

**山东大学（威海）**

**工资管理系统**

|  |  |
| --- | --- |
| 班级 | 21未来网络 |
| 学号 | 202100810120 |
| 姓名 | 于靖怿 |
| 指导老师 | 王文玉 |
| 设计完成日期 | 2023.4.18 |

编写时间：2023年4月3日

目录

[一、 选题说明及需求简介 5](#_Toc133245792)

[1.1 选题说明 5](#_Toc133245793)

[1.1.1 编写目的 5](#_Toc133245794)

[1.1.2 项目背景 5](#_Toc133245795)

[1.2 需求介绍 5](#_Toc133245796)

[1.2.1 功能性需求介绍 5](#_Toc133245797)

[1.2.2 非功能性需求介绍 6](#_Toc133245798)

[二、 系统功能模块划分 7](#_Toc133245799)

[2.1 需求分析 7](#_Toc133245800)

[2.1.1 考勤子系统 7](#_Toc133245801)

[2.1.2 员工管理子系统 7](#_Toc133245802)

[2.1.3 工资管理子系统 7](#_Toc133245803)

[2.2 数据流图分析 8](#_Toc133245804)

[2.2.1 顶层数据流图 8](#_Toc133245805)

[2.2.2 一层数据流图 9](#_Toc133245806)

[2.2.3 二层数据流图——用户管理系统 10](#_Toc133245807)

[2.2.4 二层数据流图——员工管理子系统 11](#_Toc133245808)

[2.2.5 二层数据流图——考勤子系统 11](#_Toc133245809)

[2.2.6 二层数据流图——工资管理子系统 12](#_Toc133245810)

[2.3 功能模块图 12](#_Toc133245811)

[2.3.1 用户管理系统的系统结构图 12](#_Toc133245812)

[2.3.2 员工管理系统的系统结构图 13](#_Toc133245813)

[2.3.3 考勤系统的系统结构图 15](#_Toc133245814)

[2.3.4 工资管理系统的系统结构图 16](#_Toc133245815)

[2.3.5 总体系统结构图 17](#_Toc133245816)

[三、 数据库概念结构设计 18](#_Toc133245817)

[3.1 用户表 18](#_Toc133245818)

[3.2 日考勤表 18](#_Toc133245819)

[3.3 月薪表 18](#_Toc133245820)

[3.4 工资配置表 19](#_Toc133245821)

[3.5 部门表 19](#_Toc133245822)

[3.6 员工表 19](#_Toc133245823)

[3.7 全局E-R图 20](#_Toc133245824)

[四、 数据库逻辑结构设计 21](#_Toc133245825)

[4.1 表DailyCheckStat 21](#_Toc133245826)

[4.2 表Depart 21](#_Toc133245827)

[4.3 表Emp 21](#_Toc133245828)

[4.4 表MonthlyWage 21](#_Toc133245829)

[4.5 表Users 22](#_Toc133245830)

[4.6 表WageConfig 22](#_Toc133245831)

[4.7 视图DepartmentManager 22](#_Toc133245832)

[4.8 视图EmpMonthlyWage 22](#_Toc133245833)

[4.9 SQL server关系图 23](#_Toc133245834)

[4.9.1 数据库关系图 23](#_Toc133245835)

[4.9.2 视图关系图 24](#_Toc133245836)

[五、 系统实现过程及完成效果介绍 25](#_Toc133245837)

[5.1 系统实现过程 25](#_Toc133245838)

[5.1.1 数据库 25](#_Toc133245839)

[5.1.2 后端SpringBoot 27](#_Toc133245840)

[5.2 完成效果介绍 28](#_Toc133245841)

[5.2.1 部分表格数据可视化（非前端） 28](#_Toc133245842)

[5.2.2 SQL server数据库部分数据展示 29](#_Toc133245843)

[5.2.3 数据库结构展示 31](#_Toc133245844)

[六、 总结 33](#_Toc133245845)

[七、 参考资料 34](#_Toc133245846)

[7.1 数据库脚本 34](#_Toc133245847)

[7.2 参考资料及网页 62](#_Toc133245848)

[八、 致谢 63](#_Toc133245849)

# 选题说明及需求简介

## 选题说明

### 编写目的

工资管理系统是一种应用软件，它可以帮助企业管理员工的工资、奖金、福利等信息，以及进行相关的统计和分析。工资管理系统的核心是一个数据库，它存储了员工的基本信息、工作时间、考勤情况、薪酬标准、税率等数据。我选择工资管理系统作为数据库课程设计的目的有以下几点：

* 工资管理系统是一个实用性强、功能丰富、数据量大的软件，它可以让我充分运用所学的数据库知识和技能，如数据模型设计、关系代数运算、SQL语言编程、存储过程和触发器等。
* 工资管理系统是一个涉及多个领域和专业的软件，它可以让我拓宽视野，了解不同行业和部门的需求和规范，如人力资源管理、财务会计、税务法规等。
* 工资管理系统是一个具有挑战性和创新性的软件，它可以让我锻炼思维能力和解决问题能力，如如何保证数据的安全性、完整性和一致性，如何提高数据的查询效率和处理速度，如何设计友好的用户界面和交互方式等。

综上，我认为选择工资管理系统作为数据库课程设计是一个合理而有意义的决定，它可以帮助我提高数据库方面的理论水平和实践能力，为将来从事软件开发或相关领域的工作打下坚实的基础。

### 项目背景

随着计算机技术的发展，企业管理越来越趋于自动化，而工资管理是企业管理的重要组成部分，它与企业的财务管理、人事管理有着密切的联系。由于企业员工工资结构存在一定的规律性，工资发放在时间上和操作上存在一定的重复性，这使得企事业单位工资管理计算机化成为可能。本系统在此基础上实现了工资信息查询、人员信息管理、报表生成等功能。用户只要使用浏览器访问相关站点，输入正确的用户名和密码成功登录后，普通用户便可查询自己的工资情况和其他相关信息。对管理员来说使用起来也非常的方便快捷，可以进行科学、合理的对员工工资和人员进行管理。

## 需求介绍

### 功能性需求介绍

我管理系统可以实现包括但不限于以下的功能：

1. 员工信息管理

系统可以录入、修改、删除和查询员工的基本信息，如姓名、性别、年龄、职位、部门、入职时间、图片等。

1. 员工考勤统计

系统可以录入员工的考勤情况，预留相关硬件的输入接口，从而合理、精确计算出每位员工的工资详情。

1. 工资计算管理

系统可以根据员工的基本信息和考勤情况，按照企业的工资制度，自动计算员工的应发工资、实发工资和各项扣除项，如个人所得税、社会保险等。同时也预留相关自定义工资发放的接口。

1. 工资发放管理

系统可以生成员工的工资条，打印或发送给员工，并记录员工的工资发放情况，如发放日期、发放方式等。

1. 工资统计管理

系统可以对员工的工资数据进行统计分析，生成各种报表，如部门工资报表、个人工资报表、月度工资报表等，以便于企业进行财务管理和审计。

### 非功能性需求介绍

工资管理系统的主要用户是企业的人力资源部门和财务部门，他们可以通过系统的图形用户界面或网页界面进行操作。系统应该具有以下非功能性需求：

1. 可靠性

系统应该能够保证数据的完整性和一致性，避免数据丢失或错误。

1. 安全性

系统应该能够保护数据的机密性和完整性，防止数据被未授权的用户访问或修改。

1. 易用性

系统应该提供友好和简洁的用户界面，方便用户进行操作和查询。

1. 可维护性

系统应该遵循软件工程的规范和标准，方便后期的维护和升级。

# 系统功能模块划分

## 需求分析

经过合理的分析和研究，我系统最终确定了每个模块的功能，整个系统划分为三个子系统，每个子系统的功能如下：

### 考勤子系统

#### 考勤信息的导入

考虑到工资系统的使用人员水平及相关需求，系统预留了考勤信息的读入接口，因为考勤信息通常是硬件生成的，所以这里只是预留硬件接口。相关部门管理人员处理硬件数据后，即可通过导入excel表格的方式向系统传入考勤信息。其中excel表格的模板由系统提供。然后系统会将用户导入的excel做进一步处理，将其转化成月考勤表供工资子系统使用。

#### 考勤信息管理

上传excel文件后，相关管理员还可以对考勤信息进行修改，系统提供了修改功能和重新上传等功能，方便管理人员操作。

### 员工管理子系统

#### 员工基本信息管理

该功能可以录入和修改员工的姓名、性别、年龄、身份证号、联系方式、入职日期、离职日期等基本信息，以及员工所属的部门、岗位、职级等组织结构信息。该功能还可以根据不同的条件查询员工的基本信息，如按姓名、部门、岗位等进行模糊查询或精确查询，以及导出员工的基本信息表格或打印员工的基本信息卡片。

#### 员工职务信息管理

该功能可以录入和修改员工的职务、部门等信息。

### 工资管理子系统

#### 工资计算

根据员工的基本信息、考勤情况、绩效评价、奖惩记录等，按照工资制度和政策，自动计算员工的应发工资、实发工资和各项扣除。

#### 工资统计

对员工的工资数据进行汇总、分析和报表生成，提供各种维度的工资报告，如个人工资明细、部门工资汇总、公司工资总表等。

#### 工资查询

提供员工和管理者对个人或部门的工资情况进行查询的功能，支持多种查询条件和查询方式，如按月份、按姓名、按部门等。

#### 工资核对

对员工的工资数据进行审核和核对，确保工资的准确性和合法性，处理各种异常和差错，如漏发、错发、多发等。

#### 基本工资信息管理

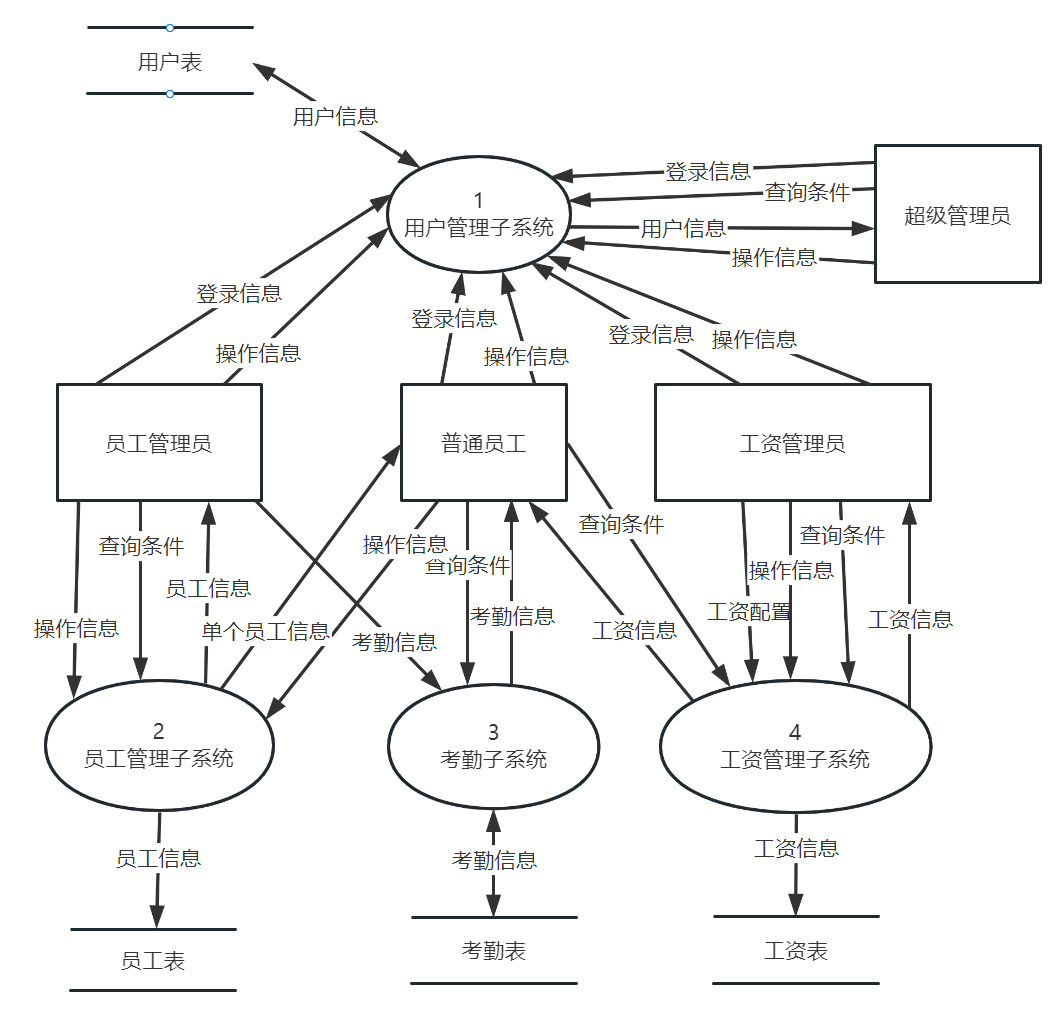
该功能允许管理员对基本工资情况进行查看、修改和添加。此处基本工资情况意为职务的默认工资，即默认工资、默认加班费用等。当然也可以使用自定义值。

