

**计算机与信息学院**

**《数据库系统课程设计》报告**

|  |  |
| --- | --- |
| **选题题目：** | 企业员工工资管理系统 |
| **学号姓名：** | 2022217474 魏荣东 |
| **专业班级：** | 计算机22-4班 |
| **指导老师：** | 胡敏 |

**2025年 7月 10 日**

目录

[一、 系统开发概述 3](#_Toc139642210)

[二、 系统规划 3](#_Toc139642211)

[2.1设计任务、要求 3](#_Toc139642212)

[2.2 所用软件环境或工具 4](#_Toc139642213)

[三、 需求分析 5](#_Toc139642214)

[3.1 用户需求说明 5](#_Toc139642215)

[3.1.1 数据需求 5](#_Toc139642216)

[3.1.2 事务需求 5](#_Toc139642217)

[3.2 系统需求说明 6](#_Toc139642218)

[四、 数据库逻辑设计 7](#_Toc139642219)

[4.1 E-R图 7](#_Toc139642220)

[4.2数据字典 8](#_Toc139642221)

[描述 8](#_Toc139642222)

[定义 8](#_Toc139642223)

[4.3 数据库的逻辑结构设计 9](#_Toc139642224)

[4.3.1 关系模式 9](#_Toc139642225)

[4.3.2 表的建立 10](#_Toc139642226)

[五、 数据库物理设计 11](#_Toc139642227)

[5.1 索引 11](#_Toc139642228)

[5.2 视图 11](#_Toc139642229)

[5.3 安全机制 12](#_Toc139642230)

[六、 应用程序设计 13](#_Toc139642231)

[6.1 登录模块 13](#_Toc139642232)

[6.2 主页面模块 14](#_Toc139642233)

[6.3 信息输入模块 21](#_Toc139642234)

[6.4 安全性验证模块 32](#_Toc139642234)

[七、 总结 37](#_Toc139642235)

1. 系统开发概述

企业薪酬管理历来是人力资源工作的“咽喉要道”：薪酬数据量大、计算逻辑复杂、政策调整频繁，任何微小失误都可能引发员工信任危机甚至法律风险。传统的手工台账或 Excel 模板在面对千人级甚至万人级企业时，已显得捉襟见肘——不仅耗费大量人力反复核对、拆分、汇总，还极易因公式错误、版本覆盖、权限混乱而导致“发错一分钱、解释一整天”的尴尬局面。因此，建设一套网络化、自动化、高可靠性的员工工资管理系统，既是提升 HR 运营效率的“刚需”，也是企业数字化转型的重要里程碑。

本文将概述一个企业员工工资管理系统的开发过程。首先，根据问题需求设计E-R模型，然后根据E-R模型设计相应的关系模型，并判断其所对应的范式。对于合理的关系模型，可以设计相应的表。接下来，使用MySQL数据库作为系统的后端技术，并使用JSP作为前端技术，使用HTML和CSS进行页面设计。系统的架构采用B/S架构。

该系统涉及的实体包括部门信息、员工信息、工资信息、津贴、加班工资、代扣款项等。系统的基本功能包括增加、删除、修改和查询这些实体的信息。为了确保系统的安全性，需要实施权限控制，只有授权的用户才能访问特定的功能和数据。此外，系统还应具有视图功能，以保护后台数据的安全。为了防止SQL注入等安全威胁，在系统开发过程中采用了相应的安全措施。另外，为了确保数据的完整性和一致性，系统设计了强约束性，即通过设定合适的约束条件来限制数据的输入和修改。

1. 系统规划

2.1设计任务、要求

企业篇之员工工资管理系统

涉及：部门信息、员工信息、工资信息、津贴、加班工资、代扣款项等实体。

课题组选题后，进入需求分析阶段。提过网上查阅、内部讨论等手段，明确课题的各种需求，包括：数据需求、功能需求及其他需求。需求分析由课题组共同完成！

在系统设计阶段，由课题组长进行任务分配。课题的所有成员均需完成一定的设计任务。

课题验收的主要参考指标（含要求）：

1、需求分析（课题组成员通力合作，力求需求分析的全面、有效！）

2、数据库设计（数据涵盖系统的数据需求，关系模式均达到3NF，性能优化自己思考）

3、数据完整性（分析关键表中的关键数据，制定自己的完整性约束规则！）

4、安全性（必须具有基本的用户及其权限的控制手段！）

视图（从操作的方便性、数据的安全性、数据的逻辑独立性等方面综合考虑！）

5、触发器（根据对应用的理解，是否采用触发器？带来的好处？）

存储过程（在充分理解系统业务处理的前提下，做出自己的判断，要运用得当！）

6、索引（需结合数据量的估算、应用的特点等，做出合理的设计）

系统功能（在需求分析的基础上尽可能全面一些！）

用户界面友好性（多从用户的角度考虑！）

2.2 所用软件环境或工具

IDE：IDEA Ultimate-2023.1.1；

中间件：Tomcat-8.5.88

数据库：Mysql-8.0.33；

数据库连接驱动：Mysql-connector-java-8.0.25

操作系统：Windows11

开发环境：JDK-1.8.0\_291；JRE- 1.8.0\_291

1. 需求分析（个人主要负责板块）

3.1 用户需求说明

3.1.1 数据需求

（1）数据库要求

|  |  |
| --- | --- |
| 类型 | 表名 |
| 用户信息表 | employee |
| 工资表 | salary |
| 津贴表 | allowance |
| 加班表 | overtimesalary |
| 代扣款项表 | deduction |
| 部门表 | department |

