福建工程学院信息科学与工程学院

**数据库应用课程设计**

题 目： 人事管理系统

专 业： 计算机科学与技术 班 级： 1801

学 号： 3181911112 姓 名： 王佩鑫

分 工： 数据库需求设计、数据库表设计、应用系统的控制端开发 、文档编辑

同组同学：

班 级： 1801 学 号： 3181911113

姓 名： 周永其

分 工： 数据库表设计、数据库代码的编写、应用系统的数据段和服务端开发 、文档编辑

**2020年6 月 4 日**

**摘要**

1. **背景：**

随着计算机技术的飞速发展,计算机已深入到各个领域，并且形成了功能强大、覆盖全球的信息传输网络。各个领域都向系统化、规范化、自动化的方向发展，使得工作效率、工作成绩和生活水平都日益提高。人事管理是很多厂矿、公同、个体事业单位所须的，人事信息管理系统包括对人事信息的统计、查询、更新、打印输出等,如果靠人，工作量将很大，若公司人数有几万甚至更多，人工统计将变得不可想象，仅一些简单的操作便可及时、准确地通过计算机获得需要的信息。计算机在企业管理中应用的普及,利用计算机实现企业人事管理势在必行。计算机管理所无法比拟的优点检索迅速、查找方便、可靠性高、存储量大、

保密性好、寿命长、成本低等。这些优点能够极大地提高人事管理的效率，也是企业的科学化、正规化管理与世界接轨的重要条件。不同的企业具有不同的人事管理制度，这就决定了不同的企业需要不同的人事管理系统。

1. **主要功能：**

系统能够提供对人事信息输入、查询、编辑以及其他管理设定、查询、修改; 可自主设定条件从而达到对工资数据的多角度查询功能;方便导入、导出数据及输出报表。部门人员以管理员身份登录,对本系统的可登录人员进行管理;实现了人事资源部门对公司人员系统的集中管理，保证了系统的安全性。

1. **开发环境：**

使用SQL Server2008、java开发工具。

**目录**

1. 系统分析

1.1系统定义.....................................................................................................4

1.2需求分析.....................................................................................................4

1.2.1功能要求..........................................................................................5

1.2.2性能要求..........................................................................................5

1.2.3数据流图..........................................................................................5

1.2.3数据字典..........................................................................................8

1.3系统模块设计.............................................................................................15

1. 数据库设计

2.1数据库概念设计.........................................................................................16

2.1.1分E-R图.........................................................................................16

2.1.2系统基本E-R图.............................................................................20

2.2数据库逻辑结构设计.................................................................................21

2.2.1E-R图向关系模型的转换...............................................................21

2.2.2数据模型的优化.............................................................................21

2.3数据库物理设计.........................................................................................22

2.3.1索引的设计......................................................................................22

2.3.2索引的建立.....................................................................................22

1. 数据库的实现与调试

3.1建立人事管理系统数据库...........................................................22

3.2建立数据库的各基本表...............................................................22

3.3插入数据.......................................................................................25

3.4建立索引.......................................................................................27

3.5视图...............................................................................................27

3.6存储过程.......................................................................................28

3.7触发器..........................................................................................33

3.8数据库功能的实现......................................................................36

3.9前台的设计....................................................................................41

1. 总结......................................................................................................43
2. 参考文献...............................................................................................44
3. **系统分析**

**1.1系统定义**

一个正常运作的公司离不开一个健全的人事管理系统，来管理员工的信息，包括员工的基本信息、工龄、工资以及奖惩情况。除了这些基本的用处以外，此系统还提供员工信息更改和删除等的功能，使得公司的人事管理方便快捷而又清晰。面对各种不同种类的信息，需要合理的数据库结构来保存数据信息以及有效地程序结构支持各种数据操作的执行。

以下及为本系统的目标：

1. 支持企业实现规范化管理。
2. 让企业高效率完成对人事的管理，比如新部门的增加及员工的加入时人事档案的建立（及员工的各种信息的输入），员工职务变动、辞退、退休记录及员工信息的查询和修改等。
3. 方便企业对员工进行奖惩。

**1.2需求分析**

**1.2.1功能要求**

1. 员工档案管理
2. 对员工进行统一编号，将每一位员工的信息保存在员工档案记录中。
3. 对新员工，将其信息加入到员工档案记录中。
4. 当员工的信息发生变动时，修改员工档案记录中相应的属性。
5. 员工职务管理

对公司所有员工的职务记录进行统一编号；登记员工调动前后的具体职务，以及调动时间。

1. 员工人事调动管理
2. 对于解聘和退休的员工，将其信息从员工档案中删除，并做调动记录。

2）对于更换部门的员工，更改档案，并做调动记录。

1. 奖惩管理

对公司里所有员工的奖惩记录进行统一编号;登记员工奖惩的原因，所受奖惩类型，奖惩时间。

1. 考勤管理

对员工上班考勤的记录进行统一编号;登记员工上班时间、签到类型(准时、迟到)；登记员工下班时间、签到类型(准时、早退)。

1. 工资管理

对公司里所有员工的工资记录进行统一编号，建立工资记录表。

1. 部门管理

对公司所有的部门]进行编号，建立部门记录。

（8）请假管理

1)对员工请假的记录进行统一编号，登记员工请假时间、请假类型。

2)对员工销假的记录进行统一编号;登记员工销假时间、请假类型，检查是否超期。

**1.2.2性能与运行要求**

目标：开发一个功能完善、操作方便的的人事管理系统。实现对每个员工的业绩考核。人机界面要友好，尽可能的减少软件对于计算机资源的占用。数据库存取效率高、空间利用率高、数据库运行管理效率高。

