
请输入所要查询的员工编号:
111
staff_id      first_name    last_name     email          phone_number  hire_date      employment_id  salary      commission_pct
-----
111           Yongding    Tao           MYEMAIL        123.456.7890  2002-10-19 00:00:00 IT_PROG       88888.88      null
查询成功!
```

CSDN @Asker_CXQ

• 功能3

```
请输入数字进入对应功能:
1 查看所有员工基本信息 (选择按员工编号升序排列, 或者按工资降序排列)
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计各部门员工最高工资, 最低工资以及平均工资
5 查询各部门基本信息
6 根据部门编号修改部门名称
7 查询各工作地点基本信息
8 增加新的工作地点
9 按员工编号查询员工工作信息, 包括其历史工作信息, 返回员工编号, 职位编号和部门编号
10 退出系统
```

```
3
请输入所要查询的员工姓名(例如Yongding Tao):
Yongding Tao
staff_id      first_name    last_name     email          phone_number  hire_date      employment_id  salary      commission_pct
-----
111           Yongding    Tao           MYEMAIL        123.456.7890  2002-10-19 00:00:00 IT_PROG       88888.88      null
```

CSDN @Asker_CXQ

• 功能4

请输入数字进入对应功能:

- 1 查看所有员工基本信息 (选择按员工编号升序排列, 或者按工资降序排列)
- 2 按员工编号查询员工基本信息
- 3 按员工姓名查询员工基本信息
- 4 统计各部门员工最高工资, 最低工资以及平均工资
- 5 查询各部门基本信息
- 6 根据部门编号修改部门名称
- 7 查询各工作地点基本信息
- 8 增加新的工作地点
- 9 按员工编号查询员工工作信息, 包括其历史工作信息, 返回员工编号, 职位编号和部门编号
- 10 退出系统

```
4
section_id      highest_salary  lowest_salary  average_salary
-----
10              5400.00        5400.00        5400.0000000000000000
20              13000.00       6000.00        9500.0000000000000000
40              6500.00        6500.00        6500.0000000000000000
50              3600.00        3600.00        3600.0000000000000000
60              88888.88       4200.00        19614.813333333333
70              10000.00       10000.00       10000.0000000000000000
90              24000.00       17000.00       19333.333333333333
100             12000.00       8200.00        9733.333333333333
110             12000.00       8300.00        10150.0000000000000000
```

CSDN @Asker_CXQ

• 功能5

- 5 查询各部门基本信息
- 6 根据部门编号修改部门名称
- 7 查询各工作地点基本信息
- 8 增加新的工作地点
- 9 按员工编号查询员工工作信息, 包括其历史工作信息, 返回员工编号, 职位编号和部门编号
- 10 退出系统

```
5
section_id      section_name    manager_id     place_id
-----
10              Administration 200            1700
20              Marketing      201            1800
30              Purchasing     114            1700
50              Shipping       121            1500
60              IT             103            1400
70              Public Relations 204            2700
80              Sales          145            2500
90              Executive      100            1700
100             Finance        108            1700
110             Accounting     205            1700
120             Treasury       null           1700
130             Corporate Tax  null           1700
140             Control And Credit null           1700
150             Shareholder Services null           1700
160             Benefits       null           1700
170             Manufacturing   null           1700
180             Construction   null           1700
190             Contracting    null           1700
200             Operations     null           1700
210             IT Support     null           1700
220             NOC            null           1700
230             IT Helpdesk    null           1700
240             Government Sales null           1700
250             Retail Sales   null           1700
260             Recruiting     null           1700
270             Payroll        null           1700
40              human_resources 203            2400
```

CSDN @Asker_CXQ

• 功能6

- 1 查看所有员工基本信息 (选择按员工编号升序排列, 或者按工资降序排列)
- 2 按员工编号查询员工基本信息
- 3 按员工姓名查询员工基本信息
- 4 统计各部门员工最高工资, 最低工资以及平均工资
- 5 查询各部门基本信息
- 6 根据部门编号修改部门名称
- 7 查询各工作地点基本信息
- 8 增加新的工作地点
- 9 按员工编号查询员工工作信息, 包括其历史工作信息, 返回员工编号, 职位编号和部门编号
- 10 退出系统

