

《数据库系统》实验报告

年级、专业、班级	2021 级计算机科学与技术(卓越)02 班	姓名	文红兵
实验题目	GaussDB 云数据库应用编程		
实验时间	2023.4.17-2022.5.14	实验地点	DS3401
实验成绩		实验性质	<input type="checkbox"/> 验证性 <input type="checkbox"/> 设计性 <input checked="" type="checkbox"/> 综合性
评价教师签名： <input type="checkbox"/> 算法/实验过程正确； <input type="checkbox"/> 源程序/实验内容提交； <input type="checkbox"/> 程序结构/实验步骤合理； <input type="checkbox"/> 实验结果正确； <input type="checkbox"/> 语法、语义正确； <input type="checkbox"/> 报告规范； 其他： <div>评价教师:</div>			

报告完成时间: 2023 年 5 月 8 日

目录

1 实验目的	4
2 实验项目内容	4
3 实验过程及结果	5
3.1 需求分析	5
3.1.1 概念模型	5
3.1.2 物理数据模型	6
3.1.3 系统功能模块图	6
3.2 数据库的初始化	7
3.2.1 数据库的创建	7
3.2.2 触发器	7
3.3 企业员工管理系统	8
3.3.1 主页面	8
3.3.2 员工管理模块	10
3.3.3 部门管理模块	11
3.3.4 职位管理模块	13
3.3.5 工资管理模块	14
3.3.6 项目管理模块	15
A 数据库初始化 SQL 代码	16
A.1 数据库创建代码	16
A.2 触发器代码	19
A.2.1 删除员工触发器	19
A.2.2 删除部门触发器	20
A.2.3 删除职位触发器	21
A.2.4 删除项目触发器	21

B 企业系统管理系统 JAVA 代码	22
B.1 企业员工管理系统主页面代码	22
B.2 员工管理模块代码	28
B.2.1 增加员工代码	28
B.2.2 删除员工代码	30
B.2.3 修改员工代码	30
B.2.4 查询员工代码	32
B.3 部门管理代码	34
B.3.1 增加部门代码	34
B.3.2 删除部门代码	35
B.3.3 修改部门代码	36
B.3.4 查询部门代码	36
B.4 职位管理代码	38
B.4.1 增加职位代码	38
B.4.2 删除职位代码	39
B.4.3 修改职位代码	40
B.4.4 查询职位代码	41
B.5 工资管理代码	42
B.5.1 增加工资条代码	42
B.5.2 删除工资条代码	43
B.5.3 修改工资条代码	44
B.5.4 查询工资条代码	45
B.6 项目管理代码	46
B.6.1 增加项目代码	46
B.6.2 删除项目代码	47
B.6.3 修改项目代码	48
B.6.4 查询项目代码	49

1 实验目的

(1) 掌握数据库建模工具 PowerDesigner 的使用, 利用 PowerDesigner 工具建立具体需求的概念数据模型和物理数据模型, 并转换成数据库对象。

(2) 学习通过应用程序连接 GaussDB 云数据库, 进行数据的操作。主要考查学生以数据库知识为主线, 结合“程序设计技术”、“软件工程”等多门课程知识的综合应用。

2 实验项目内容

要求设计并编程实现一个小型的企业员工管理系统。

具体要求:

(1) 对企业员工管理系统进行需求分析, 使用 PowerDesigner 设计 E-R 模型, 详细描述实体的属性和实体之间的联系, 消除不必要的冗余; 实现 E-R 图向关系模型的转换, 定义主键、外键, 同时设计出系统功能模块图。

(2) 基于 GaussDB 云数据库搭建企业员工管理系统数据库。通过 java、python 等程序设计语言编程实现企业员工管理系统, 重点是实现数据库的连接和各种 CRUD 操作; 必须有对数据的增、删、改、查功能。界面自选, 可以是命令行形式, web 页面形式或者 GUI。