安全性：分为用户和管理员，用户只能浏览自己的信息不能修改，管理员才拥有修改权限。

**1.2.3数据流图**

**顶层层：**

****

图1.1顶层数据流图

**第0层：**

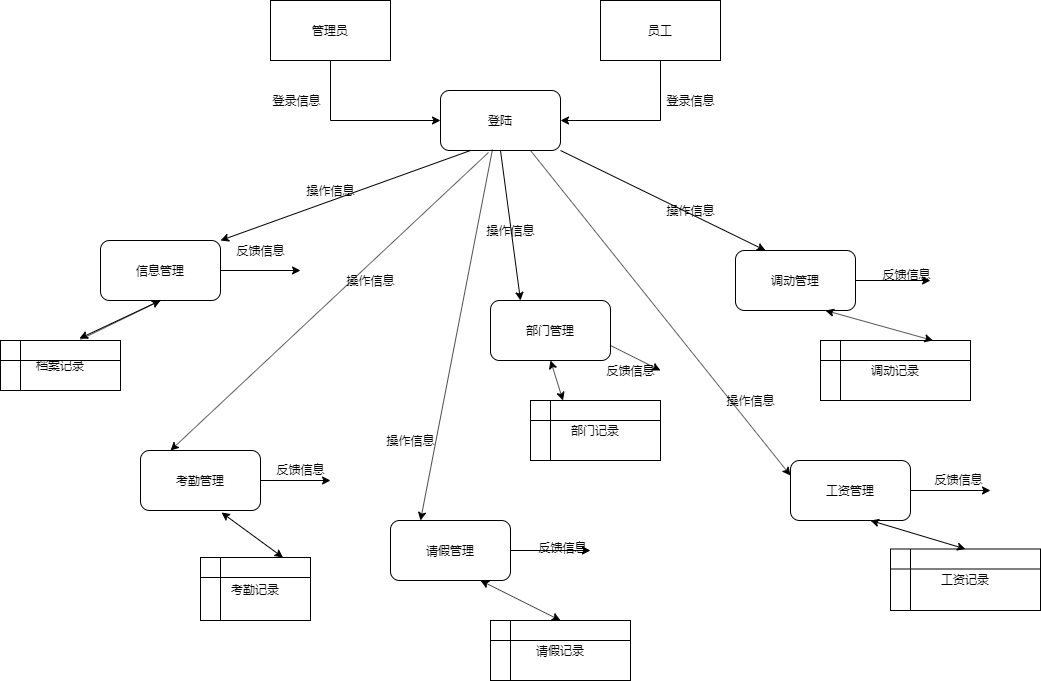


图1.2第0层数据流图

**第1层**

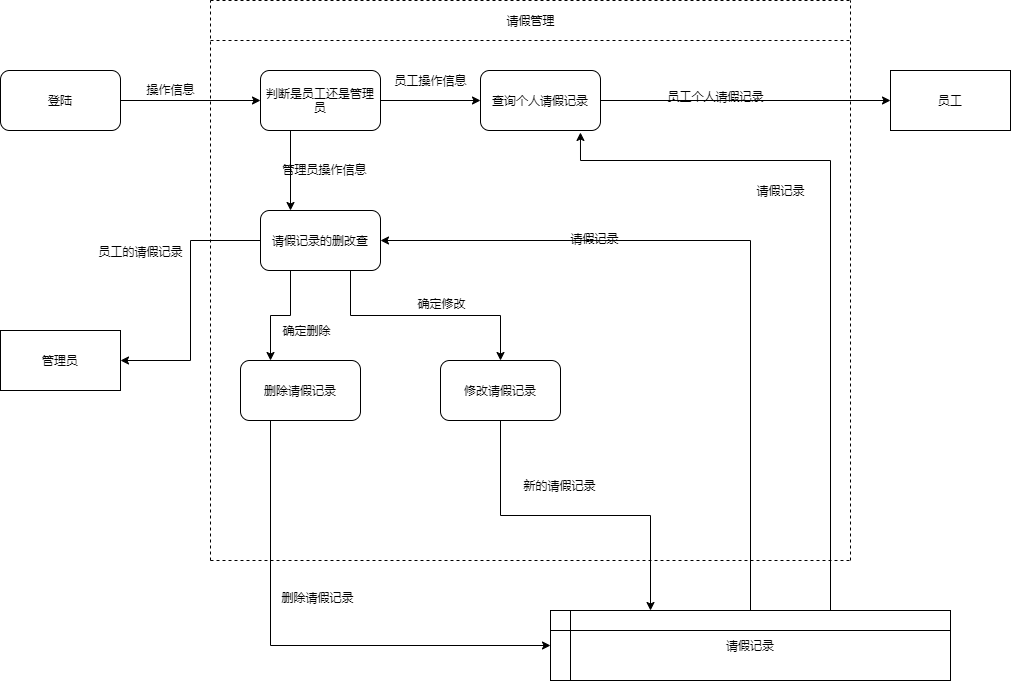


图1.3请假管理数据流图

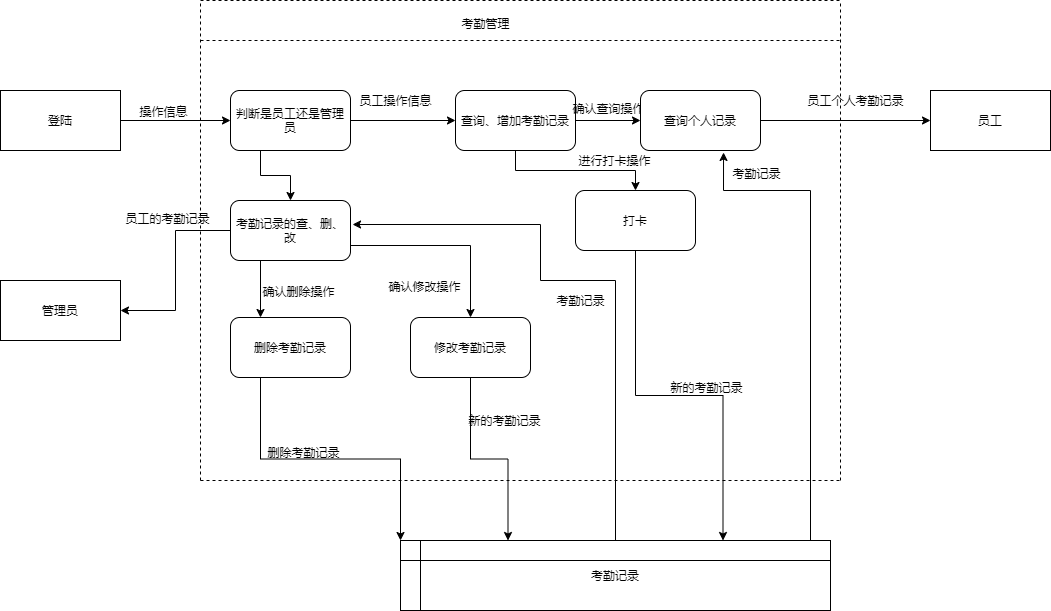


图1.4考勤管理数据流图

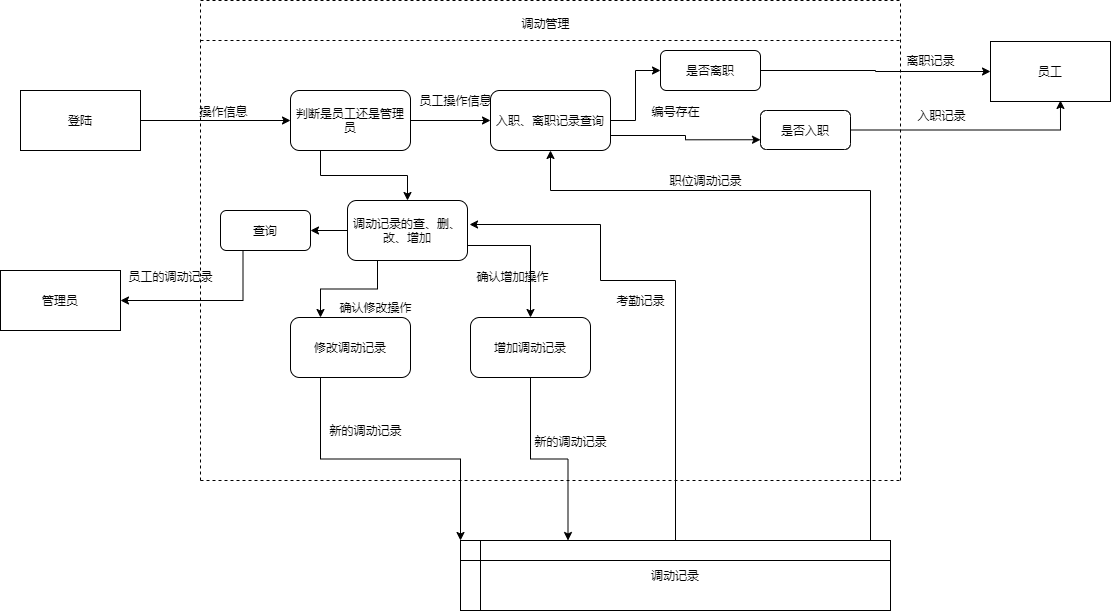


图1.5调动管理数据流图

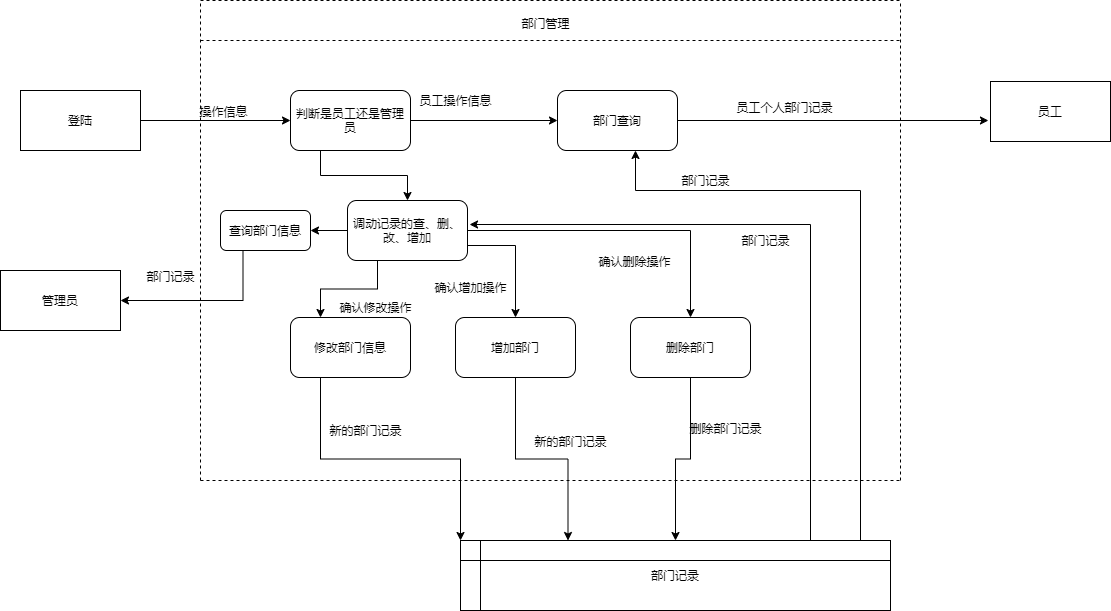


图1.6部门管理数据流图

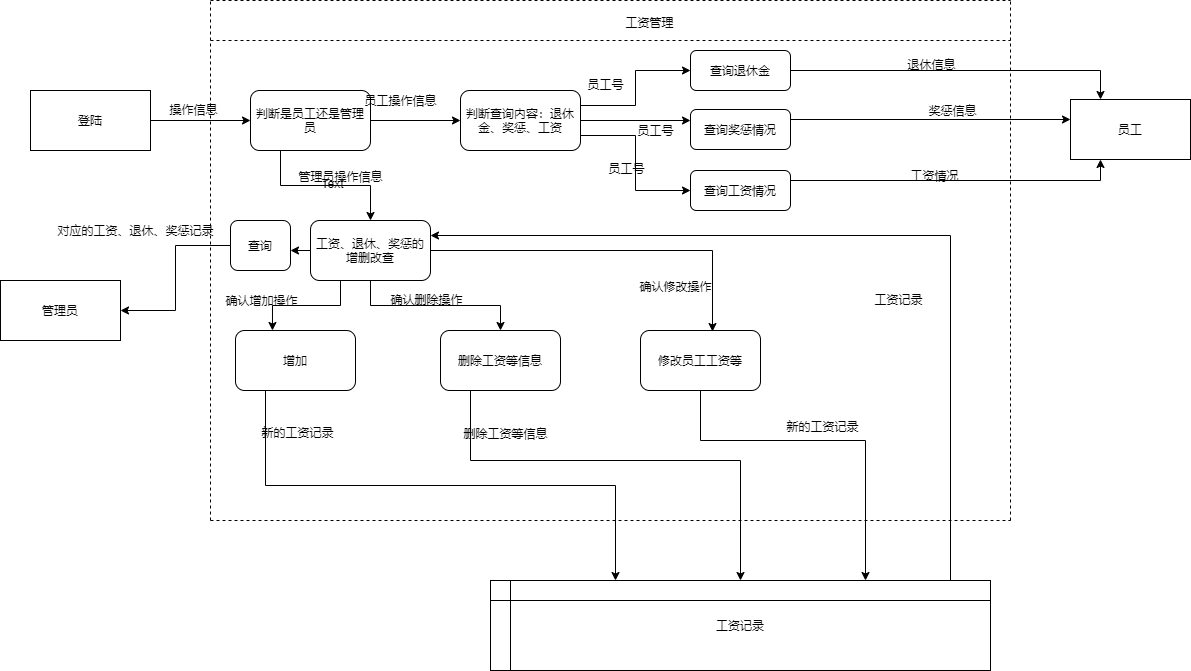


图1.7工资管理数据流图

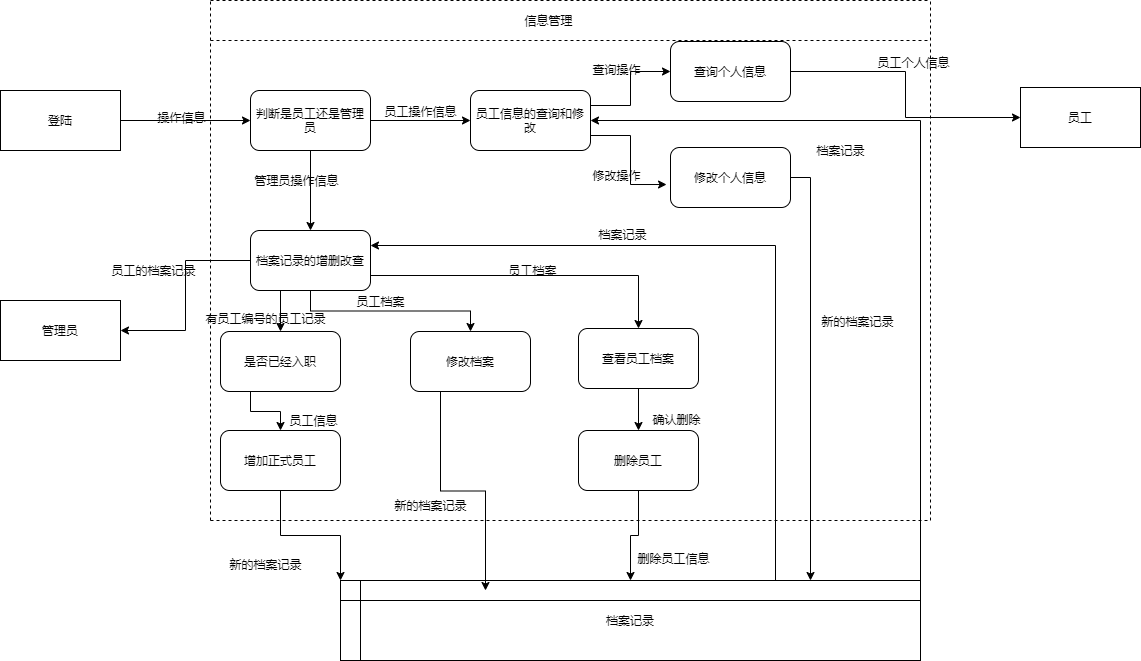
****

图1.8信息管理数据流图

**1.2.3数据字典**

**主要的数据流说明：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据流名称 | 组成 | 数据流来源 | 数据流去向 | 说明 |
| 操作信息 | 判断员工和管理员的标志、操作者的编号 | 登陆 | 判断员工还是管理员 | 无 |
| 员工操作信息 | 员工号、员工的操作方式 | 判断员工还是管理员 | 员工信息的查询和修改 | 员工号唯一，操作方式只有一种 |
| 档案记录 | 员工号、职位、部门编号、学历、姓名、婚姻状态、性别 | 档案记录 | 员工信息的查询和修改、档案记录的增删改查 | 无 |
| 员工档案 | 员工号、职位、部门编号、学历、姓名、婚姻状态、性别 | 档案记录的增删改查 | 查看员工档案、修改档案 | 无 |
| 有员工编号的员工记录 | 员工号 | 档案记录的增删改查 | 是否已经入职 | 无 |
| 新的档案记录 | 员工号、职位、部门编号、学历、姓名、婚姻状态、性别 | 增加正式员工、修改档案 | 档案记录 | 员工号唯一 |
| 员工信息 | 员工号、学历、姓名、婚姻状态、性别 | 员工 | 增加正式员工 | 新入职的员工号唯一 |
| 反馈信息 | 各种记录 | 人事管理系统 | 管理员和员工 | 无 |
| 登录信息 | 员工或者管理员的账号密码 | 管理员、员工 | 登陆 | 无 |
| 员工的考勤记录 | 上班时间、下班时间、日期、员工号 | 考勤记录的增删改查 | 管理员 | 通过日期和员工号可以得到唯一的考勤记录 |
| 员工个人考勤记录 | 上班时间、下班时间、日期 | 查询个人记录 | 员工 | 无 |
| 请假记录 | 开始时间、结束时间、部门号、经理号、批准状态、员工号、申请时间、请假编号 | 请假记录 | 查询个人请假记录、请假记录的删改查 | 请假编号唯一，可得到唯一的请假记录 |
| 工资记录 | 员工号、奖惩金额、退休金额、基本工资、发放时间、工资编号、部门编号 | 工资记录 | 工资记录的增删改查 | 工资编号唯一 |
| 奖惩信息 | 员工号、奖惩原因、奖惩类型、奖惩时间、奖惩金额、部门号、奖惩编号 | 奖惩的查询 | 员工 | 对每一次奖惩进行唯一编号 |
| 退休信息 | 员工号、退休时间、退休金、部门号、退休编号 | 退休的查询 | 员工 | 对退休记录进行唯一编号 |
| 工资信息 | 员工号、奖惩金额、基本工资、发放时间、工资编号、部门编号 | 工资信息的查询 | 员工 | 工资编号唯一 |
| 部门记录 | 部门号、部门人数、部门名称、部门记录 | 部门记录 | 部门记录的增删改查 | 部门号唯一 |
| 调动记录 | 员工号、调动编号、新部门编号、原部门编号、经理号 | 调动记录 | 调动记录的增删改查 | 调动编号唯一 |
| 离职记录 | 员工号、辞职编号、辞职时间、部门名称 | 是否离职 | 员工 | 辞职编号唯一 |
| 入职记录 | 员工号、入职时间、职位、部门 | 是否入职 | 员工 | 无 |

**主要的数据存储说明：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据存储编号 | 输入 | 输出 | 数据结构 | 说明 | 存取方式 |
| 档案记录 | 信息管理， | 信息管理 | 姓名、性别、员工编号、学历、职位、婚姻状况、部门号、状态、 | 员工编号具有唯一性和非空性；性别只能是男或女；主码是员工编号 | 联机处理；检索和更新；以随机检索为主 |
| 考勤记录 | 考勤管理 | 考勤管理 | 员工号、上班时间、下班时间、日期（年月日） | 主码是员工号和日期；员工号是外码，参照表是员工记录 | 联机处理；检索和更新；以更新操作为主 |
| 工资记录 | 工资管理 | 工资管理 | 员工号、基本工资、奖惩金额、发放时间、工资编号、退休编号、发放部门编号 | 主码为员工号和发放时间；员工号是外码，参照表是员工记录 | 联机处理；检索和更新；以更新操作为主 |
| 部门记录 | 部门管理 | 部门管理 | 部门号、部门名称、部门经理、部门人数 | 主码为部门号 | 联机处理；检索和更新；以检索操作为主 |
| 调动记录 | 调动管理 | 调动管理 | 员工号，调动编号、调新部门编号、原部门编号、经理编号 | 调动编号唯一 | 联机处理；检索和更新；以更新操作为主 |
| 请假记录 | 请假管理 | 请假管理 | 开始时间、结束时间、请假编号、部门编号、经理编号、批准状态、员工号、申请时间 | 请假编号唯一 | 联机处理；检索和更新；以更新操作为主 |

**主要的处理过程：**

|  |  |  |  |
| --- | --- | --- | --- |
| 处理过程名 | 输入 | 输出 | 处理说明 |
