武汉工程大学
计算机科学与工程学院

综合设计报告

设计名称： 数据库应用综合设计

设计题目： 清朝名人数据库系统设计与实现

专业班级： 2016 计算机科学 03

学生学号： 1605121312

学生姓名： 李美扬

指导教师（职称）： 郭　炜（讲　师）

学业导师（职称）： 黄文芝（副教授）

学生成绩：

完成时间： 2018.11.26-2018.12.14

武汉工程大学计算机科学与工程学院　　制

说明：

1、报告中的第一、二、三项由综合设计负责人在综合设计开始前填写并发给每个学生。

2、学业导师负责批改学生的设计报告，并给出相应的得分。同时，就设计报告质量撰写评语。

3、指导教师就学生在设计期间的表现及设计完成情况分别给出相应的得分。同时，就此两项情况撰写评语。

4、设计的总评成绩由上述各部分累加得出，由指导教师汇总，并填写于报告的封面。

5、设计报告正文字数一般应不少于 5000 字，也可由综合设计负责人根据本项综合设计的具体情况酌情增加字数或内容。

6、此表格式为武汉工程大学计算机科学与工程学院提供的基本格式（适用于学院各项课程设计），各专业也可根据本项综合设计的特点及内容做适当的调整，并上报学院批准。

成绩评定表

学生姓名：李美扬　　学号：1605121312　　班级 计科 03 班

类别	合计分值	各项分值	评分标准	实际得分	评语
报告质量	30	10	报告格式规范，表述清晰，章节内容组织恰当。符号统一，图表完备，符合规范要求。参考文献数量在 5 篇以上，格式及引用符合要求。		学业导师（签字）：
		10	报告内容翔实，结构严谨合理。课题背景介绍清楚，综述充分。设计与实现等主要过程完整，论述具体透彻。能运用所学专业知识对问题加以分析和求解。无抄袭现象。		
		10	设计报告对整个设计过程进行了全面总结，体现了收获，得出了有价值的结论或结果。		
平时表现	20	20	遵守学习纪律，表现良好，积极完成课程设计任务，无旷课、迟到、早退等情况。		指导教师（签字）：
设计完成情况	50	30	按照要求完成设计内容，方案合理，功能完善，设计工作量饱满，能运用专业知识和技能去发现与解决实际问题。		
		20	在设计过程中展现出了较强的学习能力、动手实践能力、团队协作能力和创新意识。		
总评成绩					

一、综合设计目的、条件、任务和内容要求：

数据库应用综合设计是计算机类专业的重要综合实践环节，对于培养学生的软件设计能力、信息素养等具有重要作用。其目的在于加深对数据库基础理论和基本知识的理解，培养学生的数据库设计与应用开发等实践能力。

要求能够自觉运用数据库系统课程学习的理论知识进行软件设计；掌握信息管理系统的开发方法和步骤。

整个应用系统的设计严格按照数据库设计的方法来进行，包括数据库的设计和应用程序的设计，两部分相辅相成。

1、数据库设计过程包含以下步骤：

(1)需求分析：系统的目的、用户的各种可能要求、业务流程图、数据流程图。

(2)概念结构设计：用 E-R 图来描述实体及实体间的联系。

(3)逻辑结构设计：确定关系模式（包括关系模式优化），各种约束的声明，如主外码约束、唯一性约束、非空约束等。同时给出系统的功能模块组成图，系统各模块功能。

(4)物理结构设计。

(5)数据库实施。

2．数据库的实施阶段：数据库用 SQL SERVER 等创建，前端开发使用 JAVA、NET 或 VB、VC、C#等实现。

3．通过此次综合设计提高自己独立分析问题、解决问题的能力。掌握从需求分析、数据库设计（概念结构设计、逻辑结构设计、物理结构设计）、编写程序、测试分析，撰写文档的整个过程。

二、进度安排：

2018.11.26：进行概念模型设计，画出 ER 图，找出各实体之间的联系。

2018.11.27-2018.12.9：进行数据模型设计，用 SQL SERVER 实现该模型，并用高级程序语言实现界面设计。

2018.12.10-2018.12.14：进行系统联调与测试，撰写综合设计报告。

目 录

摘　要

清朝是中国历史最后一个大一统封建王朝，共传十帝，享国二百七十六年。为了能够将古代名人及各个朝代的皇帝的信息统一起来，使用数据库信息管理系统是很常见的方法。数据库与 JAVA Swing 结合起来，既可以做出一个直观、简约的界面，也可以很容易实现添加、删除、查询和修改操作。同时也可以存储相当可观的资料。首先，利用 java 做出必要的界面，其中包括登录以及增删查改五个界面，其次，使用 JDBC 将已经存入名人、皇帝资料库的数据库连接起来，最后再加入指令代码，即可实现名人数据库管理系统。在名人数据库管理系统设计完成之后，它必须具备界面中所对应的功能，即可以完成登录、注册、添加、删除、查询和修改操作，并且在数据库中会得到相应的反映。名人数据库管理系统相对于之前的管理方法及途径更方便管理，易于更新，最大化的简化了管理员的管理工作，同时也让使用者更容易接受该系统。

关键词：古代名人；管理系统；javaswing 开发；mysql 储存

Abstract

The Qing Dynasty was the last unified feudal dynasty in Chinese history. It was handed down to ten emperors for 276 years. In order to unify the information of ancient celebrities and emperors of different dynasties, the use of database information management system is a common method. The combination of database and JAVA Swing can not only make an intuitive and concise interface, but also make it easy to add, delete, query and modify operations. At the same time, considerable data can be stored. Firstly, we use java to make the necessary interfaces, including login, add, delete and modify five interfaces. Secondly, we use JDBC to connect the databases already stored in celebrities and emperors'databases. Finally, we add instruction codes to realize the celebrity database management system. After the design of celebrity database management system is completed, it must have corresponding functions in the interface, that is, it can complete login, registration, addition, deletion, query and modification operations, and it will be reflected in the database accordingly. Celebrity database management system is easier to manage and update than the previous management methods and ways, which simplifies the management of administrators and makes users more receptive to the system.