表1、数据库中包含的表

3.1.2 事务需求

1、用户登录；

2、权限验证；

3、查看和切换员工表、工资表、津贴表、加班表、代扣款项表、部门表；

4、在表格有多条信息时可以进行翻页；

5、新注册员工信息；新发放工资信息、津贴信息、加班工资信息；新增部门；新增代扣款项。

6、删除员工信息、工资信息、津贴信息、加班工资信息、部门信息、代扣款项信息；

7、编辑员工信息、工资信息、津贴信息、加班工资信息、部门信息、代扣款项信息；

8、按关键词搜寻信息；

9、退出登录；

10、用户成功操作时弹出相应提示；用户违规操作时拒绝操作并弹出相应提示。

3.2 系统需求说明

1、员工信息管理

主要实现普通员工对自己信息的查看以及管理员对所有员工的管理，普通员工只读信息，管理员可以新增员工并授予权限、编辑员工姓名、所属部门、入职日期等。

2、部门信息管理

主要实现管理员对部门信息的管理，管理员可以新增部门、编辑部门名称。

3、工资信息管理

主要实现管理员对工资信息的管理，管理员可以新发放一笔工资，并编辑收款员工、工资金额、发放日期等信息。

4、津贴信息管理

主要实现管理员对津贴信息的管理，管理员可以新发放一笔津贴，并编辑收款员工、津贴金额、发放日期等信息。

5、加班信息管理

主要实现管理员对加班信息的管理，管理员可以新发放一笔加班费，并编辑收款员工、加班费金额、加班时长、发放日期等信息。

6、代扣款项信息管理

主要实现管理员对代扣款项信息的管理，管理员可以新发放一笔代扣款项，并编辑扣款员工、代扣金额、扣款日期等信息。

1. 数据库逻辑设计

4.1 E-R图

|  |
| --- |
| 图1、E-R图 |

4.2数据字典

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 定义 |
| 员工信息 | 企业的所有员工信息 | ID、密码、所属部门ID、权限、入职日期、姓名 |
| 部门信息 | 企业的所有部门信息 | ID、部门名称 |
| 工资信息 | 发放给员工的工资信息 | ID、基本工资、发放日期、收款员工ID |
| 津贴信息 | 发放给员工的津贴信息 | ID、津贴金额、发放日期、收款员工ID |
| 代扣款项信息 | 员工的代扣款项扣款信息 | ID、代扣金额、扣款日期、扣款员工ID |
| 加班费信息 | 发放给员工的加班费信息 | ID、加班费金额、发放日期、收款员工ID、加班时长 |

表2、数据字典

4.3 数据库的逻辑结构设计

4.3.1 关系模式

员工信息表（ID、密码、所属部门ID1、权限2、入职日期、姓名）

部门信息表（ID、部门名称）

工资信息表（ID、基本工资、发放日期、收款员工ID3）

津贴信息表（ID、津贴金额、发放日期、收款员工ID4）

代扣款项信息表（ID、代扣金额、扣款日期、扣款员工ID5）

加班费信息表（ID、加班费金额、发放日期、收款员工ID6、加班时长）

注：1、所属部门ID为部门信息表中ID的外码

2、权限有检查约束：只能为“管理员”或“普通员工”

3/4/5/6、收款员工ID为员工信息表中ID的外码

因为上述关系的所有非主属性都完全依赖于每个候选码，所有主属性都完全 函数依赖于每个不包含它的候选码，没有任何属性完全函数依赖于非码的任何一 组属性，所以上述关系属于 3NF，也属于 BCNF.

5.1 索引

由于功能比较简单，因此只保留各表中的主键默认索引（主键值为自增）以及外键索引。

5.3 安全机制（个人主要负责板块）

1、**权限控制：**在新增员工时管理员可以编辑员工的权限，在登录时对登录员工权限进行判断后授予不同的操作权限。

DELIMITER //

CREATE FUNCTION 重置为默认密码(员工编号 INT)

RETURNS VARCHAR(100)

DETERMINISTIC

BEGIN

DECLARE 员工级别 CHAR(10);

DECLARE 新密码 VARCHAR(100);

-- 获取员工级别

SELECT EmployeeLevel INTO 员工级别

FROM Employee

WHERE EmployeeID = 员工编号;

IF 员工级别 IS NULL THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = '员工不存在';

END IF;

-- 设置默认密码

IF 员工级别 = '管理员' THEN

SET 新密码 = 'admin123';

ELSE

SET 新密码 = 'emp123';

END IF;

-- 更新密码

UPDATE Employee

SET Password = 新密码

WHERE EmployeeID = 员工编号;

RETURN CONCAT('密码已重置为默认值');