| 信息管理 | 档案记录、操作信息 | 档案记录 | 根据员工的聘用或删除员工记录，以及在后续的过程中，以管理员身份对员工  的记录进行修改。 |
| 考勤管理 | 操作信息、考勤记录 | 考勤记录 | 根据上下班时间判断是否早退或者迟到，方便经理查看考勤记录 |
| 调动管理 | 调动记录、操作信息 | 调动记录 | 记录员工的部门调动以及辞职等人事调动情况，方便经理管理并查询 |
| 工资管理 | 工资记录、操作信息 | 工资记录 | 根据员工目前的工资情况，进行调薪的操作。 |
| 部门管理 | 部门记录、操作信息 | 部门记录 | 根据上级的调度，对现在公司的部门]进行领导人的调换，或者是增加或删除某一部门。 |
| 请假管理 | 操作信息、请假记录 | 请假记录 | 根据员工的请假申请和批复情况，进行请假表的登记 |

**主要的数据结构**

|  |  |  |
| --- | --- | --- |
| 数据结构名 | 含义说明 | 组成 |
| 员工 | 是人事管理系统的主体数据结构，定义了一个员工的有关信息 | 姓名、性别、职称、学历、婚姻状况、部门号、状态、员工号、年龄 |
| 管理员 | 是人事管理系统的主体数据结构，定义了一个管理员的有关信息 | 员工号、管理员账号、密码 |

员工表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 员工编号 | varchar | 20 | 主键 | No |
| 姓名 | varchar | 20 |  | No |
| 性别 | varchar | 2 | 默认为0 | No |
| 年龄 | smallint | 3 | 大于0，小于150 | No |
| 学历 | varchar | 5 | 本科、硕士、博士、小学、初中、高中、大专，中专 | No |
| 婚姻状况 | int | 2 | 0-2之间 |  |
| 职位 | varchar | 20 |  |  |
| 所在部门编号 | varchar | 20 | 外键 |  |
| 在职状态 | varchar | 10 |  | No |
| 禁用 | int |  | 默认为0 | No |

管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 管理员账号 | varchar | 20 | 主键 | No |
| 密码 | varchar | 100 |  | No |
| 权限 | smallint |  | default 1 |  |

考勤表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 考勤编号 | varchar | 20 | 主键 | No |
| 员工编号 | varchar | 20 |  | No |
| 上班时间 | time |  |  | No |
| 下班时间 | time |  |  | No |
| 日期 | date |  | default getdate（） |  |

奖惩表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 奖惩编号 | varchar | 20 | 主键 | No |
| 员工编号 | varchar | 20 | 外键 | No |
| 部门编号 | varchar | 20 | 外键 | No |
| 金额 | float |  | 大于0 |  |
| 奖惩原因 | varchar | 40 |  |  |

调动表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 员工工号 | varchar | 20 | 外键 | No |
| 调动编号 | varchar | 20 | 主键 | No |
| 新部门编号 | varchar | 20 | 外键 | No |
| 管理员编号 | varchar | 20 | 外键 | No |
| 原部门编号 | varchar | 20 | 外键 | No |
| 调动时间 | date |  | default getdate（） |  |
| 调动原因 | varchar | 100 |  | No |

工资表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 工资编号 | varchar | 20 | 主键 | No |
| 员工工号 | varchar | 20 | 外键 | No |
| 发放时间 | datetime |  |  |  |
| 基本工资 | flaot |  | 大于0 |  |
| 加班工资 | float |  | 大于0 |  |
| 迟到扣除 | float |  |  |  |

部门表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 部门编号 | varchar | 20 | 主键 | No |
| 部门名称 | varchar | 30 |  | No |
| 人数 | int | 5 | 大于0 |  |
| 禁用 | int |  | 默认为0 |  |

请假表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 开始时间 | datetime |  |  |  |
| 请假编号 | varchar | 20 | 主键 | No |
| 申请时间 | datetime |  |  |  |
| 员工编号 | varchar | 20 | 外键 | No |
| 批准状态 | varchar | 10 |  |  |
| 管理员编号 | varchar | 20 | 外键 | No |
| 结束时间 | datetime |  |  |  |
| 请假原因 | varchar | 100 |  | No |