3 实验过程及结果

3.1 需求分析

3.1.1 概念模型

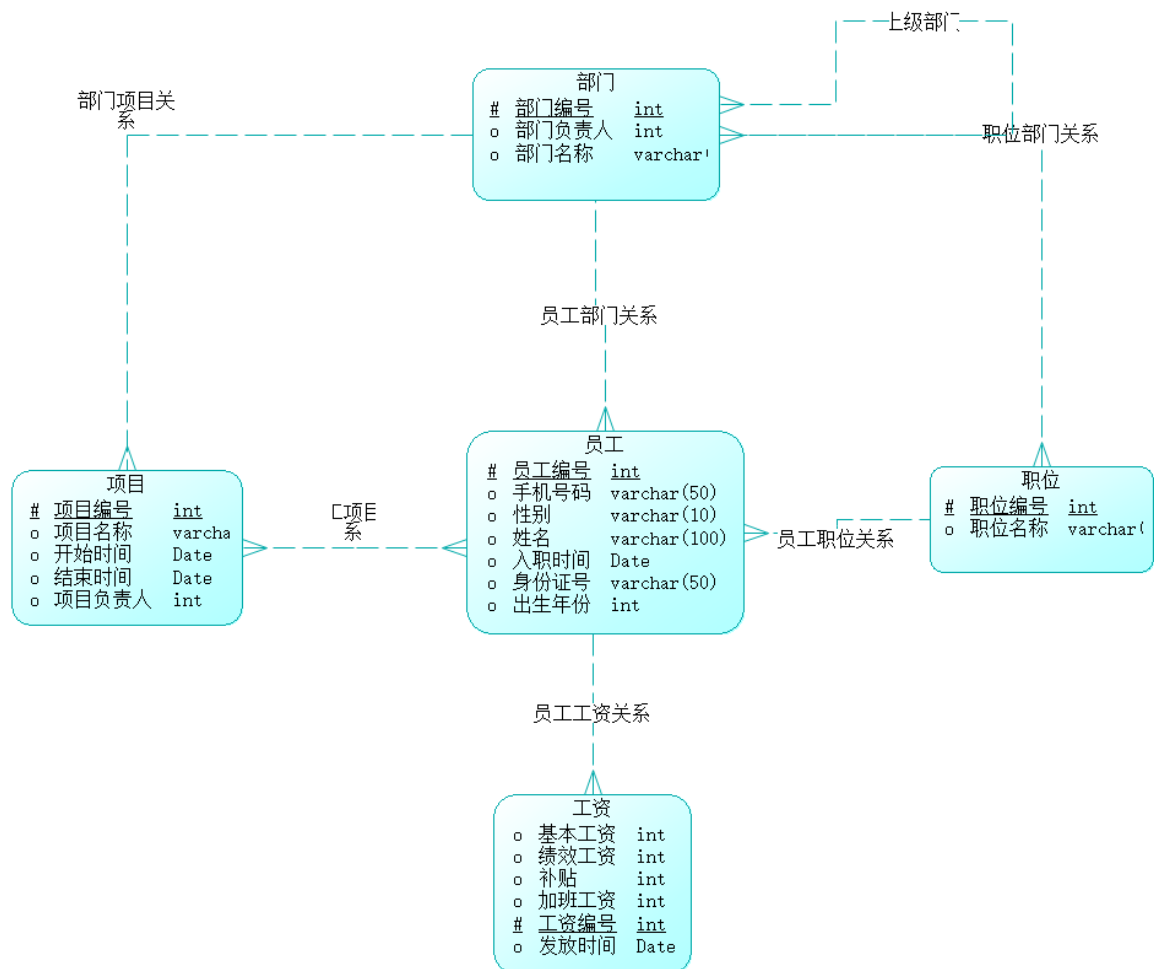


图 1: 概念模型

3.1.2 物理数据模型

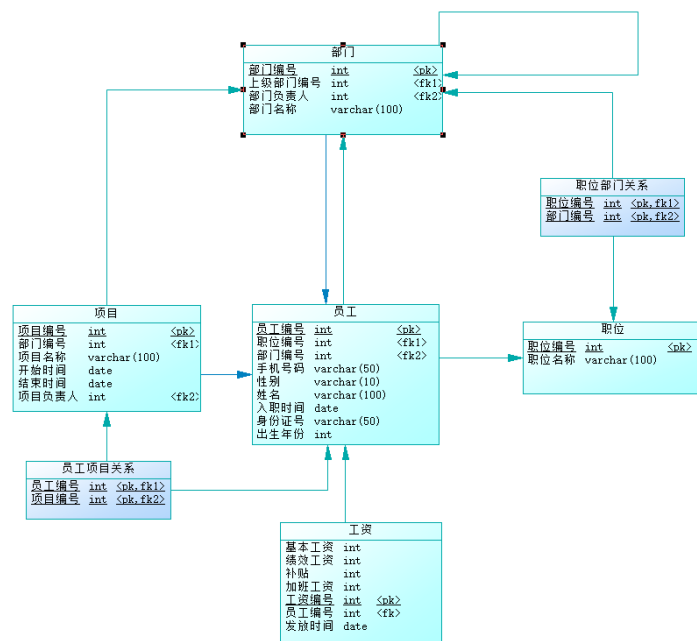


图 2: 物理模型

3.1.3 系统功能模块图

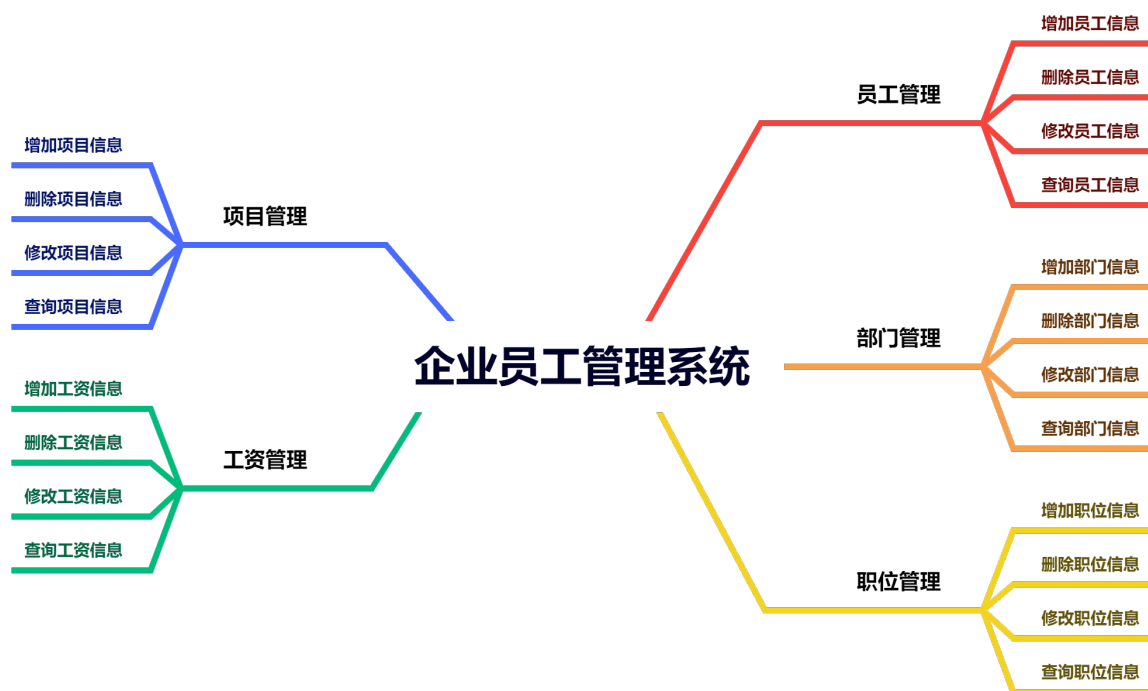


图 3: 系统功能模块图

3.2 数据库的初始化

3.2.1 数据库的创建

根据 PowerDesigner 生成的物理数据模型导出 SQL 代码创建数据库。

具体 SQL 代码位于 A.1。

该 SQL 代码用于创建一个员工管理系统的数据库,包括员工、部门、职位、项目、薪资等表格。

- 部门表格包含部门 ID、上级部门 ID、部门负责人、部门名称等信息。
- 职位表格包含职位 ID、职位名称等信息。
- 员工表格包含员工 ID、职位 ID、部门 ID、电话、性别、姓名、入职时间、身份证号、工龄等信息。
- 项目表格包含项目 ID、部门 ID、项目名称、开始时间、结束时间、项目负责人等信息。
- 薪资表格包含基本工资、绩效工资、津贴、加班工资、薪资 ID、员工 ID、发放时间等信息。

同时,还包含了外键约束,确保了表格之间的关系正确性。

3.2.2 触发器

参考: GaussDB 开发指南 (点击跳转)

删除员工触发器: 删除 employee 表的触发

- 删除 employee_project 匹配的列
- 修改 department 的 department_header 为 null
- 修改 project 的 project_header 为 null
- 删除 salary 的所匹配的项

具体 SQL 代码位于 A.2.1。

删除部门触发器: 删除 department 表的时候触发

- 修改 department 中上级部门匹配的项为 NULL
- 修改 employee 中的 department_id 为 null
- 删除 post_department 中的 department_id 匹配的项
- 修改 project 中 department_id 为 null

具体 SQL 代码位于 A.2.2。

删除职位触发器: 删除 post 表的时候触发

- 修改 employee 中的 post_id 为 NULL
- 删除 post_department 中匹配的项

具体 SQL 代码位于A.2.3。

删除项目触发器:删除 project 表的时候触发

- 删除 employee_project 中的匹配的项

具体 SQL 代码位于A.2.4。

3.3 企业员工管理系统

本项目的界面为命令行形式,完整项目请查看附件,接下来会展现代码逻辑和执行结果。

3.3.1 主页面

- 使用 JDBC 连接数据库。
- 创建五个功能模块的 Class 类,并初始化。
- 创建命令行界面和调用模块的功能。

```
*****
欢迎来到企业员工管理系统!
1、员工管理
2、部门管理
3、职位管理
4、工资管理
5、项目管理
6、退出系统
*****
请选择操作:
```

图 4: 主页面选择功能

```
*****
员工管理
1、添加员工
2、删除员工
3、修改员工信息
4、查询员工信息
5、返回上一级菜单
*****
请选择操作: |
```

图 5: 员工管理页面功能选择


```
*****
工资管理
1、添加工资记录
2、删除工资记录
3、修改工资记录
4、查询工资记录
5、返回上一级菜单
*****
请选择操作: |
```

图 6: 工资管理页面功能选择

```
*****
职位管理
1、添加职位
2、删除职位
3、修改职位信息
4、查询职位信息
5、返回上一级菜单
*****
请选择操作:
```

图 7: 职位管理页面功能选择

```
*****
部门管理
1、添加部门
2、删除部门
3、修改部门信息
4、查询部门信息
5、返回上一级菜单
*****
请选择操作: |
```

图 8: 部门管理页面功能选择

```
*****
项目管理
1、添加项目
2、删除项目
3、修改项目信息
4、查询项目信息
5、返回上一级菜单
*****
请选择操作: |
```

图 9: 项目管理页面功能选择

具体 JAVA 代码位于B.1。

接下来每个模块的增删改查实现逻辑相似,这里就只展示员工管理模块的实现逻辑,其他模块实现逻辑相似。

3.3.2 员工管理模块

增加员工

1. 根据传入的参数查询对应的职位和部门的 id。
2. 如果查询不到对应的职位和部门,返回 false。
3. 如果查询到了职位和部门的 id,就将员工信息插入到 employee 表中。
4. 如果插入成功,返回 true,否则返回 false。

```
=====:  
输入新增员工信息:  
员工编号 | 职位 | 部门 | 电话 | 性别 | 姓名 | 入职时间 | 身份证号 | 出生年  
11 保安 安保部门 15310865625 男 沈快 2015-02-15 510298744023561237 1995  
添加成功!  
=====:
```

图 10: 员工管理页面增加员工

具体 JAVA 代码位于B.2.1。

删除员工

1. 根据传入的参数拼接 SQL 语句。
2. 执行 SQL 语句删除对应的员工信息。
3. 如果删除成功,返回 true,否则返回 false。

```
=====:  
需要删除的员工编号:  
11  
删除成功!  
=====:
```

图 11: 员工管理页面删除员工

具体 JAVA 代码位于B.2.2。

修改员工信息

1. 根据传入的参数查询对应的职位和部门的 id。
2. 如果查询不到对应的职位和部门,返回 false。
3. 如果查询到了职位和部门的 id,就将员工信息更新到 employee 表中。
4. 如果修改成功,返回 true,否则返回 false。

```

=====
需要更新的员工编号:
11
依次更新后的信息:
职位 | 部门 | 电话 | 性别 | 姓名 | 入职时间 | 身份证号 | 出生年
保安 安保部门 15310865625 男 沈快 2015-02-15 510298744023561237 1993
更新成功!
=====