## 数据流图分析

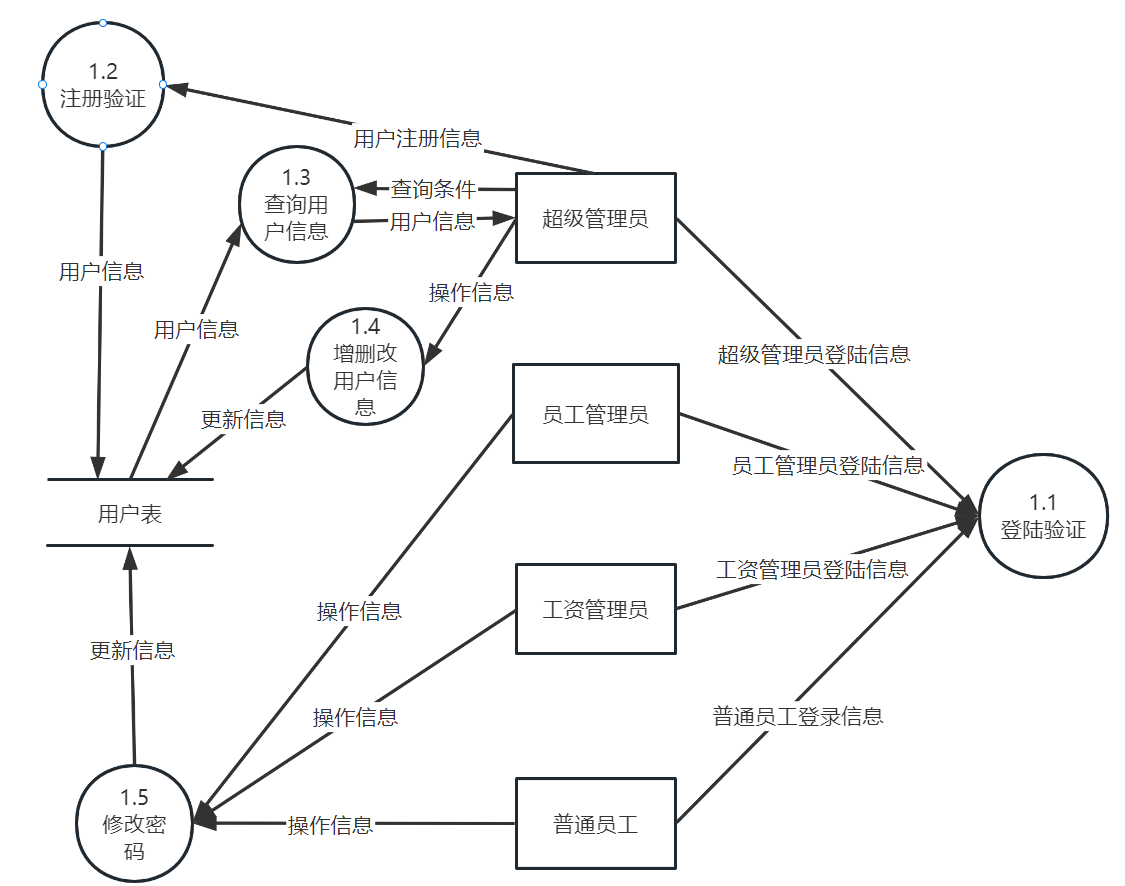
### 顶层数据流图



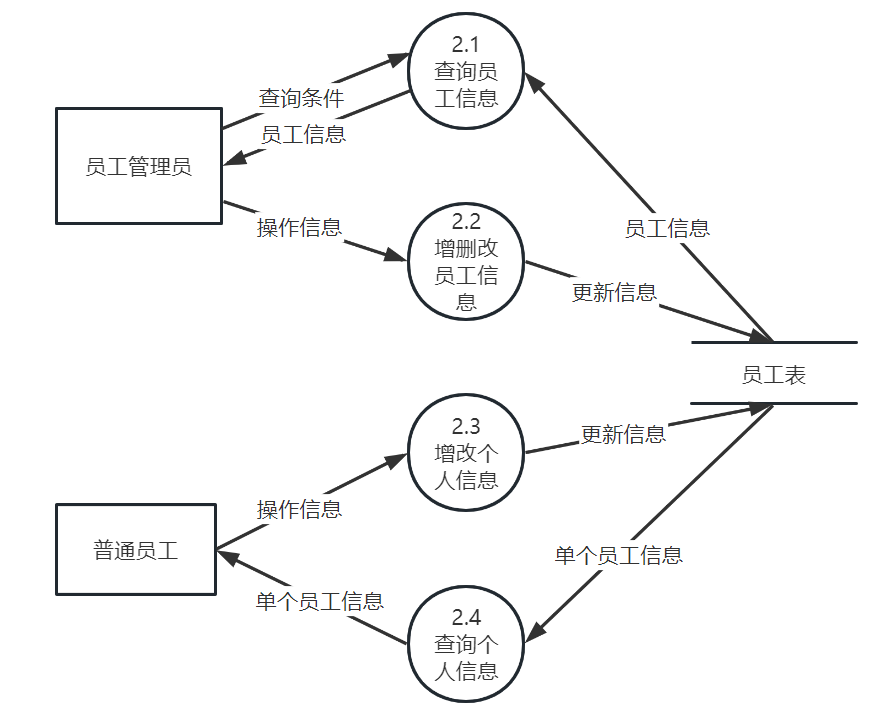
### 一层数据流图



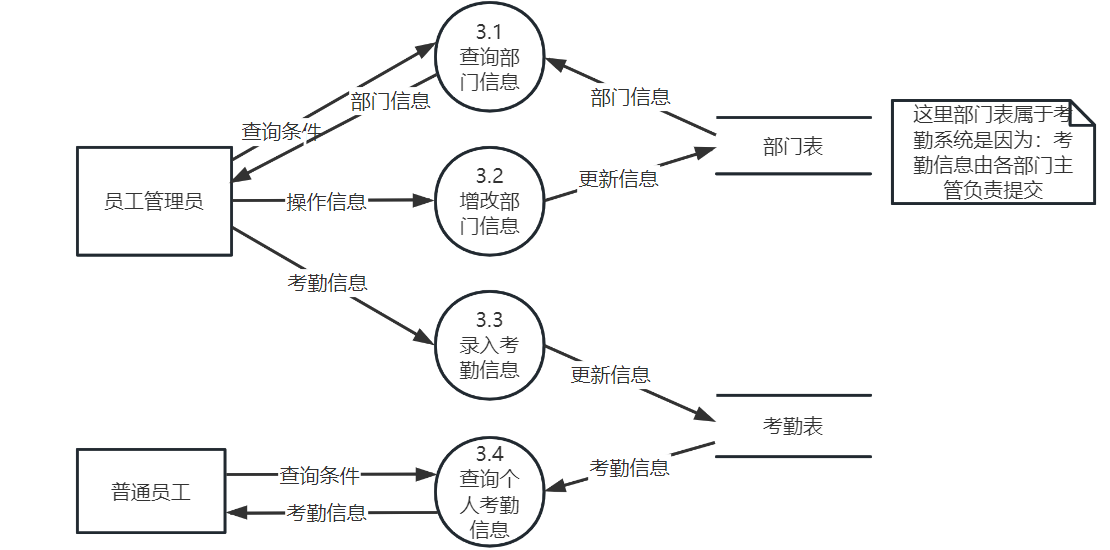
### 二层数据流图——用户管理系统



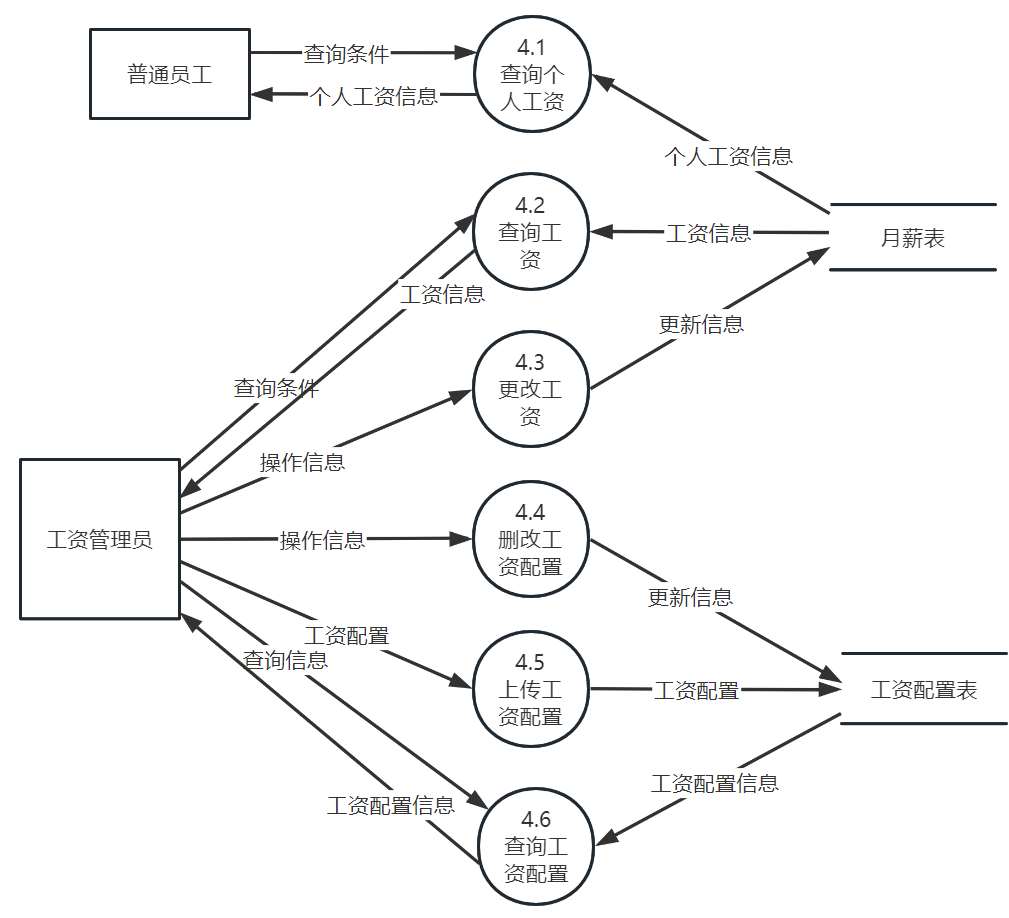
### 二层数据流图——员工管理子系统



### 二层数据流图——考勤子系统



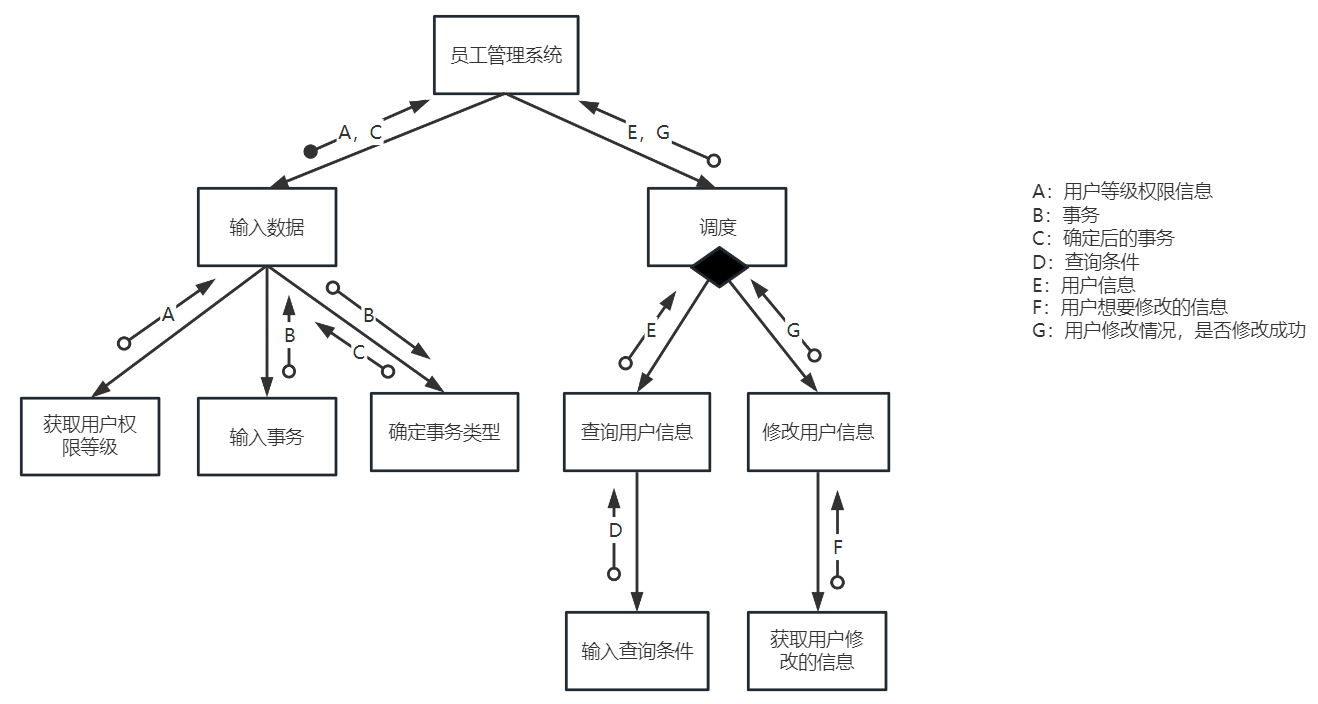
### 二层数据流图——工资管理子系统



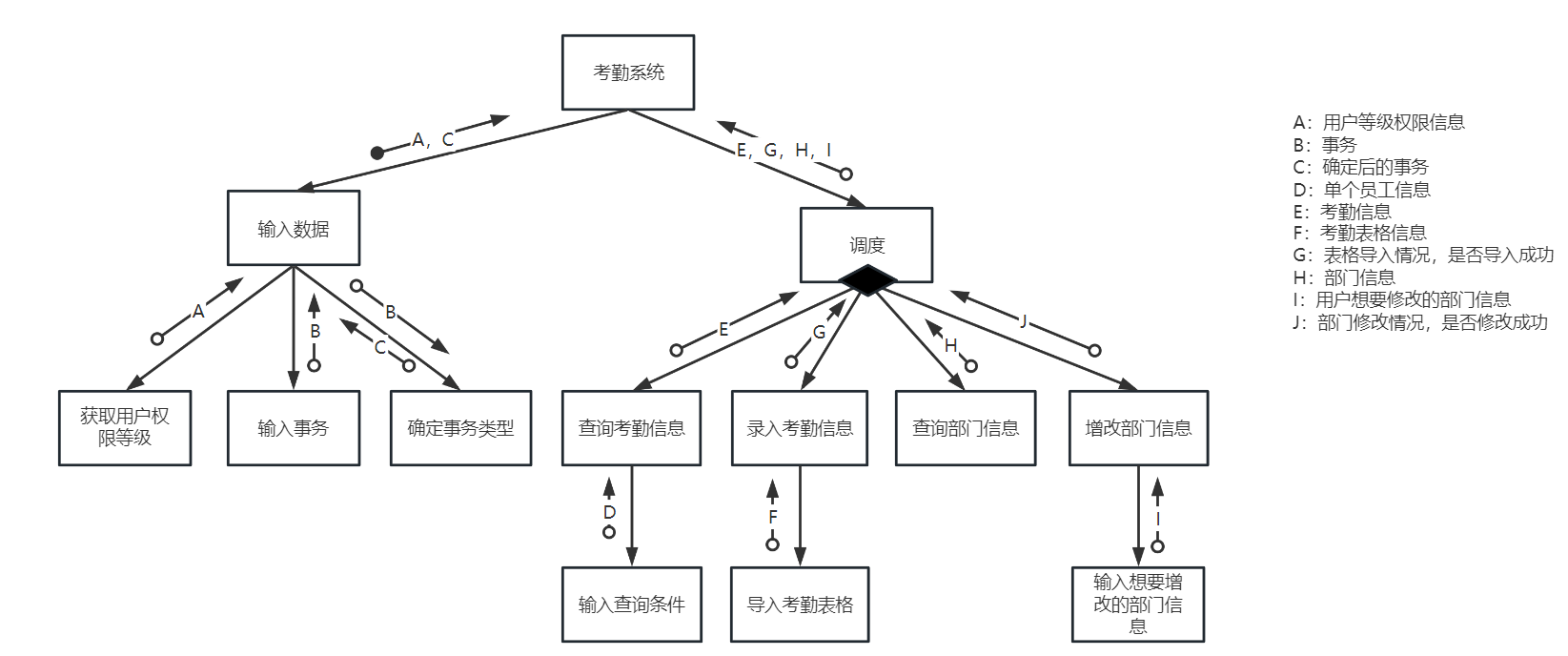
## 功能模块图

### 用户管理系统的系统结构图

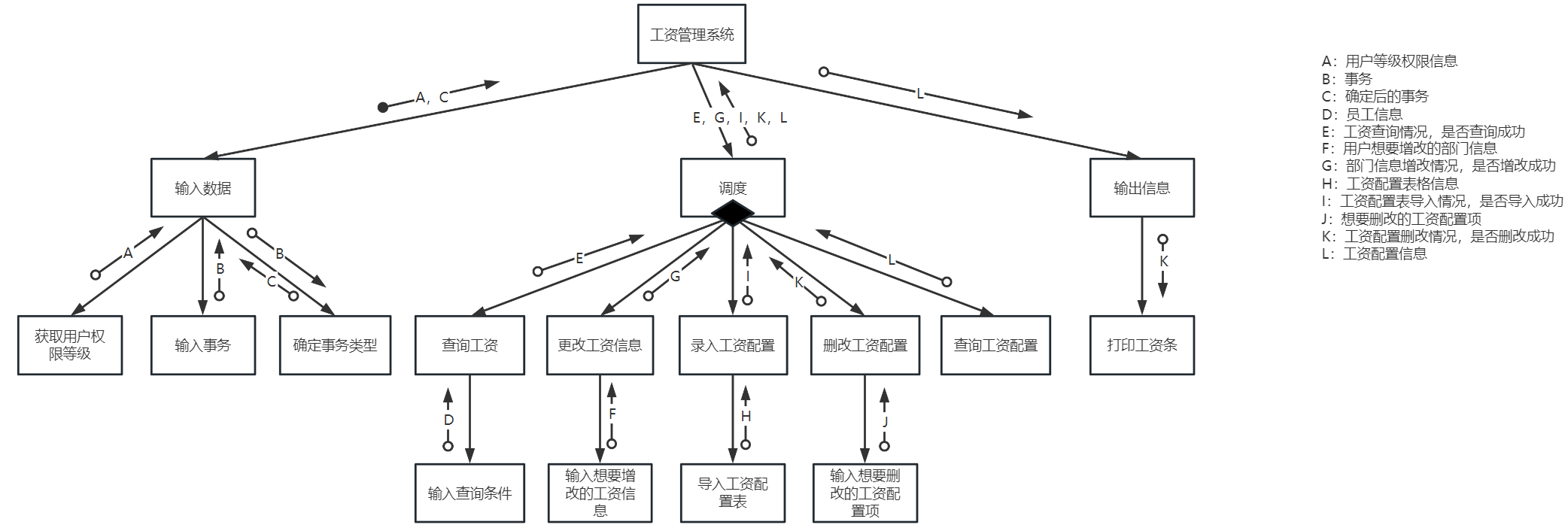
### 员工管理系统的系统结构图



### 考勤系统的系统结构图



### 工资管理系统的系统结构图

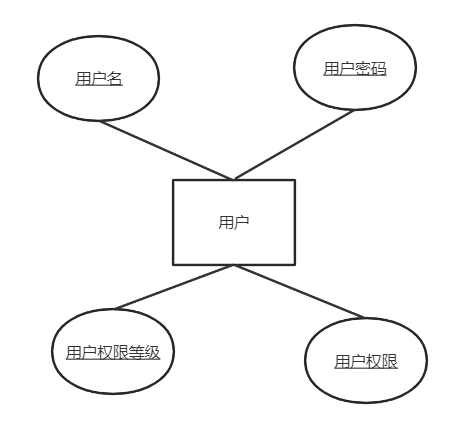


### 总体系统结构图

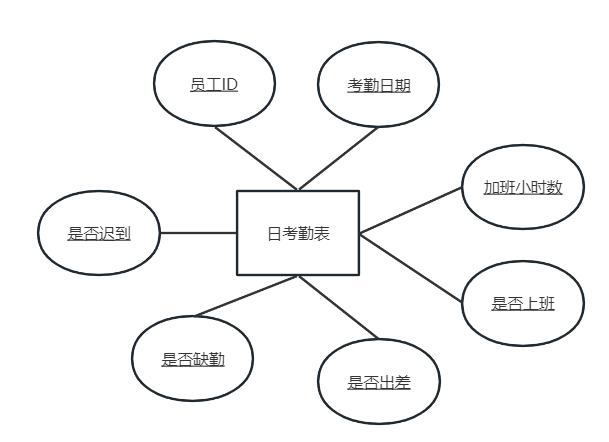


# 数据库概念结构设计

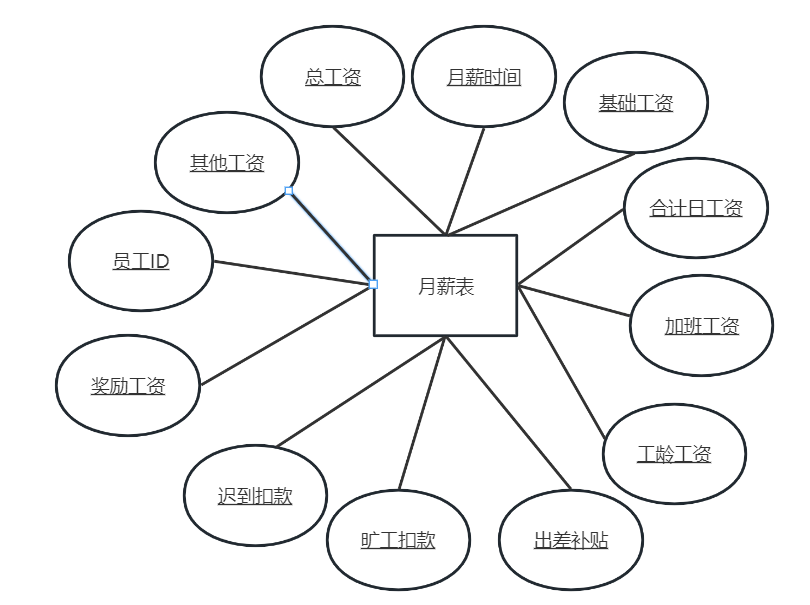
## 用户表



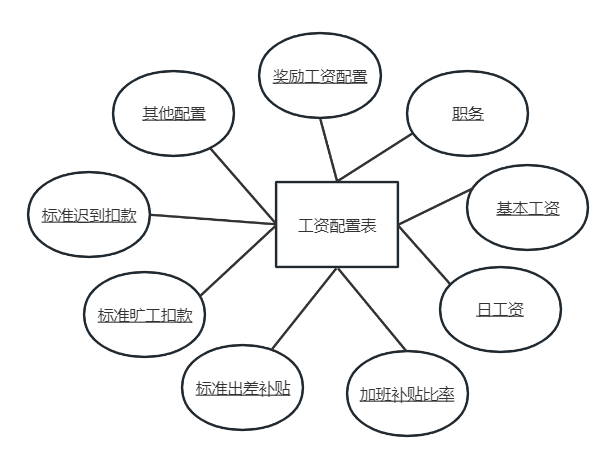
## 日考勤表



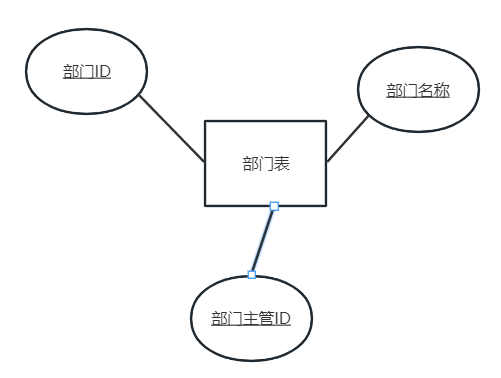
## 月薪表



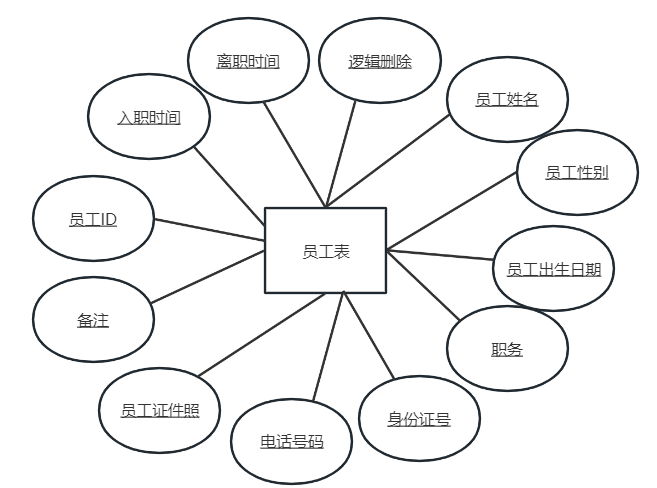
## 工资配置表



## 部门表



## 员工表



## 全局E-R图



# 数据库逻辑结构设计

## 表DailyCheckStat

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **主键** | **默认值** | **说明** |
| 1 | EmpID | int | 10 | 0 | N | **Y** |  | 员工ID |
| 2 | CheckDate | date | 10 | 0 | N | **Y** |  | 每日考勤日期 |
| 3 | OvertimeHour | float | 53 | 0 | Y | N |  | 加班小时数 |
| 4 | WorkDay | tinyint | 3 | 0 | Y | N |  | 是否上班 |
| 5 | BussinessTripDay | tinyint | 3 | 0 | Y | N |  | 是否出差 |
| 6 | AbsenceDay | tinyint | 3 | 0 | Y | N |  | 是否旷工 |
| 7 | LateDay | tinyint | 3 | 0 | Y | N |  | 迟到次数 |

## 表Depart

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **主键** | **默认值** | **说明** |
| 1 | DepartID | int identity | 10 | 0 | N | **Y** |  | 部门ID |
| 2 | DepartName | nvarchar | 20 | 0 | N | N |  | 部门名称 |
| 3 | DepartManagerID | int | 10 | 0 | Y | 外键 |  | 部门主管ID |

## 表Emp

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **主键** | **默认值** | **说明** |
| 1 | EmpID | int identity | 10 | 0 | N | **Y** |  |  |
| 2 | EmpName | nvarchar | 5 | 0 | N | N |  |  |
| 3 | EmpSex | nchar | 4 | 0 | N | N |  |  |
| 4 | EmpBirth | date | 10 | 0 | N | N |  |  |
| 5 | Duty | nvarchar | 20 | 0 | N | 外键 |  | 职务 |
| 6 | DepartID | int | 10 | 0 | Y | 外键 |  | 所属部门ID |
| 7 | IdentityID | varchar | 20 | 0 | N | N |  | 身份证号 |
| 8 | EmpTele | varchar | 20 | 0 | N | N |  |  |
| 9 | EmpPic | image | 2147483647 | 0 | Y | N |  | 员工图片 |
| 10 | EmpRemark | nvarchar | 200 | 0 | Y | N |  | 备注 |
| 11 | HireDate | date | 10 | 0 | N | N | (getdate()) | 入职时间 |
| 12 | LeaveDate | date | 10 | 0 | Y | N |  | 离职时间 |
| 13 | LogicalDelete | tinyint | 3 | 0 | Y | N | ((0)) | 逻辑删除标志 |

## 表MonthlyWage

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **主键** | **默认值** | **说明** |
| 1 | EmpID | int | 10 | 0 | N | **Y** |  |  |
| 2 | WageDate | date | 10 | 0 | N | **Y** |  |  |
| 3 | BaseWage | money | 19 | 4 | Y | N |  | 基本工资 |
| 4 | DayWage | money | 19 | 4 | Y | N |  | 日薪合计 |
| 5 | OvertimeWage | money | 19 | 4 | Y | N |  | 加班工资 |
| 6 | SeniorityWage | money | 19 | 4 | Y | N |  | 工龄工资 |
| 7 | BussinessTripWage | money | 19 | 4 | Y | N |  | 出差补贴 |