退休表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 退休编号 | varchar | 20 | 主键 | No |
| 员工编号 | varchar | 20 | 外键 | No |
| 退休时间 | datetime |  |  | No |
| 退休原因 | varchar | 40 |  | No |
| 退休金 | float |  | 大于0 |  |

离职表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 类型 | 长度 | 约束 | 是否可空 |
| 离职编号 | varchar | 20 | 主键 | No |
| 离职时间 | datetime |  |  |  |
| 员工编号 | varchar | 20 | 外键 | No |
| 离职原因 | varchar | 40 |  |  |

**1.3系统功能模块设计**

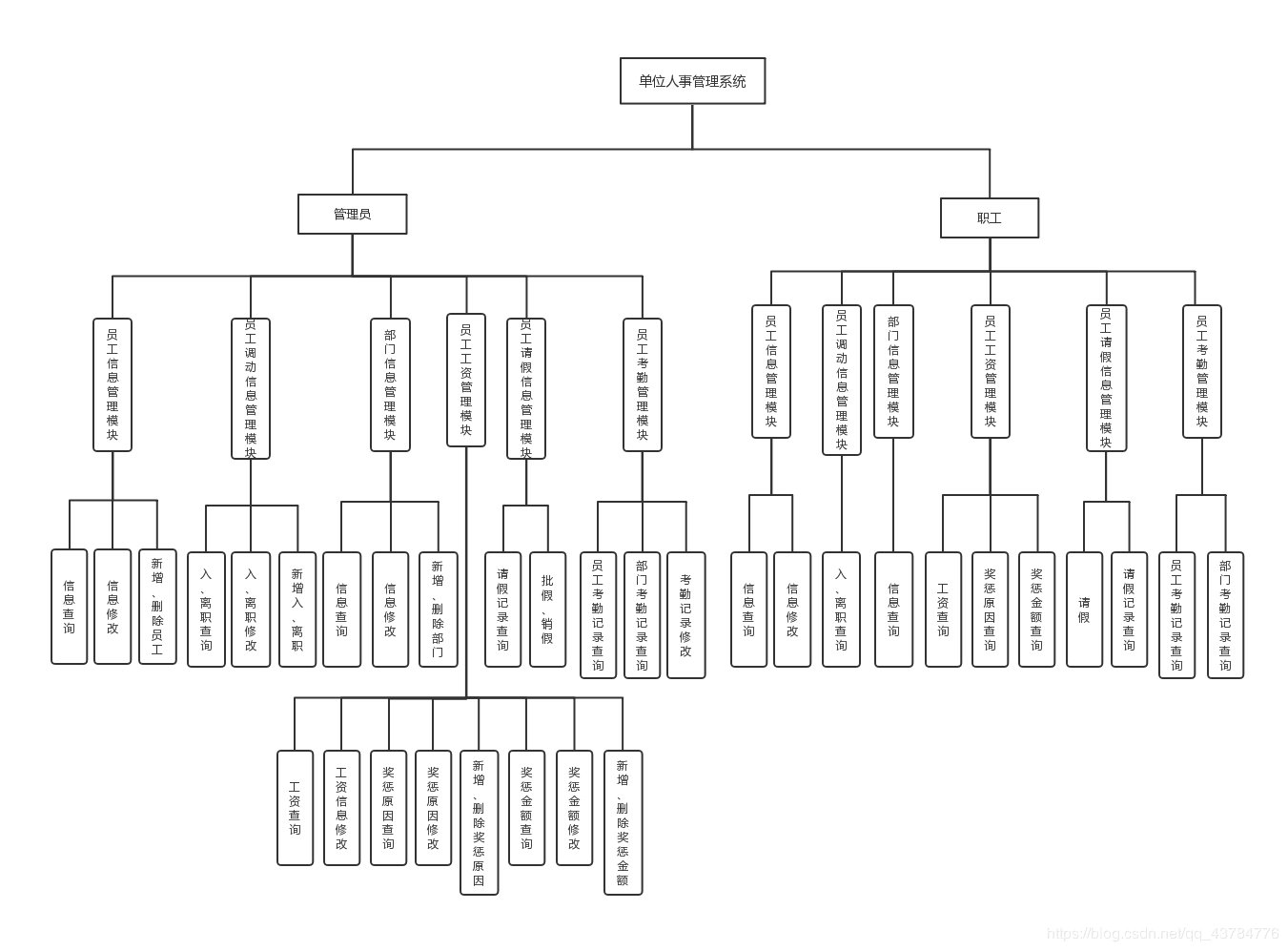
****

图1.9功能模块图

1. **数据库设计**

**2.1数据库概念设计**

**2.1.1分E-R图**

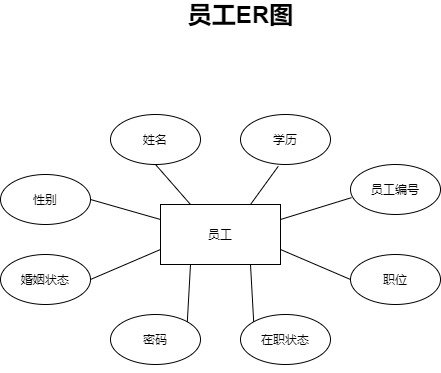
****

图2.1

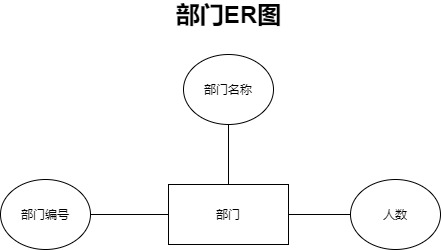
****

图2.2

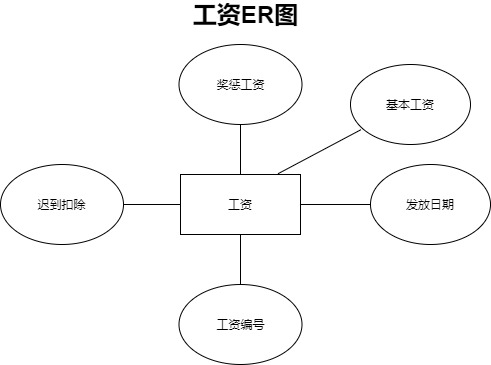
****

图2.3

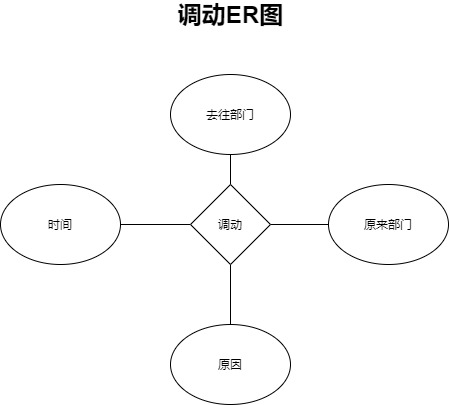
****

图2.4

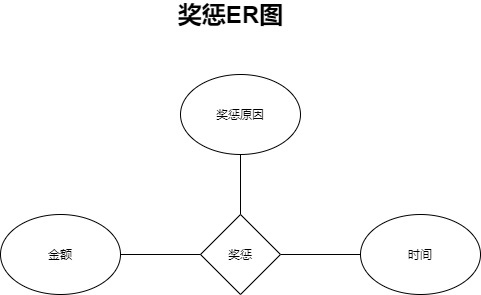
****

图2.5

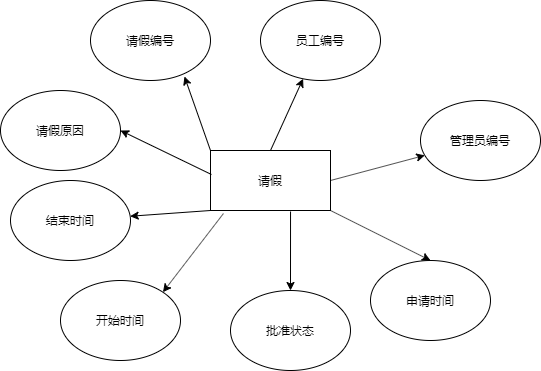
****

图2.6

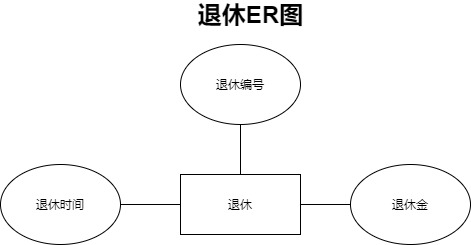
****

图2.7

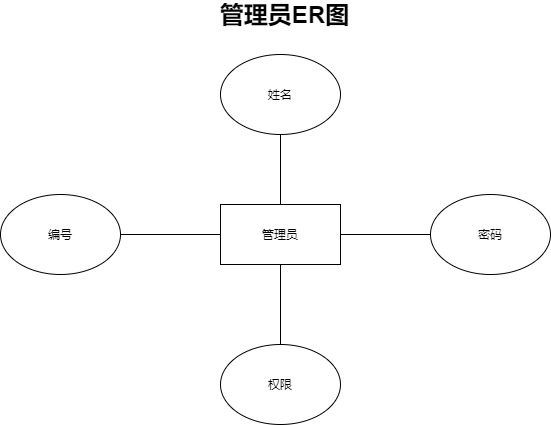
****

图2.8

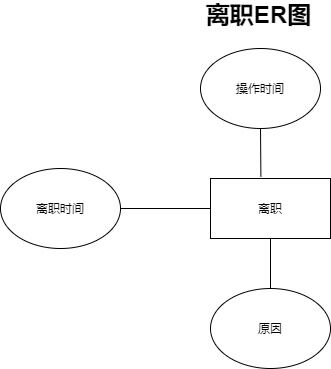


图2.9

**2.1.2系统基本E-R图**

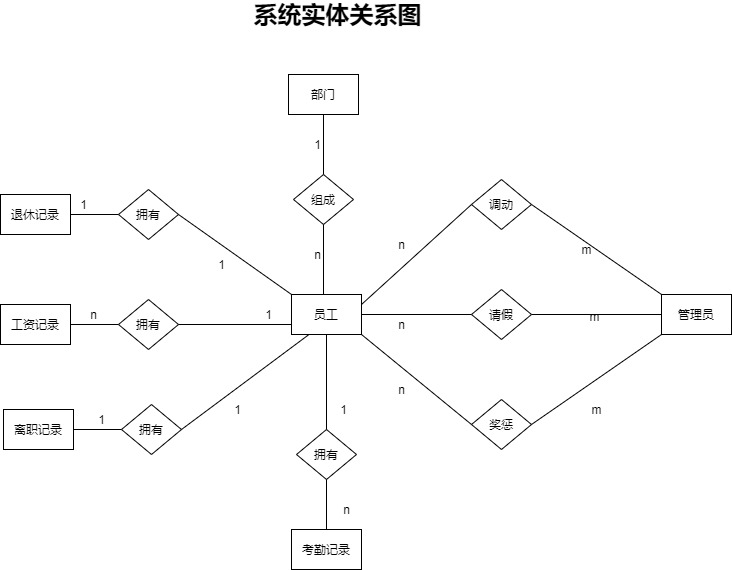
****

图2.10

**2.2数据库逻辑结构设计**

**2.2.1E-R图向关系模型的转换**

部门(部门编号，部门名称，人数)