```

图 12: 员工管理页面修改员工

具体 JAVA 代码位于B.2.3。

查询员工信息

1. 根据传入的参数查询对应的员工信息。
2. 如果查询成功,打印结果并返回 true。
3. 如果查询失败,返回 false。

```

=====
依次输入信息(未知请写0,空格作为分隔符,回车结束输入):
员工编号 | 职业 | 部门 | 性别 | 姓名
0 0 0 0 0
=====
员工编号 | 职位 | 部门 | 电话 | 性别 | 姓名 | 入职时间 | 身份证号 | 出生年
=====
1 | 后端工程师 | 销售部门 | 15310865763 | 男 | 张三 | 2023-03-21 | 51092120050217587X | 2003
5 | 后端工程师 | 销售部门 | 15310865763 | 男 | 张三 | 2023-03-21 | 51092120050217587X | 2003
3 | 会计 | 财务部门 | 13752904509 | 女 | 李四 | 2013-05-15 | 510921200504587656 | 1972
6 | 财务 | 财务部门 | 15310865562 | 女 | 王五 | 2015-05-12 | 540981300564789412 | 2001
11 | 保安 | 安保部门 | 15310865625 | 男 | 沈快 | 2015-02-15 | 510298744023561237 | 1995
=====
查询成功!
=====

```

图 13: 员工管理页面查询员工

查询未知的信息就输入 0, 也就是说全输入 0 就是查看全部员工信息, 如果按条件查找就在对应位置输入信息。

具体 JAVA 代码位于B.2.4。

3.3.3 部门管理模块

增加部门

```

=====
输入新增部门信息:
部门编号 | 上级部门编号 | 部门负责人编号 | 部门名称
9 2 6 安卓开发部门
添加成功!
=====

```

图 14: 部门管理页面增加部门

具体 JAVA 代码位于B.3.1。

删除部门

```
=====
需要删除的部门编号:
9
删除成功!
=====
```

图 15: 部门管理页面删除部门

具体 JAVA 代码位于B.3.2。

修改部门

```
=====
需要更新的部门编号:
9
依次更新后的信息:
上级部门编号 | 部门负责人编号 | 部门名称
2 3 安卓开发部门
更新成功!
=====
```

图 16: 部门管理页面修改部门

具体 JAVA 代码位于B.3.3。

查询部门

三表联查, 将对应的 id 替换成为相应的姓名

```
=====
依次输入信息(未知请写0,空格作为分隔符,回车结束输入):
部门编号 | 上级部门编号 | 部门负责人编号 | 部门名称
0 0 0 0
-----
部门编号 | 部门名称 | 上级部门 | 部门负责人
-----
1 | 销售部门 | null | null
2 | 开发部门 | null | null
3 | 安保部门 | null | null
4 | 测试部门 | null | null
5 | 后勤部门 | null | null
6 | 财务部门 | null | null
7 | 苹果开发部门 | 开发部门 | 张三
8 | 门卫处 | 安保部门 | null
9 | 安卓开发部门 | 开发部门 | 王五
-----
查询成功!
=====
```

图 17: 部门管理页面查询部门

具体 JAVA 代码位于B.3.4。

3.3.4 职位管理模块

增加职位

```
=====
输入新增职位信息：
职位编号 | 职位名称
10 111
添加成功！
=====
```

图 18: 职位管理页面增加职位

具体 JAVA 代码位于B.4.1。

删除职位

```
=====
需要删除的职位编号：
10
删除成功！
=====
```

图 19: 职位管理页面删除职位

具体 JAVA 代码位于B.4.2。

修改职位

```
=====
需要更新的职位编号：
10
依次更新后的信息：
职位名称
111大叔
更新成功！
=====
```

图 20: 职位管理页面修改职位

具体 JAVA 代码位于B.4.3。

查询职位

```

=====
依次输入信息(未知请写0,空格作为分隔符,回车结束输入):
职位编号 | 职位名称
0 0
-----
职位编号 | 职位名称
-----
1 | 后端工程师
2 | 经理
3 | 售货员
4 | 前端工程师
5 | 测试员
6 | 清洁员
7 | 会计
8 | 财务
9 | 保安
10 | 门卫
-----
查询成功!
=====

```

图 21: 职位管理页面查询职位

具体 JAVA 代码位于B.4.4。

3.3.5 工资管理模块

增加工资条

```

=====
输入新增工资条信息:
工资条编号 | 基本工资 | 绩效工资 | 补贴 | 加班工资 | 员工编号 | 发放时间
3 3000 3000 3000 3000 1 2013-05-15
添加成功!
=====

```

图 22: 工资管理页面增加工资条

具体 JAVA 代码位于B.5.1。

删除工资条

```

=====
需要删除的工资条编号:
3
删除成功!
=====

```

图 23: 工资管理页面删除工资条

具体 JAVA 代码位于B.5.2。

修改工资条

```
=====
需要更新的工资条编号:
3
依次更新后的信息:
基本工资 | 绩效工资 | 补贴 | 加班工资 | 员工编号 | 发放时间
3300 3300 3300 3300 1 2013-05-15
更新成功!
=====
```

图 24: 工资管理页面修改工资条

具体 JAVA 代码位于B.5.3。

查询工资条

多表联查,将员工 id 替换成员工姓名。

```
=====
依次输入信息(未知请写0,空格作为分隔符,回车结束输入):
工资条编号 | 基本工资 | 绩效工资 | 补贴 | 加班工资 | 员工编号 | 发放时间
0 0 0 0 0 0 0
-----
工资条编号 | 基本工资 | 绩效工资 | 补贴 | 加班工资 | 员工姓名 | 发放时间
-----
1 | 1000 | 1000 | 1000 | 1000 | 张三 | 2013-03-14
2 | 2000 | 2000 | 2000 | 2000 | 张三 | 2013-04-15
3 | 3000 | 3000 | 3000 | 3000 | 张三 | 2013-05-15
-----
查询成功!
=====
```

图 25: 工资管理页面查询工资条

具体 JAVA 代码位于B.5.4。

3.3.6 项目管理模块

增加项目

```
=====
输入新增项目信息:
项目编号 | 所属部门编号 | 项目名称 | 开始时间名称 | 结束时间 | 项目负责人编号
4 1 王老板合作销售项目 2023-01-12 2023-03-15 3
添加成功!
=====
```

图 26: 项目管理页面增加项目

具体 JAVA 代码位于B.6.1。

删除项目

```

=====
需要删除的项目编号:
4
删除成功!
=====

```

图 27: 项目管理页面删除项目

具体 JAVA 代码位于B.6.2。

修改项目

```

=====
需要更新的项目编号:
4
依次更新后的信息:
所属部门编号 | 项目名称 | 开始时间名称 | 结束时间 | 项目负责人编号
1 文老板合作项目 2023-01-12 2023-03-15 3
更新成功!
=====

```

图 28: 项目管理页面修改项目

具体 JAVA 代码位于B.6.3。

查询项目

三表联查, 将对应的 id 替换成为相应的姓名

```

=====
依次输入信息(未知请写0,空格作为分隔符,回车结束输入):
项目编号 | 所属部门编号 | 项目名称 | 开始时间名称 | 结束时间 | 项目负责人编号
0 0 0 0 0 0

-----
项目编号 | 所属部门 | 项目名称 | 开始时间名称 | 结束时间 | 项目负责人
-----
4 | 销售部门 | 王老板合作销售项目 | 2023-01-12 | 2023-03-15 | 李四
3 | 开发部门 | 王者荣耀 | 2023-02-14 | 2023-03-14 | null
1 | 开发部门 | 微信开发 | 2021-02-12 | 2023-03-14 | null
2 | 开发部门 | QQ开发 | 2021-02-12 | 2023-03-14 | null
-----
查询成功!
=====

```

图 29: 项目管理页面查询项目

具体 JAVA 代码位于B.6.4。

A 数据库初始化 SQL 代码

A.1 数据库创建代码


```

drop table if exists department;

drop table if exists employee;

drop table if exists employee_project;

drop table if exists post;

drop table if exists post_department;

drop table if exists project;

drop table if exists salary;

/*=====*/
/* Table: department */
/*=====*/
create table department
(
    department_id      int not null ,
    dep_department_id  int ,
    department_header   int ,
    department_name     varchar(100) ,
    primary key (department_id)
);

/*=====*/
/* Table: employee */
/*=====*/
create table employee
(
    id                  int not null ,
    post_id             int ,
    department_id       int ,
    tel                 varchar(50) ,
    gender              varchar(10) ,
    name                varchar(100) ,
    entry_time          date ,
    id_card              varchar(50) ,
    year                int ,
    primary key (id)
);

/*=====*/
/* Table: employee_project */
/*=====*/
create table employee_project
(
    id                  int not null ,