| 8 | AbsenceWage | money | 19 | 4 | Y | N |  | 旷工扣款 |
| 9 | LateWage | money | 19 | 4 | Y | N |  | 迟到扣款 |
| 10 | BonusWage | money | 19 | 4 | Y | N |  | 奖励工资 |
| 11 | EtcWage | money | 19 | 4 | Y | N |  | 其他工资 |
| 12 | TotalWage | money | 19 | 4 | Y | N |  | 工资总和 |

## 表Users

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **主键** | **默认值** | **说明** |
| 1 | UserName | nvarchar | 20 | 0 | N | **Y** |  |  |
| 2 | UserPassword | varchar | 20 | 0 | N | N |  |  |
| 3 | UserRight | nvarchar | 10 | 0 | Y | N |  | 用户权限 |
| 4 | UserRightLevel | int | 10 | 0 | N | N | ((0)) | 用户权限等级 |

## 表WageConfig

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **主键** | **默认值** | **说明** |
| 1 | Duty | nvarchar | 20 | 0 | N | **Y** |  | 职务 |
| 2 | BaseWage | money | 19 | 4 | N | N |  | 基本工资 |
| 3 | DayWage | money | 19 | 4 | N | N |  | 日薪 |
| 4 | OvertimeRatio | float | 53 | 0 | N | N |  | 加班补贴比率 |
| 5 | BussinessTripStd | money | 19 | 4 | N | N |  | 标准出差补贴 |
| 6 | AbsenceStd | money | 19 | 4 | N | N |  | 标准旷工扣款 |
| 7 | LateStd | money | 19 | 4 | N | N |  | 标准迟到扣款 |
| 8 | Bonus | money | 19 | 4 | N | N |  | 标准奖励 |
| 9 | EtcStd | money | 19 | 4 | N | N |  | 其他 |

## 视图DepartmentManager

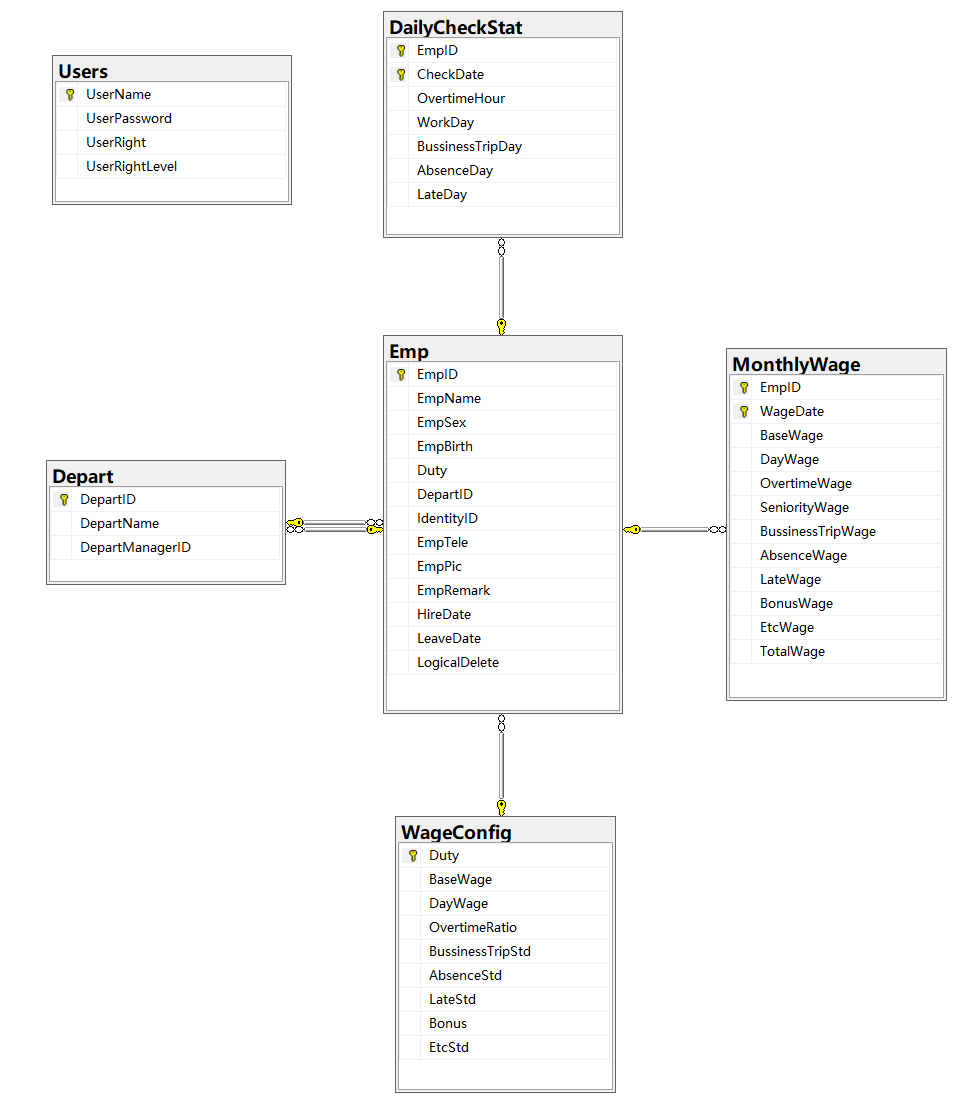
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **默认值** | **说明** |
| 1 | DepartID | nvarchar | 20 | 0 | N |  | 职务 |
| 2 | DepartName | nvarchar | 20 | 0 | N |  | 部门名称 |
| 3 | EmpID | int identity | 10 | 0 | N |  |  |
| 4 | EmpName | nvarchar | 5 | 0 | N |  |  |

## 视图EmpMonthlyWage

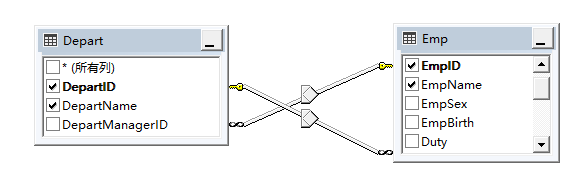
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **名称** | **数据类型** | **长度** | **小数位** | **允许空值** | **默认值** | **说明** |
| 1 | EmpID | int | 10 | 0 | N |  |  |
| 2 | EmpName | nvarchar | 5 | 0 | N |  |  |
| 3 | DepartName | nvarchar | 20 | 0 | N |  |  |
| 4 | Duty | nvarchar | 20 | 0 | N |  |  |
| 5 | WageMonth | date | 10 | 0 | N |  |  |
| 6 | TotalWage | money | 19 | 4 | N |  | 总工资计算列 |
| 7 | BaseWage | money | 19 | 4 | N |  | 基本工资 |
| 8 | DayWage | money | 19 | 4 | N |  | 日薪 |
| 9 | OvertimeWage | money | 19 | 4 | N |  | 加班工资 |
| 10 | SeniorityWage | money | 19 | 4 | Y |  | 工龄工资 |
| 11 | BussinessTripWage | money | 19 | 4 | Y |  | 出差补贴 |
| 12 | AbsenceWage | money | 19 | 4 | Y |  | 旷工扣款 |
| 13 | LateWage | money | 19 | 4 | Y |  | 迟到扣款 |
| 14 | BonusWage | money | 19 | 4 | Y |  | 奖励工资 |
| 15 | EtcWage | money | 19 | 4 | Y |  | 其他工资 |
| 16 | WageDate | date | 10 | 0 | N |  |  |

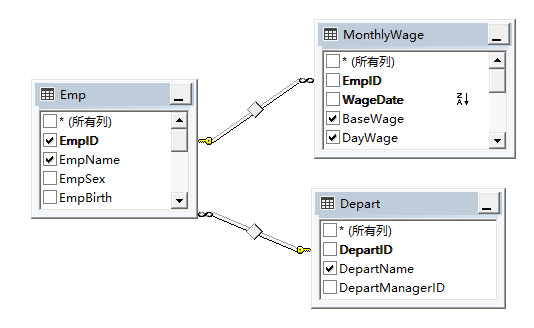
## SQL server关系图

### 数据库关系图



### 视图关系图





# 系统实现过程及完成效果介绍

## 系统实现过程

### 数据库

#### 表功能细节



图 数据库功能实现的脚本文件

建立数据库后直接编写脚本完成数据库的相关功能，包括但不限于各类函数、触发器、存储过程等。从这里的修改日期可以看出这些工作不是一蹴而就的。

#### 表数据生成

由于没有相关真实数据，所以数据库的数据来源于其他途径，包括但不限于ChatGPT、new Bing，以及使用其他编程语言随机生成。以下是一部分数据的生成途径：

public class **RandomInsertDCS** {  
 public static void main(**String**[] args) throws **IOException** {  
 int Iter = 100000;  
 **Random** random = new Random();  
  