员工(员工编号，姓名，性别，学历，年龄，婚姻状态，毕业时间，部门编号，职位，在职状态，密码)

管理员(管理员编号，姓名，密码,权限)

考勤(考勤编号，上班时间，下班时间，员工编号，日期)

调动(员工编号，管理员编号，调动原因，旧部门编号，新部门编号,管理员编号)

工资(工资编号，员工编号，日期，基本工资，加班工资，迟到扣除)

奖惩(员工编号，管理员编号，奖惩原因，管理员编号，奖惩金额)

请假(员工编号，管理员编号,申请时间，批准状态，开始时间，结束时间)

离职(离职编号，离职时间，员工编号，离职原因)

退休(员工编号，退休时间，退休原因，退休金)

**2.2.2数据模型的优化**

部门(部门编号，部门名称，人数，禁用)。

员工(员工编号，姓名，性别，学历，婚姻状态，部门编号，职位，在职状态，禁用)。部门编号是外键。

管理员(账号，密码,权限)。

考勤(考勤编号，上班时间，下班时间，员工编号，日期)。

调动(调动编号，员工编号，管理员编号，调动时间，调动原因，旧部门编号，新部门编号)。员工编号，管理员编号，部门编号是外键。

工资(工资编号，员工编号，日期，基本工资，加班工资，迟到扣除)。员工编号是外键。

奖惩(奖惩编号，员工编号，管理员编号，奖惩原因，奖惩金额)。员工编号，管理员编号是外键。

请假(请假编号，员工编号，管理员编号,申请时间，批准状态，开始时间，结束时间，请假原因)。员工编号，管理员编号是外键。

离职(离职编号，离职时间，员工编号，离职原因)。员工编号是外键。

退休(退休编号、员工编号，退休时间，退休原因，退休金)。员工编号是外键

**用户子模式：**

员工账号(员工编号，密码)

员工调动(调动编号，调动原因，新部门，旧部门)

员工请假(请假编号，员工编号，申请时间，批准状态)

**2.3数据库物理设计**

**2.4.1索引的设计**

1. 唯一性索引的值是唯一的，可以更快的通过该索引来确定某条记录，主键索引和唯一键索引，在查询中使用是效率最高的。
2. 为经常需要排序，分组和联合操作的字段建立索引。
3. 为常作为where查询条件的字段建立索引。
4. 限制索引的数目，索引不是越多越好，每个索引都需要占用磁盘空间。
5. 不是所有的表都适合建立索引，只有数据量大表才适合建立索引。

**2.4.2索引的建立**

由索引的设计，对部门名称、新部门号、旧部门号，禁用字段建立索引。

1. **数据库的实现与调试**

**3.1建立人事管理系统数据库**

create database PeoData

on primary -- 主数据文件

(

name='PeoData',

filename='d:\db\PeoData.mdf',

size=3MB,

maxsize=UNLIMITED,

filegrowth=2MB

)

log on -- 日志文件

(

name='PeoLog',

filename='d:\db\ PepLog.ldf',

size=1MB,

maxsize=10MB,

filegrowth=10%

)

**3.2建立数据库的各基本表**

**create table department(**

**dno varchar(20) primary key, --部门编号**

**dname varchar(30) not null, --部门名称**

**dnum int check(dnum>0) --部门人数**

**deleted int default 0**

**);**

**--建立员工表**

**create table staff(**

**sno varchar(20) PRIMARY key, --员工编号**

**sname varchar(20) not null, --姓名**

**ssex varchar(5) check(ssex in (‘男’,’女’)), --性别**

**sxl varchar(5) not null CHECK(sxl in('本科','硕士','博士','小学','初中','高中','大专','中专','其他')),**

**smarrige varchar(5) check(smarrige in (‘已婚’,’未婚’)), --婚姻状态**

**sposition varchar(20), --职位**

**sstatus varchar(10) , --在职状态**

**sdept varchar(20) REFERENCES department(dno), --部门编号**

**deleted int default 0**

**)**

**--建立管理员表**

**create table admin(**

**ano varchar(20) PRIMARY key, --管理员账号**

**pwd varchar(100) not null, --密码**

**pvl smallint DEFAULT 1 --权限**

**);**

**--建立考勤表**

**create table attend(**

**ano varchar(20) primary key, --考勤编号**

**stime TIME(0) not null, --上班时间**

**etime TIME(0) not null, --下班时间**

**sno varchar(20) REFERENCES staff(sno), --员工编号**

**adate DATE default GETDATE() --日期**

**);**

**--建立调动表**

**create table transfer(**

**tno varchar(20) primary key, --调动编号**

**sno varchar(20) REFERENCES staff(sno), --员工编号**

**ano varchar(20) REFERENCES admin(ano), --管理员编号**

**ttime date default GETDATE(), --调动时间**

**treason varchar(100) not null, --调动原因**

**odno varchar(20) REFERENCES department(dno), --旧部门号**

**ndno varchar(20) REFERENCES department(dno) --新部门号**

**);**

**--工资表**

**create table salary(**

**sano varchar(20) primary key, --工资编号**

**sno varchar(20) references staff(sno), --员工编号**

**sa\_time datetime, --日期**

**sa\_basic float check(sa\_basic > 0), --基本工资**

**sa\_jiaban float check(sa\_jiaban > 0), --加班工资**

**sa\_late float --迟到扣除**

**）**

**--奖惩表**

**create table reward(**

**jcno varchar(20) primary key, --奖惩编号**

**sno varchar(20) not null, --员工编号**

**ano varchar(20) not null, --管理员编号**

**reason varchar(40), --奖惩原因**

**kind varchar(5) not null check(kind in (‘奖励’,’惩罚’)),**

**money float check(money > 0), --奖惩金额**

**foreign key(sno) references staff(sno),**

**foreign key(ano) references admin(ano)**

**)**

**--建立请假表**

**create table leave(**

**lno varchar(20) primary key, --请假编号**

**sno varchar(20) not null, --员工编号**

**ano varchar(20) not null, --管理员编号**

**reason varchar(100) not null, --请假原因**

**l\_time date , --申请时间**

**l\_start date , --开始时间**

**l\_end date , --结束时间**

**foreign key(sno) references staff(sno),**

**foreign key(ano) references admin(ano)**

**)**

**--建立离职表**

**create table quit(**

**qno varchar(20) primary key, --离职编号**

**q\_time date, --离职时间**

**sno varchar(20) not null, --员工编号**

**q\_reason varchar(40) --离职原因**

**foreign key(sno) references staff(sno)**

**)**

**--建立退休表**

**create table retire(**

**rno varchar(20) primary key, --退休编号**

**sno varchar(20) not null, --员工编号**

**r\_time date not null, --退休时间**

**r\_money float check(r\_money > 0) --退休金**

**foreign key(sno) references staff(sno)**

**)**

**3.3插入数据**

管理员表：

insert into admin values('123456','520',2)

insert into admin values('159753','250',1)

insert into admin values('159159','3600',2)

insert into admin values('126578','156',1)

insert into admin values('147895','111',2)

员工表：

insert into staff values('1234567890','老王','男','小学','已婚','服务员','工作','123','200010')

insert into staff values('1234567891','张三','男','本科','已婚','电工','休假','124','200011')

insert into staff values('1234567892','阿花','女','硕士','未婚','销售人员','工作','125','200012')

insert into staff values('1234567893','李四','男','博士','已婚','hr','工作','126','200013')

insert into staff values('1234567894','王舞','女','本科','已婚','副总经理','休假','127','200014')

insert into staff values('1234567895','赵六','男','硕士','已婚','会计师','工作','128','200015')

insert into staff values('1234567896','老周','男','本科','未婚','总经理','工作','127','200016')

insert into staff values('1234567897','小二','男','本科','已婚','前端工程师','工作','124','200017')

insert into staff values('1234567898','老八','男','初中','已婚','销售人员','辞职','123','200018')

insert into staff values('1234567899','宝儿姐','女','硕士','已婚','财务部长','工作','128','200019')

insert into staff values('1234567900','瓜子哥','男','本科','已婚','销售组长','工作','125','200020')

insert into staff values('1234567901','彬彬姐','女','高中','已婚','销售人员','工作','125','200021')

insert into staff values('1234567902','老三','男','硕士','已婚','会计师','工作','128','200022')

insert into staff values('1234567903','小四','男','博士','已婚','算法工程师','工作','124','200023')

insert into staff values('1234567904','海海','男','博士','已婚','后端工程师','工作','124','200024')

insert into staff values('1234567905','麟麟','男','本科','已婚','后端工程师','工作','124','200025')

部门表：

insert into department values('123','后勤部','20')

insert into department values('124','技术部','15')

insert into department values('125','销售部','40')

insert into departm ent values('126','人事部','10')

insert into department values('127','行政部','8')

insert into department values('128','财务部','6')

考勤表：

insert into attend values('1234','07:32:08','18:50:20','1234567890','2020-06-01')

insert into attend values('1235','07:23:08','17:50:20','1234567890','2020-06-02')

insert into attend values('1236','08:32:08','18:56:20','1234567891','2020-06-01')

insert into attend values('1237','07:05:08','20:23:20','1234567891','2020-06-02')

insert into attend values('1238','07:01:08','22:21:20','1234567903','2020-06-01')

insert into attend values('1239','07:06:08','22:27:20','1234567903','2020-06-02')

insert into attend values('1240','08:01:08','17:21:20','1234567896','2020-06-01')