```
6
请输入部门编号:
40
当前部门名称为:
section_name
-----
human_resources
请输入新的部门名称:
Human_Resources
更新成功!
```

CSDN @Asker_CXQ

5	section_id	section_name	manager_id	place_id
	10	Administration	200	1700
	20	Marketing	201	1800
	30	Purchasing	114	1700
	50	Shipping	121	1500
	60	IT	103	1400
	70	Public Relations	204	2700
	80	Sales	145	2500
	90	Executive	100	1700
	100	Finance	108	1700
	110	Accounting	205	1700
	120	Treasury	null	1700
	130	Corporate Tax	null	1700
	140	Control And Credit	null	1700
	150	Shareholder Services	null	1700
	160	Benefits	null	1700
	170	Manufacturing	null	1700
	180	Construction	null	1700
	190	Contracting	null	1700
	200	Operations	null	1700
	210	IT Support	null	1700
	220	NOC	null	1700
	230	IT Helpdesk	null	1700
	240	Government Sales	null	1700
	250	Retail Sales	null	1700
	260	Recruiting	null	1700
	270	Payroll	null	1700
	40	Human Resources	203	2400

CSDN @Asker_CXQ

• 功能7

7	place_id	street_address	postal_code	city	state_province	state_id
	1000	1297 Via Cola di Rie	00989	Roma	null	IT
	1100	93091 Calle della Testa	10934	Venice	null	IT
	1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
	1300	9450 Kamiya-cho	6823	Hiroshima	null	JP
	1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
	1500	2011 Interiors Blvd	99236	South San Francisco	California	US
	1600	2007 Zagora St	50090	South Brunswick	New Jersey	US
	1700	2004 Charade Rd	98199	Seattle	Washington	US
	1800	147 Spadina Ave	MSV 2L7	Toronto	Ontario	CA
	1900	6092 Boxwood St	YSW 9T2	Whitehorse	Yukon	CA
	2000	40-5-12 Laogianggen	190518	Beijing	null	CN
	2100	1298 Vileparle (E)	490231	Bombay	Maharashtra	IN
	2200	12-98 Victoria Street	2901	Sydney	New South Wales	AU
	2300	198 Clementi North	540198	Singapore	null	SG
	2400	8204 Arthur St	null	London	null	UK
	2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK
	2600	9702 Chester Road	09629850293	Stretford	Manchester	UK
	2700	Schwanthalerstr.	703180925	Munich	Bavaria	DE
	2800	Rua Frei Caneca	1360 01307-002	Sao Paulo	Sao Paulo	BR
	2900	20 Rue des Corps-Saints	1730	Geneva	Geneve	CH
	3000	Murtenstrasse	921 3095	Bern	BE	CH
	3100	Pieter Breughelstraat	8373029SK	Utrecht	Utrecht	NL
	3200	Mariano Escobedo	999111932	Mexico City	Distrito Federal	MX

CSDN @Asker_CXQ

• 功能8

8
您想要插入一个新地点，请根据引导完成数据读入
请输入部门编号：
1001
请输入部门所在地街道地址：
123 Main St
请输入部门所在地邮政编码：
12345
请输入部门所在城市：
New York
请输入部门所在省：
Ny
请输入国家简写（如CN）
US
更新成功！

CSDN @Asker_CXQ

7	place_id	street_address	postal_code	city	state_province	state_id
	1000	1297 Via Cola di Rie	00989	Roma	null	IT
	1100	93091 Calle della Testa	10934	Venice	null	IT
	1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
	1300	9450 Kamiya-cho	6823	Hiroshima	null	JP
	1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
	1500	2011 Interiors Blvd	99236	South San Francisco	California	US
	1600	2007 Zagora St	50090	South Brunswick	New Jersey	US
	1700	2004 Charade Rd	98199	Seattle	Washington	US
	1800	147 Spadina Ave	MSV 2L7	Toronto	Ontario	CA
	1900	6092 Boxwood St	YSW 9T2	Whitehorse	Yukon	CA
	2000	40-5-12 Laogianggen	190518	Beijing	null	CN
	2100	1298 Vileparle (E)	490231	Bombay	Maharashtra	IN
	2200	12-98 Victoria Street	2901	Sydney	New South Wales	AU
	2300	198 Clementi North	540198	Singapore	null	SG
	2400	8204 Arthur St	null	London	null	UK
	2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK
	2600	9702 Chester Road	09629850293	Stretford	Manchester	UK
	2700	Schwanthalerstr.	703180925	Munich	Bavaria	DE
	2800	Rua Frei Caneca	1360 01307-002	Sao Paulo	Sao Paulo	BR
	2900	20 Rue des Corps-Saints	1730	Geneva	Geneve	CH
	3000	Murtenstrasse	921 3095	Bern	BE	CH
	3100	Pieter Breughelstraat	8373029SK	Utrecht	Utrecht	NL
	3200	Mariano Escobedo	999111932	Mexico City	Distrito Federal	MX
	1001	123 Main St	12345	New York	Ny	US

CSDN @Asker_CXQ

• 功能9