 **SimpleDateFormat** sdf = new SimpleDateFormat("yyyy-MM-dd");  
 **Format** decimalFormat = new DecimalFormat("#0.0");  
  
 **BufferedWriter** bw = new BufferedWriter(new FileWriter("./RandomInsertDCS.sql", true));  
  
 while (Iter-- > 0) {  
 int workDay = random.nextInt(2);  
 int bussinessTripDay = random.nextInt(2);  
 int absenceDay = random.nextInt(2);  
 int lateDay = random.nextInt(2);  
 double overtimeHour = random.nextDouble(5);  
 if ((workDay == 1 && bussinessTripDay == 0 && absenceDay == 0) ||  
 (workDay == 0 && bussinessTripDay == 1 && absenceDay == 0) ||  
 (workDay == 0 && bussinessTripDay == 0 && absenceDay == 1)) {  
 bw.write("INSERT INTO DailyCheckStat (EmpID, CheckDate, OvertimeHour, WorkDay, BussinessTripDay, AbsenceDay, LateDay) VALUES (" + (random.nextInt(8) + 41) +  
 ", '" + sdf.format(*between*(new Date(120, **Calendar**.*JANUARY*, 1), new Date(123, **Calendar**.*MARCH*, 29))) +  
 "', " + decimalFormat.format(workDay \* overtimeHour) +  
 ", " + workDay +  
 ", " + bussinessTripDay +  
 ", " + absenceDay +  
 ", " + lateDay + ");**\n**");  
 }  
 }  
 }  
  
 public static **Date** between(**Date** start, **Date** end) {  
 long startMillis = start.getTime();  
 long endMillis = end.getTime();  
 long randomMillis = **ThreadLocalRandom** .*current*()  
 .nextLong(startMillis, endMillis);  
  
 return new Date(randomMillis);  
 }

}

代码 使用Java随机生成考勤数据

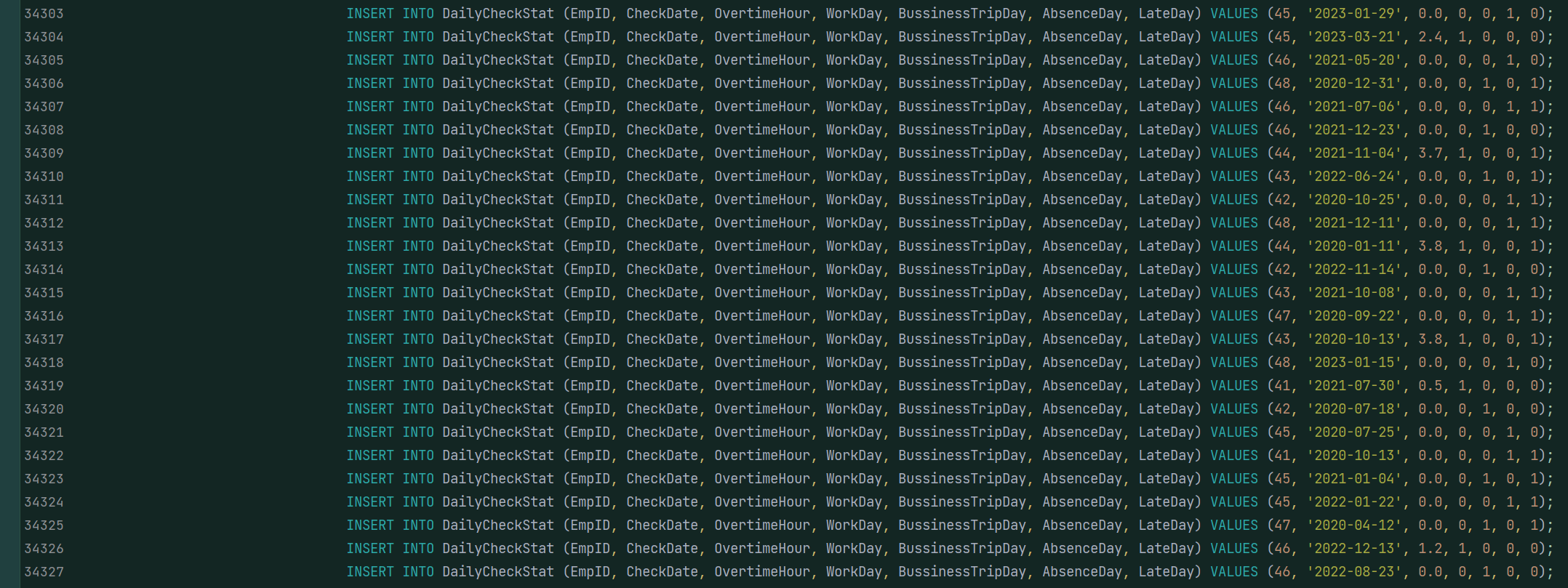


图 Java随机生成50000条考勤记录（部分）



图 使用ChatGPT生成模拟员工信息

### 后端SpringBoot

#### 逆向工程

后端SpringBoot采用AutoGenerator自动根据数据库表结构，逆向生成相关Java类对象，自动生成Controller、Mapper、Service模板。