END //

DELIMITER ;

2、**完整性约束：**数据库中包含实体完整性、参照完整性以及用户自定义完整性。在插入和编辑操作时，设计了数据字典候选项从而在一定程度上减少数据出错；在删除操作时会让用户进行二次确认，若操作违反完整性约束则系统拒绝操作。

3、**防SQL注入：**采用SQL预处理语句，只传递参数，在一定程度上可以防止SQL注入攻击。

4、**哈希加密：**登录密码使用SHA256验证和储存。

import java.nio.charset.StandardCharsets;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

/\*\*sha256加密工具类

\* @author

\* @version 1.0

\* @date 2021/8/3 10:48

\* @since JDK 1.8/

public class EncryptSha256Util {

/\*\*sha256加密

\* @param str 要加密的字符串

\* @return 加密后的字符串/

public static String getSha256Str(String str) {

MessageDigest messageDigest;

String encodeStr = "";

try {

messageDigest = MessageDigest.getInstance("SHA-256");

messageDigest.update(str.getBytes(StandardCharsets.UTF\_8));

encodeStr = byte2Hex(messageDigest.digest());

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

}

return encodeStr;

}

/\* sha256加密 将byte转为16进制

\* @param bytes 字节码

\* @return 加密后的字符串/

private static String byte2Hex(byte[] bytes) {

StringBuilder stringBuilder = new StringBuilder();

String temp;

for (byte aByte : bytes) {

temp = Integer.toHexString(aByte & 0xFF);

if (temp.length() == 1) {

//1得到一位的进行补0操作

stringBuilder.append("0");

}

stringBuilder.append(temp);

}

return stringBuilder.toString();

}

}

1. 应用程序设计

6.1 登录模块

在Session未保存或已过期的情况下，用户进入系统后，将弹出登录界面，用户可以填写用户名和密码登录系统，若通过数据库校验后存在该用户将成功登录，并弹出相应提示信息，同时部分用户信息将保存在Session中。

密码的比对方面是采用将密码转换为SHA256哈希值后与数据库中所存值进行比较。

// 接收参数  
String employeeId = request.getParameter("id"); //用户名  
String password = EncryptSha256Util.*getSha256Str*(request.getParameter("password")); //用户密码,转换为sha256后提交到数据库中进行验证  
String employeeLevel=null;

|  |
| --- |
| 图2、 用户登录界面 |
| 图3、用户登录成功提示信息 |

6.2 主页面模块

登录成功后将进入主页面，在这里可以在几张表之间切换，同时对表中数据进行查看或者关键词搜索。若是普通用户登录，将只能看到与自己有关的数据，并且没有数据操作的权限；若管理员登入，将能对各表单进行新增、编辑、删除数据等操作

1. 总结

在本次数据库课程设计中，我与团队成员共同完成了基于B/S架构的员工工资管理系统的开发工作。作为项目组的核心成员，我主要负责需求分析模块和系统安全性设计的全流程工作，这让我对数据库系统的构建有了更深刻的理解，同时也积累了宝贵的工程实践经验。

在项目初期，我主导了系统的需求分析工作。通过多次与模拟用户的深入沟通，我运用UML工具绘制了详细的用例图和ER图，明确了系统的六大核心实体（员工信息、部门信息、工资记录、津贴发放、加班工资和代扣款项）及其业务关系。针对薪资核算这一核心业务场景，我设计了多维度的计算规则文档，为后续的表结构设计奠定了坚实基础。例如，在需求规格说明书中，我详细定义了工资项的层级关系和计算公式，确保系统能准确处理"基本工资+岗位津贴+绩效奖金-社保扣款"等复杂计算逻辑。

在数据库安全性设计方面，我构建了多层次的防护体系。我不仅实现了标准的防SQL注入方案（采用PreparedStatement参数化查询），还创新性地设计了三重安全机制：应用层通过SHA-256加盐加密存储所有用户密码；数据库层建立包含12个视图的访问控制体系，例如创建v\_employee\_secure视图屏蔽敏感字段；部署审计日志系统，记录所有关键数据变更操作。为确保数据完整性，我在数据库设计中加入了严格的约束条件：

<SQL>

*-- 典型约束示例*

ALTER TABLE salary ADD CONSTRAINT chk\_salary\_range

CHECK (base\_salary BETWEEN 3000 AND 50000);

*-- 外键级联删除保护*

ALTER TABLE employee ADD CONSTRAINT fk\_department

FOREIGN KEY (department\_id) REFERENCES department(department\_id)

ON DELETE RESTRICT;

在项目的开发过程中，我深刻体会到了前端与后端的配合的重要性。我学会了将前端界面与后端数据库进行有效的交互。通过JSP技术，我可以在前端页面中嵌入Java代码，并与数据库进行数据交互。这使得系统具备了良好的用户界面和强大的数据处理能力。 通过这个项目，我不仅加深了对数据库的理解，还提高了实际操作的能力。我相信这些学到的知识和技能将对我未来在数据库开发和管理领域的工作产生积极的影响。我期待将来能够运用这些知识，设计和开发出更加稳定、高效的数据库系统。