```
1 查看所有员工基本信息（选择按员工编号升序排列，或者按工资降序排列）
2 按员工编号查询员工基本信息
3 按员工姓名查询员工基本信息
4 统计各部门员工最高工资，最低工资以及平均工资
5 查询各部门基本信息
6 根据部门编号修改部门名称
7 查询各工作地点基本信息
8 增加新的工作地点
9 按员工编号查询员工工作信息，包括其历史工作信息，返回员工编号，职位编号和部门编号
10 退出系统
9
请输入所要查询的员工编号：
101
staff_id      start_date      end_date      employment_id  section_id
-----
101           1989-09-21 00:00:00 1993-10-27 00:00:00 AC_ACCOUNT    110
101           1993-10-28 00:00:00 1997-03-15 00:00:00 AC_MGR        110
查询成功!
```

CSDN @Asker_CXQ

总结与感悟

最终我成功完成了人力资源管理系统编写，虽然还存在许多的不足，比如，没有使用前端页面进行展示，后端也没有相应地采用SpringBoot等框架规范我的编写，但对于之前没有接触过Java语言的我，已经是一个很大的进步。

通过这个项目，我初步学会了使用Java编程语言，了解了数据库的连接和操作。学会了使用JDBC（Java Database Connectivity）来与数据库进行交互，包括建立数据库连接、执行SQL查询和更新语句等操作。

在开发过程中，我学会了使用JDBC的基本组件，如Connection、Statement和ResultSet，并将其应用于人力资源管理系统中的各种功能，包括员工信息的查询、修改和统计等。

除了数据库操作，我还熟悉了Java的面向对象编程思想，通过定义和使用类、继承和重写方法等概念，实现了系统中的不同角色的功能。在我的编写过程中，我多次重构我的代码，将原来冗长的代码改为用多个类来编写，从而简化代码，理清思路。

这个项目的完成不仅锻炼了我的编程能力，还培养了我的问题解决能力和学习能力。在遇到问题时，我广泛查阅文档和资料，并通过调试和试错的方式找到解决方案。这种持续学习和不断探索的精神是非常重要的，尤其在软件开发领域。

通过这个项目，我也体验到了软件开发的整个过程，包括需求分析、系统设计、编码实现和测试等环节。这对于我进一步了解和掌握软件开发流程和方法论非常有帮助。

总的来说，我在完成这个人力资源管理系统过程中取得了很大的进步，并掌握了许多有价值的技能和知识，为我打开了一扇新的窗口。

最后，感谢孙老师在课程过程中的悉心教导，也感谢在课堂上和我一起讨论学习的同学们。