
华侨大学

信息科学与工程学院

课程设计报告

课程名称 数据库技术课程设计

题 目

KTV 点歌系统

院(系)别

信息科学与工程学院

专 业

自动化

级 别

学 号

姓 名

指导老师

2019 年 11 月 24 日

KTV 点歌系统设计

摘 要

现如今随着经济文化水平的显著提高，人们对生活质量及工作环境的要求也越来越高。同时，随着生活节奏的加快，每个人都处于忙碌烦乱的社会当中，不论是在家庭、工作场所或是学校中，无时无刻充满着各方面的压力。在闲时，找到一种能够缓解压力、释放疲劳的娱乐方式，已成为大家共同的愿望。由于受到工作条件和时间的限制，又由于去 KTV 既方便省事，又能很好地娱乐放松，所以越来越多的人选择去 KTV 唱歌跳舞。

本文便基于 SQL server 和 VB.NET 设计了一套 KTV 点歌系统。本系统的使用 SQL server 创建了数据库，并建立了歌曲信息表、歌手信息表、会员信息表和管理员信息表等 KTV 点歌系统数据库关系表。同时，本系统使用 VB.NET 设计了美观且友好的人机交互界面，用户可以通过数字点歌、歌星点歌、拼音点歌和歌名点歌四种方式点播歌曲，也可以方便的查看歌曲排行榜信息。而管理员也可以很方便的对整个 KTV 系统的信息进行管理。

关键词：KTV 点歌系统，数据库，SQL Server，VB.NET



目 录

引言	1
第 1 章 概述	1
1.1 数据库设计背景.....	1
1.2 数据库设计意义.....	1
1.3 数据库设计内容.....	1
第 2 章 KTV 点歌系统设计	3
2.1 需求分析.....	3
2.1.1 信息要求分析.....	3
2.1.2 处理要求分析.....	3
2.2 概念模式设计.....	5
2.2.1 E-R 模型的基本概念	5
2.2.2 局部 E-R 图设计	6
2.2.3 全局 E-R 图设计	11
2.3 逻辑模式设计.....	11
2.3.1 E-R 图到关系模式的转换	12
2.3.2 关系模式的规范化处理	13
第 3 章 数据库的实施与 VB.NET 界面开发	14
3.1 数据库基本表设计	14
3.2 登录模块界面设计及程序分析	16
3.3 前台点歌模块界面设计及程序分析	18
3.3.1 开始点歌窗口.....	19
3.3.2 用户积分查询和密码修改.....	29
3.4 后台管理模块界面设计及程序分析	31
3.4.1 用户信息管理.....	31
3.4.2 歌曲信息管理和歌手信息管理.....	36
3.4.3 管理员密码修改.....	43
3.4.4 超级管理员.....	44
总结	50
参考文献	51

引 言

《数据库课程设计》是自动化专业的实践课程。在较为系统地学完《数据库技术及应用》后，通过此课程设计，一方面可以增强学生对数据库的理解；另一方面可以培养学生查找资料、分析问题和解决问题的能力。

关系数据库设计过程分为以下 5 个阶段：需求分析、概念模式设计、逻辑模式设计、数据库实施、数据库运行维护。也就是说，数据库的设计过程是先进行认真细致的需求分析，在清楚用户的要求后抽象出实体和实体之间的关系，用建模方法 E-R 图表示出来，然后根据具体实施的 DBMS 平台将其转换为相应平台所支持的数据库逻辑模式并进行规范化处理，最后进行建库建标等物理模式的设计，从而完成整个数据库模式的设计过程。

本次课程设计是对 KTV 点歌系统的设计，设计过程也严格遵循以上 5 个阶段。在设计完成的数据库中，应具备以下功能：对会员用户而言，登录到前台点歌系统后可以根据歌曲编号、歌手姓名、歌曲拼音缩写或歌曲名称检索歌曲，用户点击歌曲即可将歌曲加入到我的歌单中。进入播放列表后可以实现播放我的歌单中的歌曲，同时可以移除所选歌曲、清空我的歌单、暂停播放和重新播放等功能。在开始点歌窗口，用户还可以进入歌曲排行榜对歌曲的点播热度进行查询，同时也可以在歌曲排行榜中将歌曲添加到我的歌单中。每当一首歌被添加进我的歌单中时，歌曲的总点击率便会加 1；对于系统管理员而言，登录到 KTV 后台管理系统后可以对用户信息、歌曲信息和歌手信息进行添加、删除、查询、更新等操作。同时设置超级管理员，若管理员拥有超级管理员权限则可以对系统所有的管理员信息进行管理。也就是，数据库的设计应该具备基本的增删改查功能。

在第一部分，我们将介绍 KTV 点歌系统的设计背景、设计意义和设计内容。第二部分是本报告的重点，基于 KTV 点歌系统，对数据库设计的 5 个阶段，即需求分析、概念模式设计、逻辑模式设计、数据库实施、数据库运行与维护。

第 1 章 概述

1.1 数据库设计背景

由于经济文化的高速发展，人们对生活质量及工作环境的要求也越来越高，追求也越来越高。同时，随着生活节奏的加快，每个人都处于忙碌的社会当中，不论是在家庭，工作场所，或是学校，无时无刻充满着生活和学习上的压力。工作之余，找到一种释放压力，释放疲劳的娱乐方式，已成为大家共同的愿望。

然而，受到工作条件和时间的限制，越来越多的人选择了去 KTV 厅唱 K 休闲娱乐。唱歌，自然免不了点歌的繁琐。传统的点歌设备主要是 CD 和固放设备组成，虽然基本满足了人们的需求，但是也带来了设备成本高，点歌过程繁琐，更新麻烦等不少问题。随着科学的发展和信息时代的到来，点歌系统应运而生，KTV 点歌系统已经成为了娱乐场所必不可少的一部分。KTV 点歌系统不仅简化了用户操作，而且极大化满足了人们的需求和使用习惯。

1.2 数据库设计意义

KTV 点歌系统是娱乐场所的一项管理措施，因此开发 KTV 点歌系统具有较大的社会意义，同时点歌系统是一娱乐软件的开端，它具有简单的软件特征，系统结构与现实生活紧密结合，具体直观，具有典范的便捷软件特点。KTV 点歌系统主要用于对音乐的各项查找，操作简单，直观。用起来极为方便。减少了手工操作对工作人员带来的不便和繁琐，使每个用户都能根据自己爱好选择自己喜欢的音乐。因此，KTV 点歌系统是企业化、智能化、科学化、正规化不可缺少的管理软件。

1.3 数据库设计内容

运用所学的数据库技术理论知识，对 KTV 点歌系统进行数据库设计。设计应包含需求分析、概念模式设计、逻辑模式设计、数据库实施、数据库运行和维护五个阶段。本次 KTV 点歌系统的开发主要包含前台点歌系统和后台管理系统模块的开发。

前台点歌系统包含以下功能：

1. 歌曲点播（包含歌星点歌、数字点歌、拼音点歌、歌名点歌、全部歌曲）

-
2. 歌曲排行榜
 3. 将选中的歌曲添加至播放列表并进行播放
 4. 用户使用用户名和密码登录
 5. 用户积分查询和密码修改

后台管理系统包含以下功能：

1. 用户信息管理（增加、删除、更新、修改用户信息）
2. 歌曲信息管理（增加、删除、更新、修改歌曲信息）
3. 歌手信息管理（增加、删除、更新、修改歌手信息）
4. 超级管理员可以对管理员信息进行管理
5. 管理员密码修改