```

```

        project_id          int not null ,
        primary key (id , project_id)
);

/*=====*/
/* Table: post */
/*=====*/
create table post
(
    post_id          int not null ,
    post_name        varchar(100) ,
    primary key (post_id)
);

/*=====*/
/* Table: post_department */
/*=====*/
create table post_department
(
    post_id          int not null ,
    department_id     int not null ,
    primary key (post_id , department_id)
);

/*=====*/
/* Table: project */
/*=====*/
create table project
(
    project_id        int not null ,
    department_id     int ,
    project_name       varchar(100) ,
    start_time        date ,
    end_time          date ,
    project_header     int ,
    primary key (project_id)
);

/*=====*/
/* Table: salary */
/*=====*/
create table salary
(
    basic_salary      int ,
    merit_salary      int ,
    allowance         int ,
    overtime_salary   int ,
    salary_id         int not null ,
    id               int ,

```

```

        release_time          date ,
        primary key (salary_id)
    );

alter table department add constraint FK_Reference_11 foreign key (
    department_header)
    references employee (id) on delete restrict on update restrict;

alter table department add constraint FK_department_upper foreign key (
    dep_department_id)
    references department (department_id) on delete restrict on update restrict;

alter table employee add constraint FK_employee_department foreign key (
    department_id)
    references department (department_id) on delete restrict on update restrict;

alter table employee add constraint FK_employee_post foreign key (post_id)
    references post (post_id) on delete restrict on update restrict;

alter table employee_project add constraint FK_employee_project foreign key (id)
    references employee (id) on delete restrict on update restrict;

alter table employee_project add constraint FK_employee_project2 foreign key (
    project_id)
    references project (project_id) on delete restrict on update restrict;

alter table post_department add constraint FK_post_department foreign key (post_id)
    references post (post_id) on delete restrict on update restrict;

alter table post_department add constraint FK_post_department2 foreign key (
    department_id)
    references department (department_id) on delete restrict on update restrict;

alter table project add constraint FK_Reference_10 foreign key (project_header)
    references employee (id) on delete restrict on update restrict;

alter table project add constraint FK_department_project foreign key (department_id
    )
    references department (department_id) on delete restrict on update restrict;

alter table salary add constraint FK_employee_salary foreign key (id)
    references employee (id) on delete restrict on update restrict;

```

A.2 触发器代码

A.2.1 删除员工触发器

```

# 删除employee表的时候
# 删除employee_project匹配的列
# 修改department的department_header为NULL
# 修改project的project_header为NULL
# 删除salary的id
CREATE OR REPLACE FUNCTION tri_employee_del_func() RETURNS TRIGGER AS
$$
DECLARE
BEGIN
    delete from employee_project where employee_project.id= OLD.id ;
    update department set department_header=null where department.
        department_header=OLD.id ;
    update project set project_header=null where project.project_header=OLD.id
        ;
    delete from salary where salary.id = OLD.id ;
    RETURN OLD;
END
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER tri_employee_del
BEFORE DELETE ON employee
FOR EACH ROW
EXECUTE PROCEDURE tri_employee_del_func();

```

A.2.2 删除部门触发器

```

# 删除department表的时候触发
# 修改department中上级部门匹配的项为NULL
# 修改employee中的department_id为NULL
# 删除post_department中的department_id匹配的项
# 修改project中department_id为NULL
CREATE OR REPLACE FUNCTION tri_department_del_func() RETURNS TRIGGER AS
$$
DECLARE
BEGIN
    update department set dep_department_id = null where department.
        dep_department_id = OLD.department_id ;
    update employee set department_id = null where employee.department_id = OLD.
        department_id ;
    delete FROM post_department WHERE post_department.department_id = OLD.
        department_id ;
    update project set department_id=null where project.department_id = OLD.
        department_id ;
    RETURN OLD;
END
$$ LANGUAGE PLPGSQL;

```

```
CREATE TRIGGER tri_department_del
BEFORE DELETE ON department
FOR EACH ROW
EXECUTE PROCEDURE tri_department_del_func();
```

A.2.3 删除职位触发器

```
# 删除 post 表的时候触发
# 修改 employee 中的 post_id 为 NULL
# 删除 post_department 中匹配的项
CREATE OR REPLACE FUNCTION tri_post_del_func() RETURNS TRIGGER AS
$$
DECLARE
BEGIN
    update employee set post_id = null where employee.post_id=OLD.post_id ;
    delete FROM post_department WHERE post_department.post_id=OLD.post_id ;
    RETURN OLD;
END
$$ LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER tri_post_del
BEFORE DELETE ON post
FOR EACH ROW
EXECUTE PROCEDURE tri_post_del_func();
```

A.2.4 删除项目触发器

```
# 删除 project 表的时候触发
# 删除 employee_project 中的匹配的项
CREATE OR REPLACE FUNCTION tri_project_del_func() RETURNS TRIGGER AS
$$
DECLARE
BEGIN
    delete from employee_project where employee_project.project_id = OLD.
        project_id ;
    RETURN OLD;
END
$$ LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER tri_project_del
BEFORE DELETE ON project
FOR EACH ROW
EXECUTE PROCEDURE tri_project_del_func();
```

B 企业系统管理系统 JAVA 代码

B.1 企业员工管理系统主页面代码

```
package FunctionDemo;

import ClassDemo.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Scanner;

public class EmployeeManagementSystem {
    static final String JDBC_DRIVER = "com.huawei.opengauss.jdbc.Driver";
    static final String DB_URL = "jdbc:opengauss://110.41.126.115:8000/CQU_DB2021?
        ApplicationName=app1";
    // 数据库的用户名与密码
    static final String USER = "db_user2021_24";
    static final String PASSWORD = "db_user@123";
    static Connection conn ;
    static ProjectManage projectmanage ;
    static DepartmentManage departmentmanage ;
    static EmployManage employeemanage ;
    static PostManage postmanage ;
    static SalaryManage salarymanage ;
    static{
        try {
            Class.forName(JDBC_DRIVER) ;
            conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        projectmanage = new ProjectManage(conn);
        departmentmanage = new DepartmentManage(conn) ;
        employeemanage = new EmployManage(conn) ;
        postmanage = new PostManage(conn) ;
        salarymanage = new SalaryManage(conn) ;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("
                *****
            ");
            System.out.println("欢迎来到企业员工管理系统！");
```

```

System.out.println("1、员工管理");
System.out.println("2、部门管理");
System.out.println("3、职位管理");
System.out.println("4、工资管理");
System.out.println("5、项目管理");
System.out.println("6、退出系统");
System.out.println("
*****
");
System.out.print("请选择操作: ");
choice = scanner.nextInt();
System.out.println("");

switch (choice) {
    case 1:
        try {
            employeeManagement();
        } catch (SQLException throwables) {
            System.out.println("操作错误!");
        }
        break;
    case 2:
        try {
            departmentManagement();
        } catch (SQLException throwables) {
            System.out.println("操作错误!");
        }
        break;
    case 3:
        try {
            positionManagement();
        } catch (SQLException throwables) {
            System.out.println("操作错误!");
        }
        break;
    case 4:
        try {
            salaryManagement();
        } catch (SQLException throwables) {
            System.out.println("操作错误!");
        }
        break;
    case 5:
        try {
            projectManagement();
        } catch (SQLException throwables) {
            System.out.println("操作错误!");
        }
        break;
}

```

```

        case 6:
            System.out.println("感谢使用企业员工管理系统，再见！");
            break;
        default:
            System.out.println("请输入正确的操作编号！");
            break;
    }
} while (choice != 6);
}

private static void employeeManagement() throws SQLException{
    Scanner scanner = new Scanner(System.in);
    int choice;
    do {
        System.out.println("");
        System.out.println("
*****
");
        System.out.println("员工管理");
        System.out.println("1、添加员工");
        System.out.println("2、删除员工");
        System.out.println("3、修改员工信息");
        System.out.println("4、查询员工信息");
        System.out.println("5、返回上一级菜单");
        System.out.println("
*****
");
        System.out.print("请选择操作：");
        choice = scanner.nextInt();
        System.out.println("");

        switch (choice) {
            case 1:
                employeemanage.employee_add();
                break;
            case 2:
                employeemanage.employee_del();
                break;
            case 3:
                employeemanage.employee_update();
                break;
            case 4:
                employeemanage.employee_query();
                break;
            case 5:
                return;
            default:
                System.out.println("请输入正确的操作编号！");
                break;
        }
    } while (choice != 5);
}