Keywords：Ancient celebrities; Management system; JavaSwing development; Mysql storage

第一章　课题背景

以前对古代名人管理的主要方式是基于文本、表格等纸介质的手工处理，对于数据信息处理工作量大，容易出错；随着网络的发展很多资料由纸质迁移到了网络上，但是由于数据繁多，杂乱且良莠不齐，不易查找[1]。采用数据库技术生成的名人管理系统将会极大地方便古代名人管理和使用人员，使得需要查找整理此类资料的人员从繁忙、复杂的工作进入到一个简单、高效的工作中。

1.1 课题介绍

进入 21 世纪，随着计算机和网络技术的飞速发展，数字资源越来越显现其重要作用，依托计算机中的资源，建立具有文化特色的专题资源数据库，已成为许多机构的重要任务之一[1]。名人数据库就是其中的一种以收集、整理、开发、利用具有公众效应的名人资料，并对名人资源进行重组、整合和分层次加工，实现多途径的管理和深层次地揭示名人资料的一种专题数据库[2]。

名人数据库是一种以收集、整理、开发、利用具有公众效应的名人资料，并对名人资源进行重组、整合和分层次加工，利用多个软件实现检索和增删查改的功能。建设名人数据库应具备资源、数据、技术和网络等方面的基础[2]。名人专题数据库的组织结构应以人为主线，并充分发挥 JAVA 的性能，集多功能为一体，具体结构框架可由 MYSQL 中建立的数据库、JAVA 的功能和 JDBC 连接构成。

网络中虽然存贮大量古代名人资料，但是其中的内容也是鱼龙混杂。所以让当用户们在网上查询名人资料时，虽然网页上囊括的知识比较多，但是事实上见效甚微[3]。为了能够使得用户们可以更高效的查询到自己想要的知识内容，名人数据库管理系统应运而生。现代计算机可以帮助人们实现这些看似并不复杂的而实际操作起来非常不顺心的工作。试想一下，当用户想要查询某个人的资料时，只需要输入人物名字，就可以看到该人的详细资料。设计名人数据库的目的便在于在计算机软件支持下，实现对名人资料信息采集、输入、输出，便于管理,便于检索的技术系统[4]。

1.2 工具介绍

1.2.1 java 语言

Java 是一门面向对象编程语言，不仅吸收了 C++语言的各种优点，还摒弃了 C++里难以理解的多继承、指针等概念，因此 Java 语言具有功能强大和简单易用两个特征。Java 语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程[5]。

Java 具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等。

1.2.2 Eclipse 介绍

Eclipse 是著名的跨平台的自由集成开发环境（IDE）。最初主要用来 Java 语言开发，通过安装不同的插件 Eclipse 可以支持不同的计算机语言，比如 C++和 Python 等开发工具。Eclipse 的本身只是一个框架平台，但是众多插件的支持使得 Eclipse 拥有其他功能相对固定的 IDE 软件很难具有的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE[6]。

Eclipse 最初由 OTI 和 IBM 两家公司的 IDE 产品开发组创建，起始于 1999 年 4 月。IBM 提供了最初的 Eclipse 代码基础，包括 Platform、JDT 和 PDE。Eclipse 项目 IBM 发起，围绕着 Eclipse 项目已经发展成为了一个庞大的 Eclipse 联盟，有 150 多家软件公司参与到 Eclipse 项目中，其中包括 Borland、Rational Software、Red Hat 及 Sybase 等。Eclipse 是一个开放源码项目，它其实是 Visual Age for Java 的替代品，其界面跟先前的 Visual Age for Java 差不多，但由于其开放源码，任何人都可以免费得到，并可以在此基础上开发各自的插件，因此越来越受人们关注[7]。随后还有包括 Oracle 在内的许多大公司也纷纷加入了该项目，Eclipse 的目标是成为可进行任何语言开发的 IDE 集成者，使用者只需下载各种语言的插件即可。

1.2.3 SQL Server 简介

SQL Server 是由 Microsoft 开发和推广的关系数据库管理系统（DBMS），目前最新版本是 2012 年 3 月份推出的 SQL SERVER 2012。SQL 是英文（Structured Query Language）的缩写，意思为结构化查询语言。SQL 语言的主要功能就是同各种数据库建立联系，进行沟通。SQL 被作为关系型数据库管理系统的标准语言[8]。SQL 语句可以用来执行各种各样的操作，例如更新数据库中的数据，从数据库中提取数据等。

1.3 章节安排

本报告总共分为四个章节：
第一章：主要写的是名人管理系统的背景、设计目的、使用的工具介绍等；
第二章：主要介绍该系统的实现的主要功能以及相应的模块；
第三章：主要介绍 MySQL 中主要的数据格式，以及实现各个功能的伪码；
第四章：主要写的是该系统的测试以及结果分析。

第二章　设计简介及设计方案论述

整个应用系统的设计严格按照数据库设计的方法来进行，包括数据库的设计和应用程序的设计，两部分相辅相成[9]。数据库设计过程包含以下步骤：需求分析：系统的目的、用户需求、功能流程图；概念结构设计：用 E-R 图来描述实体及实体间的联系；逻辑结构设计：确定关系模式，各种约束的声明，同时给出系统的功能模块组成图，系统各模块功能；物理结构设计。数据库的实施阶段：数据库用 SQL　SERVER 等创建，前端开发使用 JAVA 实现。

2.1 需求分析

所谓"需求分析"，是指对要解决的问题进行详细的分析，弄清楚问题的要求，包括需要输入什么数据，要得到什么结果，最后应输出什么。可以说需求分析是做系统之前必做的。需求分析是软件工程中的一个关键过程[10]。在这个过程中，系统分析员和软件工程师确定顾客的需要。只有在确定了这些需要后，设计者才能够分析和寻求新系统的解决方法。需求分析阶段的任务是确定软件系统功能。

2.1.1 用户需求

在构造系统时，首先从需求出发构造数据库表，然后再由数据库表结合需求划分系统功能模块。这样，就把一个大的系统分解成了几个小系统。这里把系统划分为了三个模块：登录模块，管理员模块，用户模块。模块分别能够实现以下功能：

（1）登录模块，实现登录功能，　注册功能。
（2）管理员模块，实现管理员对名人的增删改减功能。
（3）读者模块，实现用户查询功能。

2.1.2 系统目标

根据需求分析及用户的沟通，该系统应达到以下目标：

（1）界面设计友好，美观。
（2）数据存储安全，可靠。
（3）信息分类清晰，准确。
（4）强大的查询功能，保证数据查询的灵活性。
（5）操作简单易用，界面清晰大方。
（6）系统安全稳定。
（7）开发技术先进，功能完备，扩展性强。
（8）占用资源少，对硬件要求低。
（9）提供灵活，方便的权限设置功能，使整个系统的管理分工明确。