华侨大学
HUAQIAO UNIVERSITY

第 2 章 KTV 点歌系统的设计

本次对于 KTV 点歌系统的设计，应该包含以下 5 个阶段：需求分析、概念模式式设计、逻辑模式式设计、数据库实施、数据库运行和维护，如图 2-1 所示。

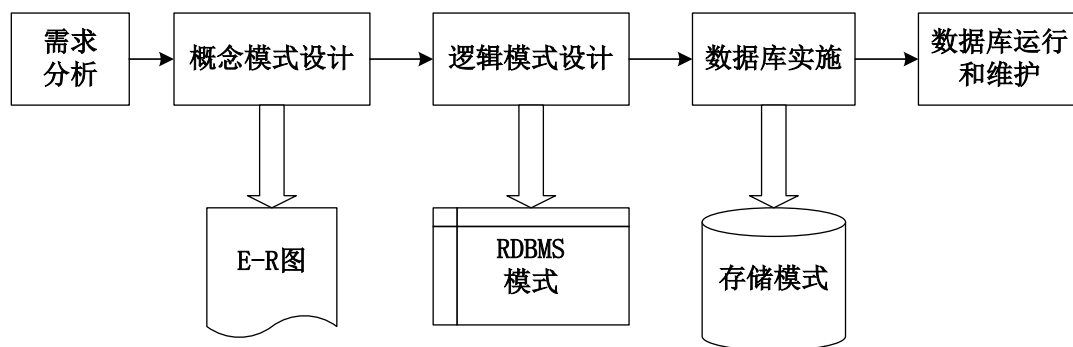


图 2-1 数据库的设计过程

2.1 需求分析

2.1.1 信息要求分析

在此次《数据库课程设计》中，根据课程内容安排，所选题目为：KTV 点歌系统。经过对 KTV 点歌系统的结构化分析，得出信息要求如下：

- 1、歌手信息：歌手编号、姓名、姓名拼音缩写、性别、出生日期、国籍。
- 2、歌曲信息：歌曲编号、歌曲时长、歌曲存储。
- 3、歌曲排行榜：歌曲编号。
- 4、我的歌单：歌曲编号。
- 5、会员信息：会员编号、注册时间、会员密码、积分。
- 6、管理员信息：管理员编号。
- 7、超级管理员：超级管理员编号。
- 8、后台操作记录：操作时间。

2.1.2 处理要求分析

根据以上 8 个表的属性，可以总结出，设计出来的数据库应该面向用户和管

理员。对于用户，需要会员账号和密码登入前台点歌系统，进入后可以查询个人账户的积分以及修改账户密码。还可通过多种方式进行歌曲点播，如根据演唱歌星的名字、歌曲名称编号、歌曲名称拼音缩写以及歌曲名称等检索歌曲。同时，可以查看歌曲排行榜信息。对于管理员，可使用管理员编号和密码进行登入后台操作系统，其进入后可对歌曲、歌手以及会员用户信息进行各类操作，如增加、删除、更新、修改等，也可以修改管理员账号的密码。同时，设置超级管理员，其可使用超级管理员的账号密码进入管理员信息管理系统对管理员信息进行各类操作。

上面表的结构关系较为混乱，如果没有进行必要的修改，将导致数据冗余、修改异常、插入异常、删除异常等情况，所以接下来将根据需求分析所需要的功能做后续的分析与设计，图 2-2 为 KTV 点歌系统结构图：

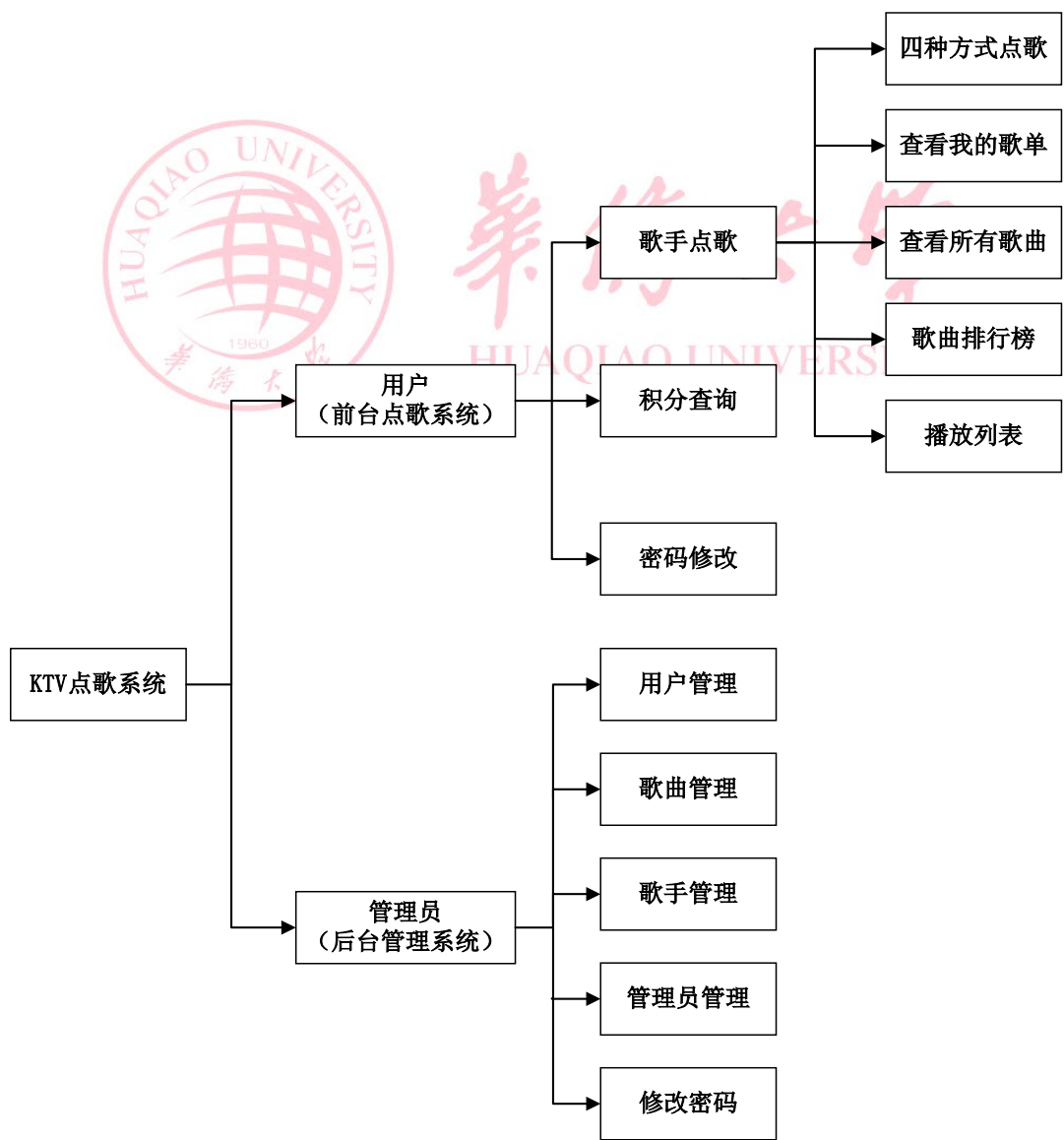


图 2-2 KTV 点歌系统结构图

2.2 概念模式设计

直接把现实世界事务及其联系抽象为数据世界的数据库模型过于复杂，因此一般是先将现实世界的事务及其联系抽象为信息世界实体及其联系，然后再将其转化为计算机世界的数据库模型，如关系数据库模型。

概念模式设计主要任务是完成对需求分析报告中描述现实世界的建模，即用一种数据库模型来实现对现实世界的抽象表达。这种建模与具体的机器、DBMS 无关，是现实世界到信息世界的第一层抽象，所用的数据库模型是用户与数据库设计人员之间进行交流的最重要的某种语言或表示方法。因此，用于表达概念模式的数据库模型一方面应该具有较强的语义表达能力，能够方便、直接地表达实际应用中的各种语义知识，另一方面它还应该简单、清晰、易于用户理解。常用的数据库模型就是实体 - 联系模型（E - R 模型）。

2.2.1 E-R 模型的基本概念

实体 - 联系模型（E - R 模型）是一个面向问题的概念性数据库模型。该模型将现实世界的要求转化成实体、联系、属性等几个基本概念，以及它们之间的两种基本联接关系，并且可以用实体联系图非常直观地表示出来。

E-R 模型的基本元素如下：

- （1） 实体：客观存在并可相互区别的事物称为实体。
- （2） 属性：实体所具有的某一特征称为实体的属性。
- （3） 键：也称关键字或码，唯一标识实体的最小的属性集称为实体的键。
- （4） 联系：现实世界的事物彼此是有联系的，反映在信息世界就是实体之间的联系。

实体间的相互关系称为联系。联系在数据库中的反映是实体集之间存在着这样和那样的联系，这种联系实际上表示了实体集之间的某种函数映射关系。可分为以下三类：

1. 1:1 联系：已知实体集 A 和 B, 若其中每个实体集中任一实体至多与另一实体集中的一个实体有联系，则称为 A 和 B 的联系为 1:1 联系。
2. 1:n 联系：已知实体集 A 和 B, 若 A 中每个实体可与 B 中任意实体有联系，而 B 中每个实体至多与 A 中的一个实体有联系，则称为 A 和 B 的联系为“1 对多联系”，简记为 1:n 联系。
3. m:n 联系：已知实体集 A 和 B, 若其中每个实体集中任一实体可与另一实体集有联系，则称 A 和 B 的联系为“多对多联系”，简记为 m:n 联系。