│ │ │ ├─controller

│ │ │ ├─entity

│ │ │ ├─mapper

│ │ │ │ └─xml

│ │ │ └─service

│ │ │ └─impl

图 SpringBoot后端代码结构

上述目录支持一键自动生成。

#### 常规工具类



图 后端SpringBoot工具类

从黄色的修改时间中可以看出，这些工作不是一蹴而就的。

#### 部分后端接口

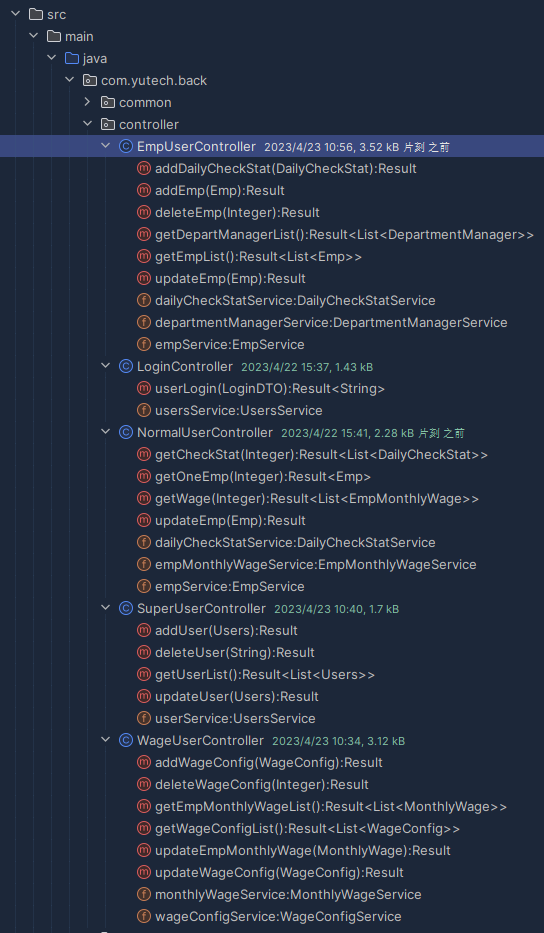


图 后端控制器结构

## 完成效果介绍

### 部分表格数据可视化（非前端）

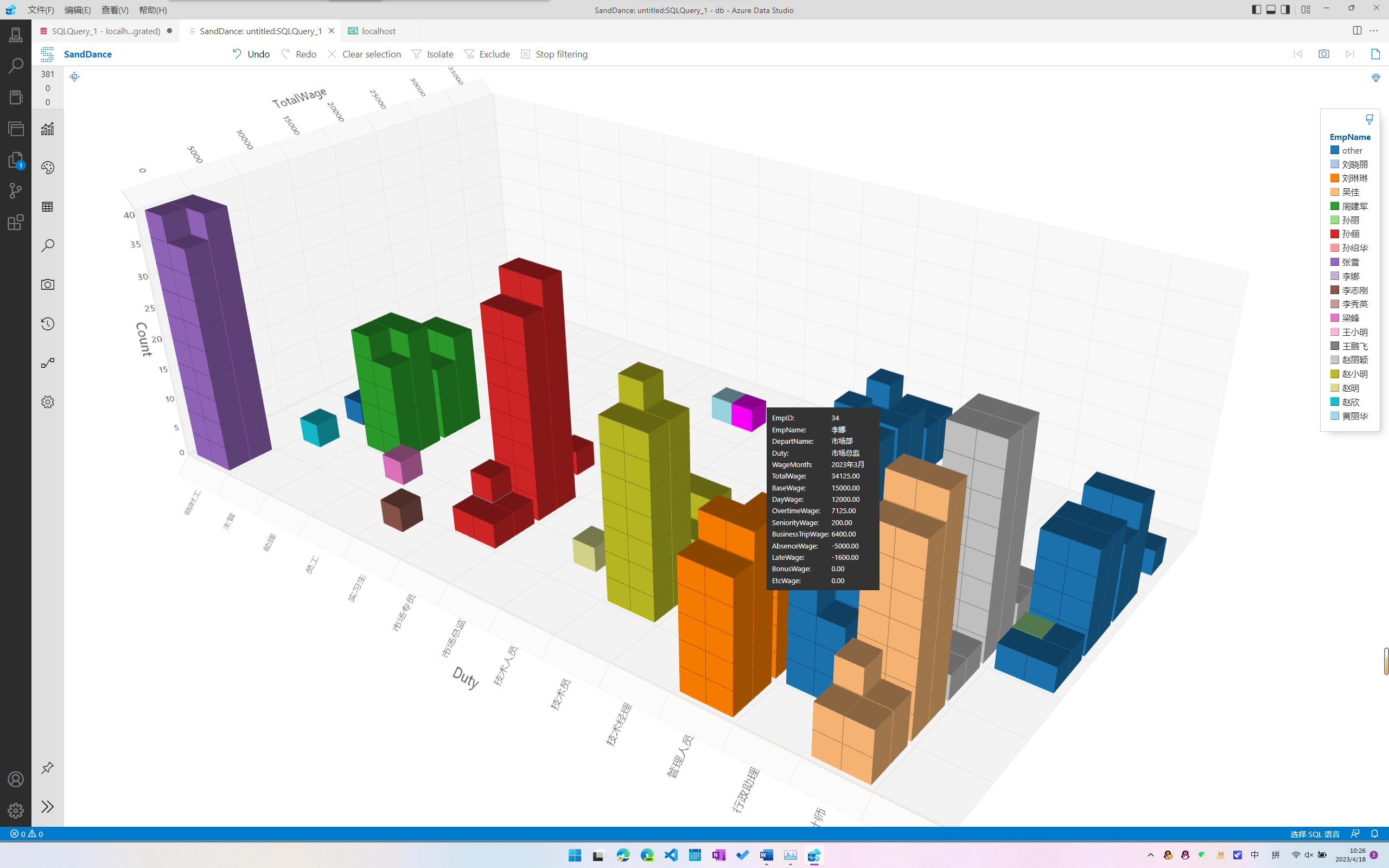


图 月薪表视图

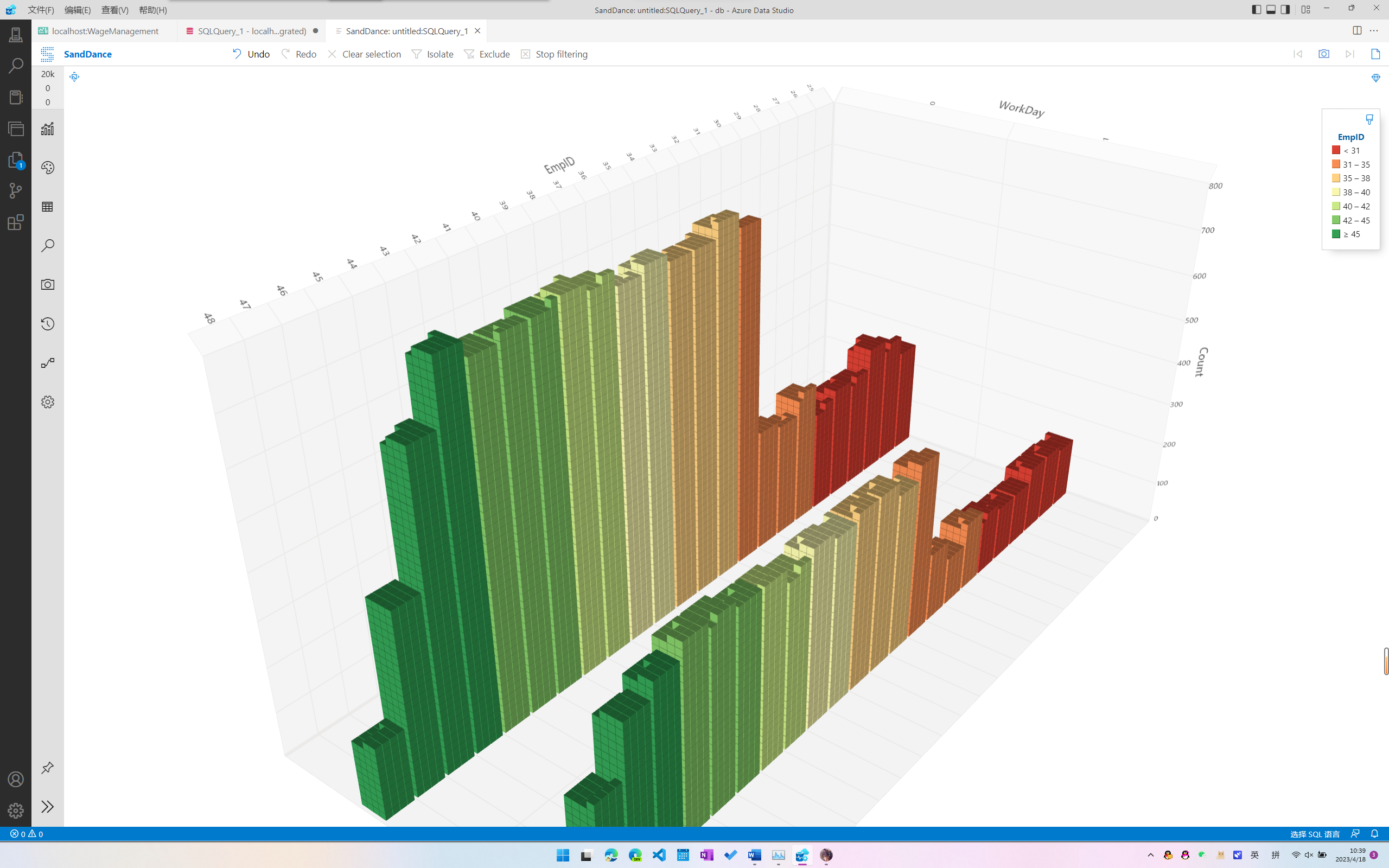


图 员工上班天数与未上班天数

### SQL server数据库部分数据展示

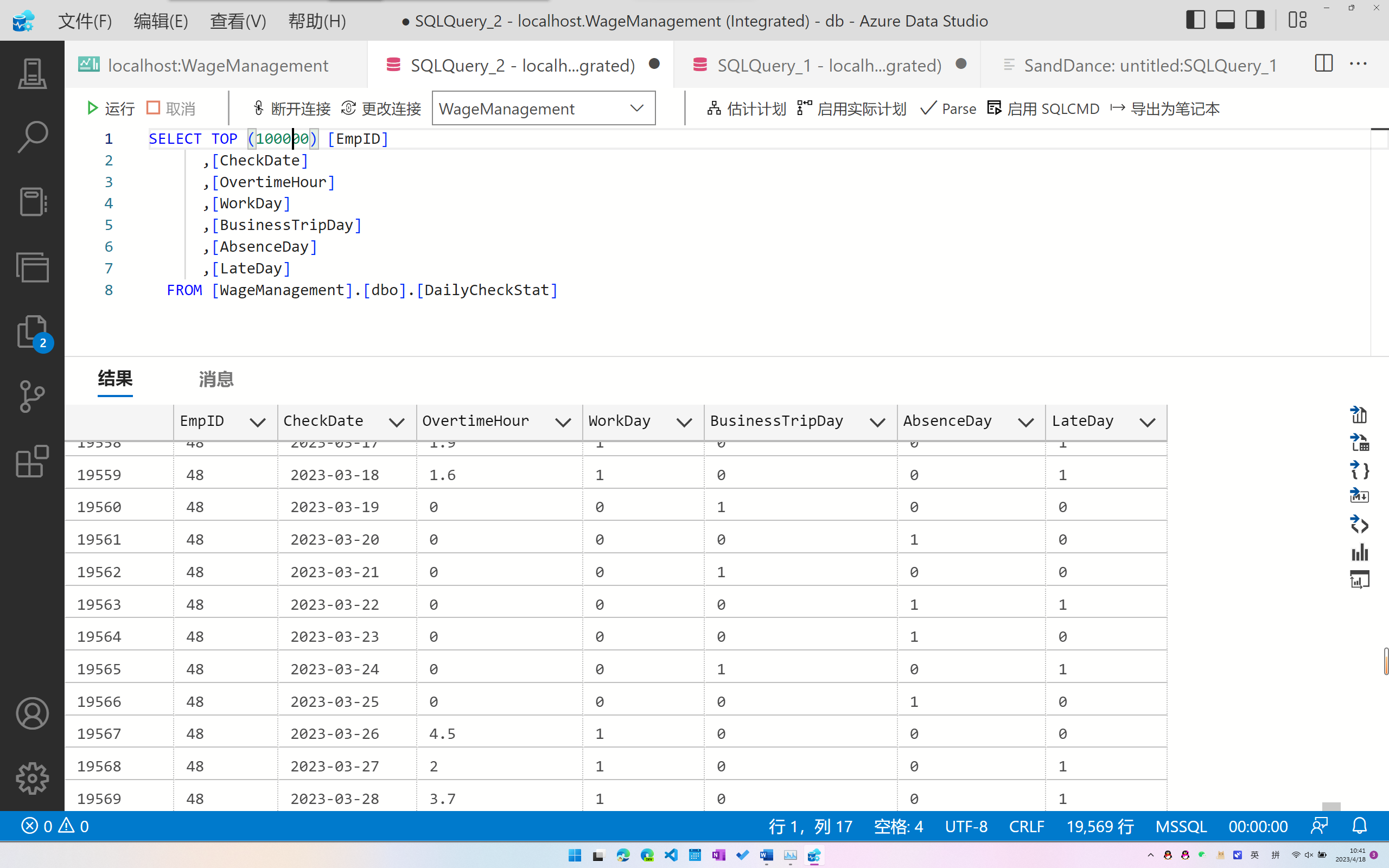


图 考勤表中有上万条数据

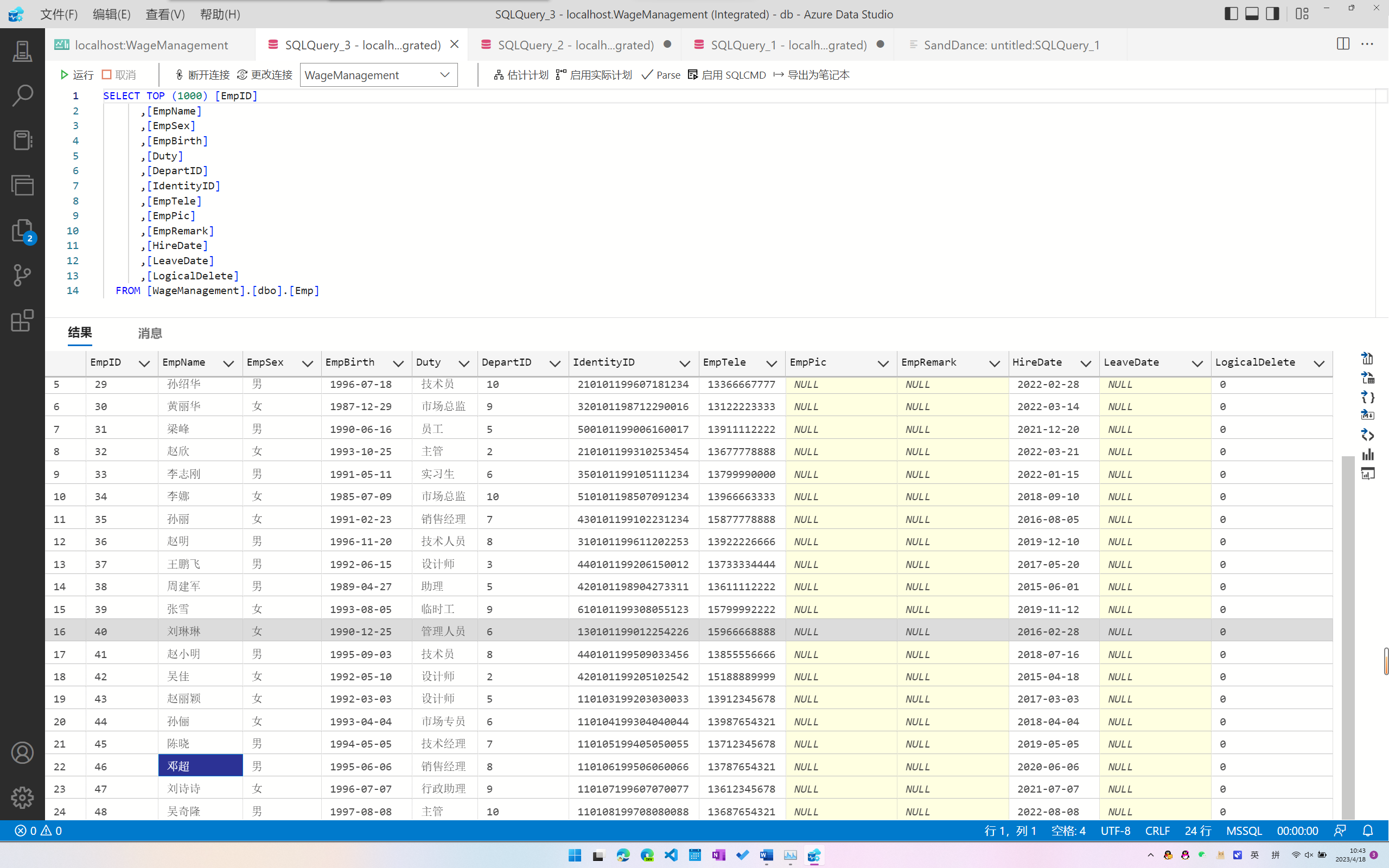


图 员工表中的数据

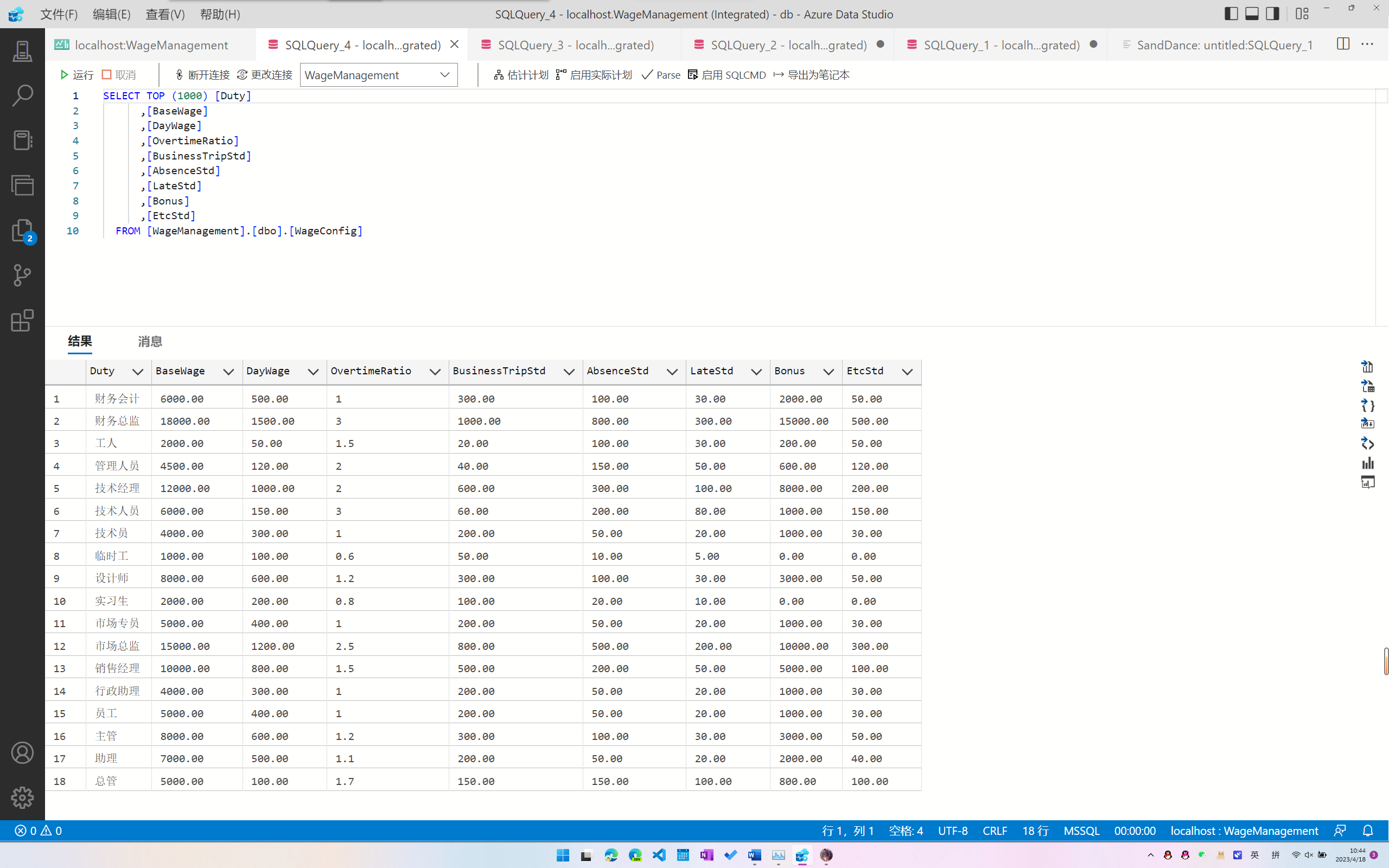


图 工资配置表中的数据

### 数据库结构展示

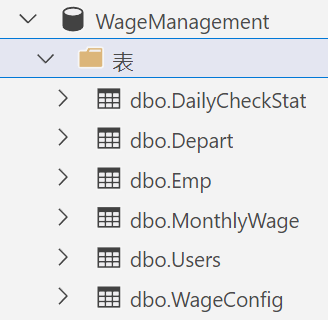


图 表结构展示



图 视图结构展示



图 存储过程展示

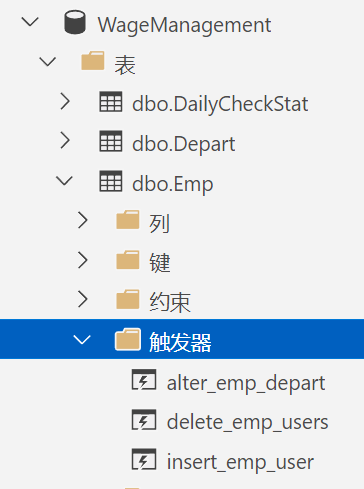


图 表级触发器展示1

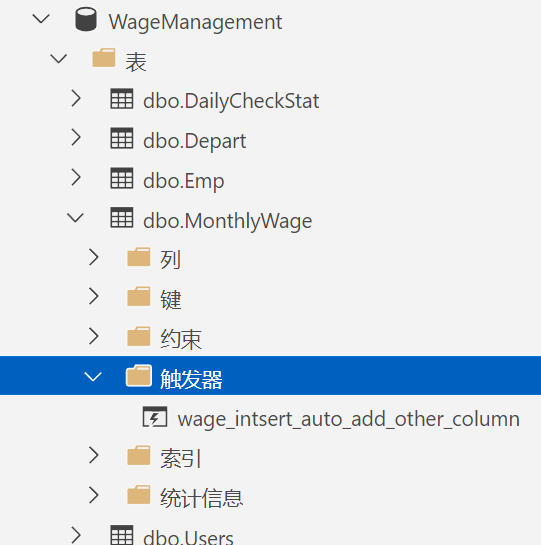


图 表级触发器展示2

### 前端网页展示

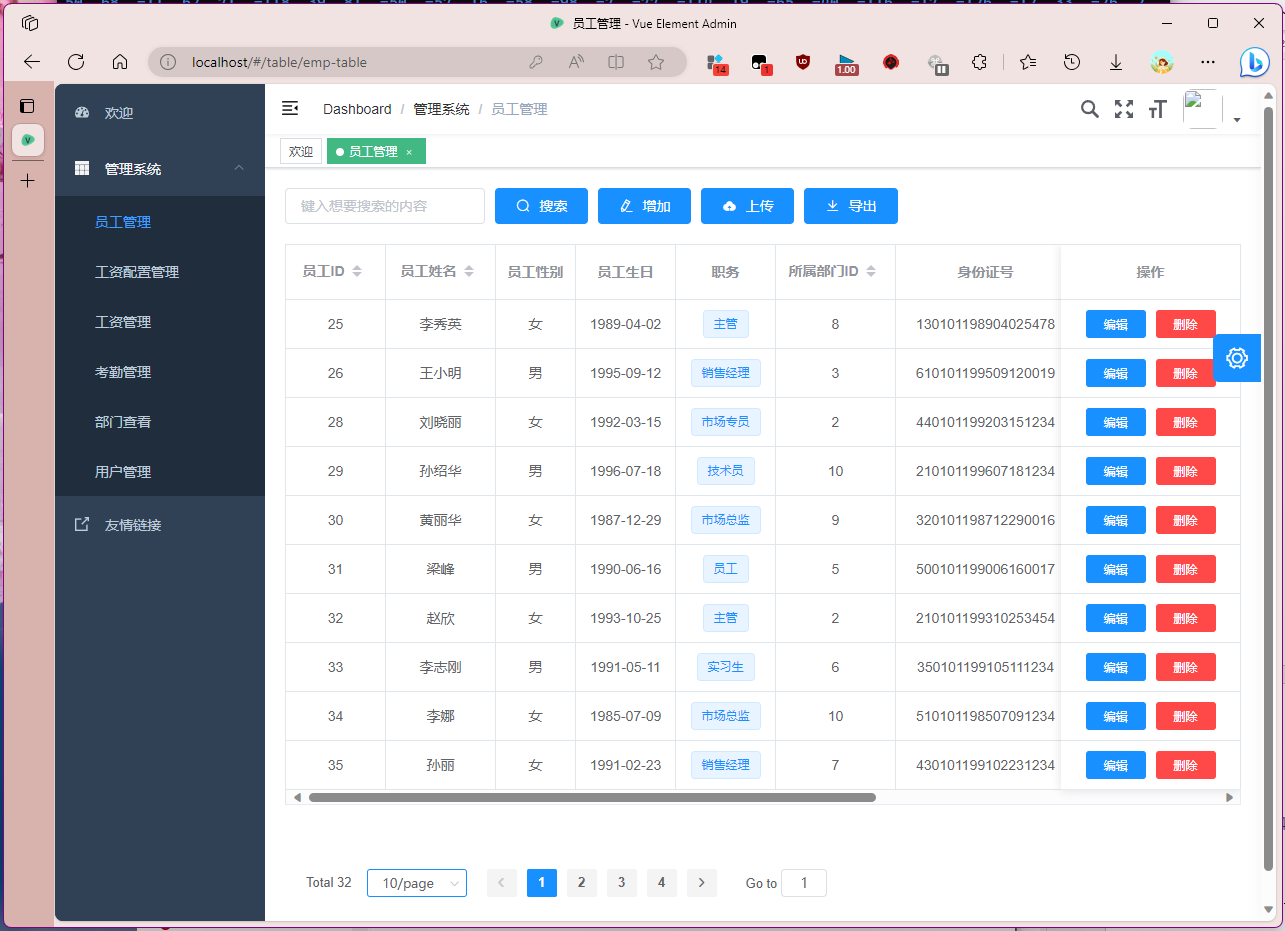