2.2 概要设计

数据库系统主要使用面向对象的分析方法,采用 JAVA、SQL 语句来详尽地描述清朝历代皇帝和名人的详细资料，名人数据库管理系统可以分为两个大型模块：用户登录和管理员登录[11]。

用户登录是名人数据库系统中的一个重要部分，它虽然不能像管理员一样对数据库进行更新操作，但是管理员的存在也正是服务于用户，管理员的操作能够让用户查询到相对准确的信息，同时也提高了用户查询的效率。同时给予用户查询功能可以防止一些不良用户对系统的恶意使用，进而保证了系统的安全性[12]。

Swing 程序表示 JAVA 的客户端窗体程序，除了通过手动编写代码的方式设计 Swing 程序之外，Eclipse 中还提供了一种 Window Builder 工具，该工具是一种非常好用的 Swing 可视化开发工具，有了它，开发人员就可以通过拖放组件的方式编写 Swing 程序了。

当名人、皇帝的资料表导入 MYSQL 之后，需要在 Eclipse 中做出几个必要的界面：登录界面和增加、删除、查询、修改界面，同时也需要必须的管理员选择界面。在界面中设计采用 Window Builder 插件，进而在后台自动生成代码。

数据库管理系统是在管理员对系统进行更新操作之后，数据库内的数据随之发生改变。登录操作已经完成，其中包括用户登录与管理员登录。两种身份登陆之后出现的界面是不同的，用户登录成功之后只能进入查询界面，而管理员登录之后进入一个选择界面，可以自主选择增加、删除、查询、修改操作。

登录操作已经完成，其中包括用户登录与管理员登录。两种身份登陆之后出现的界面是不同的，用户登录成功之后只能进入查询界面，而管理员登录之后进入一个选择界面，可以自主选择增加、删除、查询、修改操作。

在管理员登录成功之后，进入一个选择界面，在这个界面中包含增删查改四个功能。在添加界面中，必须先输入人物的名字，并在文字提示框后输入相应的内容，进而通过系统将输入的信息录入数据库。

增加、删除、修改三个操作中，无论其中哪一种进行操作，数据库中的数据也需要发生相应改变，这一点可以在基本实现系统中进行操作，操作完成之后打开 MYSQL 的表进行验证。

管理员登录的数据需求分析用户的需求具体体现在各种信息的提供、保存、更新和查询上，这就要求数据库结构能充分满足各种信息的输出和输入。其主要模块示意图如图 2.1 所示。

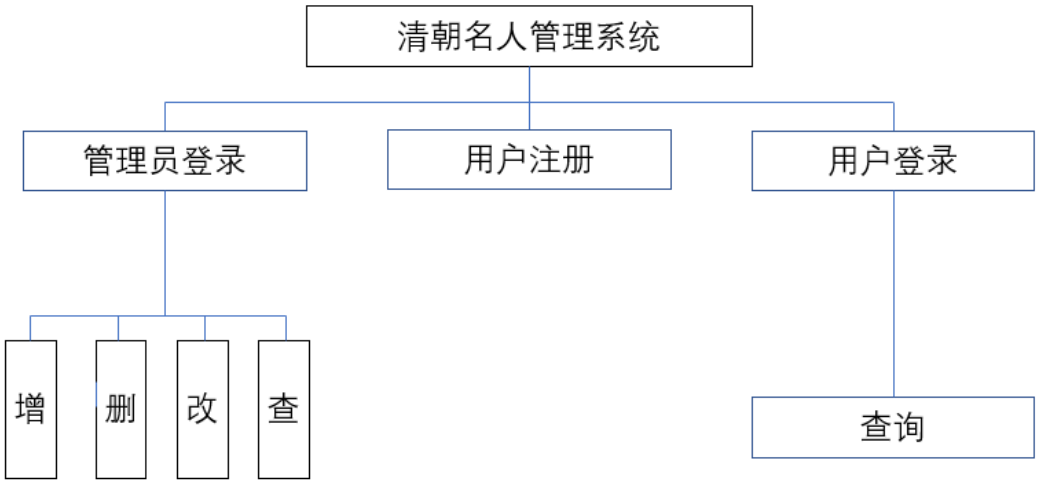

图 2.1 模块示意图

（1）增加信息（管理员身份）：在界面上输入相应的名人信息后，经过系统确认即可以在数据库中存储名人的信息，并且在查询界面可以查询到相应的信息。

（2）查询信息（管理员身份）：输入人物名字，选择该人物的身份（皇帝或者名人），经过系统查询之后即可在界面上显示相应内容。

（3）删除名人（管理员身份）：在相应界面中输入需要删除的名人姓名，确认之后系统将删除该名人信息，并在数据库中清除该名人信息。

（4）修改信息（管理员身份）：在修改界面中输入名人姓名及相应的信息，系统确认之

后即可修改信息，并且在数据库中得以修改，在查询界面亦可查询到更新后的信息。
（5）查询信息（用户身份）：输入人物名字，选择该人物的身份（皇帝或者名人），经过系统查询之后即可在界面上显示相应内容。
（6）登录验证（用户身份）：输入用户名字和密码，点击登录，通过验证即可登录跳转用户查询界面，假如密码错误，则显示“该用户不存在”。
（7）登录验证（管理员身份）：输入管理员名字和密码，点击登录，通过验证即可登录跳转用户查询界面，假如密码错误，则显示“该管理员不存在”。
（8）用户注册（用户身份）：在用户登录界面点击用户注册，输入用户名和密码以及 Id，点击确定，即可注册成功。

第三章　详细设计

概念设计是由分析用户需求到生成概念产品的一系列有序的、可组织的、有目标的设计活动，它表现为一个由粗到精、由模糊到清晰、由抽象到具体的不断进化的过程。概念设计即是利用设计概念并以其为主线贯穿全部设计过程的设计方法。概念设计是完整而全面的设计过程，它通过设计概念将设计者繁复的感性和瞬间思维上升到统一的理性思维从而完成整个设计。

3.1 逻辑设计

逻辑设计就是把一种计划、规划、设想通过视觉的形式通过概念、判断、推理、论证来理解和区分客观世界的思维传达出来的活动过程。逻辑设计比物理设计更理论化和抽象化，关注对象之间的逻辑关系，提供了更多系统和子系统的详细描述。