E-R 图的常用的基本图符如下：

1. 矩形框：表示实体型。
2. 菱形框：表示联系。
3. 椭圆形框：表示实体或联系类型的属性。
4. 直线：联系类型与其所涉及的实体之间用直线连接，实体与实体之间用直线连接。

2.2.2 局部 E-R 图设计

(1) 歌曲和歌手的局部 E-R 图

分析：图 2-3 为歌曲和歌手的局部 E-R 图。从图中可以看出：一首歌曲可以由多位不同的歌手演唱，一位歌手可以演唱多首歌曲。所以二者是多对多的关系，即 $m:n$ 的关系。其中歌曲的主键：歌曲编号；歌手的主键：歌手编号。

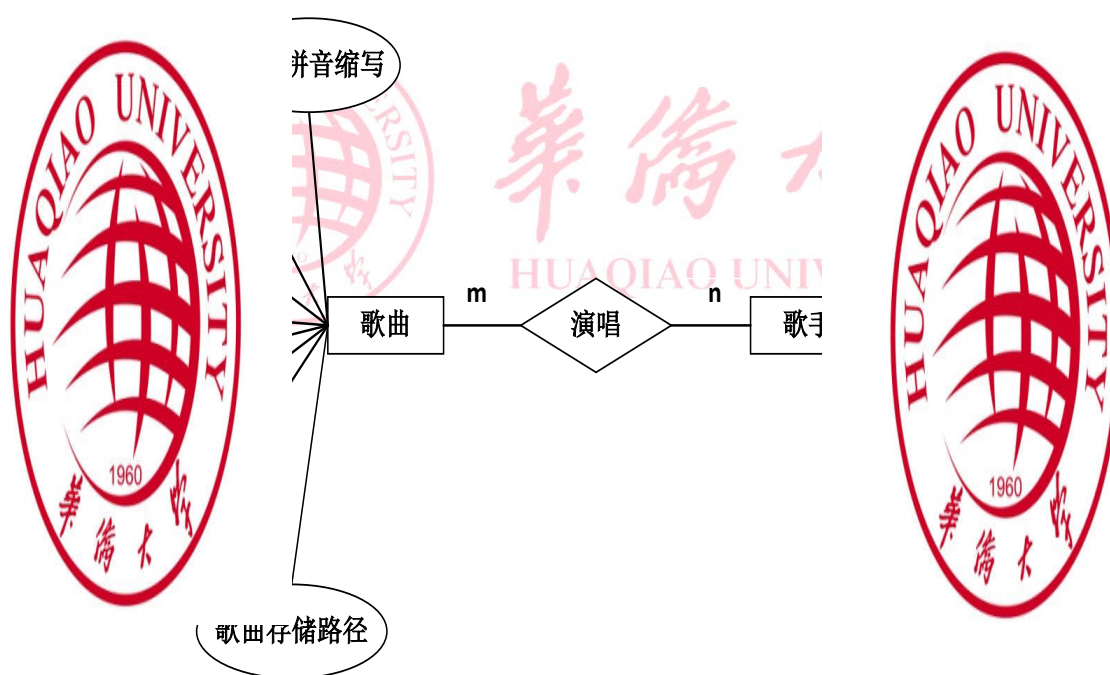


图 2.3 歌曲和歌手的局部 E-R 图

(2) 歌曲和用户的局部 E-R 图

分析：图 2-4 为歌曲和用户的局部 E-R 图。从图中可以看出：一首歌曲可以被多位用户点播，一位用户可以点播多首歌曲。所以二者是多对多的关系，即 $m:n$ 的关系。其中歌曲的主键：歌曲编号；用户的主键：会员编号。

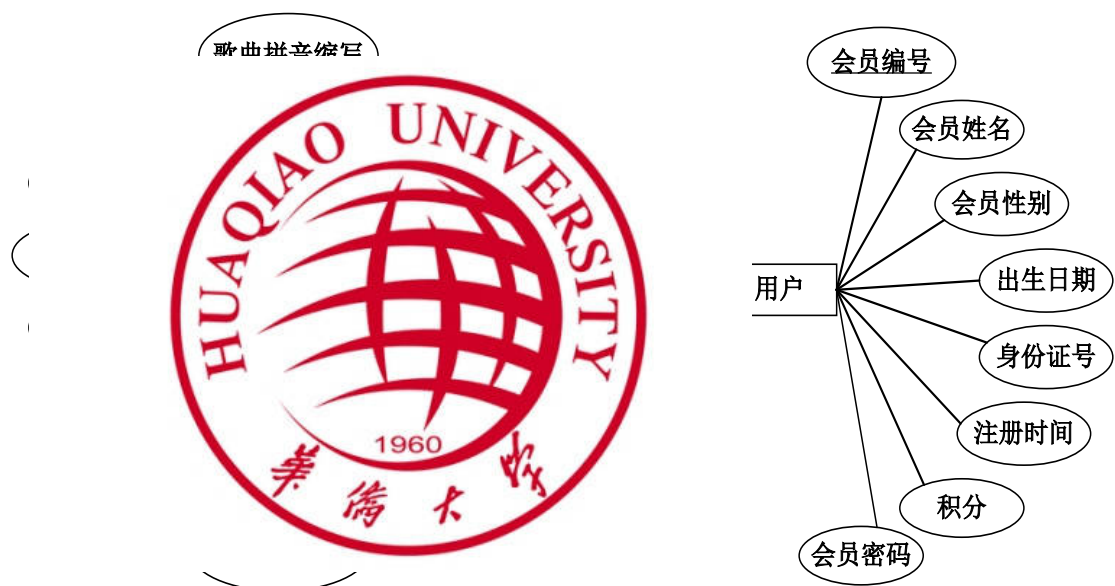


图 2-4 歌曲和用户的局部 E-R 图

(3) 歌曲和管理员的局部 E-R 图

分析：图 2-5 为歌曲和管理员的局部 E-R 图。从图中可以看出：一首歌曲可由多位管理员管理，一名管理员可以管理数据库中所有的歌曲。所以二者是多对多的关系，即 $m:n$ 的关系。其中歌曲的主键：歌曲编号；管理员的主键：管理员编号。