图 管理表格展示

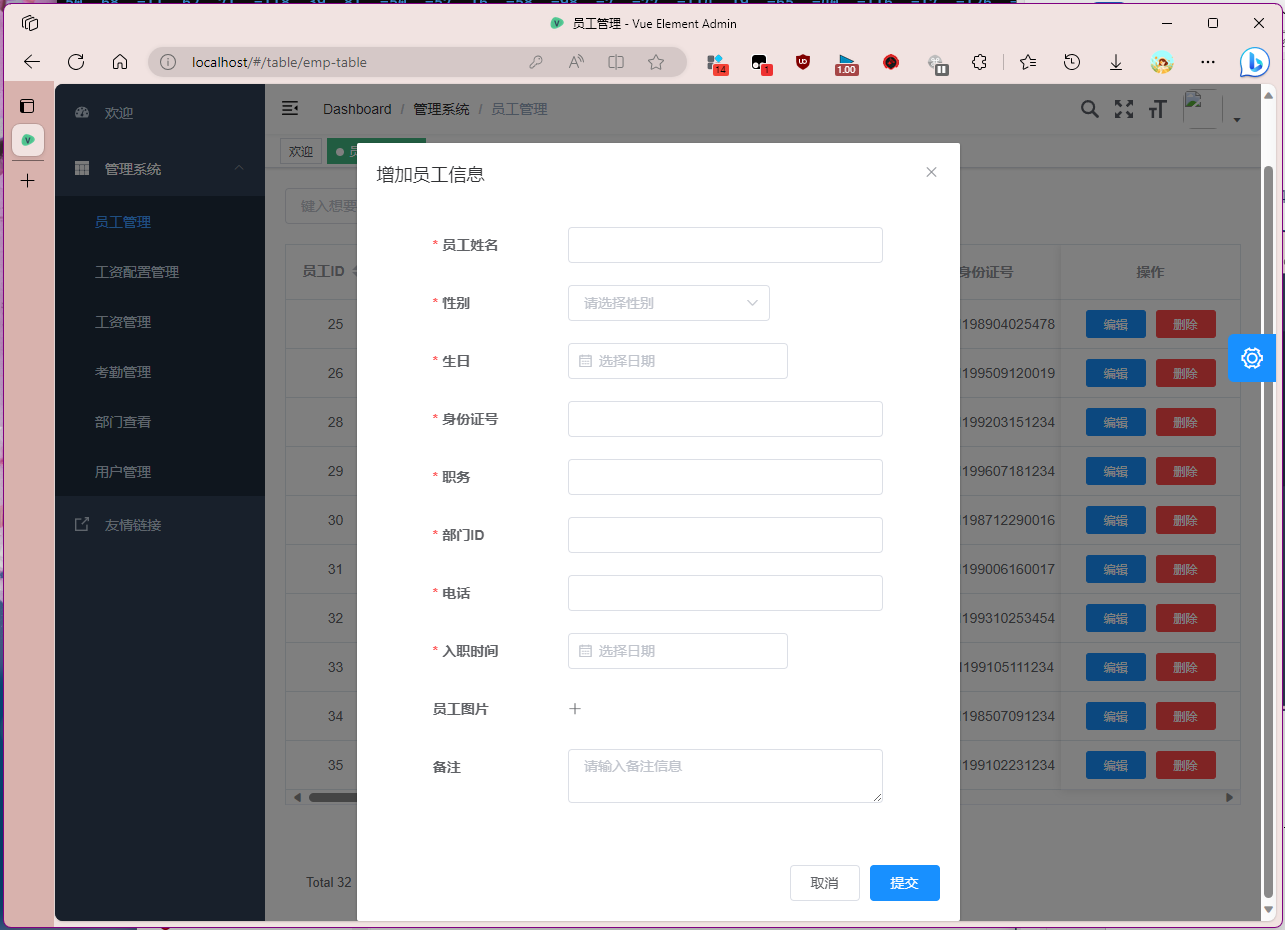


图 增加表单展示

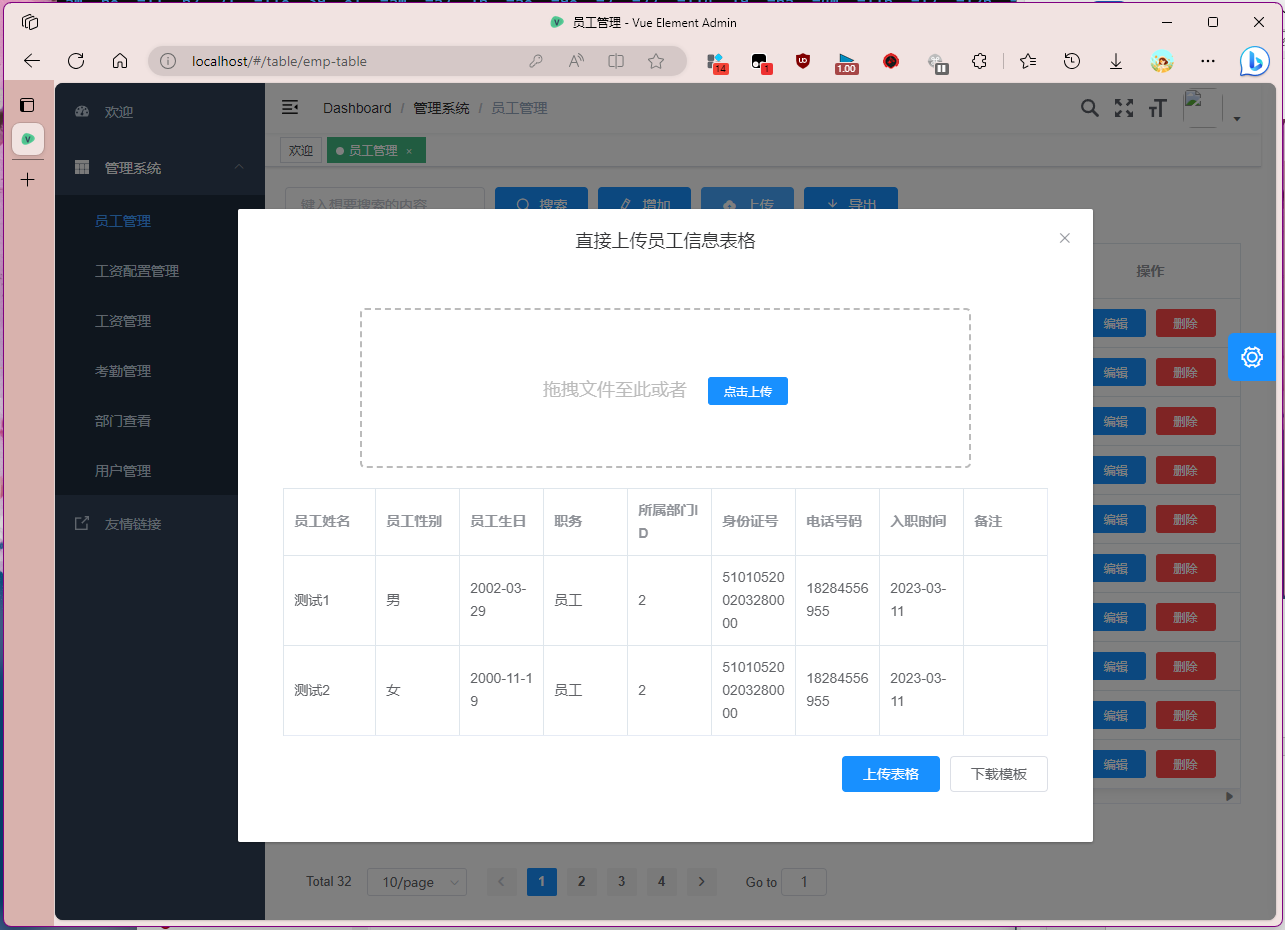


图 Excel表格直接上传模块展示

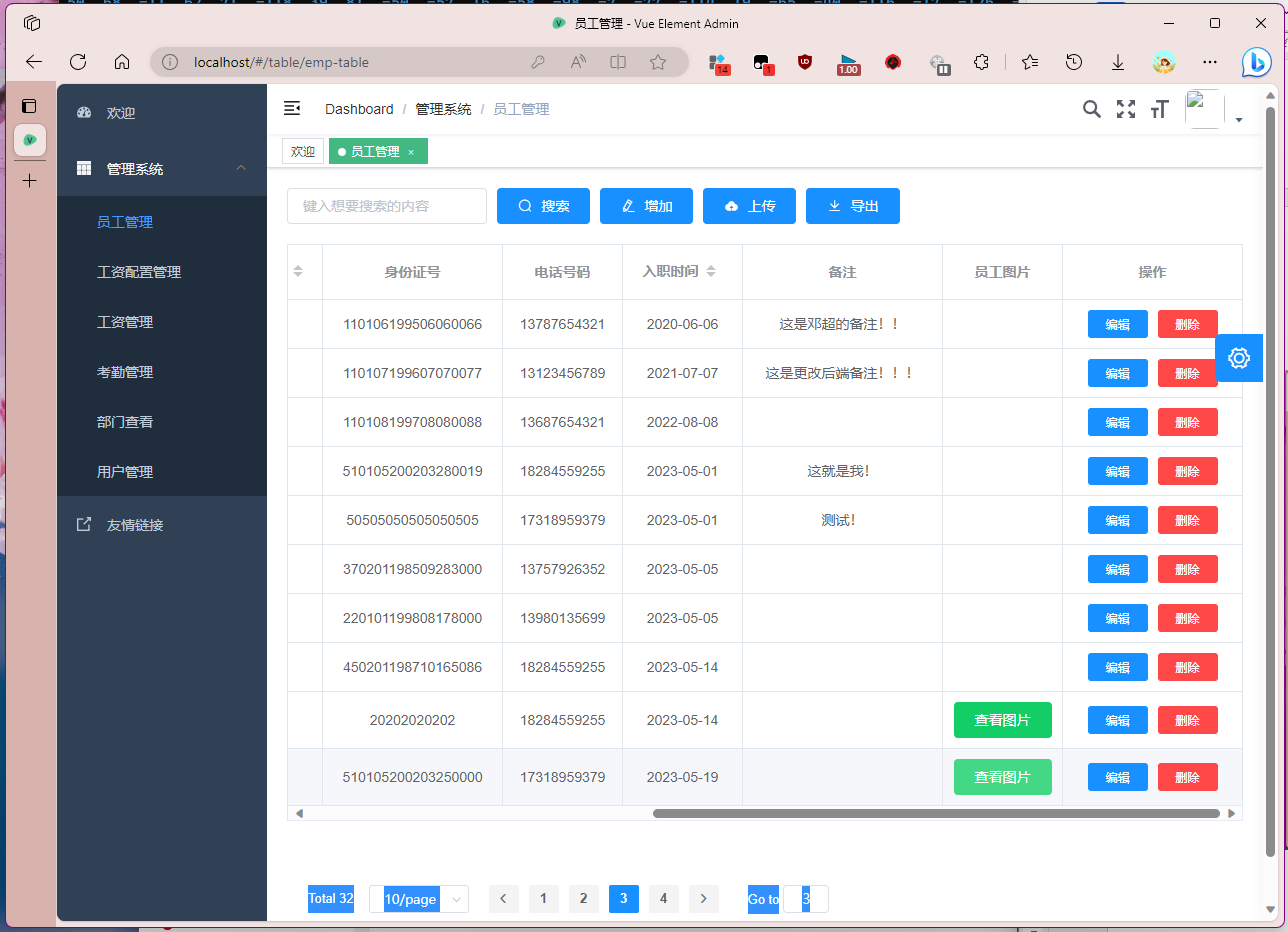


图 图片查看按钮展示1

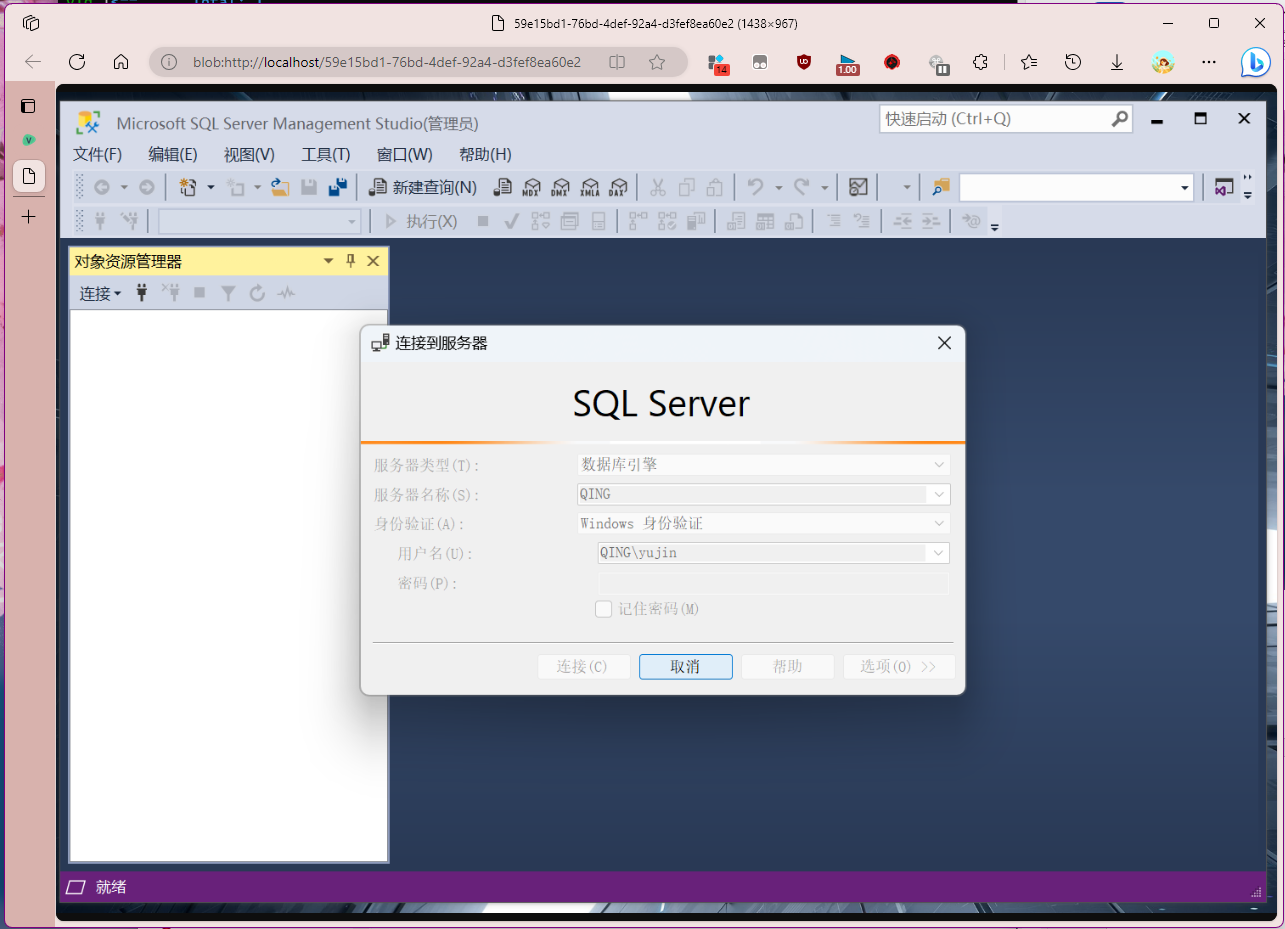


图 图片查看按钮展示2

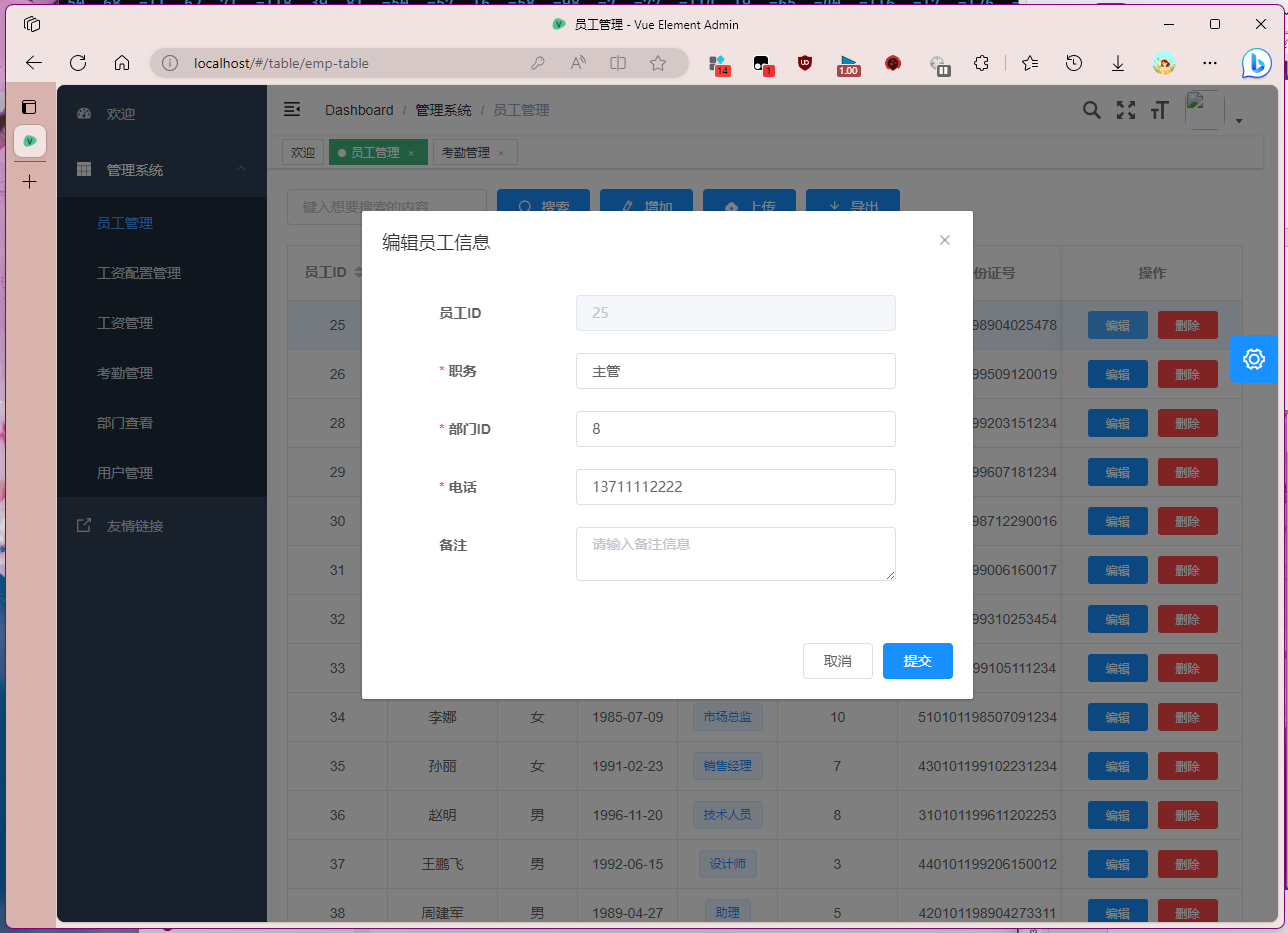


图 编辑表单展示

# 总结

本文档是工资管理系统软件开发的最终报告，旨在总结本项目的主要目标、功能、设计、实现、测试和评估。工资管理系统软件是为某公司的人力资源部门开发的，用于管理员工的基本信息、考勤情况、工资计算和发放等。本系统采用了B/S架构，使用了Java语言、SQL server数据库和HTML/CSS/JavaScript/vue等前端技术。本系统具有以下功能：

* 员工信息管理：管理员可以录入、修改和删除员工的基本信息，如姓名、性别、年龄、职位、部门等。
* 考勤管理：管理员可以设置每月的考勤规则，如上下班时间、请假类型、加班费率等。员工可以通过网页或手机端打卡签到和签退，也可以申请请假、加班等。系统会根据考勤规则自动计算员工的出勤天数、缺勤天数、迟到次数、早退次数、加班时长等。
* 工资管理：管理员可以设置每月的工资规则，如基本工资、奖金、扣款等。系统会根据员工的信息和考勤情况自动计算员工的应发工资、实发工资和个人所得税等。管理员可以通过网页或手机端查看和导出员工的工资明细和汇总表，也可以通过银行卡或支付宝等方式给员工发放工资。
* 系统管理：管理员可以设置系统的基本参数，如公司名称、logo、联系方式等。管理员也可以管理用户的账号和权限，如添加、修改和删除用户，分配用户的角色和功能。