```



```

    }
} while (true);
}

private static void departmentManagement() throws SQLException{
    Scanner scanner = new Scanner(System.in);
    int choice;
    do {
        System.out.println("");
        System.out.println("
*****
");
        System.out.println("部门管理");
        System.out.println("1、添加部门");
        System.out.println("2、删除部门");
        System.out.println("3、修改部门信息");
        System.out.println("4、查询部门信息");
        System.out.println("5、返回上一级菜单");
        System.out.println("
*****
");
        System.out.print("请选择操作: ");
        choice = scanner.nextInt();
        System.out.println("");

        switch (choice) {
            case 1:
                departmentmanage.department_add();
                break;
            case 2:
                departmentmanage.department_del();
                break;
            case 3:
                departmentmanage.department_update();
                break;
            case 4:
                departmentmanage.department_query();
                break;
            case 5:
                return;
            default:
                System.out.println("请输入正确的操作编号!");
                break;
        }
    } while (true);
}

private static void positionManagement() throws SQLException{
    Scanner scanner = new Scanner(System.in);

```

```

int choice;

do {
    System.out.println("");
    System.out.println("
*****
");
    System.out.println("职位管理");
    System.out.println("1、添加职位");
    System.out.println("2、删除职位");
    System.out.println("3、修改职位信息");
    System.out.println("4、查询职位信息");
    System.out.println("5、返回上一级菜单");
    System.out.println("
*****
");
    System.out.print("请选择操作: ");
    choice = scanner.nextInt();
    System.out.println("");

    switch (choice) {
        case 1:
            postmanage.post_add();
            break;
        case 2:
            postmanage.post_del();
            break;
        case 3:
            postmanage.post_update();
            break;
        case 4:
            postmanage.post_query();
            break;
        case 5:
            return;
        default:
            System.out.println("请输入正确的操作编号!");
            break;
    }
} while (true);
}

```

```

private static void salaryManagement() throws SQLException{
    Scanner scanner = new Scanner(System.in);
    int choice;

    do {
        System.out.println("");

```

```

        System.out.println("
        *****
        ");
        System.out.println("工资管理");
        System.out.println("1、添加工资记录");
        System.out.println("2、删除工资记录");
        System.out.println("3、修改工资记录");
        System.out.println("4、查询工资记录");
        System.out.println("5、返回上一级菜单");
        System.out.println("
        *****
        ");
        System.out.print("请选择操作: ");
        choice = scanner.nextInt();
        System.out.println("");

        switch (choice) {
            case 1:
                salarymanage.salary_add();
                break;
            case 2:
                salarymanage.salary_del();
                break;
            case 3:
                salarymanage.salary_update();
                break;
            case 4:
                salarymanage.salary_query();
                break;
            case 5:
                return;
            default:
                System.out.println("请输入正确的操作编号!");
                break;
        }
    } while (true);
}

private static void projectManagement() throws SQLException{
    Scanner scanner = new Scanner(System.in);
    int choice;
    do {
        System.out.println("");
        System.out.println("
        *****
        ");
        System.out.println("项目管理");
        System.out.println("1、添加项目");
        System.out.println("2、删除项目");
    }
}

```

```

System.out.println("3、修改项目信息");
System.out.println("4、查询项目信息");
System.out.println("5、返回上一级菜单");
System.out.println("
    *****
");
System.out.print("请选择操作: ");
choice = scanner.nextInt();
System.out.println("");

switch (choice) {
    case 1:
        projectmanage.project_add();
        break;
    case 2:
        projectmanage.project_del();
        break;
    case 3:
        projectmanage.project_update();
        break;
    case 4:
        projectmanage.project_query();
        break;
    case 5:
        return;
    default:
        System.out.println("请输入正确的操作编号!");
        break;
}
} while (true);
}
}

```

B.2 员工管理模块代码

B.2.1 增加员工代码

```

private boolean employee_add_helper(    String id, // 新增员工
                                       String post,
                                       String department,
                                       String tel,
                                       String gender,
                                       String name,
                                       String entry_time,
                                       String id_card,
                                       String year) throws SQLException{

    // 查询post的id

```

```

Statement sta = conn.createStatement() ;
String sql = String.format("SELECT post_id FROM post WHERE post_name='%s'",
    post) ;
ResultSet rs = sta.executeQuery(sql) ;
int post_id ;
if(rs.next()) post_id = rs.getInt("post_id") ;
else return false ;

// 查询department的id
sql = String.format("SELECT department_id FROM department WHERE department_name
    ='%s'", department) ;
rs = sta.executeQuery(sql) ;
int department_id ;
if(rs.next()) department_id = rs.getInt("department_id") ;
else return false ;

// 插入员工
sql = String.format("INSERT INTO employee values(%s,%d,%d,'%s','%s','%s','%s'
    ,'%s',%s) ;",
    id, post_id, department_id, tel, gender, name, entry_time, id_card, year ) ;
sta.executeUpdate(sql) ;
return true ;
}

public void employee_add() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

    ");
    System.out.println("输入新增员工信息:\n员工编号 | 职位 | 部门 | 电话 | 性别 |
        姓名 | 入职时间 | 身份证号 | 出生年");
    String s = sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split("\\s") ;
    if(info.length != 9) {
        System.out.println("添加失败!");
        System.out.println("

        ");
        return ;
    }
    if(employee_add_helper(info[0], info[1], info[2], info[3], info[4], info[5],
        info[6], info[7], info[8])) {
        System.out.println("添加成功!");
    } else {
        System.out.println("添加失败!");
    }
    System.out.println("

    ");

```

```
}
```

B.2.2 删除员工代码

```
private boolean employee_del_helper(String id) throws SQLException {
    Statement sta = conn.createStatement(); String sql ;
    sql = String.format("DELETE FROM employee WHERE employee.id = %s", id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void employee_del() throws SQLException { // 删除员工
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
=====
    ");
    System.out.println("需要删除的员工编号：");
    String id = sc.nextLine() ;
    if(employee_del_helper(id)){
        System.out.println("删除成功!");
    } else {
        System.out.println("删除失败!");
    }
    System.out.println("
=====
    ");
}
```

B.2.3 修改员工代码

```
public boolean employee_update_helper(String id, // 更新员工信息
    String post,
    String department,
    String tel,
    String gender,
    String name,
    String entry_time,
    String id_card,
    String year) throws SQLException {
    Statement sta ; String sql ; ResultSet rs ;
    sta = conn.createStatement() ; int post_id = 0, department_id = 0 ;
    // 查询post的id
    if(!post.equals("")) {
        sql = String.format("SELECT post_id FROM post WHERE post_name='%s'", post)
        ;
        rs = sta.executeQuery(sql) ;
    }
```

```

        if(rs.next()) post_id = rs.getInt("post_id") ;
        else return false ;
    }
    // 查询department的id
    if(!department.equals("")) {
        sql = String.format("SELECT department_id FROM department WHERE
            department_name='%s'", department) ;
        rs = sta.executeQuery(sql) ;
        if(rs.next()) department_id = rs.getInt("department_id") ;
        else return false ;
    }

    sql = String.format("update employee set post_id=%d,department_id=%d,tel='%s',
        gender='%s',name='%s',entry_time='%s',id_card='%s',year=%s WHERE id=%s",
        post_id, department_id, tel, gender, name, entry_time, id_card, year,
        id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void employee_update() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

");
    System.out.println("需要更新的员工编号：");
    String s = sc.nextLine() + " ";
    System.out.println("依次更新后的信息:\n职位 | 部门 | 电话 | 性别 | 姓名 | 入职
        时间 | 身份证号 | 出生年");
    s += sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split("\\s") ;
    if(info.length != 9) {
        System.out.println("更新失败!");
        System.out.println("

");
        return ;
    }
    if(employee_update_helper(info[0], info[1], info[2], info[3], info[4], info[5],
        info[6], info[7], info[8])){
        System.out.println("更新成功!");
    } else {
        System.out.println("更新失败!");
    }
    System.out.println("

");
}