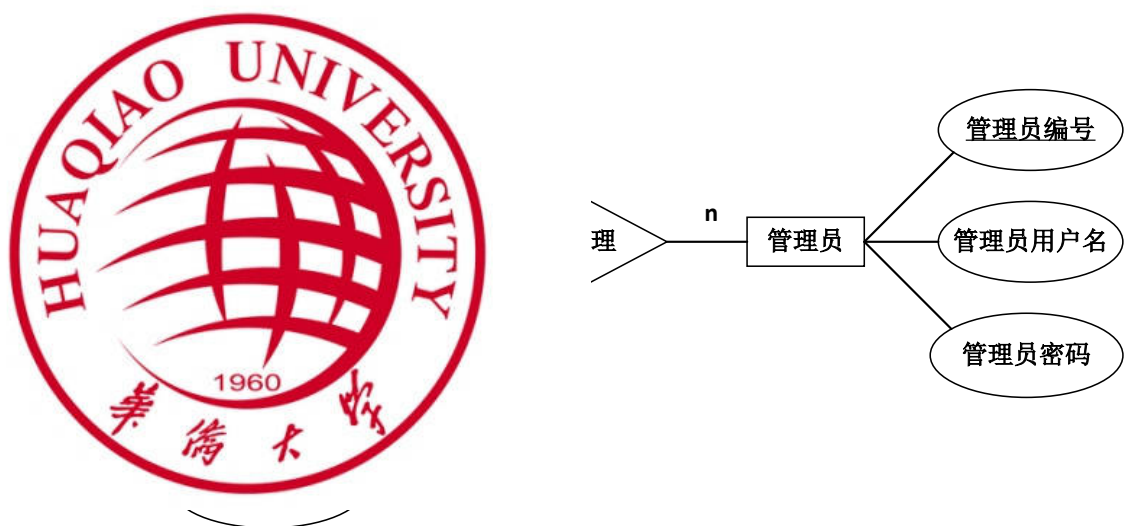


图 2-5 歌曲和管理员的局部 E-R 图

(4) 用户和管理员的局部 E-R 图

分析：图 2-6 为用户和管理员的局部 E-R 图。从图中可以看出：一名用户可由多位管理员管理，一名管理员可以管理数据库中所有的用。所以二者是多对多

的关系，即 $m:n$ 的关系。其中用户的主键：会员编号；管理员的主键：管理员编号。

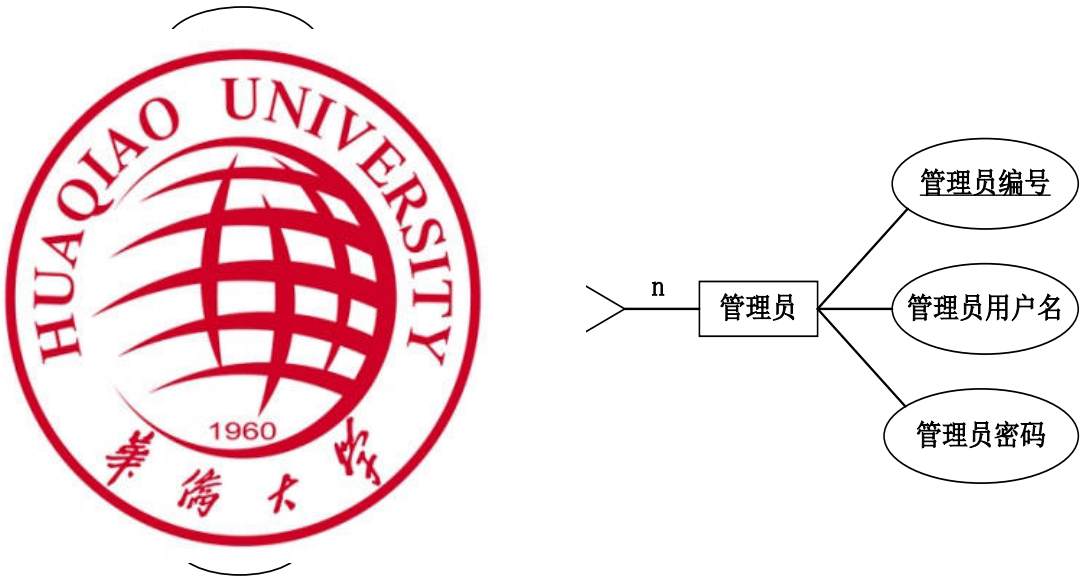


图 2-6 用户和管理员的局部 E-R 图

(5) 用户和歌手的局部 E-R 图

分析：图 2-7 为用户和歌手的局部 E-R 图。从图中可以看出：一名用户可检索多位歌手，一名歌手可被多名用户检索。所以二者是多对多的关系，即 $m:n$ 的关系。其中用户的主键：会员编号；歌手的主键：歌手编号。



图 2-7 用户和歌手的局部 E-R 图

(6) 歌手和管理员的局部 E-R 图

分析：图 2-8 为歌手和管理员的局部 E-R 图。从图中可以看出：一名歌手可由多位管理员管理，一名管理员可管理数据库中所有歌手。所以二者是多对多的关系，即 $m:n$ 的关系。其中歌手的主键：歌手编号；管理员的主键：管理员编号。



图 2-8 歌手和管理员的局部 E-R 图

(7) 超级管理员和管理员的局部 E-R 图

分析：图 2-9 为超级管理员和管理员的局部 E-R 图。从图中可以看出：一名管理员可由多位超级管理员管理，一名超级管理员可管理所有的管理员。所以二者是多对多的关系，即 $m:n$ 的关系。其中超级管理员的主键：超级管理员编号；管理员的主键：管理员编号。

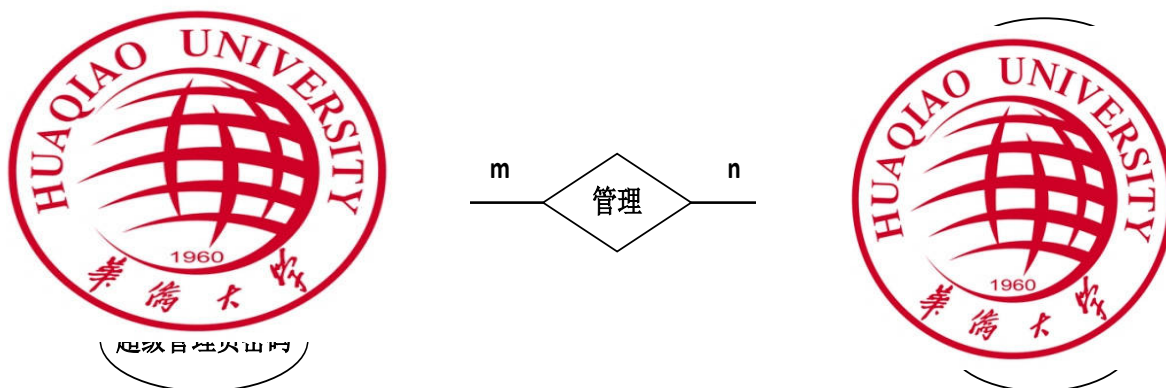


图 2-9 超级管理员和管理员的局部 E-R 图

(8) 歌曲排行榜实体 E-R 图

分析：图 2-10 为歌曲排行榜实体 E-R 图。实体歌曲排行榜中一共有三个属性，分别是歌曲编号、歌曲名称和总点击率，每首歌都具有唯一的歌曲编号，因此将歌曲编号作为歌曲排行榜的主键。



图 2-10 歌曲排行榜实体 E-R 图

(9) 我的歌单实体 E-R 图

分析：图 2-11 为我的歌单实体 E-R 图。实体我的歌单中一共有两个属性，分别是歌曲编号和歌曲名称，歌曲编号作为歌曲排行榜的主键。



图 2-11 我的歌单实体 E-R 图

(10) 后台操作记录实体 E-R 图



图 2-12 后台操作记录实体 E-R 图

2.2.3 全局 E-R 图设计

在设计完所有实体之间的联系后,即局部 E-R 图设计,需将局部 E-R 图合并,在这个过程需要去掉多个实体和属性之间的命名冲突和联系冲突等,从而生成一个完整的、满足应用需求的全局 E-R 图。图 2-13 即为 KTV 点歌系统的全局 E-R 图。通过图 2-13 可以清晰的看出各个实体之间的关系。



图 2-13 KTV 点歌系统的全局 E-R 图

2.3 逻辑模式设计

数据库概念模式设计完成后,数据库的设计进入到逻辑模式设计阶段。这时的数据库设计与具体的机器世界是相关联的。

逻辑模式设计阶段有两大任务:

- 1、按一定的规则将 E-R 图转换成关系模式。
- 2、关系模式的规范化处理。所谓关系模式的规范化处理,就是对关系式进行某种处理,使它满足一定的规范化要求。规范化的主要目的就是

要尽可能地消除关系操作过程中的异常情况。

2.3.1 E-R 图到关系模式的转换

关系模型的逻辑结构是一组关系模式的集合。E-R 图则是由实体、实体的属性和实体之间的联系三个要素组成的。所以将 E-R 图转换为关系模型实际上就是要将实体、实体的属性和实体之间的联系转换为关系模式的集合。

将 E - R 图转换为关系模式的转换规则如下：

规则 1：将 E - R 图中的一个实体集转换为一个同名关系模式。实体集的属性就是关系模式的属性。实体集的键码就是关系模式的主键。

规则 2：由于实体集之间的联系有 3 种类型：1:1、1: n 和 n:m，每一种类型转换都不一样，现就每种类型予以说明。

若两实体集间的联系为 1:1，则可以在两个实体集转换成的两个关系模式中的任意一个关系模式的属性集中加入另一个关系模式的主键和联系自身的属性，由此来完成 1:1 联系到关系模式的转换。

若两实体集间的联系为 1: n，则可以在多端实体集转换成的关系模式中加入 1 端实体集的键码和联系自身的属性，由此来完成 1: n 联系到关系模式的转换。

若两实体集间的联系为 m:n，则将联系转换成一个独立的关系模式，其属性为两端实体集的键码加上联系自身的属性，联系关系模式的主键为复合键，由两端实体集键码组合而成。