3.1.1 关系模型

概念结构设计所得的 E-R 模型是对用户需求的一种抽象的表达形式，它独立于任何一种具体的数据模型，因而也不能为任何一个具体的 DBMS 所支持。为了能够建立起最终的物理系统，还需要将概念结构进一步转化为某一 DBMS 所支持的数据模型，然后根据逻辑设计的准则、数据的语义约束、规范化理论等对数据模型进行适当的调整和优化，形成合理的全局逻辑结构，并设计出用户子模式。这就是数据库逻辑设计所要完成的任务。

数据库逻辑结构的设计分为两个步骤：首先将概念设计所得的 E-R 图转换为关系模型；然后对关系模型进行优化，如图 3.1 所示。

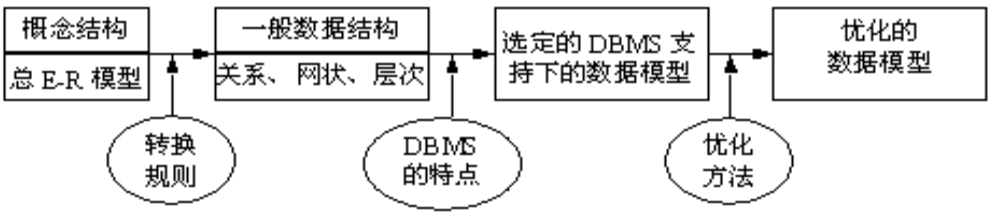

图 3.1 逻辑结构设计过程

关系模型是由一组关系(二维表)的结合，而 E-R 模型则是由实体、实体的属性、实体间的关系三个要素组成。所以要将 E-R 模型转换为关系模型，就是将实体、属性和联系都要转换为相应的关系模型。

本系统的关系模型转换如下：

user (id,username,password)

people(id,name,national,place,time,work,message)

emperor(id,name,national,yearname,place,miaohao,overtime,message,shihao)

3.1.2 系统功能总框

图书馆管理系统功能总框图，如图 3.2 所示。

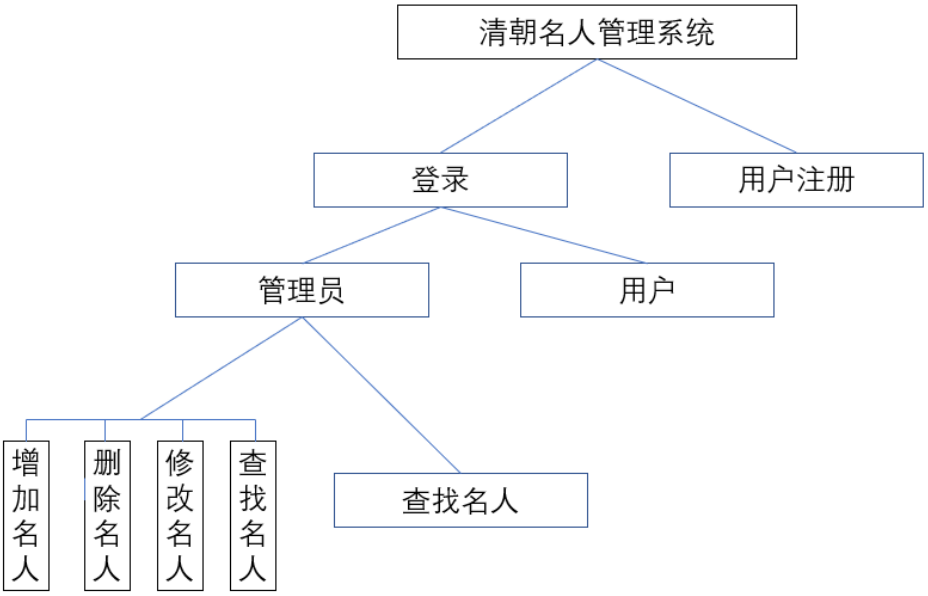

图 3.2　系统功能框架图

3.2 物理设计

数据库在物理上的存储结构与存储方法称为数据库的物理结构，它依赖于选定的数据库管理系统。为一个给定的逻辑数据模型选取一个最适合应用要求的物理结构的过程，就是物理设计。

3.2.1 基本表设计

由于名人数量过于繁多，需要先从网络中收集名人信息，并按照一定的属性存放在 Excel 工作表中。清朝名人分为皇帝、名人两个表，并在收集工作完成之后导入 MYSQL 中，在 MYSQL 同样需要两个表进行存储[10]。

皇帝表需要的属性有 Id、姓名、在位时间、民族、年号、庙号、成就，在 MYSQL 存储方式如表 3-1 所示。

表 3-1 皇帝表

属性	存储类型	是否可为空
Id	Var char(10)	否
姓名	Var char(10)	是
在位时间	Var char(10)	是
民族	Var char(10)	是
年号	Var char(10)	是
庙号	Var char(10)	是
成就	Long text	是

名人表需要的属性有 Id、姓名、时期、民族、官职、出生地、成就，在 MYSQL 存储方式如表 3-2 所示。

表 3-2 名人表

属性	存储类型	是否可为空
民族	Var char(10)	是
Id	Var char(10)	否
时期	Var char(10)	是
姓名	Var char(10)	是
官职	Var char(10)	是
成就	Long text	是
出生地	Var char(10)	是