```

B.2.4 查询员工代码

```
private boolean employee_query_helper( String id,    // 查询员工信息
                                       String post,
                                       String department,
                                       String tel,
                                       String gender,
                                       String name,
                                       String entry_time,
                                       String id_card,
                                       String year) throws SQLException {
    Statement sta ; String sql ; ResultSet rs ;
    sta = conn.createStatement() ;
    // 查询post的id
    if(!post.equals("")) {
        sql = String.format("SELECT post_id FROM post WHERE post_name='%s'", post)
        ;
        rs = sta.executeQuery(sql) ;
        if(rs.next()) post = rs.getString("post_id") ;
        else return false ;
    }
    // 查询department的id
    if(!department.equals("")) {
        sql = String.format("SELECT department_id FROM department WHERE
        department_name='%s'", department) ;
        rs = sta.executeQuery(sql) ;
        if(rs.next()) department = rs.getString("department_id") ;
        else return false ;
    }
    // 拼接查询条件
    String con = "WHERE employee.post_id = post.post_id AND employee.department_id
    = department.department_id " ;
    if(!id.equals("")) {
        con += String.format("AND employee.id=%s ", id) ;
    }
    if(!post.equals("")) {
        con += String.format("AND employee.post_id='%s' ", post) ;
    }
    if(!department.equals("")) {
        con += String.format("AND employee.department_id='%s' ", department) ;
    }
    if(!tel.equals("")) {
        con += String.format("AND employee.tel='%s' ", tel) ;
    }
    if(!gender.equals("")) {
        con += String.format("AND employee.gender='%s' ", gender) ;
    }
    if(!name.equals("")) {
        con += String.format("AND employee.name='%s' ", name) ;
    }
}
```



```

    }
    if(!entry_time.equals("")) {
        con += String.format("AND employee.entry_time='%s' ", entry_time) ;
    }
    if(!id_card.equals("")) {
        con += String.format("AND employee.id_card='%s' ", id_card) ;
    }
    if(!year.equals("")) {
        con += String.format("AND employee.year=%s ", year) ;
    }
    // 查询, 执行sql语句
    sql = "SELECT id,post_name, department_name,tel ,gender ,name,entry_time ,id_card ,
        year FROM employee, post ,department " + con ;
    rs = sta.executeQuery(sql) ;
    // 打印结果
    System.out.println("
        -----
        ");
    System.out.println("员工编号 | 职位 | 部门 | 电话 | 性别 | 姓名 | 入职时间 | 身
        份证号 | 出生年");
    System.out.println("
        -----
        ");
    while(rs.next()) {
        System.out.printf("    %s    | %s | %s | %s | %s | %s | %s | %s | %s\n",
            rs.getString("id"), rs.getString("post_name"), rs.getString("
                department_name"), rs.getString("tel"),
            rs.getString("gender"), rs.getString("name"), rs.getString("
                entry_time"), rs.getString("id_card"),
            rs.getString("year"));
    }
    System.out.println("
        -----
        ");
    return true ;
}

public void employee_query()throws SQLException{
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
        =====
        ");
    System.out.println("依次输入信息(未知请写0,空格作为分隔符,回车结束输入):\n员工
        编号 | 职业 | 部门 | 性别 | 姓名 ");
    String s = sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split(" ") ;
    if(info.length != 5) {
        System.out.println("查询失败!");
    }

```

```

        System.out.println("
    ");
    return ;
}
for (int i = 0 ; i < info.length ; ++ i) {
    if(info[i].equals("0")) info[i] = "*" ;
};
if(employee_query_helper(info[0], info[1], info[2], "*", info[3], info[4], "*",
    "*", "*")) {
    System.out.println("查询成功！") ;
} else {
    System.out.println("查询失败！") ;
}
System.out.println("
");
}

```

B.3 部门管理代码

B.3.1 增加部门代码

```

private boolean department_add_helper( String department_id, // 新增部门
    String dep_department_id,
    String department_header_id,
    String department_name) throws SQLException
{
    Statement sta = conn.createStatement() ;
    String sql ;

    // 插入员工
    sql = String.format("INSERT INTO department values(%s,%s,%s,'%s') ;",
        department_id, dep_department_id, department_header_id, department_name
    ) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void department_add() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
");
    System.out.println("输入新增部门信息:\n部门编号 | 上级部门编号 | 部门负责人编号
    | 部门名称") ;
}

```

```

String s = sc.nextLine() ;
// 处理输入参数
String[] info = s.split("\\s") ;
if(info.length != 4) {
    System.out.println("添加失败！");
    System.out.println("

");
    return ;
}
if(department_add_helper(info[0], info[1], info[2], info[3])) {
    System.out.println("添加成功！");
} else {
    System.out.println("添加失败！");
}
System.out.println("

");
}

```

B.3.2 删除部门代码

```

private boolean department_del_helper(String id) throws SQLException {
    Statement sta = conn.createStatement(); String sql ;
    sql = String.format("DELETE FROM department WHERE department_id = %s", id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void department_del() throws SQLException { // 删除部门
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

");
    System.out.println("需要删除的部门编号：");
    String id = sc.nextLine() ;
    if(department_del_helper(id)){
        System.out.println("删除成功！");
    } else {
        System.out.println("删除失败！");
    }
    System.out.println("

");
}

```

B.3.3 修改部门代码

```
private boolean department_update_helper(    String department_id, // 更新部门
                                           String dep_department_id,
                                           String department_header_id,
                                           String department_name) throws
                                           SQLException {

    Statement sta ; String sql ;
    sta = conn.createStatement() ;

    sql = String.format("update department set dep_department_id=%s,
        department_header=%s, department_name='%s' WHERE department_id=%s;",
        dep_department_id, department_header_id, department_name, department_id
        ) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void department_update() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

=====

");
    System.out.println("需要更新的部门编号：");
    String s = sc.nextLine() + " ";
    System.out.println("依次更新后的信息:\n上级部门编号 | 部门负责人编号 | 部门名称
    ");
    s += sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split("\\s") ;
    if(info.length != 4) {
        System.out.println("更新失败!");
        System.out.println("

=====

");
        return ;
    }
    if(department_update_helper(info[0], info[1], info[2], info[3])){
        System.out.println("更新成功!");
    } else {
        System.out.println("更新失败!");
    }
    System.out.println("

=====

");
}
```

B.3.4 查询部门代码

```

private boolean department_query_helper( String department_id, // 查询部门
                                         String dep_department_id,
                                         String department_header_id,
                                         String department_name) throws
                                         SQLException {

    Statement sta ; String sql ; ResultSet rs ;
    sta = conn.createStatement() ;
    // 拼接查询条件
    String con = "WHERE " ;
    if(!department_id.equals("")) {
        con += String.format("AND department.department_id=%s ", department_id) ;
    }
    if(!dep_department_id.equals("")) {
        con += String.format("AND department.dep_department_id=%s ",
                               dep_department_id) ;
    }
    if(!department_header_id.equals("")) {
        con += String.format("AND department.department_header=%s ",
                               department_header_id) ;
    }
    if(!department_name.equals("")) {
        con += String.format("AND department.department_name='%s' ",
                               department_name) ;
    }
    // 查询, 执行sql语句, 三表联查, 将对应的id替换成为相应的姓名
    sql = "SELECT d.department_id, d.department_name, d2.department_name AS
           dep_department_name, e.name AS department_header_name\n" +
           "FROM department d\n" +
           "LEFT JOIN department d2 ON d.dep_department_id = d2.department_id\n" +
           "LEFT JOIN employee e ON d.department_header = e.id\n" ;
    if(!con.equals("WHERE ")) sql += con ;
    rs = sta.executeQuery(sql) ;
    // 打印结果
    System.out.println("
    -----
    ");
    System.out.println("部门编号 | 部门名称 | 上级部门 | 部门负责人");
    System.out.println("
    -----
    ");
    while(rs.next()) {
        System.out.printf(" %s | %s | %s | %s\n",
                           rs.getString("department_id"), rs.getString("department_name"),
                           rs.getString("dep_department_name"), rs.getString("
                           department_header_name"));
    }
    System.out.println("
    -----
    ");

```

```

        return true ;
    }
    public void department_query() throws SQLException{
        Scanner sc = new Scanner(System.in) ;
        System.out.println("
=====
");
        System.out.println("依次输入信息(未知请写0,空格作为分隔符,回车结束输入):\n部门
        编号 | 上级部门编号 | 部门负责人编号 | 部门名称");
        String s = sc.nextLine() ;
        // 处理输入参数
        String[] info = s.split(" ") ;
        if(info.length != 4) {
            System.out.println("查询失败! ");
            System.out.println("
=====
");
            return ;
        }
        for (int i = 0 ; i < info.length ; ++ i) {
            if(info[i].equals("0")) info[i] = "*" ;
        };
        if(department_query_helper(info[0], info[1], info[2], info[3])){
            System.out.println("查询成功! ") ;
        } else {
            System.out.println("查询失败! ") ;
        }
        System.out.println("
=====
");
    }
}