根据以上 E-R 模型向关系模型转换规则，对上述的 E-R 图进行转换，转换结果如下（主键用下划线标出，外键用波浪线标出）：

1) 歌手（歌手编号）

每位歌手

所以选择

2) 歌曲（歌曲编号）

编号）

每首歌

所以选择

信息表

3) 用户（用户编号）

会员密

每位用

户，所

4) 管理员



每位管理员的管理员编号都是唯一的，通过属性管理员编号即可对应唯一的一位管理员，所以选择管理员编号作为主键。

- 5) 超级管理员（超级管理员编号、超级管理员用户名、超级管理员密码）

每位超级管理员的编号都是唯一的，通过属性超级管理员编号即可对应唯一的一位超级管理员，所以选择超级管理员编号作为主键。

- 6) 演唱（歌手编号、歌曲编号）

演唱是歌曲和歌手这两个实体之间的联系，联系关系模式的属性是由两端实体集的键码和联系自身的属性构成，所以其属性为歌手编号和歌曲编号。同时，联系关系模式的键码为复合键码，由两端实体集键码以及相关属性组合而成，所以其主键为歌手编号和歌曲编号。以下六个联系关系模式集合的分析与此同理。

- 7) 点播（歌曲编号、会员编号）

- 8) 检索（歌手编号、会员编号）

- 9) 管理（歌曲编号、管理员编号）

2.3.2 关系模式的规范化处理

在数据库逻辑模式设计阶段，完成 E-R 图到关系模式的转换后，就要进入关系模式的规范化处理阶段。这一过程主要研究关系模式内各属性之间的依赖关系，保持属性间好的数据依赖关系，消除导致异常的依赖关系，使得关系模式由低一级的规范模式上升到高一级的规范模式。我们设计的最终目的是使关系模式规范化，所以最低应该达到第三范式。

首先，上述的关系中每一个属性都是不可再分的数据，所以均符合第一范式。

其次，第二范式要求关系模式必须属于第一范式，且每个非主属性都是完全函数依赖于主键。在上述关系中，“歌手”、“歌曲”、“用户”、“管理员”和“超级管理员”都只有一个主键，所以不存在部分依赖关系，属于第二范式。“演唱”、“点播”、“检索”和“管理”等七个关系中，均有两个主键，但是这七个关系均不存在非主属性，所以不存在部分函数依赖，也属于第二范式。

最后，第三范式要求关系模式必须属于第一范式，且每个非主属性都不传递函数依赖于主键。在上述关系中，“歌手”、“歌曲”、“用户”、“管理员”和“超级管理员”都不存在传递函数依赖关系，属于第三范式。“演唱”、“点播”、“检索”和“管理”等七个关系中，均有两个主键，但是这七个关系均不存在非主属性，所以不存在传递函数依赖，也属于第三范式。

第3章 数据库的实施与 VB.NET 界面开发

数据库实施部分首先运用 SQL Server 建立 KTV 系统管理数据，再将各关系模型转化为实际的 SQL 数据表。VB.NET 界面开发部分主要是利用 VB.NET 进行 KTV 前台点歌系统和后台管理系统人机交互界面的设计与制作。

3.1 数据库基本表设计

表 3-1 歌手信息表

字段名	数据类型	主键/允许空
歌手编号	INT	PRIMARY KEY

表 3-2 歌曲信息表

字段名	数据类型	主键/允许空
歌曲编号	INT	PRIMARY KEY

表 3-3 用户信息表

字段名	数据类型	主键/允许空
会员编号	INT	PRIMARY KEY

表 3-4 管理员信息表

字段名	数据类型	主键/允许空
管理员编号	INT	PRIMARY KEY
管理员用户名	VARCHAR(20)	Null
管理员密码	VARCHAR(30)	Null

表 3-5 超级管理员信息表

表 3-6 歌曲排行榜

字段名	数据类型	主键/允许空

表 3-7 我的歌单

表 3-8 后台操作记录

字段名	数据类型	主键/允许空

3.2 登录模块界面设计及程序分析

登录模块由登录页面、用户登录页面和管理员登录页面三个窗口组成。登录页面在 KTV 点歌系统中起到了欢迎界面的作用，以构建一个美观舒适的人机交互环境，如图 3-1 所示。在该窗口有三个功能按钮，分别是用户登录、管理员登录和退出系统。其中点击用户登录按钮可以进入用户登录界面，即进入前台点歌系统，如图 3-2 所示；点击管理员登录按钮可以进入管理员登录界面，即进入后台管理系统，如图 3-3 所示；点击退出，则系统会关闭。



图 3-1 登录页面



图 3-2 用户登录页面

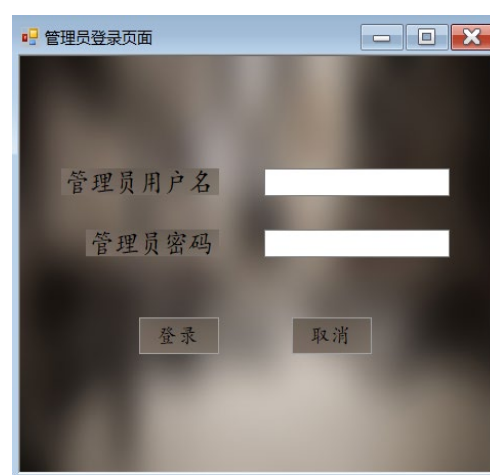


图 3-3 管理员登录页面

在用户登录页面和管理员登录页面，用户和管理员必须输入正确用户名和密码才能成功登录前台点歌系统，如图 3-4 和图 3-5 所示。若点击取消则会返回登录页面。



图 3-4 用户登录成功



图 3-5 管理员登录成功

为了提高系统的安全性，当用户或者管理员输入错误的用户名或密码的次数超过两次时，我们认为正在进行登录操作的人员不是我们的会员用户和系统管理员，系统执行退出指令，禁止该人员登录。如图 3-6 和图 3-7 所示，管理员登录页面的设计在这点上与用户登录页面设计一致。

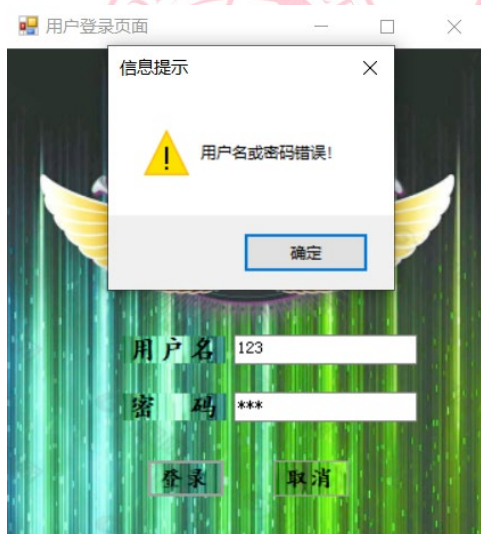


图 3-6 错误输入用户名或密码

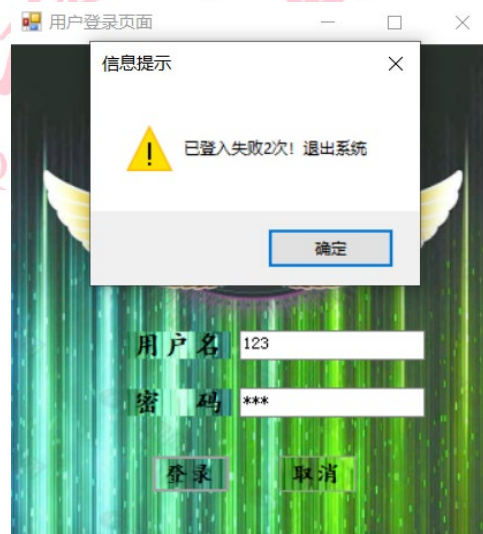


图 3-7 登录失败两次

```
Dim n As Integer = 0
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
    '定义SqlConnection对象/windows身份验证模式
    Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data
Source = .;Integrated Security = True;")
    '定义SqlDataAdapter对象
    Dim mysql As String = "select * from 会员信息表 where 会员姓名='" +
TextBox_用户名.Text + "' and 会员密码='" + TextBox_密码.Text + "'"
```

```

Dim myadapter As New SqlDataAdapter(mysql, myConnection)
' 定义DataSet对象, 用mysql命令串执行的结果填充数据集mydataset
Dim mydataset As New DataSet
myadapter.Fill(mydataset, "会员信息表")
If mydataset.Tables("会员信息表").Rows.Count = 0 Then
    n += 1
    If n < 2 Then
        MsgBox("用户名或密码错误!", MsgBoxStyle.OkOnly +
MsgBoxStyle.Exclamation, "信息提示")
        TextBox_用户名.Text = ""
        TextBox_密码.Text = ""
        TextBox_用户名.Focus() ' 设置输入焦点
    Else
        MsgBox("已登入失败2次! 退出系统", MsgBoxStyle.OkOnly +
MsgBoxStyle.Exclamation, "信息提示")
        Me.Close()
        登录页面.Close()
    End If
Else
    MsgBox("欢迎光临!")
    用户操作界面.Show()
    Me.Close()
End If
End Sub