对于数据库中的皇帝表和名人表存在一定联系，其中的实体与属性的联系如图 3.1 所示。

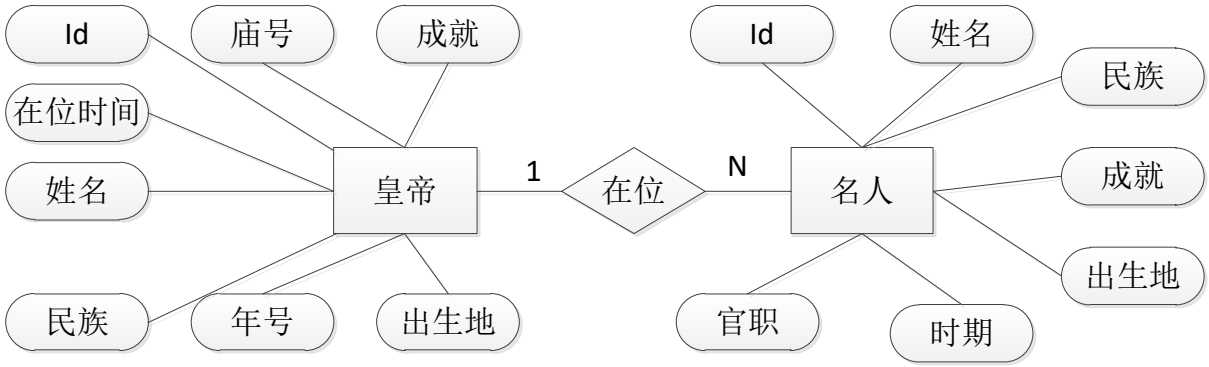

图 3.3 皇帝名人关系 ER 图

3.2.2 系统功能设计

当名人、皇帝的资料表导入 MYSQL 之后，需要在 Eclipse 中做出几个必要的界面：用户和管理员登录界面、用户注册界面、增加、删除、查询、修改界面，同时也需要必须的管理员选择界面。在界面中设计采用 Window Builder 插件，进而在后台自动生成代码[10]。

在用户和管理员登录界面中，在键盘上输入用户名和密码，如果为空或者用户名与密码不匹配都会给出错误提示并无法登录，且在登陆之前需要选择登录身份，即选择“管理员”或者“用户”。如果登陆错误次数超过一定数目，同样会导致无法登陆。如果登录成功则进入相应的界面。故可以写出登录伪码：

```
If 用户名=空 Then
    输出 "登录用户名不能为空，请重新填写！"
    Exit
End If
If 密码= 空 Then
    输出"登录用户密码不能为空，请重新填写！"
End If
If 选择身份=空 Then
    输出 "你没有选择用户身份，请选择！"
```

Exit
End If
连接数据库之后，核对过账号密码；
输出"祝贺你！你已经成功登录！"
"如果账号密码不匹配，输出"对不起！你输入的用户密码不正确，请重新输入！"
次数 + 1；
重新输入；
如果连续三次不正确，则输出“对不起，你已经连续三次输入错误，不能继续登录程序就退出！"
End
ELSE　If 身份 = "管理员" Then
出现选择界面
Else If 身份= "用户" Then
出现查询界面
End If

此时，登录操作已经完成，其中包括用户登录与管理员登录。两种身份登陆之后出现的界面是不同的，用户登录成功之后只能进入查询界面，而管理员登录之后进入一个选择界面，可以自主选择增加、删除、查询、修改操作。

在管理员登录成功之后，进入一个选择界面，在这个界面中包含增删查改四个功能。在添加界面中，必须先输入人物的名字，并在文字提示框后输入相应的内容，进而通过系统将输入的信息录入数据库。

If 身份= "皇帝" Then
查找皇帝名字；
更新信息；
End If
If 身份 = "名人" Then
更新信息
If 信息错误
输出"添加失败",
Else
输出"添加成功"
End If

对于增加操作，实际上是在系统中键入皇帝或者名人的资料，在 UTF-8 格式中的信息会被系统识别，所以在文本框内添加信息会被系统录入。在选择了身份之后，系统会根据代码将信息录入对应的表中，进而将存入的信息放入数据库中，在添加之后，就可以在查询界面中查到相应结果。

同样的，在实现删除功能之前，首先要通过 SQL 语句连接上数据库，在管理员输入了人物姓名之后，系统查找出相应的人物，进而删除掉该任务的所有信息，有删除界面伪代码如下：

连接数据库；
If 身份= "皇帝" Then
查找皇帝名字；
删除；
End If

If 身份 = "名人" Then
删除；
If 信息错误
输出"添加失败"
Else
输出"添加成功"
End If

删除功能实际为系统在识别查询出管理员键入的人物名字之后，在数据库中将相应的信息更新为空，同时将姓名置空，所以在删除之后，数据库中该人物的资料已经被清除，无法再被查询出来。

在修改功能中，管理员首先通过输入人物名字并选择相对应的身份（皇帝或者名人），在相对应的文本框内输入更改后的信息，在输入完成之后，点击修改，系统则会自动修改数据库内的人物信息。修改界面伪代码如下：

连接数据库；
If 身份= "皇帝" Then
查找皇帝名字；
修改信息；
End If
If 身份 = "名人" Then
修改信息；
If 信息错误
输出"修改失败"
Else
输出"修改成功"
End If

修改功能中同样是经过识别之后，找到相应的人物，将其中的信息更新为管理员键入的新内容，伪代码中的“修改信息”实际上是将管理员键入的内容代替原来的资料，从而做到信息更改。

用户或者管理员都可以使用查询界面，在此界面中，需要在文本框内输入人物名字，再选择人物的身份（皇帝或者名人），点击查询，系统则会输出对应的人物信息，如果信息错误则无法输出。查询界面伪代码如下：

连接数据库；
If 身份= "皇帝" Then
查找皇帝名字；
输出信息；
End If
If 身份 = "名人" Then
输出信息；
If 姓名错误
输出"查询失败"
Else
输出"查询成功"
End If

查询信息的运作方式与其他操作类似，在用户或者管理员选择人物身份之后，系统

在数据库的表中查询对应的人物名字，查询成功之后在界面上显示出内容。伪代码中的“输出信息”实为将表中的某一项内容特定输出，在不同的文本框内输出不同的内容。

增加、删除、修改三个操作中，无论其中哪一种进行操作，数据库中的数据也需要发生相应改变，这一点可以在基本实现系统中进行操作，操作完成之后打开 MYSQL 的表进行验证。

四种操作已经基本完成，也就是 MYSQL 和 Eclipse 两个软件的工作已经完成，此时需要使用 JDBC 将二者进行连接，才能在 Eclipse 中的界面实现对应的功能。打开 NAVICAT FOR MYSQL 后，建立连接，新建数据库，在 JDBC 的连接 MYSQL。即完成了 Eclipse 与 MYSQL 的连接。