调动表：

insert into transfer values('12','1234567890','123456','2020-05-31','上班撸猫','125','123')

insert into transfer values('13','1234567896','159753','2020-05-31','家里有关系','125','127')

insert into transfer values('14','1234567890','123456','2020-05-30','上班不认真','124','125')

insert into transfer values('15','1234567895','147895','2020-05-26','上班积极','123','128')

insert into transfer values('16','1234567900','123456','2020-05-31','业绩很好','125','125')

insert into transfer values('17','1234567903','126578','2020-05-12','很有潜力','123','124')

insert into transfer values('18','1234567893','123456','2020-05-31','用人很准','126','127')

工资表：

insert into salary values('20','1234567890','2020-05-01 18:00:00',3000.2,52.1,0)

insert into salary values('21','1234567891','2020-05-01 18:00:00',7260,1000,0)

insert into salary values('22','1234567892','2020-05-01 18:00:00',4500,569,100)

insert into salary values('23','1234567893','2020-05-01 18:00:00',13200,2600,0)

insert into salary values('24','1234567894','2020-05-01 18:00:00',20000,2300,200)

insert into salary values('25','1234567895','2020-05-01 18:00:00',8520,900,100)

insert into salary values('26','1234567896','2020-05-01 18:00:00',30000,4623,500)

奖惩表：

insert into reward values('1','1234567890','123456','上班撸猫','罚款',1000)

insert into reward values('2','1234567894','159753','长的漂亮','奖励',520)

insert into reward values('3','1234567899','123456','最近一段时间财务部效率很高','奖励',2000)

insert into reward values('4','1234567903','159159','算法效率高','奖励',1500)

insert into reward values('5','1234567898','123456','销售冠军','奖励',800)

insert into reward values('6','1234567897','123456','页面做的很漂亮','奖励',1100)

请假表：

insert into leave values('01','1234567890','123456','回家撸猫','2020-05-21','不允许','2020-05-22','2020-05-25')

insert into leave values('02','1234567897','159159','病假','2020-05-24','允许','2020-05-25','2020-05-28')

insert into leave values('03','1234567894','123456','产假','2020-05-04','允许','2020-05-05','2020-05-25')

insert into leave values('04','1234567892','147895','回家探亲','2020-05-15','允许','2020-05-20','2020-05-22')

insert into leave values('05','1234567900','123456','度蜜月','2020-05-21','不允许','2020-06-01','2020-06-08')

insert into leave values('06','1234567890','123456','在不撸猫我要死了','2020-06-05','允许','2020-06-10','2020-06-15')

离职表：

insert into quit values('857','2020-06-08','1234567898','找到更好的工作了')

退休表：

insert into retire values('111','1234567904','2020-04-23','头发没了',2000)

**3.4建立索引**

create index department\_dname on department(dname)

create index department\_dnum on department(dnum)

create index transfer\_odno on transfer(odno)

create index transfer\_ndno on transfer(ndno)

**3.5视图**

1. 员工调动（调动编号，调动原因，新部门，旧部门）

create view view\_depart\_1 as

select dno, dname from department

create view view\_attend\_1 as

select tno , treason ,d.dname newdepart, vd.dname olddepart

from transfer,department d,view\_depart\_1 vd

where odno = vd.dno and ndno = d.dno



图3.1

**3.6存储过程**

1. 添加员工信息的存储过程

create procedure staff\_insert

@sname varchar(20),

@ssex varchar(5),

@sxl varchar(5),

@smarrige varchar(5),

@sposition varchar(20),

@sdept varchar(20)

as

begin

declare @maxsno varchar(20),@max int

select @maxsno = MAX(sno) from staff

set @max = CONVERT(int,@maxsno)

set @max+=1

set @maxsno = CONVERT(varchar(20),@max)

insert into staff

values(@maxsno,@sname,@ssex,@sxl,@smarrige,@sposition,'工作',@sdept,0)

end

1. 管理员更新员工的信息

create procedure admin\_update\_staff

@sno varchar(20),

@sname varchar(20),

@ssex varchar(5),

@sxl varchar(5),

@smarrige varchar(5),

@sposition varchar(20),

@sstatus varchar(10)

as

begin

update staff

set sno = @sno,sname = @sname,ssex = @ssex,sxl = @sxl

,smarrige = @smarrige,sposition = @sposition,

sstatus = @sstatus

where sno = @sno

end

1. 考勤记录的增加

create procedure att\_insert

@stime time(0),

@etime time(0),

@sno varchar(20)

as

begin

declare @maxano varchar(20),@max int

select @maxano = MAX(ano) from attend

set @max = CONVERT(int,@maxano)

set @max+=1

set @maxano = CONVERT(varchar(20),@max)

insert into attend

values(@maxano,@stime,@etime,@sno,GETDATE())

end

1. 调动表的增加

create procedure transfer\_insert

@sno varchar(20),

@ano varchar(20),

@treason varchar(100),

@ndno varchar(20)

as

begin

declare @maxtno varchar(20),@max int,@odno varchar(20)

select @maxtno = MAX(tno) from transfer

set @max = CONVERT(int,@maxtno)

set @max+=1

set @maxtno = CONVERT(varchar(20),@max)

select @odno = sdept from staff where sno = @sno

insert into transfer

values(@maxtno,@sno,@ano,GETDATE(),@treason,@odno,@ndno)

end

1. 工资记录的增加

create procedure salary\_insert

@sno varchar(20),

@basic float

as

begin

declare @late float,@ltime time(0)

--进行迟到扣款计算

begin

declare findlate cursor

for select stime from attend

where sno = @sno

open findlate

fetch next from findlate into @ltime

set @late = 0

while @@FETCH\_STATUS = 0

begin

if @ltime > '09:00:00'

set @late+=50

fetch next from findlate into @ltime

end

close findlate

deallocate findlate

end

--进行加班工资计算

declare @jiaban float,@etime time(0)

begin

declare findjiaban cursor

for select etime from attend

where sno = @sno

open findjiaban

fetch next from findjiaban into @etime

set @jiaban = 0

while @@FETCH\_STATUS = 0

begin

if @etime > '18:00:00' and @etime <= '19:00:00'

set @jiaban += 30

if @etime > '19:00:00' and @etime <= '20:00:00'

set @jiaban += 60

if @etime > '20:00:00' and @etime <= '21:00:00'

set @jiaban += 90

if @etime > '21:00:00' and @etime <= '22:00:00'

set @jiaban += 120

if @etime > '22:00:00'

set @jiaban += 180

fetch next from findjiaban into @etime

end

close findjiaban

deallocate findjiaban

end

--准备给工资表插入数据

declare @maxsano varchar(20),@max int

select @maxsano = MAX(sano) from salary

set @max = CONVERT(int,@maxsano)

set @max+=1

set @maxsano = CONVERT(varchar(20),@max)

insert into salary

values(@maxsano,@sno,GETDATE(),@basic,@jiaban,@late)

end

1. 添加奖惩的存储过程

create procedure reward\_insert

@sno varchar(20),

@ano varchar(20),

@reason varchar(40),

@kind varchar(5),

@money float

as

begin

declare @maxjcno varchar(20),@max int

select @maxjcno =jcno from reward

set @max = CONVERT(int,@maxjcno)

set @max+=1

set @maxjcno = CONVERT(varchar(20),@max)

insert into reward

values(@maxjcno,@sno,@ano,@reason,@kind,@money)

end

1. 添加请假的存储过程

create procedure lea\_insert

@sno varchar(20),

@ano varchar(20),

@reason varchar(100),

@l\_start date,

@l\_end date

as

begin

declare @maxlno varchar(20),@max int

select @maxlno = MAX(lno) from leave

set @max = CONVERT(int,@maxlno)

set @max+=1

set @maxlno = CONVERT(varchar(20),@max)

insert into leave

values(@maxlno,@sno,@ano,@reason,GETDATE(),@l\_start,@l\_end)

end

1. 增加离职的存储过程

create procedure quit\_insert

@sno varchar(20),

@q\_reason varchar(40)

as

begin

declare @maxqno varchar(20),@max int

select @maxqno = MAX(qno) from quit

set @max = CONVERT(int,@maxqno)

set @max+=1

set @maxqno = CONVERT(varchar(20),@max)

update staff set exist = 1 where sno = @sno

insert into quit

values(@maxqno,GETDATE(),@sno,@q\_reason)

end

1. 增加退休的存储过程

create procedure re\_insert

@sno varchar(20),

@r\_time date,

@r\_money float

as

begin

declare @maxrno varchar(20),@max int

select @maxrno = MAX(rno) from retire

set @max = CONVERT(int,@maxrno)

set @max+=1

set @maxrno = CONVERT(varchar(20),@max)

update staff set exist = 1 where sno = @sno

insert into retire

values(@maxrno,@sno,@r\_time,@r\_money)

end

1. 增加部门的存储过程

create procedure department\_insert

@dname varchar(30)

as

begin

declare @dno varchar(20),@max int

select @dno = MAX(dno) from department

set @max = CONVERT(int,@dno)

set @max+=1

set @dno = CONVERT(varchar(20),@max)

insert into department

values(@dno,@dname,1,0)

end

**3.7触发器**

1. 当新加入员工时，对应部门的人数加1

create trigger Depart\_Insert on staff for insert

as

declare @depart\_num int, @dno1 varchar(20)

select @dno1 = sdept from inserted

select @depart\_num = dnum from department where dno = @dno1

set @depart\_num+=1

update department set dnum = @depart\_num where dno = @dno1

1. 当有员工辞职时，对应部门的人数减1

create trigger Depart\_Insert2 on quit for insert

as

declare @sno varchar(20), @dno varchar(20),@dnum int

select @sno = sno from inserted

select @dno = sdept from staff where sno = @sno

select @dnum = dnum from department where dno = @dno

set @dnum-=1

update department set dnum = @dnum where dno = @dno

1. 当有员工辞职时，对应的员工表的‘状态’属性变为辞职

create trigger Staff\_Update on quit for insert

as

declare @sno varchar(20),@sstatus varchar(10)

select @sno = sno from inserted

set @sstatus = '离职'

update staff set sstatus = @sstatus where sno = @sno

1. 当有员工调动时，对应的新部门人数加一，旧部门人数减一。并且员工表中部门号自动修改。

create trigger Depart\_Update on transfer for insert

as

declare @ndno varchar(20),@odno varchar(20)

declare @sno varchar(20),@ndnum int,@odnum int

--获取插入的调动表中员工号、新旧部门号

select @sno = sno from inserted

select @ndno = ndno from inserted

select @odno = odno from inserted

--对员工表的部门号进行更新

update staff set sdept = @ndno where sno = @sno

--获取新旧部门对应的人数

select @ndnum = dnum from department where dno = @ndno

select @odnum = dnum from department where dno = @odno

--新部门+1，旧部门-1

set @ndnum+=1

set @odnum-=1

--更新新旧部门的人数

update department set dnum = @ndnum where dno = @ndno

update department set dnum = @odnum where dno = @odno

1. 当员工退休时，对应部门人数减一，并且员工表的‘状态’改为退休

create trigger Retire\_Insert on retire for insert

as

declare @sno varchar(20),@dnum int,@sstatus varchar(10)

declare @dno varchar(20)

set @sstatus = '退休'

select @sno = sno from inserted

update staff set sstatus = @sstatus where sno = @sno

select @dno = sdept from staff where sno = @sno

select @dnum = dnum from department where dno = @dno

set @dnum-=1

update department set dnum = @dnum where dno = @dno

1. 当员工迟到时，自动生成相应的奖惩

create trigger Attend\_Insert on attend for insert

as

declare @time time(0),@sno varchar(20)

declare @maxano varchar(20),@max int

--获取奖惩表编号的最大编号

select @maxano = MAX(jcno) from reward

--转换为整型加一再转换为字符

set @max = CONVERT(int,@maxano)

set @max+=1

set @maxano = CONVERT(varchar(20),@max)

select @sno = sno from inserted

select @time = stime from inserted

if @time > '09:00:00'

insert into reward values(@maxano,@sno,'123456','上班迟到','罚款',50)