```

B.4 职位管理代码

B.4.1 增加职位代码

```

private boolean post_add_helper( String post_id, // 新增职位
                                String post_name) throws SQLException{

    Statement sta = conn.createStatement() ;
    String sql ;

    // 插入员工
    sql = String.format("INSERT INTO post values(%s,'%s') ;",
        post_id, post_name) ;
    sta.executeUpdate(sql) ;
    return true ;
}

```

```

}
public void post_add() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
=====
");
    System.out.println("输入新增职位信息:\n职位编号 | 职位名称") ;
    String s = sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split("\\s") ;
    if(info.length != 2) {
        System.out.println("添加失败! ");
        System.out.println("
=====
");
        return ;
    }
    if(post_add_helper(info[0], info[1])) {
        System.out.println("添加成功! ");
    } else {
        System.out.println("添加失败! ");
    }
    System.out.println("
=====
");
}

```

B.4.2 删除职位代码

```

private boolean post_del_helper(String id) throws SQLException {
    Statement sta = conn.createStatement(); String sql ;
    sql = String.format("DELETE FROM post WHERE post_id = %s", id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void post_del() throws SQLException { // 删除职位
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
=====
");
    System.out.println("需要删除的职位编号: ");
    String id = sc.nextLine() ;
    if(post_del_helper(id)){
        System.out.println("删除成功!");
    } else {
        System.out.println("删除失败!");
    }
}

```

```

    }
    System.out.println("

```

```

        ");
}

```

B.4.3 修改职位代码

```

private boolean post_update_helper( String post_id, // 更新部门
                                   String post_name) throws SQLException {

    Statement sta ; String sql ;
    sta = conn.createStatement() ;

    sql = String.format("update post set post_name='%s' WHERE post_id=%s;",
                        post_name, post_id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void post_update() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

```

```

        ");
    System.out.println("需要更新的职位编号: ");
    String s = sc.nextLine() + " ";
    System.out.println("依次更新后的信息:\n职位名称") ;
    s += sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split("\\s") ;
    if(info.length != 2) {
        System.out.println("更新失败!");
        System.out.println("

```

```

            ");
        return ;
    }
    if(post_update_helper(info[0], info[1])){
        System.out.println("更新成功!");
    } else {
        System.out.println("更新失败!");
    }
    System.out.println("

```

```

        ");
}

```


B.4.4 查询职位代码

```
private boolean post_query_helper(String post_id, // 查询部门
                                   String post_name) throws SQLException {
    Statement sta ; String sql ; ResultSet rs ;
    sta = conn.createStatement() ;
    // 拼接查询条件
    String con = "WHERE " ;
    if (!post_id.equals("*")) {
        con += String.format("AND post.post_id=%s ", post_id) ;
    }
    if (!post_name.equals("*")) {
        con += String.format("AND post.post_name='%s' ", post_name) ;
    }
    // 查询, 执行sql语句
    sql = "SELECT post_id,post_name FROM post " ;
    if (!con.equals("WHERE ")) sql += con ;
    rs = sta.executeQuery(sql) ;
    // 打印结果
    System.out.println("
-----
");
    System.out.println("职位编号 | 职位名称") ;
    System.out.println("
-----
");
    while(rs.next()) {
        System.out.printf("    %s    | %s \n",
                           rs.getString("post_id"), rs.getString("post_name"));
    }
    System.out.println("
-----
");
    return true ;
}

public void post_query() throws SQLException{
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
=====
");
    System.out.println("依次输入信息(未知请写0,空格作为分隔符,回车结束输入):\n职位
    编号 | 职位名称") ;
    String s = sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split(" ") ;
    if(info.length != 2) {
        System.out.println("查询失败!") ;
        System.out.println("
=====
") ;
    }
}
```

```

        ");
        return ;
    }
    for (int i = 0 ; i < info.length ; ++ i) {
        if(info[i].equals("0")) info[i] = "*" ;
    };
    if(post_query_helper(info[0], info[1])){
        System.out.println("查询成功！") ;
    } else {
        System.out.println("查询失败！") ;
    }
    System.out.println("
");
}

```

B.5 工资管理代码

B.5.1 增加工资条代码

```

private boolean salary_add_helper( String salary_id, // 新增工资条
                                   String basic_salary,
                                   String merit_salary,
                                   String allowance,
                                   String overtime_salary,
                                   String id,
                                   String release_time) throws SQLException{

    Statement sta = conn.createStatement() ;
    String sql ;

    // 插入工资条
    sql = String.format("INSERT INTO salary values(%s,%s,%s,%s,%s,%s,'%s') ;",
        basic_salary, merit_salary, allowance, overtime_salary, salary_id, id,
        release_time) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void salary_add() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
");
    System.out.println("输入新增工资条信息:\n工资条编号 | 基本工资 | 绩效工资 | 补
    贴 | 加班工资 | 员工编号 | 发放时间") ;
    String s = sc.nextLine() ;
}

```

```

// 处理输入参数
String[] info = s.split("\\s") ;
if(info.length != 7) {
    System.out.println("添加失败！");
    System.out.println("
=====
");
    return ;
}
if(salary_add_helper(info[0], info[1], info[2], info[3], info[4], info[5], info
[6])) {
    System.out.println("添加成功！");
} else {
    System.out.println("添加失败！");
}
System.out.println("
=====
");
}

```

B.5.2 删除工资条代码

```

private boolean salary_del_helper(String id) throws SQLException {
    Statement sta = conn.createStatement(); String sql ;
    sql = String.format("DELETE FROM salary WHERE salary_id = %s", id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void salary_del() throws SQLException { // 删除工资信息
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
=====
");
    System.out.println("需要删除的工资条编号：");
    String id = sc.nextLine() ;
    if(salary_del_helper(id)){
        System.out.println("删除成功！");
    } else {
        System.out.println("删除失败！");
    }
    System.out.println("
=====
");
}

```

B.5.3 修改工资条代码

```
private boolean salary_update_helper( String salary_id, // 更新工资条
                                     String basic_salary,
                                     String merit_salary,
                                     String allowance,
                                     String overtime_salary,
                                     String id,
                                     String release_time) throws SQLException {

    Statement sta ; String sql ;
    sta = conn.createStatement() ;
    sql = String.format("update salary set basic_salary=%s,merit_salary=%s,
        allowance=%s,overtime_salary=%s,id=%s,release_time='%s' WHERE salary_id=%s ;
        ",
        basic_salary,merit_salary,allowance,overtime_salary,id,release_time,
        salary_id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void salary_update() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

");

    System.out.println("需要更新的工资条编号：");
    String s = sc.nextLine() + " ";
    System.out.println("依次更新后的信息:\n基本工资 | 绩效工资 | 补贴 | 加班工资 |
        员工编号 | 发放时间");
    s += sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split("\\s") ;
    if(info.length != 7) {
        System.out.println("更新失败!");
        System.out.println("

");
        return ;
    }
    if(salary_update_helper(info[0], info[1], info[2], info[3], info[4], info[5],
        info[6])){
        System.out.println("更新成功!");
    } else {
        System.out.println("更新失败!");
    }
    System.out.println("

");
}
```