第四章　设计结果及分析

系统有 3 个模块：登录模块，管理员模块和用户模块。登录模块实现登录功能， 注册功能；管理员模块实现名人的增删改查；用户模块实现名人查找功能。

4. 1 功能测试

在 Eclipse 打开工程，即可进入用户登录界面，登录界面显示如图 4.1 所示。

图 4.1 用户登录验证图

选择用户身份进行登录操作时，点击注册用户，注册用户账号密码，即可完成注册，显示如图 4.2 所示。

图 4.2 用户注册界面图

注册用户账号密码及序号填写完毕后，即可完成注册，具体结果显示如图 4.3 所示。

图 4.3 用户添加结果图

使用数据库中已经存储的用户账号密码，进行登录，登录成功界面如图 4.4 所示。

图 4.4 查询选项图

查询界面选项界面，在此界面中，需要选择人物的身份（皇帝或者名人），在文本框内输入人物名字，再点击查询，显示如图 4.5、4.6 所示。

图 4.5 皇帝信息查询图

图 4.6 名人信息查询图

选择人物的身份（皇帝或者名人）之后，在文本框内输入人物名字，再点击查询，系统则会输出对应的人物信息。显示如图 4.7、4.8 所示。

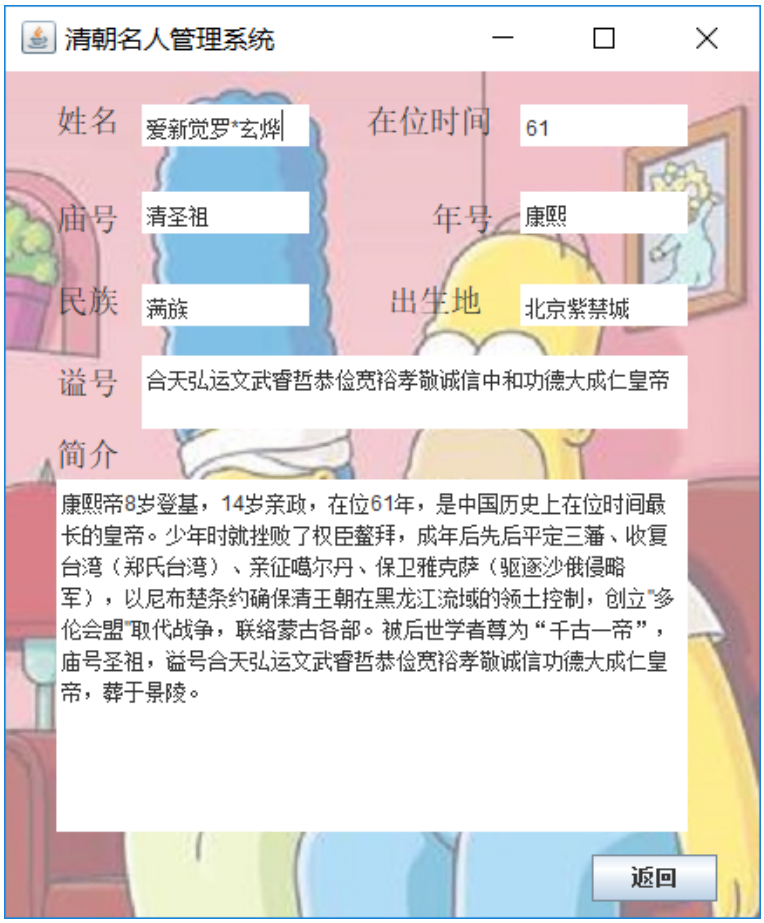

图 4.7 皇帝信息显示图

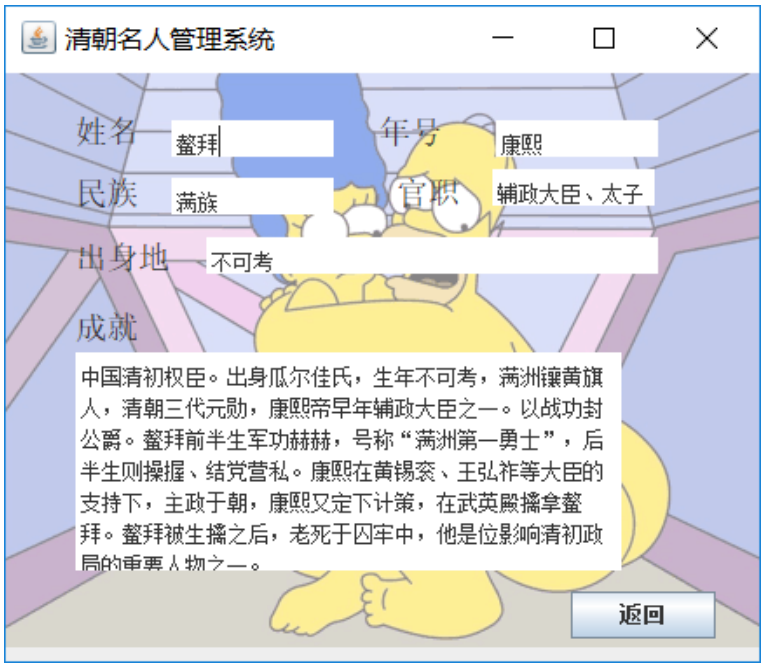

图 4.8 名人信息显示图

选择以管理员身份进入系统，显示管理员登录验证界面，输入管理员帐号和密码，点击登录按键，如图 4.9 所示。

图 4.9 管理员登录验证图

管理员帐号、密码和数据库管理员信息表比对后，验证成功，即可出现管理员主界面如图 4.10 所示。

图 4.10 管理员主界面图

选择皇帝中的查询，即可看到添加窗口，名人和此界面一样，如图 4.11，图 4.12 所示。

图 4.11　皇帝添加界面

13	新的测试	皇帝	

图 4.12　皇帝添加成功的数据库

点击添加皇帝，即可成功添加皇帝 选择皇帝中的查询，即可看到查询窗口，名人和此界面一样，如图 4.13 所示。

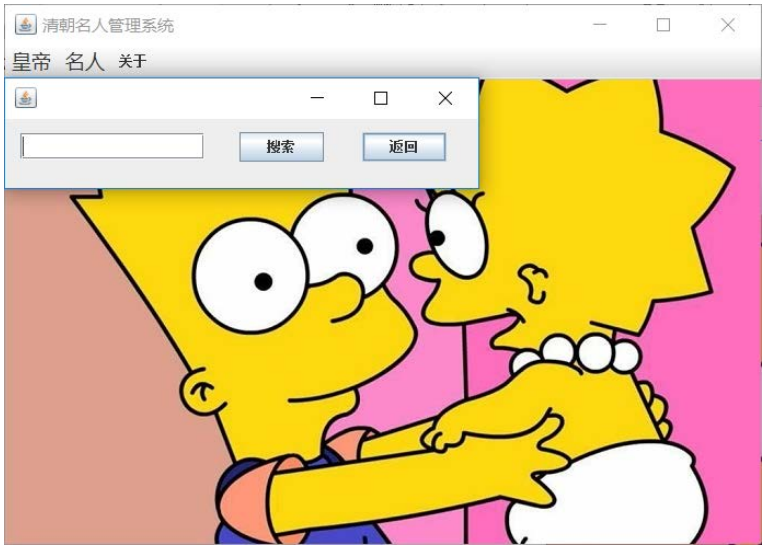

图 4.13 皇帝查询界面

搜索刚才的添加皇帝，即可进入皇帝的删，改界面，如图 4.14 所示。

图 4.14 皇帝修改删除界面

加入新的文字，或者修改旧信息，即可做到修改皇帝的功能，如图 4.15，图 4.16，图 4.17 所示。

图 4.15 皇帝修改界面

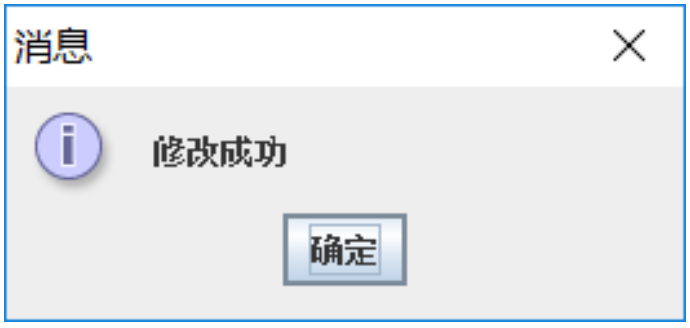

图 4.16 修改成功提示

13	新的测试	哈哈	皇帝		哈哈	122	哈哈	哈哈

图 4.17 修改成功的数据库

直接点击删除按钮，即可做到删除当前页面皇帝的功能，如图 4.18，图 4.19 所示。

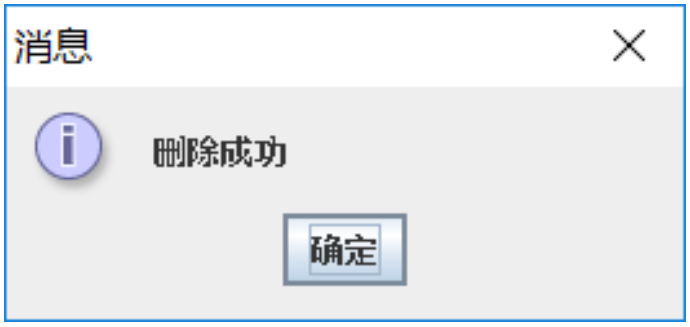

图 4.18 删除成功的提示

4.2 结果分析

一款好的名人数据库管理系统是在管理员对系统进行更新操作之后，数据库内的数据随之发生改变。在这一款清朝名人管理系统中，经过增加、删除、更改之后，再次查询人物信息都能得到正确的结果，说明在各种操作之后，数据库中的数据同样改变，可以评价为这一款名人管理系统是相对比较成功的。