**3.8数据库功能的实现**

1. 考勤的增删改查

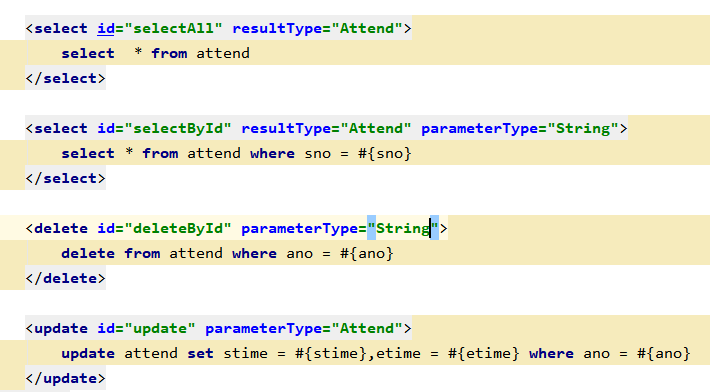


图3.2

1. 部门的增删改查

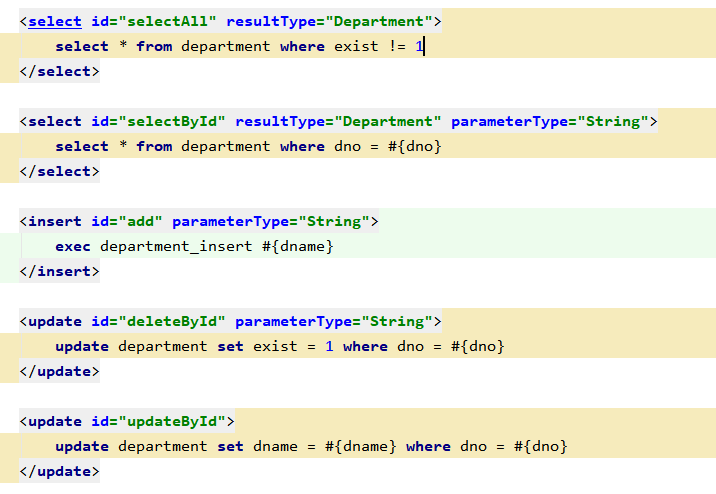


图3.3

1. 请假表的增删改查

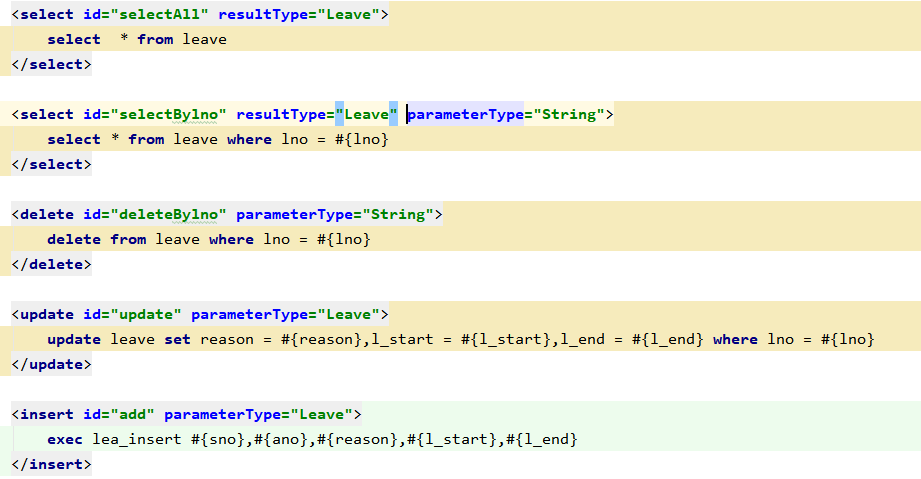


图3.4

1. 离职表的增删改查

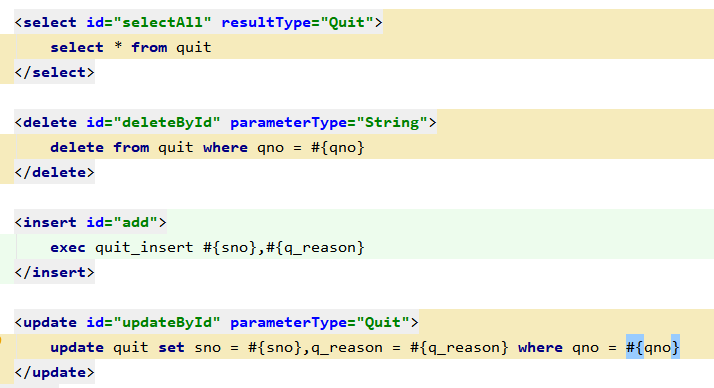


图3.5

1. 退休表的增删改查

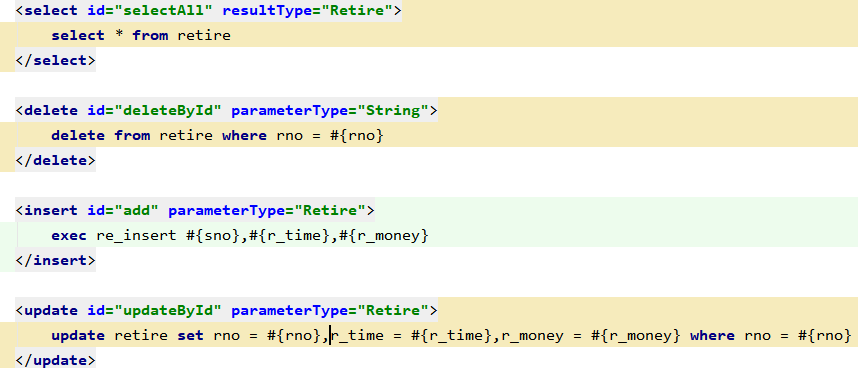


图3.6

1. 奖惩表的增删改查

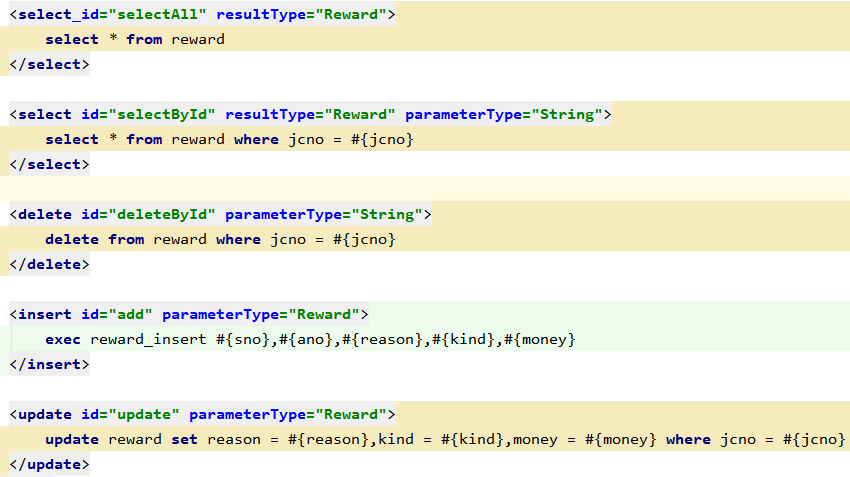


图3.7

1. 工资表的增删改查

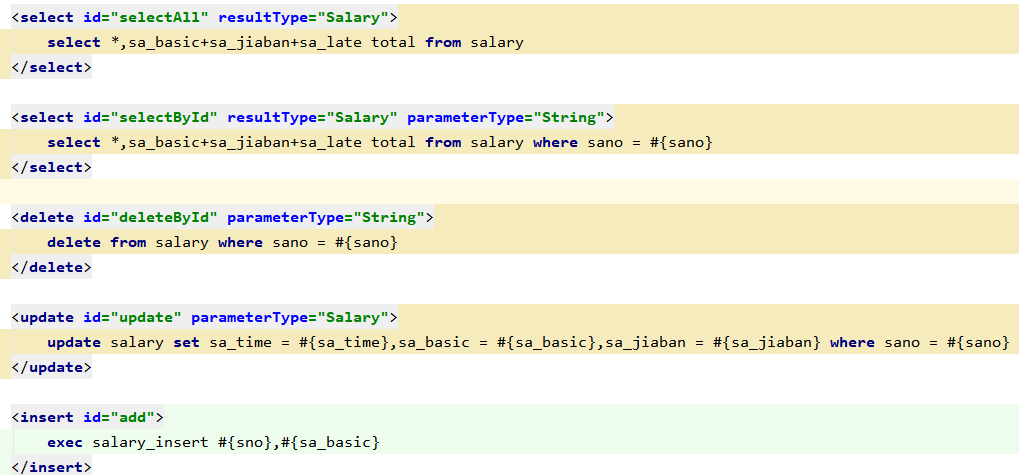


图3.8

1. 员工表的增删改查

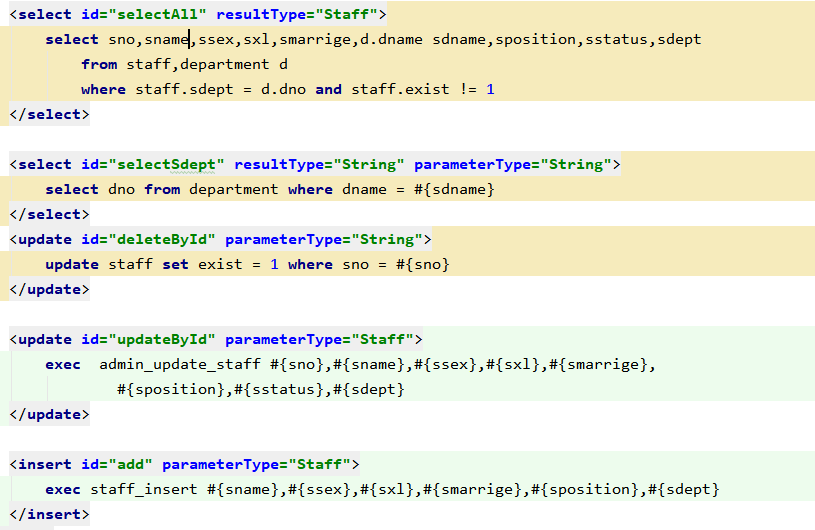


图3.9

1. 调动表的增删改查

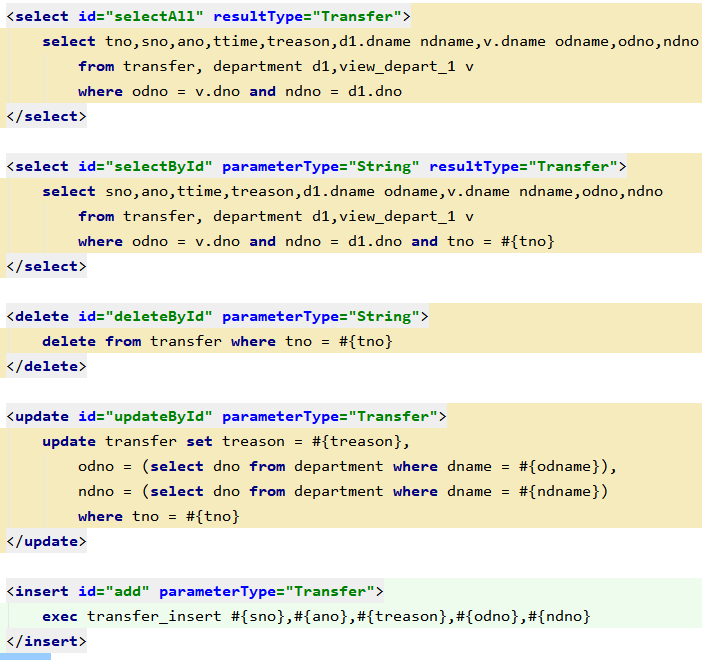


图3.10

**3.9前台的设计**



图3.11



图3.12

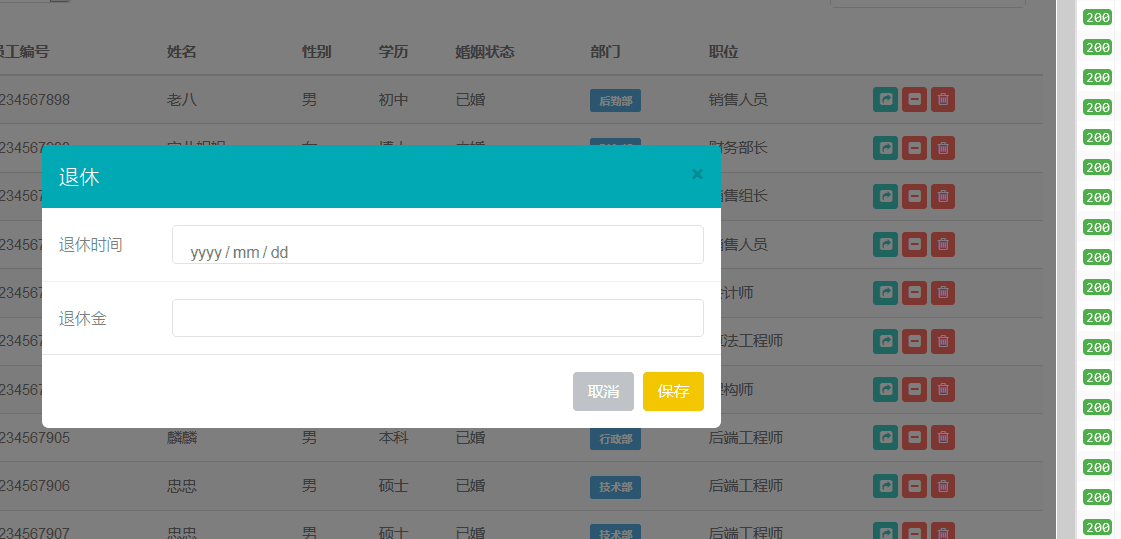


图3.13

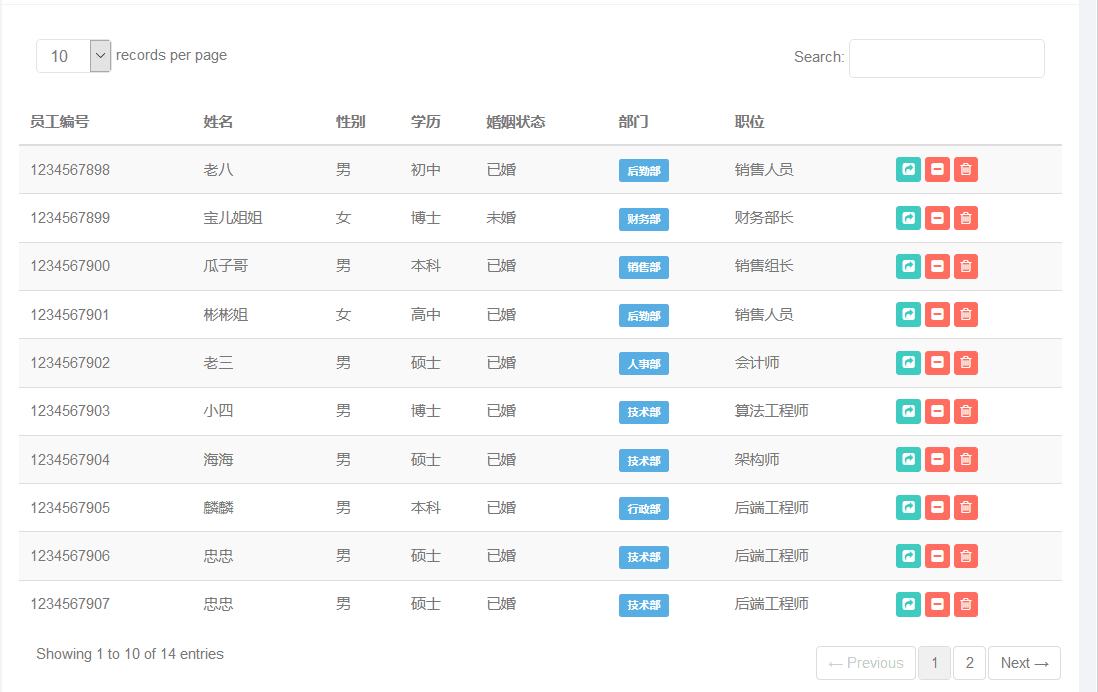


图3.14

1. **总结**

分工：

王佩鑫：人事管理系统的需求分析、人事管理系统E-R图的主要设计、数据字典建立、关系模式的转化以及优化、数据库表的设计以及审查、应用系统的前端开发以及控制端的编写。

周永其：人事管理系统的需求分析、人事管理系统E-R的设计、数据字典建立、数据表表的设计、数据库代码的编写、应用系统的数据层和服务层编写、报告的整理。

总结：

周永其：

总结一下这次专周。

主要遇见的问题：