```

3.3 前台点歌模块界面设计及程序分析

会员用户在用户登录界面登录成功之后便可进入前台点歌模块的用户操作界面, 在用户操作界面用户可以进行开始点歌、积分查询、密码修改和退出账号四项操作, 如图 3-8 所示。



图 3-8 用户操作界面

在前台点歌系统模块中进行操作时将会涉及到歌曲信息表、歌手信息表、歌曲排行榜和我的歌单这四个 SQL 数据库关系表。

3.3.1 开始点歌窗口

用户点击开始点歌便会进入开始点歌窗口，该窗口是 KTV 点歌系统前台点歌模块的主要用户操作界面，如图 3-9 所示。



图 3-9 开始点歌窗口

(1) 歌曲搜索

在开始点歌窗口，用户可以通过输入关键字实现数字点歌（通过检索歌曲编号进行歌曲搜索）、歌星点歌（通过检索歌手姓名进行歌曲搜索）、拼音点歌（通过检索歌曲拼音缩写进行歌曲搜索）和歌名点歌（通过检索歌曲名称进行歌曲搜索）。同时，这四种歌曲搜索方式均可以实现模糊搜索。

1) 数字点歌

```
If 关键字.Text = "" Then
    MsgBox("请输入关键字进行搜索！", MsgBoxStyle.0kOnly, "信息提示")
Else
    Dim mycmd As New SqlCommand
        ' 定义SqlConnection对象/windows身份验证模式
    Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data
Source = .;Integrated Security = True;")
    Dim mysql2 As String = "SELECT * FROM 歌曲信息表 WHERE 歌曲编号
LIKE '%" & 关键字.Text & "%'"
    mycmd.Connection = myConnection ' 指定mycmd对象的数据连接
```

```

' 定义mycmd对象要执行的SQL语句
mycmd.CommandText = mysql2
' 定义myadapter2对象, 用于满足一定条件的数据填充
Dim myadapter2 As New SqlDataAdapter(mycmd)
' 定义数据集mydataset
Dim mydataset2 As New DataSet
' 填充数据
myadapter2.Fill(mydataset2, "歌曲信息表")
Me.DataGridView1.DataSource=mydataset2.Tables("歌曲信息表")
End If

```

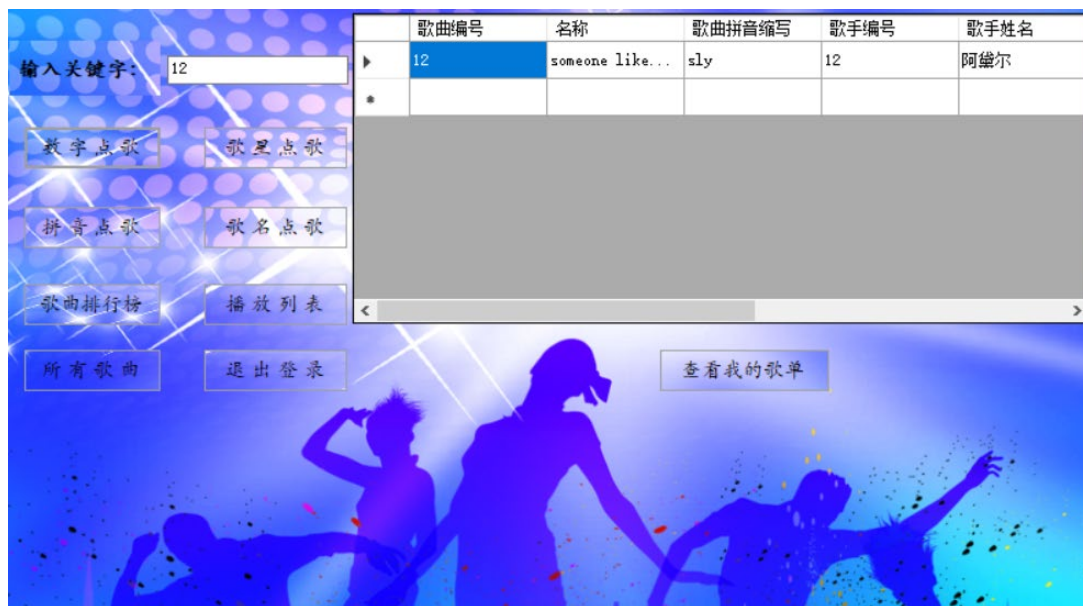


图 3-10 数字搜索

2) 歌星搜索

```

If 关键字.Text = "" Then
    MsgBox("请输入关键字进行搜索!", MsgBoxStyle.OkOnly, "信息提示")
Else
    Dim mycmd As New SqlCommand
    ' 定义SqlConnection对象/windows身份验证模式
    Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data
Source = .;Integrated Security = True;")
    Dim mysql1 As String = "SELECT * FROM 歌曲信息表 WHERE 歌手编号 in (SELECT 歌手
编号 FROM 歌手信息表 WHERE 姓名 Like '%" & 关键字.Text & "%')"
    mycmd.Connection = myConnection ' 指定mycmd对象的数据连接
    mycmd.CommandText = mysql1 ' 定义mycmd对象要执行的SQL语句
    Dim myadapter1 As New SqlDataAdapter(mycmd)
' 定义myadapter2对象, 用于满足一定条件的数据填充
    Dim mydataset1 As New DataSet
    myadapter1.Fill(mydataset1, "歌曲信息表")

```

```

Me.DataGridView1.DataSource=mydataset1.Tables("歌曲信息表")
End If

```



图 3-11 歌星搜索

3) 拼音搜索

```

If 关键字.Text = "" Then
    MsgBox("请输入关键字进行搜索!", MsgBoxStyle.OkOnly, "信息提示")
Else
    Dim mycmd As New SqlCommand
' 定义SqlConnection对象/windows身份验证模式
    Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data
Source = .;Integrated Security = True;")
    Dim mysql3 As String = "SELECT * FROM 歌曲信息表 WHERE 歌曲拼音缩写
LIKE '%" & 关键字.Text & "%'"
    mycmd.Connection = myConnection
' 定义mycmd对象要执行的SQL语句
    mycmd.CommandText = mysql3
    Dim myadapter3 As New SqlDataAdapter(mycmd)
    Dim mydataset3 As New DataSet
    myadapter3.Fill(mydataset3, "歌曲信息表")
    Me.DataGridView1.DataSource=mydataset3.Tables("歌曲信息表")
End If

```



图 3-12 拼音搜索

4) 歌名搜索

```

If 关键字.Text = "" Then
    MsgBox("请输入关键字进行搜索!", MsgBoxStyle.OkOnly, "信息提示")
Else
    Dim mycmd As New SqlCommand
    ' 定义SqlConnection对象/windows身份验证模式
    Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data
Source = .;Integrated Security = True;")
    Dim mysql4 As String = "SELECT * FROM 歌曲信息表 WHERE 名称 LIKE
    "

    mycmd.Connection = myConnection
    mycmd.CommandText = mysql4
    Dim myadapter4 As New SqlDataAdapter(mycmd)
    Dim mydataset4 As New DataSet
    myadapter4.Fill(mydataset4, "歌曲信息表")
    Me.DataGridView1.DataSource =
End If

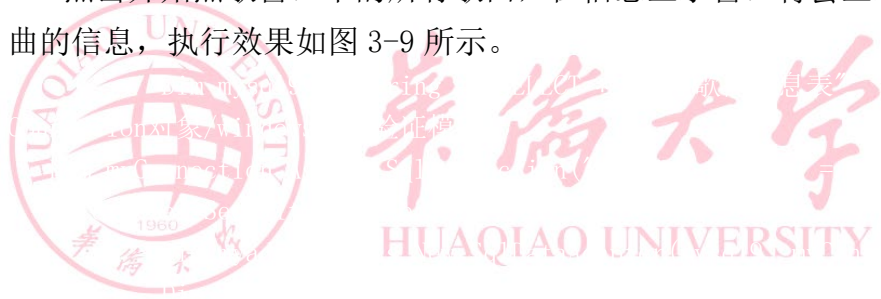
```