B.5.4 查询工资条代码

```
private boolean salary_query_helper( String salary_id, // 查询工资条
                                     String basic_salary,
                                     String merit_salary,
                                     String allowance,
                                     String overtime_salary,
                                     String id,
                                     String release_time) throws SQLException {
    Statement sta ; String sql ; ResultSet rs ;
    sta = conn.createStatement() ;
    // 拼接查询条件
    String con = "WHERE " ;
    if(!salary_id.equals("")) {
        con += String.format("AND salary.salary_id=%s ", salary_id) ;
    }
    if(!basic_salary.equals("")) {
        con += String.format("AND salary.basic_salary=%s ", basic_salary) ;
    }
    if(!merit_salary.equals("")) {
        con += String.format("AND salary.merit_salary=%s ", merit_salary) ;
    }
    if(!allowance.equals("")) {
        con += String.format("AND salary.allowance=%s ", allowance) ;
    }
    if(!overtime_salary.equals("")) {
        con += String.format("AND salary.overtime_salary=%s ", overtime_salary) ;
    }
    if(!id.equals("")) {
        con += String.format("AND salary.id=%s ", id) ;
    }
    if(!release_time.equals("")) {
        con += String.format("AND salary.release_time='%s' ", release_time) ;
    }
    // 查询, 执行sql语句, 多表联查, 将员工id替换成员工姓名
    sql = "SELECT salary_id,basic_salary,merit_salary, allowance,overtime_salary,e.
        name as name,release_time \n" +
        "FROM salary s\n" +
        "LEFT JOIN employee e ON s.id = e.id";
    if(!con.equals("WHERE ")) sql += con ;
    rs = sta.executeQuery(sql) ;
    // 打印结果
    System.out.println("
    -----
    ");
    System.out.println("工资条编号 | 基本工资 | 绩效工资 | 补贴 | 加班工资 | 员工姓
        名 | 发放时间") ;
    System.out.println("
    -----
    ");
}
```

```

        ");
while(rs.next()) {
    System.out.printf("    %s    | %s | %s | %s | %s | %s | %s\n",
        rs.getString("salary_id"), rs.getString("basic_salary"),
        rs.getString("merit_salary"), rs.getString("allowance"),
        rs.getString("overtime_salary"), rs.getString("name"),
        rs.getString("release_time"));
}
System.out.println("
-----

");
return true ;
}
public void salary_query()throws SQLException{
    Scanner sc = new Scanner(System.in) ;
    System.out.println("

");
    System.out.println("依次输入信息(未知请写0,空格作为分隔符,回车结束输入):\n工资
        条编号 | 基本工资 | 绩效工资 | 补贴 | 加班工资 | 员工编号 | 发放时间");
    String s = sc.nextLine() ;
    // 处理输入参数
    String[] info = s.split(" ") ;
    if(info.length != 7) {
        System.out.println("查询失败! ");
        System.out.println("

");
        return ;
    }
    for (int i = 0 ; i < info.length ; ++ i) {
        if(info[i].equals("0")) info[i] = "*" ;
    };
    if(salary_query_helper(info[0], info[1], info[2], info[3], info[4], info[5],
        info[6])) {
        System.out.println("查询成功! ") ;
    } else {
        System.out.println("查询失败! ") ;
    }
    System.out.println("

");
}
}

```

B.6 项目管理代码

B.6.1 增加项目代码

```

private boolean project_add_helper(    String project_id, // 新增项目
                                     String department_id,
                                     String project_name,
                                     String start_time,
                                     String end_time,
                                     String project_header) throws SQLException
    {
Statement sta = conn.createStatement() ;
String sql ;

// 插入员工
sql = String.format("INSERT INTO project values(%s,%s,'%s','%s','%s',%s) ;",
                    project_id,department_id, project_name, start_time, end_time,
                    project_header) ;
sta.executeUpdate(sql) ;
return true ;

}

public void project_add() throws SQLException {
Scanner sc = new Scanner(System.in) ;
System.out.println("

");
System.out.println("输入新增项目信息:\n项目编号 | 所属部门编号 | 项目名称 | 开
始时间名称 | 结束时间 | 项目负责人编号");
String s = sc.nextLine() ;
// 处理输入参数
String[] info = s.split("\\s") ;
if(info.length != 6) {
    System.out.println("添加失败! ");
    System.out.println("

");
    return ;
}
if(project_add_helper(info[0], info[1], info[2], info[3], info[4], info[5])) {
    System.out.println("添加成功! ");
} else {
    System.out.println("添加失败! ");
}
System.out.println("

");
}

```

B.6.2 删除项目代码

```

private boolean project_del_helper(String id) throws SQLException {

```

```

        Statement sta = conn.createStatement(); String sql ;
        sql = String.format("DELETE FROM project WHERE project_id = %s", id) ;
        sta.executeUpdate(sql) ;
        return true ;
    }

    public void project_del() throws SQLException { // 删除项目
        Scanner sc = new Scanner(System.in) ;
        System.out.println("
=====
");
        System.out.println("需要删除的项目编号：");
        String id = sc.nextLine() ;
        if(project_del_helper(id)){
            System.out.println("删除成功!");
        } else {
            System.out.println("删除失败!");
        }
        System.out.println("
=====
");
    }
}

```

B.6.3 修改项目代码

```

private boolean project_update_helper( String project_id, // 更新项目
                                       String department_id,
                                       String project_name,
                                       String start_time,
                                       String end_time,
                                       String project_header) throws SQLException
{
    Statement sta ; String sql ;
    sta = conn.createStatement() ;

    sql = String.format("update project set department_id=%s,project_name='%s',
        start_time='%s',end_time='%s',project_header=%s WHERE project_id=%s",
        department_id, project_name, start_time, end_time,project_header,
        project_id) ;
    sta.executeUpdate(sql) ;
    return true ;
}

public void project_update() throws SQLException {
    Scanner sc = new Scanner(System.in) ;
    System.out.println("
=====
");
    System.out.println("需要更新的项目编号：");
}

```



```

String s = sc.nextLine() + " ";
System.out.println("依次更新后的信息:\n所属部门编号 | 项目名称 | 开始时间名称 |
    结束时间 | 项目负责人编号");
s += sc.nextLine();
// 处理输入参数
String[] info = s.split("\\s");
if(info.length != 6) {
    System.out.println("更新失败!");
    System.out.println("
=====
");
    return ;
}
if(project_update_helper(info[0], info[1], info[2], info[3], info[4], info[5]))
{
    System.out.println("更新成功!");
} else {
    System.out.println("更新失败!");
}
System.out.println("
=====
");
}

```

B.6.4 查询项目代码

```

private boolean project_query_helper( String project_id, // 查询项目
    String department_id,
    String project_name,
    String start_time,
    String end_time,
    String project_header) throws SQLException {
Statement sta ; String sql ; ResultSet rs ;
sta = conn.createStatement();
// 拼接查询条件
String con = "WHERE ";
if(!project_id.equals("")) {
    con += String.format("AND project.project_id=%s ", project_id);
}
if(!department_id.equals("")) {
    con += String.format("AND project.department_id=%s ", department_id);
}
if(!project_name.equals("")) {
    con += String.format("AND project.project_name='%s' ", project_name);
}
if(!start_time.equals("")) {
    con += String.format("AND project.start_time='%s' ", start_time);
}
}

```

```

        if(!end_time.equals("")) {
            con += String.format("AND project.end_time='%s' ", end_time) ;
        }
        if(!project_header.equals("")) {
            con += String.format("AND project.project_header='%s' ", project_header) ;
        }
        // 查询，执行sql语句，三表联查，将对应的id替换成为响应的姓名
        sql = "SELECT project_id,d.department_name as d_name, project_name, start_time,
            end_time,e.name as e_name \n" +
            "FROM project p\n" +
            "LEFT JOIN department d ON d.department_id=p.department_id\n" +
            "LEFT JOIN employee e\n ON e.id=p.project_header\n" ;
        if(!con.equals("WHERE ")) sql += con ;
        rs = sta.executeQuery(sql) ;
        // 打印结果
        System.out.println("
            -----
            ");
        System.out.println("项目编号 | 所属部门 | 项目名称 | 开始时间名称 | 结束时间 |
            项目负责人");
        System.out.println("
            -----
            ");
        while(rs.next()) {
            System.out.printf("    %s    | %s | %s | %s | %s | %s\n",
                rs.getString("project_id"), rs.getString("d_name"),
                rs.getString("project_name"), rs.getString("start_time"),
                rs.getString("end_time"), rs.getString("e_name"));
        }
        System.out.println("
            -----
            ");
        return true ;
    }
    public void project_query()throws SQLException{
        Scanner sc = new Scanner(System.in) ;
        System.out.println("
            -----
            ");
        System.out.println("依次输入信息(未知请写0,空格作为分隔符，回车结束输入):\n项目
            编号 | 所属部门编号 | 项目名称 | 开始时间名称 | 结束时间 | 项目负责人编号");
        ;
        String s = sc.nextLine() ;
        // 处理输入参数
        String[] info = s.split(" ") ;
        if(info.length != 6) {
            System.out.println("查询失败!");
            System.out.println("
            -----
            ");

```

```

        ");
        return ;
    }
    for (int i = 0 ; i < info.length ; ++ i) {
        if(info[i].equals("0")) info[i] = "*" ;
    };
    if(project_query_helper(info[0], info[1], info[2], info[3], info[4], info[5])){
        System.out.println("查询成功！ ") ;
    } else {
        System.out.println("查询失败！ ") ;
    }
    System.out.println("
    ");
}

```