最开始分析需求的时候，没有明确系统的目的，导致想法一直不断，改了又改。然后一开始数据流程图也不知道怎么画，导致画的就和普通的流程图一样。犯的错误最多的就是E-R图的设计了，一开始对E-R的概念没太了解，对实体联系之间的关系的理解也很模糊，导致设计了一些贼烂且有错误的E-R图。

解决：

一开始是在对需求分析很迷惑，最后就在网上浏览了别人的人事系统的功能，进而有了一些启发。流程图参照了老师给了范例，然后就照葫芦画瓢，模仿着话数据流图。E-R图主要靠书和老师给的案例，翻了很多遍书，尽量规范了一些，实体联系之间也合理一些。

收获：

通过这次专周，初步对应用系统的需求分析有了一定的了解。也巩固了一些数据库知识，在后面的数据库编程中，也更加体验到了存储过程和触发器的好处，确实很方便，并且可以减少java逻辑代码的编写。其实最大的收获就是和同学一起合作写报告写代码。

王佩鑫：

这次人事管理系统的设计，让我充分明白了应用程序设计中，数据库的设计和程序设计是紧密结合在一起的。一共两周的时间里，我们大部分的时间都是用于需求分析、流程图的编写和ER图的编写。首先我们画了流程图，然后根据流程图写了数据字典，流程图刚开始我们也画的很不清晰，经过后面重新设计了模块过后，也重新画了流程图，因此发现，一个好的设计，往往在大型系统中十分重要。流程图和ER图刚开始我们对这两个的概念都不熟悉，于是反反复复的画，ER图也出现了很多问题，包括在总的流程图里面出现不合适的关系和属性。最后经过很多次修改我们才得以完成ER图的设计。我觉得这次实验最难的地方莫过于ER图了。然后就是表结构的设计，根据我们的ER图，我们将其转化为数据库的表结构，其中还遇到了问题，就是出现在表结构中的字段也许不是由一个实体构成的而是实体和它的关系一起构成的。在ER图设计过后，我开始了前台页面的设计，根据我之前画的模块图，我搭建出来了一个差不多样子的页面，然后我的伙伴写好了业务层代码，我写了视图层的控制器，进而编写详细的页面路由，实现对各个功能的Service的调用还有数据的交互。另外，这次实验采用了ssm，大大的简化了我们的web部分的开发。总结:这是一次很有意思，很有挑战性的实验，花了我们小组四天四夜才得以完成。

1. **参考文献**

数据库系统概论 第五版 --王珊、萨师煊