管理员增加、删除、修改三个操作中，无论其中哪一种进行操作，数据库中的数据也需要发生相应改变，这一点可以在基本实现系统中进行操作，操作完成之后打开 MYSQL 的表进行验证。四种操作已经基本完成，也就是 MYSQL 和 Eclipse 两个软件的工作已经完成，此时需要使用 JDBC 将二者进行连接，才能在 Eclipse 中的界面实现对应的功能[11]。

总　　结

经过两周的努力，清朝名人管理系统终于完成，经过这次综合设计，自己总结了这个名人数据库管理系统的一些问题，不过收获还是颇为丰富的，再有理论知识上结合实践，使我学到了更多知识。

首先，更进一步的了解了系统分析与设计、JAVA、JDBC 的基本操作，在这之前，系统分析与设计的学习仅仅刚开了个头，我们只是在了解一些概念性的东西。在做这个系统之前，我连基本的系统分析与设计，界面设计等这些东西都不熟练。现在对于系统中的增删改查操作比较熟练了。

对于初学者来说，比较头疼的就是对于数据库连接的处理。我的建议是如果不理解先把按照课本上正确的语句敲，然后在多次进行数据库的链接，增删改查操作中不断总结规律。

这次设计的名人数据库管理系统，全在自己所掌握的知识下，进行系统分析与设计，完全体现了自己在系统分析与设计中设计课程学习状况，充分地为自己以后更深入了数据库语言奠下深厚的基础。纵观此名人数据库管理系统的整体概况，目前，自我认为设计良好，相关功能都能够实现，功能强大，条理清晰，界面可观性比较好。并且特色在于，所设计的表单都在一个表单系统桌面中运行，比较符合系统的观念。在系统设计的过程中，我从中发现，学习系统分析与设计要细心和有耐性，并且要不断地从外界学习更多的技术才能设计出一套完美的系统。

致　　谢

在综合设计中，遇到了许多问题，特别是 Java 中代码的问题，首先感谢郭老师，张老师与孙老师，在这几周的综合设计一直辛勤地指导我们，在我的清朝名人管理系统设计中给予了我很大的帮助，解决了我本次设计中的难点，

本次综合设计中同学也给予了我很大帮助，在我们清朝名人资料收集中，尤其感谢小组同学，给了我不少名人资料。

最后感谢学院，给了我们这次难得的机会，让我们获得了宝贵的知识，在编程方面有了很大进步。

参考文献

[1] 余丽君．关于建立沈阳名人数据库的设想[J]．图书馆学刊，1999，21 (6)：19-20．
[2] 赵志刚，王伟．使用 SQL Server 管理应用程序服务数据[J]．沈阳师范大学学报(自然科学版)，2010，28(2)：233-235．
[3] 黄欣欣．浅谈权限管理系统的需求分析[J]．科技信息，2010(16)：69-70．
[4] 杨华．基于 B/S 模式的高校仓库管理系统的需求分析[J]．无线互联科技，2014(8)：71-71．
[5] 周龙．分布式数据库管理系统实现技术[M]．科学出版社，1998．15-16．
[6] 张慎明，卜凡强，姚建国．遵循 IEC61970 标准的实时数据库管理系统[J]．电力系统自动化，2002，26(24)：26-30．
[7] 廖卫东．陈梅．JAVA 程序设计[M]. 机械工业出版社，2008．268-346．
[8] 李素若．JAVA 面向对象程序设计[M]. 北京化学工业出版社，2008．126-136．
[9] 傅仕星．JAVA Swing 设计基础[M]. 清华大学出版社，2003．68-89．
[10] 刘彦明．数据库基础设计与开发[M]. 西安电子科技大学出版社，2009．56-67．
[11] 华轩逸. JAVA 面向对象程序设计（第二版）[M]. 北京：中国铁道出版社，2012. 21-29．
[12] 郑国果．JAVA 语言程序设计（第 4 版）[M]．北京：清华大学出版社，2010．33-38．