图 3-13 歌名搜索

5) 所有歌曲

点击开始点歌窗口中的所有歌曲，在信息显示窗口将会显示所有歌曲的信息，执行效果如图 3-9 所示。



(2) 歌曲排行榜

点击歌曲排行榜，则用户可以进入歌曲排行榜界面查看歌曲总点击率信息，如图 3-14 所示，点击返回按钮则会返回开始点歌界面。

当用户将选中的歌曲添加进我的歌单中时，歌曲排行榜中的总点击率便会加 1，同时将会更新到 SQL 数据库关系表中的歌曲排行榜表。SQL 数据库关系表的歌曲排行榜表的排序是按照总点击率的降序排列，所以总点击率越大则在开始点歌窗口中的歌曲排行榜中的位置越靠前，继而实现歌曲热度排名功能。

歌曲排行榜			
	歌曲编号	歌曲名称	总点击率
▶	1	说好不哭	144
	13	年少有为	135
	5	成都	131
	17	浪子回头	128
	7	我和我的祖国	123
	30	你不知道的事	99

刷新

返回

图 3-14 歌曲排行榜

点击刷新按钮后歌曲排行榜的信息将会刷新，用户便可查看到最新的总点击率。如图 3-15 和图 3-16 所示。

’将所选歌曲添加进我的歌单





	歌曲编号	歌曲名称	总点击率
	19	唯一	1
	16	勇气	1
	11	山丘	1
	21	听海	0
	26	一起走过的日子	0

图 3-15 点击歌曲前的总点击率



	歌曲编号	歌曲名称	总点击率
	19	唯一	2
	16	勇气	2
	11	山丘	2
	8	岁月神偷	1
	21	听海	1
	26	一起走过的日子	1

图 3-16 点击歌曲后的总点击率

(3) 我的歌单

点击查看我的歌单会在歌曲信息显示窗口显示出我的歌单表的所有信息，如图 3-17 所示。



图 3-17 查看我的歌单

在开始点歌窗口完成歌曲搜索之后可以直接在歌曲信息显示窗口点击想要点播的歌曲，此操作可以直接将被点击的歌曲添加进我的歌单。在该操作过程中，系统会询问用户是否要添加所选歌曲到我的歌单，若用户选择否则歌曲不会被添加。当某一首歌曲被用户添加进我的歌单中时，该歌曲在歌曲排行榜中的总点击率将会加 1。图 3-15 是未添加选定歌曲至我的歌单时的歌曲排行榜界面，图 3-16 是添加了选定歌曲至我的歌单后的歌曲排行榜界面。



图 3-18 添加歌曲后的我的歌单

另外，在歌曲排行榜界面的歌曲信息窗口点击歌曲也可以直接将歌曲添加进我的歌单，并且会询问用户是否确定要添加该歌曲，在确定添加后，相应歌曲的总点击率将会加 1，如图 3-19 和图 3-20 所示。

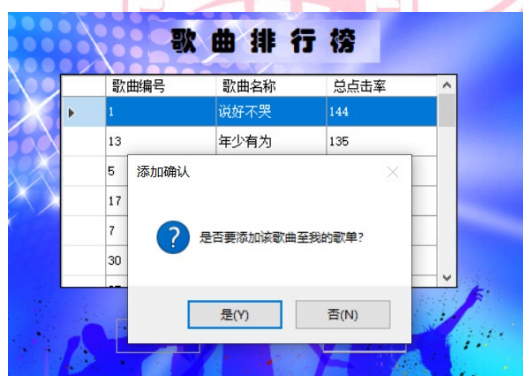


图 3-19 确认是否添加歌曲



图 3-20 添加成功后总点击率加 1

```
Private Sub DataGridView1_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView1.CellClick
    ' 将所选歌曲添加进我的歌单
    Dim song As Integer = 0
    Dim count As Integer = 0
    ' 定义SqlConnection对象/windows身份验证模式
```

```
Dim mysqlG As String = "INSERT INTO 我的歌单
```

```
' 歌曲排行榜总点击率
```

```
)
```

```
End If
```

```
End Sub
```

(4) 播放列表

点击播放列表将会进入播放列表界面。该界面显示的歌曲信息是我的歌单信息，歌曲的播放是由 Windows Media Player 实现。该界面可以实现播放选中的歌曲、暂停播放、歌曲重播、调整音量、倍速播放、将所选歌曲移出我的歌单、清空我的歌单和刷新我的歌单等功能，进行清空和移除操作是系统会询问用户是否要进行该操作，用户确认之后系统才会执行，如图 3-21 和图 3-22 所示。



图 3-21 播放列表界面

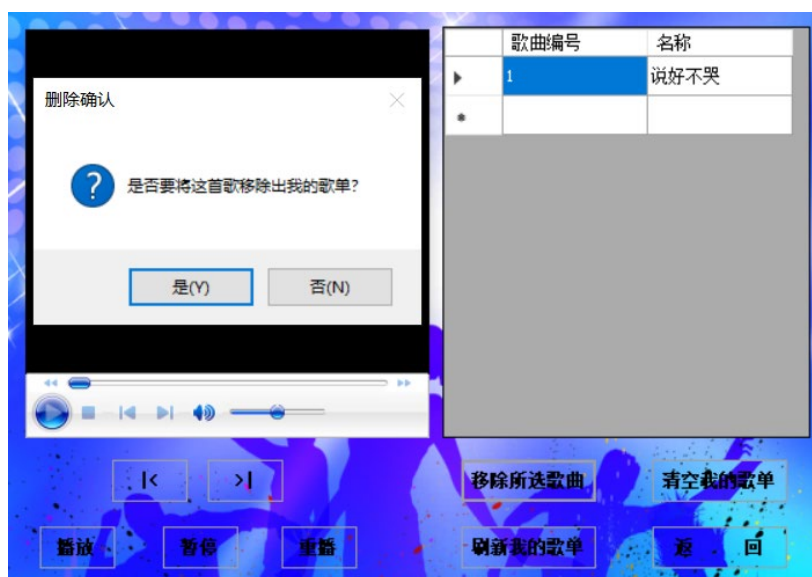


图 3-22 移除歌曲确认信息提示

```

' 定义SqlConnection对象/windows身份验证模式
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source
= .;Integrated Security = True;")
' 播放歌曲
    Dim song1 As String = DataGridView1.Item("名称",
DataGridView1.CurrentCell.RowIndex).Value
    Dim mysql1 As String = "SELECT 歌曲存储路径 FROM 歌曲信息表 WHERE 名称 = '" &
song1 & "'"
    Dim mycmd1 As New SqlCommand(mysql1, myConnection)
    myConnection.Open()
    Dim myreader As SqlDataReader = mycmd1.ExecuteReader
    While myreader.Read
        AxWindowsMediaPlayer1.URL = myreader("歌曲存储路径")
    End While
    myConnection.Close()
' 将歌曲移除出我的歌单

```

```
End If  
' 清空我的歌单
```

```
' 刷新我的歌单
```

```
' 暂停播放  
AxWindowsMediaPlayer1.Ctlcontrols.pause()  
' 重播  
AxWindowsMediaPlayer1.Ctlcontrols.stop()  
AxWindowsMediaPlayer1.Ctlcontrols.play()
```

3.3.2 用户积分查询和密码修改

在用户操作界面，用户可以很便捷的查询个人账号的积分以及对账号密码进行修改，如图 3-23 和图 3-24 所示。

```
' 积分查询
```

```
MsgBox("您的账号积分：" & integral, MsgBoxStyle.DefaultButton1, "账号积分")
End If
```



图 3-23 用户积分查询



图 3-24 用户密码修改

' 密码修改

```
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source  
= .;Integr
```

End If

3.4 后台管理模块界面设计及程序分析

在登录页面选择管理员登录，并使用正确的管理员用户名和密码登录后便可进入 KTV 后台管理系统。在后台管理系统中，管理员可以对用户信息、歌曲信息和歌手信息进行管理，也可以修改管理员账号的密码。若管理员具有超级管理员的权限，则可以使用超级管理员账号和密码获取超级管理员权限，之后便可对系统的普通管理员信息和超级管理员信息进行管理。KTV 后台管理系统操作界面如图 3-25 所示。