本项目采用了敏捷开发的方法，按照需求分析、设计、编码、测试和部署的流程进行了多个迭代。本项目的开发过程中使用了Git作为版本控制工具，使用了Jet Brain-idea作为集成开发环境，使用了Maven作为项目管理工具。本项目的开发过程中遵循了编码规范和文档规范，保证了代码的可读性和可维护性，也保证了文档的完整性和一致性。

本项目经过了充分的测试和评估，验证了系统的功能正确性和性能优越性。本项目的测试包括了单元测试、集成测试、系统测试和验收测试，覆盖了系统的所有功能和场景。本项目的评估包括了功能评估、性能评估和用户满意度评估，采用了定量和定性的方法，收集了用户的反馈和建议。本项目的测试和评估结果表明，本系统能够满足用户的需求和期望，提高了人力资源部门的工作效率和质量。

总之，本项目是一个成功的软件开发项目，展示了专业能力和创新精神。本项目也为今后的软件开发提供了宝贵的经验和启示。

# 参考资料

以下列出了设计系统所查阅的所有参考资料包括数据库脚本等信息

## 数据库脚本

USE [master]

GO

/\*\*\*\*\*\* Object:  Database [WageManagement]    Script Date: 2023/4/7 17:28:45 \*\*\*\*\*\*/

CREATE DATABASE [WageManagement]

 CONTAINMENT = NONE

 ON  PRIMARY

( NAME = N'WageManagement', FILENAME = N'C:\Project\DBS\SQL Server\课程设计\WageManagement\_data.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

 LOG ON

( NAME = N'WageManagement\_log', FILENAME = N'C:\Project\DBS\SQL Server\课程设计\WageManagement\_log.ldf' , SIZE = 139264KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

 WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

GO

ALTER DATABASE [WageManagement] SET COMPATIBILITY\_LEVEL = 150

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [WageManagement].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [WageManagement] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [WageManagement] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [WageManagement] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [WageManagement] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [WageManagement] SET ARITHABORT OFF

GO

ALTER DATABASE [WageManagement] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [WageManagement] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [WageManagement] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [WageManagement] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [WageManagement] SET CURSOR\_DEFAULT  GLOBAL

GO

ALTER DATABASE [WageManagement] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [WageManagement] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [WageManagement] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [WageManagement] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [WageManagement] SET  DISABLE\_BROKER

GO

ALTER DATABASE [WageManagement] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [WageManagement] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [WageManagement] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [WageManagement] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [WageManagement] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [WageManagement] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [WageManagement] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [WageManagement] SET RECOVERY FULL

GO

ALTER DATABASE [WageManagement] SET  MULTI\_USER

GO

ALTER DATABASE [WageManagement] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [WageManagement] SET DB\_CHAINING OFF

GO

ALTER DATABASE [WageManagement] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [WageManagement] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [WageManagement] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [WageManagement] SET ACCELERATED\_DATABASE\_RECOVERY = OFF

GO

EXEC sys.sp\_db\_vardecimal\_storage\_format N'WageManagement', N'ON'

GO

ALTER DATABASE [WageManagement] SET QUERY\_STORE = OFF

GO

USE [WageManagement]

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_AbsenceWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[get\_AbsenceWage](@Empid int,@WageDate date,@InputWage money)

returns money

as

begin

    --存储每日标准工资和标准出差工资补贴

    declare @DayWage money,@AbsenceStd money

    --存储出差工时，允许小数点

    declare @AbsenceDay float

    select @AbsenceDay=sum(AbsenceDay) from DailyCheckStat

    where Empid=@Empid and month(CheckDate)=month(@WageDate) and year(CheckDate)=year(@WageDate)

    --若查询结果为NULL则返回0

    if @AbsenceDay is NULL

    begin

        return 0

    end

    --如果没有指定默认小时出差工资，则从工资配置表中读取

    if(@InputWage is null)

    begin

        select @AbsenceStd=AbsenceStd

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        return -@AbsenceDay\*@AbsenceStd

    end

    return -@AbsenceDay\*@InputWage

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_BaseWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create function [dbo].[get\_BaseWage](@Empid int)

returns money

as

begin

    declare @Duty nvarchar(20),@BaseWage money

    select @Duty=Duty from emp where empid=@Empid

    select @BaseWage=BaseWage from WageConfig where Duty=@Duty

    return @BaseWage

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_BonusWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create function [dbo].[get\_BonusWage](@Empid int,@WageDate date,@InputWage money,@Flag bit=0)

returns money

as

begin

    --存储标准红利奖金

    declare @Bonus money

    declare @LateDay int

    select @LateDay=sum(LateDay) from DailyCheckStat

    where Empid=@Empid and month(CheckDate)=month(@WageDate) and year(CheckDate)=year(@WageDate)

    --如果没有指定红利奖金，则从工资配置表中读取

    if(@InputWage is null)

    begin

        select @Bonus=Bonus

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

    end

    else

    begin

        select @Bonus=@InputWage

    end

    --如果全勤或者Flag=1，返回红利奖金，否则返回0

    --Flag=1表示强制发放红利奖金

    if(@LateDay=0 or @Flag=1)

        return @Bonus

    return 0

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_BussinessTripWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[get\_BussinessTripWage](@Empid int,@WageDate date,@InputWage money)

returns money

as

begin

    --存储每日标准工资和标准出差工资补贴

    declare @DayWage money,@BussinessTripStd money

    --存储出差工时，允许小数点

    declare @BussinessTripDay float

    select @BussinessTripDay=sum(BussinessTripDay) from DailyCheckStat

    where Empid=@Empid and month(CheckDate)=month(@WageDate) and year(CheckDate)=year(@WageDate)

    --·如果没有出差工时，则返回0

    if @BussinessTripDay is NULL

    begin

        return 0

    end

    --如果没有指定默认小时出差工资，则从工资配置表中读取

    if(@InputWage is null)

    begin

        select @BussinessTripStd=BussinessTripStd

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        return @BussinessTripDay\*@BussinessTripStd

    end

    return @BussinessTripDay\*@InputWage

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_DayWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create function [dbo].[get\_DayWage](@Empid int,@WageDate date,@InputWage money)

returns money

as

begin

    --存储每日标准工资

    declare @DayWage money

    --存储上班日，允许小数点

    declare @WorkDay float

    select @WorkDay=sum(WorkDay) from DailyCheckStat

    where Empid=@Empid and month(CheckDate)=month(@WageDate) and year(CheckDate)=year(@WageDate)

    --·如果没有上班日，则返回0

    if @WorkDay is NULL

    begin

        return 0

    end

    --如果没有指定默认小时出差工资，则从工资配置表中读取

    if(@InputWage is null)

    begin

        select @DayWage=DayWage

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        return @WorkDay\*@DayWage

    end

    return @WorkDay\*@InputWage

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_EtcWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[get\_EtcWage](@Empid int,@WageDate date,@InputWage money,@Flag bit=0)

returns money

as

begin

    --Flag=1表示强制发放其他工资

    if(@Flag=1)

    begin

        --存储标准其他工资

        declare @EtcWage money

        --如果没有指定其他工资，则从工资配置表中读取

        if(@InputWage is null)

        begin

            select @EtcWage=EtcStd

            from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        end

        else

        begin

            select @EtcWage=@InputWage

        end

        return @EtcWage

    end

    return 0

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_LateWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[get\_LateWage](@Empid int,@WageDate date,@InputWage money)

returns money

as

begin

    --存储每日标准工资和标准迟到惩罚

    declare @LateStd money

    --存储出差工时，允许小数点

    declare @LateDay int

    select @LateDay=sum(LateDay) from DailyCheckStat

    where Empid=@Empid and month(CheckDate)=month(@WageDate) and year(CheckDate)=year(@WageDate)

    --如果没有迟到天数，则返回0

    if @LateDay is NULL

    begin

        return 0

    end

    --如果没有指定迟到惩罚工资，则从工资配置表中读取

    if(@InputWage is null)

    begin

        select @LateStd=LateStd

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        return -@LateDay\*@LateStd

    end

    return -@LateDay\*@InputWage

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_overtime\_wage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[get\_overtime\_wage](@Empid int,@WageDate date,@DefaultWage money)

returns money

as

begin

    --存储每日标准工资和加班工资比例

    declare @DayWage money,@OvertimeRatio float

    --存储加班工时，允许小数点

    declare @OvertimeHours float

    select @OvertimeHours=sum(OvertimeHour) from DailyCheckStat

    where Empid=@Empid and month(CheckDate)=month(@WageDate) and year(CheckDate)=year(@WageDate)

    --·如果没有加班工时，则返回0

    if @OvertimeHours is NULL

    begin

        return 0

    end

    --如果没有指定默认小时加班工资，则从工资配置表中读取

    if(@DefaultWage is null)

    begin

        select @DayWage=DayWage

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        select @OvertimeRatio=OvertimeRatio

        from WageConfig where Duty=(select Duty from emp where empid=@Empid)

        return @DayWage/8.0\*@OvertimeHours\*@OvertimeRatio

    end

    return @OvertimeHours\*@DefaultWage

end

GO

/\*\*\*\*\*\* Object:  UserDefinedFunction [dbo].[get\_seniority\_wage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE function [dbo].[get\_seniority\_wage](@Empid int,@WageDate date,@DefaultWage money=50)

returns int

begin

    --指的是工龄，使用年份计时

    declare @seniority int

    select @seniority=datediff(day,HireDate,@WageDate)/365

    from emp

    where empid=@Empid

    --工龄工资最多发放5年

    if(@seniority<5)

        return @seniority\*@DefaultWage

    return 5\*@DefaultWage

end

GO

/\*\*\*\*\*\* Object:  Table [dbo].[MonthlyWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[MonthlyWage](

    [EmpID] [int] NOT NULL,

    [WageDate] [date] NOT NULL,

    [BaseWage] [money] NULL,

    [DayWage] [money] NULL,

    [OvertimeWage] [money] NULL,

    [SeniorityWage] [money] NULL,

    [BussinessTripWage] [money] NULL,

    [AbsenceWage] [money] NULL,

    [LateWage] [money] NULL,

    [BonusWage] [money] NULL,

    [EtcWage] [money] NULL,

    [TotalWage]  AS (((((((([BaseWage]+[OvertimeWage])+[SeniorityWage])+[BussinessTripWage])+[AbsenceWage])+[LateWage])+[BonusWage])+[EtcWage])+[DayWage]) PERSISTED,

 CONSTRAINT [PK\_MonthlyWage] PRIMARY KEY CLUSTERED

(

    [EmpID] ASC,

    [WageDate] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object:  Table [dbo].[Depart]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Depart](

    [DepartID] [int] IDENTITY(1,1) NOT NULL,

    [DepartName] [nvarchar](20) NOT NULL,

    [DepartManagerID] [int] NULL,

 CONSTRAINT [PK\_Depart] PRIMARY KEY CLUSTERED

(

    [DepartID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object:  Table [dbo].[Emp]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Emp](

    [EmpID] [int] IDENTITY(1,1) NOT NULL,

    [EmpName] [nvarchar](5) NOT NULL,

    [EmpSex] [nchar](4) NOT NULL,

    [EmpBirth] [date] NOT NULL,

    [Duty] [nvarchar](20) NOT NULL,

    [DepartID] [int] NULL,

    [IdentityID] [varchar](20) NOT NULL,

    [EmpTele] [varchar](20) NOT NULL,

    [EmpPic] [image] NULL,

    [EmpRemark] [nvarchar](200) NULL,

    [HireDate] [date] NOT NULL,

    [LeaveDate] [date] NULL,

    [LogicalDelete] [tinyint] NULL,

 CONSTRAINT [PK\_Emp] PRIMARY KEY CLUSTERED

(

    [EmpID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

/\*\*\*\*\*\* Object:  View [dbo].[EmpMonthlyWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[EmpMonthlyWage]

AS

SELECT   TOP (100) PERCENT dbo.Emp.EmpID, dbo.Emp.EmpName, dbo.Depart.DepartName, dbo.Emp.Duty,

                { fn CONCAT({ fn CONCAT({ fn CONCAT(trim(STR(YEAR(dbo.MonthlyWage.WageDate))), '年') },

                trim(STR(MONTH(dbo.MonthlyWage.WageDate)))) }, '月') } AS WageMonth, dbo.MonthlyWage.TotalWage,

                dbo.MonthlyWage.BaseWage, dbo.MonthlyWage.DayWage, dbo.MonthlyWage.OvertimeWage,

                dbo.MonthlyWage.SeniorityWage, dbo.MonthlyWage.BussinessTripWage, dbo.MonthlyWage.AbsenceWage,

                dbo.MonthlyWage.LateWage, dbo.MonthlyWage.BonusWage, dbo.MonthlyWage.EtcWage

FROM      dbo.Emp INNER JOIN

                dbo.MonthlyWage ON dbo.Emp.EmpID = dbo.MonthlyWage.EmpID INNER JOIN

                dbo.Depart ON dbo.Emp.DepartID = dbo.Depart.DepartID

ORDER BY dbo.MonthlyWage.WageDate DESC

GO

/\*\*\*\*\*\* Object:  View [dbo].[DepartmentManager]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[DepartmentManager]

AS

SELECT   dbo.Depart.DepartID, dbo.Depart.DepartName, dbo.Emp.EmpID, dbo.Emp.EmpName

FROM      dbo.Depart LEFT OUTER JOIN

                dbo.Emp ON dbo.Depart.DepartManagerID = dbo.Emp.EmpID AND dbo.Depart.DepartID = dbo.Emp.DepartID

GO

/\*\*\*\*\*\* Object:  Table [dbo].[DailyCheckStat]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[DailyCheckStat](

    [EmpID] [int] NOT NULL,

    [CheckDate] [date] NOT NULL,

    [OvertimeHour] [float] NULL,

    [WorkDay] [tinyint] NULL,

    [BussinessTripDay] [tinyint] NULL,

    [AbsenceDay] [tinyint] NULL,

    [LateDay] [tinyint] NULL,

 CONSTRAINT [PK\_DailyCheckStat] PRIMARY KEY CLUSTERED

(

    [EmpID] ASC,

    [CheckDate] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object:  Table [dbo].[Users]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Users](

    [UserName] [nvarchar](20) NOT NULL,

    [UserPassword] [varchar](20) NOT NULL,

    [UserRight] [nvarchar](10) NULL,

    [UserRightLevel] [int] NOT NULL,

 CONSTRAINT [PK\_User] PRIMARY KEY CLUSTERED

(

    [UserName] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object:  Table [dbo].[WageConfig]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[WageConfig](

    [Duty] [nvarchar](20) NOT NULL,

    [BaseWage] [money] NOT NULL,

    [DayWage] [money] NOT NULL,

    [OvertimeRatio] [float] NOT NULL,

    [BussinessTripStd] [money] NOT NULL,

    [AbsenceStd] [money] NOT NULL,

    [LateStd] [money] NOT NULL,

    [Bonus] [money] NOT NULL,

    [EtcStd] [money] NOT NULL,

 CONSTRAINT [PK\_WageConfig] PRIMARY KEY CLUSTERED

(

    [Duty] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Emp] ADD  CONSTRAINT [DF\_Emp\_HireDate]  DEFAULT (getdate()) FOR [HireDate]

GO

ALTER TABLE [dbo].[Emp] ADD  CONSTRAINT [DF\_Emp\_LogicalDelate]  DEFAULT ((0)) FOR [LogicalDelete]

GO

ALTER TABLE [dbo].[Users] ADD  CONSTRAINT [DF\_Users\_UserRightLevel]  DEFAULT ((0)) FOR [UserRightLevel]

GO

ALTER TABLE [dbo].[DailyCheckStat]  WITH CHECK ADD  CONSTRAINT [FK\_DailyCheckStat\_Emp] FOREIGN KEY([EmpID])

REFERENCES [dbo].[Emp] ([EmpID])

GO

ALTER TABLE [dbo].[DailyCheckStat] CHECK CONSTRAINT [FK\_DailyCheckStat\_Emp]

GO

ALTER TABLE [dbo].[Depart]  WITH CHECK ADD  CONSTRAINT [FK\_Depart\_Emp] FOREIGN KEY([DepartManagerID])

REFERENCES [dbo].[Emp] ([EmpID])

GO

ALTER TABLE [dbo].[Depart] CHECK CONSTRAINT [FK\_Depart\_Emp]

GO

ALTER TABLE [dbo].[Emp]  WITH CHECK ADD  CONSTRAINT [FK\_Emp\_Emp] FOREIGN KEY([DepartID])

REFERENCES [dbo].[Depart] ([DepartID])

GO

ALTER TABLE [dbo].[Emp] CHECK CONSTRAINT [FK\_Emp\_Emp]