附录　主要程序代码

```
package com.dao;
import java.sql.ResultSet;
import com.model.Celebrity;
import com.model.Emperor;
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.PreparedStatement;
public class EmperorDao {
    /*
     * 查询皇帝
     */
    public static Emperor FindEByAll(Connection con, String p)throws Exception{
        Emperor emperor1 = new Emperor();//新建 emperor 对象
        String sql = "select * from t_emperor where id = ? or name = ? or yearname = ? or
miaohao = ?";
        PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);
         pstmt.setString(1,p);
         pstmt.setString(2,p);
         pstmt.setString(3,p);
         pstmt.setString(4,p);
            ResultSet rs=pstmt.executeQuery();
            if(rs.next())
              {
                  emperor1=new Emperor();
                  emperor1.setEid(rs.getString("id"));
                  emperor1.setEjianjie(rs.getString("message"));
                  emperor1.setExingming(rs.getString("name"));
                  emperor1.setEminzu(rs.getString("National"));
                  emperor1.setEzaiweishijian(rs.getString("overtime"));
                  emperor1.setEchushengdi(rs.getString("place"));
                  emperor1.setEmiaohao(rs.getString("miaohao"));
                  emperor1.setEnianhao(rs.getString("yearname"));
                  emperor1.setEshihao(rs.getString("shihao"));
              }
            return emperor1;
    }
    /*
     * 添加皇帝
     */
    public static int AddEmperor(Connection con, Emperor emperor2) throws Exception{
        String sql = "insert into
t_emperor(name,national,yearname,place,miaohao,overtime,message,shihao) values
(?,?,?,?,?,?,?,?)";
```

```
        PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);
        pstmt.setString(1, emperor2.getExingming());
        pstmt.setString(2, emperor2.getEminzu());
        pstmt.setString(3, emperor2.getEnianhao());
        pstmt.setString(4, emperor2.getEchushengdi());
        pstmt.setString(5, emperor2.getEmiaohao());
        pstmt.setString(6, emperor2.getEzaiweishijian());
        pstmt.setString(7, emperor2.getEjianjie());
        pstmt.setString(8, emperor2.getEshihao());

        return pstmt.executeUpdate();
    }

    /*
     * 修改
     */
    public static int ChangeEmperor(Connection con, Emperor emperor) throws Exception{
        String sql = "update t_emperor set name = ? ,national = ?,yearname = ?,place
= ?,miaohao = ?,overtime = ?,message = ?,shihao = ? where id = ?";
        PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);
        pstmt.setString(1, emperor.getExingming());
        pstmt.setString(2, emperor.getEminzu());
        pstmt.setString(3, emperor.getEnianhao());
        pstmt.setString(4, emperor.getEchushengdi());
        pstmt.setString(5, emperor.getEmiaohao());
        pstmt.setString(6, emperor.getEzaiweishijian());
        pstmt.setString(7, emperor.getEjianjie());
        pstmt.setString(8, emperor.getEshihao());
        pstmt.setString(9,emperor.getEid());

        return pstmt.executeUpdate();
    }

    /*
     * 删除
     */
    public static int DeleteEmperor(Connection con,String id) throws Exception{
        String sql = "delete from t_emperor where id = ?";
        PreparedStatement pstmt = (PreparedStatement) con.prepareStatement(sql);
        pstmt.setString(1, id);

        return pstmt.executeUpdate();
    }
```

```
}
/*
         * 添加背景
     */
    public void setBak(){
        ((JPanel)this.getContentPane()).setOpaque(false);
        ImageIcon img = new ImageIcon("src\\pic\\bg\\emperorShow.jpg"); //添加图片
        JLabel background = new JLabel(img);
        this.getLayeredPane().add(background, new Integer(Integer.MIN_VALUE));
        background.setBounds(0,0, img.getIconWidth(), img.getIconHeight());
    }
    /*
     * 修改
     */
    protected void Butchange(JTextArea textArea_jianjie) {
        String id = Emperor.getEid(); //获得 id
        String xingming = this.textField_xingming.getText();
        String zaiweishijian = this.textField_zaiweishijian.getText();
        String miaohao = this.textField_miaohao.getText();
        String nianhao = this.textField_nianhao.getText();
        String minzu = this.textField_minzu.getText();
        String chushengdi = this.textField_chushengdi.getText();
        String shihao = this.textField_shihao.getText();
        String jianjie = textArea_jianjie.getText();

        if(StringUtil.isEmpty(xingming)) {
            JOptionPane.showMessageDialog(null, "姓名不能为空");
            return;
        }
        else if(StringUtil.isEmpty(nianhao)) {
            JOptionPane.showMessageDialog(null, "年号不能为空");
            return;
        }
        Emperor emperor = new
Emperor(xingming,zaiweishijian,miaohao,nianhao,minzu,chushengdi,shihao,jianjie);
        emperor.setEid(id);

        Connection con = null;
        try {
            con = DbUtil.getCon();
            int ChangeResult = EmperorDao.ChangeEmperor(con, emperor);
            if(ChangeResult == 1) {
                JOptionPane.showMessageDialog(null,"修改成功");
            }else {
```

```
                JOptionPane.showMessageDialog(null,"修改失败，请稍后再试");
            }
        }catch(Exception e) {
            e.printStackTrace();
        }
    }
    /*
     * 删除
     */
    protected void DeleteButAction(){
        String id =Emperor.getEid();
        //System.err.println(id);
        Connection con = null;
        try {
            con = DbUtil.getCon();
            int DeleteResult = EmperorDao.DeleteEmperor(con, id);
            if(DeleteResult == 1) {
                JOptionPane.showMessageDialog(null,"删除成功");
                dispose();
                new Home().setVisible(true);
            }else {
                JOptionPane.showMessageDialog(null,"删除失败，请稍后再试");
            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }
```