GO

ALTER TABLE [dbo].[Emp]  WITH CHECK ADD  CONSTRAINT [FK\_WageConfig\_Emp\_Duty] FOREIGN KEY([Duty])

REFERENCES [dbo].[WageConfig] ([Duty])

GO

ALTER TABLE [dbo].[Emp] CHECK CONSTRAINT [FK\_WageConfig\_Emp\_Duty]

GO

ALTER TABLE [dbo].[MonthlyWage]  WITH CHECK ADD  CONSTRAINT [FK\_MonthlyWage\_Emp] FOREIGN KEY([EmpID])

REFERENCES [dbo].[Emp] ([EmpID])

GO

ALTER TABLE [dbo].[MonthlyWage] CHECK CONSTRAINT [FK\_MonthlyWage\_Emp]

GO

ALTER TABLE [dbo].[DailyCheckStat]  WITH CHECK ADD  CONSTRAINT [CK\_DailyCheckStat] CHECK  ((([bussinesstripday]=(1) OR [bussinesstripday]=(0)) AND ([absenceday]=(1) OR [absenceday]=(0)) AND [lateday]>=(0) AND [workday]>=(0)))

GO

ALTER TABLE [dbo].[DailyCheckStat] CHECK CONSTRAINT [CK\_DailyCheckStat]

GO

/\*\*\*\*\*\* Object:  StoredProcedure [dbo].[refreshAllMonthlyWage]    Script Date: 2023/4/7 17:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[refreshAllMonthlyWage](@Date date)

as

begin

    declare @empid int

    declare refreshAllMonthlyWage\_c1 cursor for

    select empid from emp where LogicalDelete=0

    open refreshAllMonthlyWage\_c1

    fetch next from refreshAllMonthlyWage\_c1 into @empid

    while @@fetch\_status=0

    begin

        --如果传入日期，则只刷新该日期的月薪

        if @Date!=''

        begin

            insert into monthlywage(empid,WageDate) values(@empid,@Date)

        end

        --如果不传入日期，则刷新所有月薪

        else

        begin

            declare @WageDate date

            declare refreshAllMonthlyWage\_c2 cursor for

            select distinct concat(trim(str(year(CheckDate))),'-',trim(str(month(CheckDate))),'-01') from DailyCheckStat

            where empid=@empid

            open refreshAllMonthlyWage\_c2

            fetch next from refreshAllMonthlyWage\_c2 into @WageDate

            while @@fetch\_status=0

            begin

                insert into monthlywage(empid,WageDate) values(@empid,@WageDate)

                fetch next from refreshAllMonthlyWage\_c2 into @WageDate

            end

            close refreshAllMonthlyWage\_c2

            deallocate refreshAllMonthlyWage\_c2

        end

        fetch next from refreshAllMonthlyWage\_c1 into @empid

    end

    close refreshAllMonthlyWage\_c1

    deallocate refreshAllMonthlyWage\_c1

end

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'每日考勤日期' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'DailyCheckStat', @level2type=N'COLUMN',@level2name=N'CheckDate'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'加班小时数' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'DailyCheckStat', @level2type=N'COLUMN',@level2name=N'OvertimeHour'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'是否上班' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'DailyCheckStat', @level2type=N'COLUMN',@level2name=N'WorkDay'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'是否出差' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'DailyCheckStat', @level2type=N'COLUMN',@level2name=N'BussinessTripDay'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'是否旷工' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'DailyCheckStat', @level2type=N'COLUMN',@level2name=N'AbsenceDay'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'迟到次数' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'DailyCheckStat', @level2type=N'COLUMN',@level2name=N'LateDay'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'部门名称' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Depart', @level2type=N'COLUMN',@level2name=N'DepartName'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'部门主管ID' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Depart', @level2type=N'COLUMN',@level2name=N'DepartManagerID'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'职务' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'Duty'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'琐事部门ID' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'DepartID'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'身份证号' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'IdentityID'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'员工图片' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'EmpPic'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'备注' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'EmpRemark'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'入职时间' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'HireDate'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'离职时间' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'LeaveDate'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'逻辑删除标志' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Emp', @level2type=N'COLUMN',@level2name=N'LogicalDelete'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'基本工资' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'BaseWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'日薪合计' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'DayWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'加班工资' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'OvertimeWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'工龄工资' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'SeniorityWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'出差补贴' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'BussinessTripWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'旷工扣款' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'AbsenceWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'迟到扣款' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'LateWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'奖励工资' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'BonusWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'其他工资' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'EtcWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'工资总和' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'MonthlyWage', @level2type=N'COLUMN',@level2name=N'TotalWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'用户权限' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Users', @level2type=N'COLUMN',@level2name=N'UserRight'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'用户权限等级' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'Users', @level2type=N'COLUMN',@level2name=N'UserRightLevel'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'职务' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'Duty'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'基本工资' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'BaseWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'日薪' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'DayWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'加班补贴比率' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'OvertimeRatio'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'标准出差补贴' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'BussinessTripStd'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'标准旷工扣款' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'AbsenceStd'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'标准迟到扣款' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'LateStd'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'标准奖励' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'Bonus'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_Description', @value=N'其他' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'WageConfig', @level2type=N'COLUMN',@level2name=N'EtcStd'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

   Begin PaneConfigurations =

      Begin PaneConfiguration = 0

         NumPanes = 4

         Configuration = "(H (1[40] 4[20] 2[20] 3) )"

      End

      Begin PaneConfiguration = 1

         NumPanes = 3

         Configuration = "(H (1 [50] 4 [25] 3))"

      End

      Begin PaneConfiguration = 2

         NumPanes = 3

         Configuration = "(H (1 [50] 2 [25] 3))"

      End

      Begin PaneConfiguration = 3

         NumPanes = 3

         Configuration = "(H (4 [30] 2 [40] 3))"

      End

      Begin PaneConfiguration = 4

         NumPanes = 2

         Configuration = "(H (1 [56] 3))"

      End

      Begin PaneConfiguration = 5

         NumPanes = 2

         Configuration = "(H (2 [66] 3))"

      End

      Begin PaneConfiguration = 6

         NumPanes = 2

         Configuration = "(H (4 [50] 3))"

      End

      Begin PaneConfiguration = 7

         NumPanes = 1

         Configuration = "(V (3))"

      End

      Begin PaneConfiguration = 8

         NumPanes = 3

         Configuration = "(H (1[56] 4[18] 2) )"

      End

      Begin PaneConfiguration = 9

         NumPanes = 2

         Configuration = "(H (1 [75] 4))"

      End

      Begin PaneConfiguration = 10

         NumPanes = 2

         Configuration = "(H (1[66] 2) )"

      End

      Begin PaneConfiguration = 11

         NumPanes = 2

         Configuration = "(H (4 [60] 2))"

      End

      Begin PaneConfiguration = 12

         NumPanes = 1

         Configuration = "(H (1) )"

      End

      Begin PaneConfiguration = 13

         NumPanes = 1

         Configuration = "(V (4))"

      End

      Begin PaneConfiguration = 14

         NumPanes = 1

         Configuration = "(V (2))"

      End

      ActivePaneConfig = 0

   End

   Begin DiagramPane =

      Begin Origin =

         Top = 0

         Left = 0

      End

      Begin Tables =

         Begin Table = "Depart"

            Begin Extent =

               Top = 6

               Left = 38

               Bottom = 127

               Right = 233

            End

            DisplayFlags = 280

            TopColumn = 0

         End

         Begin Table = "Emp"

            Begin Extent =

               Top = 86

               Left = 413

               Bottom = 226

               Right = 580

            End

            DisplayFlags = 280

            TopColumn = 1

         End

      End

   End

   Begin SQLPane =

   End

   Begin DataPane =

      Begin ParameterDefaults = ""

      End

   End

   Begin CriteriaPane =

      Begin ColumnWidths = 11

         Column = 1440

         Alias = 900

         Table = 1170

         Output = 720

         Append = 1400

         NewValue = 1170

         SortType = 1350

         SortOrder = 1410

         GroupBy = 1350

         Filter = 1350

         Or = 1350

         Or = 1350

         Or = 1350

      End

   End

End

' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'DepartmentManager'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'DepartmentManager'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]

Begin DesignProperties =

   Begin PaneConfigurations =

      Begin PaneConfiguration = 0

         NumPanes = 4

         Configuration = "(H (1[40] 4[20] 2[20] 3) )"

      End

      Begin PaneConfiguration = 1

         NumPanes = 3

         Configuration = "(H (1 [50] 4 [25] 3))"

      End

      Begin PaneConfiguration = 2

         NumPanes = 3

         Configuration = "(H (1 [50] 2 [25] 3))"

      End

      Begin PaneConfiguration = 3

         NumPanes = 3

         Configuration = "(H (4 [30] 2 [40] 3))"

      End

      Begin PaneConfiguration = 4

         NumPanes = 2

         Configuration = "(H (1 [56] 3))"

      End

      Begin PaneConfiguration = 5

         NumPanes = 2

         Configuration = "(H (2 [66] 3))"

      End

      Begin PaneConfiguration = 6

         NumPanes = 2

         Configuration = "(H (4 [50] 3))"

      End

      Begin PaneConfiguration = 7

         NumPanes = 1

         Configuration = "(V (3))"

      End

      Begin PaneConfiguration = 8

         NumPanes = 3

         Configuration = "(H (1[56] 4[18] 2) )"

      End

      Begin PaneConfiguration = 9

         NumPanes = 2

         Configuration = "(H (1 [75] 4))"

      End

      Begin PaneConfiguration = 10

         NumPanes = 2

         Configuration = "(H (1[66] 2) )"

      End

      Begin PaneConfiguration = 11

         NumPanes = 2

         Configuration = "(H (4 [60] 2))"

      End

      Begin PaneConfiguration = 12

         NumPanes = 1

         Configuration = "(H (1) )"

      End

      Begin PaneConfiguration = 13

         NumPanes = 1

         Configuration = "(V (4))"

      End

      Begin PaneConfiguration = 14

         NumPanes = 1

         Configuration = "(V (2))"

      End

      ActivePaneConfig = 0

   End

   Begin DiagramPane =

      Begin Origin =

         Top = 0

         Left = 0

      End

      Begin Tables =

         Begin Table = "Emp"

            Begin Extent =

               Top = 6

               Left = 38

               Bottom = 146

               Right = 205

            End

            DisplayFlags = 280

            TopColumn = 0

         End

         Begin Table = "MonthlyWage"

            Begin Extent =

               Top = 62

               Left = 447

               Bottom = 202

               Right = 649

            End

            DisplayFlags = 280

            TopColumn = 0

         End

         Begin Table = "Depart"

            Begin Extent =

               Top = 247

               Left = 491

               Bottom = 368

               Right = 686

            End

            DisplayFlags = 280

            TopColumn = 0

         End

      End

   End

   Begin SQLPane =

   End

   Begin DataPane =

      Begin ParameterDefaults = ""

      End

   End

   Begin CriteriaPane =

      Begin ColumnWidths = 11

         Column = 1440

         Alias = 1530

         Table = 1170

         Output = 720

         Append = 1400

         NewValue = 1170

         SortType = 1350

         SortOrder = 1410

         GroupBy = 1350

         Filter = 1350

         Or = 1350

         Or = 1350

         Or = 1350

      End

   End

End

' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'EmpMonthlyWage'

GO

EXEC sys.sp\_addextendedproperty @name=N'MS\_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'EmpMonthlyWage'

GO

USE [master]

GO

ALTER DATABASE [WageManagement] SET  READ\_WRITE

GO

--作业语句

USE [msdb]

GO

/\*\*\*\*\*\* Object:  Job [AutoDailyCheckStat]    Script Date: 2023/4/22 20:27:29 \*\*\*\*\*\*/

BEGIN TRANSACTION

DECLARE @ReturnCode INT

SELECT @ReturnCode = 0

/\*\*\*\*\*\* Object:  JobCategory [Database Maintenance]    Script Date: 2023/4/22 20:27:30 \*\*\*\*\*\*/

IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'Database Maintenance' AND category\_class=1)

BEGIN

EXEC @ReturnCode = msdb.dbo.sp\_add\_category @class=N'JOB', @type=N'LOCAL', @name=N'Database Maintenance'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)

EXEC @ReturnCode =  msdb.dbo.sp\_add\_job @job\_name=N'AutoDailyCheckStat',

        @enabled=1,

        @notify\_level\_eventlog=0,

        @notify\_level\_email=0,

        @notify\_level\_netsend=0,

        @notify\_level\_page=0,

        @delete\_level=0,

        @description=N'每天增加虚拟打卡数据',

        @category\_name=N'Database Maintenance',

        @owner\_login\_name=N'Qing\yujin', @job\_id = @jobId OUTPUT

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

/\*\*\*\*\*\* Object:  Step [循环增加虚拟打卡数据]    Script Date: 2023/4/22 20:27:30 \*\*\*\*\*\*/

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobstep @job\_id=@jobId, @step\_name=N'循环增加虚拟打卡数据',

        @step\_id=1,

        @cmdexec\_success\_code=0,

        @on\_success\_action=3,

        @on\_success\_step\_id=0,

        @on\_fail\_action=2,

        @on\_fail\_step\_id=0,

        @retry\_attempts=0,

        @retry\_interval=0,

        @os\_run\_priority=0, @subsystem=N'TSQL',

        @command=N'DECLARE @EmpID INT

DECLARE @LateDay INT

DECLARE @OvertimeHour FLOAT

declare emp\_cur CURSOR FOR SELECT EmpID

FROM Emp

where LogicalDelete=0

OPEN emp\_cur

FETCH NEXT FROM emp\_cur INTO @EmpID

WHILE @@FETCH\_STATUS = 0

BEGIN

    SET @LateDay=CAST(floor(rand()\*3) as int)

    SET @OverTimeHour=RAND()

    if @OvertimeHour>0.5

    BEGIN

        SET @OvertimeHour=@OvertimeHour\*10

    END

    ELSE

    BEGIN

        SET @OvertimeHour=0

    END

    INSERT INTO DailyCheckStat

        (EmpID, CheckDate, OvertimeHour, WorkDay, BusinessTripDay, AbsenceDay, LateDay)

    VALUES

        (@EmpID, GETDATE()-1, @OvertimeHour, 1, 0, 0, @LateDay)

    FETCH NEXT FROM emp\_cur INTO @EmpID

END',

        @database\_name=N'WageManagement',

        @flags=0

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

/\*\*\*\*\*\* Object:  Step [刷新工资数据]    Script Date: 2023/4/22 20:27:30 \*\*\*\*\*\*/

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobstep @job\_id=@jobId, @step\_name=N'刷新工资数据',

        @step\_id=2,

        @cmdexec\_success\_code=0,

        @on\_success\_action=1,

        @on\_success\_step\_id=0,

        @on\_fail\_action=2,

        @on\_fail\_step\_id=0,

        @retry\_attempts=0,

        @retry\_interval=0,

        @os\_run\_priority=0, @subsystem=N'TSQL',

        @command=N'DECLARE @DATE DATE

set @DATE=GETDATE()-1

EXEC refreshAllMonthlyWage @DATE',

        @database\_name=N'master',

        @flags=0

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_update\_job @job\_id = @jobId, @start\_step\_id = 1

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobschedule @job\_id=@jobId, @name=N'自动插入虚拟打卡数据',

        @enabled=1,

        @freq\_type=4,

        @freq\_interval=1,

        @freq\_subday\_type=1,

        @freq\_subday\_interval=0,

        @freq\_relative\_interval=0,

        @freq\_recurrence\_factor=0,

        @active\_start\_date=20230423,

        @active\_end\_date=99991231,

        @active\_start\_time=80000,

        @active\_end\_time=235959,

        @schedule\_uid=N'964013c9-7bf2-429e-b67b-756b512258be'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

EXEC @ReturnCode = msdb.dbo.sp\_add\_jobserver @job\_id = @jobId, @server\_name = N'(local)'

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

COMMIT TRANSACTION

GOTO EndSave

QuitWithRollback:

    IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION

EndSave:

GO

## 参考资料及网页

* SpringBoot整合Element UI框架的博客：<https://www.cnblogs.com/l-y-h/p/13083375.html>
* 工资管理系统设计报告：[工资管理系统课程设计报告（整体）.doc-全文可读 (book118.com)](https://max.book118.com/html/2018/0830/6012152032001214.shtm)
* 使用人工智能new bing解答部分问题：<http://www.bing.com/new>

# 致谢

本文档是工资管理系统设计的结果，该系统旨在为企业提供一个高效、安全、准确的工资计算和发放的平台。在文档的编写过程中，我们得到了许多人的帮助和支持，特此表示感谢。

首先，我要感谢我的指导老师，他们在系统设计的方向、方法和细节上给予了我们很多的指导和建议，使我们能够顺利地完成系统设计的各个阶段。

其次，我要感谢为我提出改进建议的同学们，他们在需求分析和系统测试的过程中提供了宝贵的反馈和意见，使我能够更好地满足他们的实际需求和期望。

最后，我要感谢所有参与本文档审阅和修改的人员，他们在文档的格式、内容和质量上提出了很多的意见和建议，使我们能够更加完善和规范地呈现我们的系统设计。

在此，我对所有给予我帮助和支持的人员表示衷心的感谢！