图 3-25 KTV 后台管理系统操作界面

3.4.1 用户信息管理

用户信息管理操作界面如图 3-26 所示。在不输入任何用户信息时，管理员可以通过点击查询用户信息查询所有用户的信息。填入会员编号或者会员姓名，可以查询指定用户的信息，同时还可以实现模糊搜索；填入完整的用户信息后，可以点击添加用户将新的用户添加进会员信息表，若填入的信息不完整则系统会提示填入完整的用户信息，如图 3-27 所示；若要删除某用户的信息时可以只填入会员编号或者会员姓名，如图 3-28 所示；更新某用户时必须完整的填入用户信息，然后再对用户信息进行更改。



图 3-26 用户信息管理操作界面

’ 查询用户信息

```
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source = .;Integrated Security = True;")
```

```
If 会员编号.Text = "" And 会员姓名.Text = "" Then
```

```
’ 定义SqlDataAdapter对象
```



End If

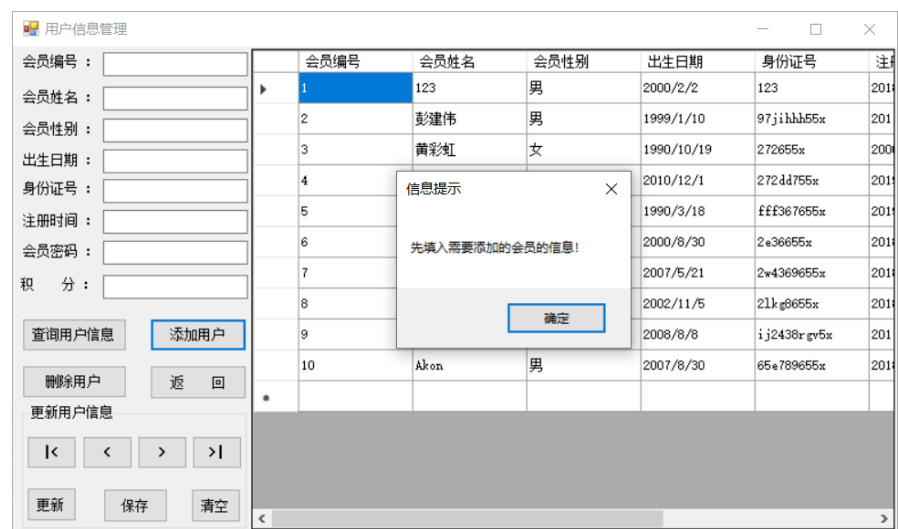


图 3-27 添加用户时的系统提醒

’添加用户信息

```
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source  
= .;Integrated Security = True;")  
Dim mydataset3 As New DataSet  
Dim mysql3 As String = "SELECT *
```

End If



图 3-28 删除会员用户

’删除用户信息

```
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source
= .;Integrated Security = True;")
Dim mydataset4 As New DataSet
Dim mysql4 As String = "DELETE FROM 会员信息表 WHERE

If 会员编号.Text = "" Then
    MsgBox("先填入需要删除的会员的信息!", MsgBoxStyle.OkOnly, "信息提示")
ElseIf MsgBox("是否要删除该会员信息?", MsgBoxStyle.YesNo +
MsgBoxStyle.Question, "删除确认") = MsgBoxResult.Yes Then
    Dim myadapter4 As New SqlDataAdapter(mysql4, myConnection)
    myadapter4.Fill(mydataset4, "会员信息表")
    MsgBox("删除成功!")
    会员编号.Text = "" : 会员姓名.Text = ""
End If
```

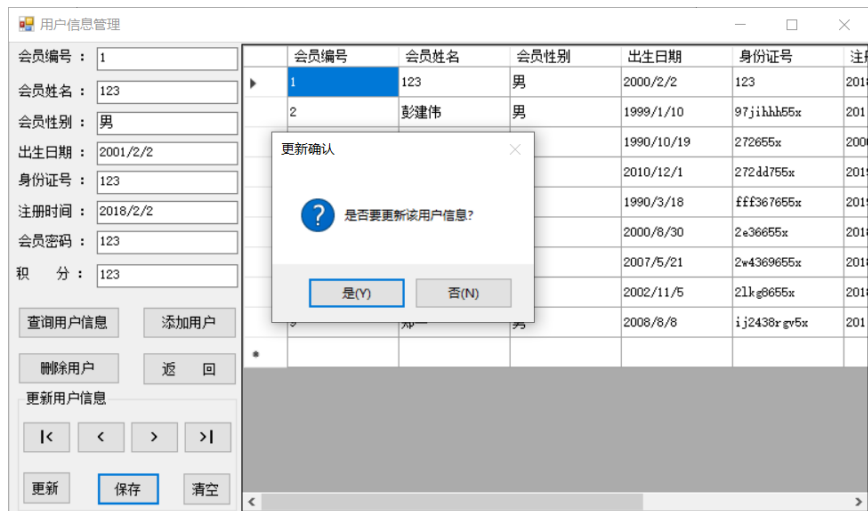


图 3-29 更新会员信息

更新用户信息

’ 定义绑定数据源对象

```
Dim mybind As New BindingSource
```

```
Dim mysql_change As String = "SELECT * FROM 会员信息表"
```

```
Dim myadapter_change As New SqlDataAdapter(mysql_change, myConnection)
```

```
Dim mydataset_change As New DataSet
```

```
Private Sub Button_用户更新_Click(sender As Object, e As EventArgs) Handles Button_
    用户更新.Click
```

’ 生成数据集

```
myadapter_change.Fill(mydataset_change, "会员信息表")
```

’ 为绑定数据源对象设置数据源

```
mybind.DataSource = mydataset_change
```

```
mybind.DataMember = "会员信息表"
```

’ 为控件设置要绑定的数据源中对应表的属性

```
会员编号.DataBindings.Add(New Binding("Text", mybind, "会员编号", True))
```

)

```
积分.DataBindings.Add(New Binding("Text", mybind, "积分", True))
```

```
End Sub
```

```
Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles 保存修
    改.Click
```

```
If MsgBox("是否要更新该用户信息?", MsgBoxStyle.YesNo + MsgBoxStyle.Question, "
    更新确认") = MsgBoxResult.Yes Then
```

```
    mybind.EndEdit()
```

```
    Dim cmd As New SqlCommandBuilder(myadapter_change)
```

```

myadapter_change.Update(mydataset_change.Tables("会员信息表"))
MsgBox("更新成功!")
End If
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles 首位.Click
mybind.MoveFirst()
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles 前一个.Click
mybind.MovePrevious()
End Sub
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles 下一个.Click
mybind.MoveNext()
End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles 末尾.Click
mybind.MoveLast()
End Sub
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
会员编号.Text = "" : 会员姓名.Text = ""
会员性别.Text = "" : 出生日期.Text = ""
身份证号.Text = "" : 注册时间.Text = ""
会员密码.Text = "" : 积分.Text = ""
End Sub

```

3.4.2 歌曲信息管理和歌手信息管理

歌曲信息管理和歌手信息管理在实现上与用户信息管理大致相同。这两个模块所要实现的功能是对歌曲信息和歌手信息进行增删改查。



图 3-30 歌曲信息管理操作界面

’ 查询歌曲信息

```
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source  
= .;Integrated Security = True;")  
If 歌曲编号.Text = "" And 名称.Text = "" Then  
    Dim mysql1 As String = "SELECT * FROM 歌曲信息表"  
    Dim myadapter1 As New SqlDataAdapter(mysql1, myConnection)  
    Dim mydataset1 As New DataSet  
    myadapter1.Fill(mydataset1, "歌曲信息表")  
    Me.DataGridView1.DataSource = mydataset1.Tables("歌曲信息表")  
Else  
    ’ 定义SqlCommand对象  
    Dim mycmd As New SqlCommand  
    If 歌曲编号.Text <> "" Then
```





華僑大學
HUAQIAO UNIVERSITY



華僑大學
HUAQIAO UNIVERSITY

End Sub



图 3-31 歌手信息管理操作界面

' 查询歌手信息

```
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source
= .;Integrated Security = True;")
```

```
If 歌手编号.Text = "" And 姓名.Text = "" Then
```

```
    ' 定义SqlDataAdapter对象
```

```
    Dim mysql1 As String = "SELECT * FROM 歌手信息表"
```

```
    Dim myadapter1 As New SqlDataAdapter(mysql1, myConnection)
```

```
    ' 定义DataSet对象, 用mysql命令串执行的结果填充数据集mydataset
```

```
    Dim mydataset1 As New DataSet
```

```
    myadapter1.Fill(mydataset1, "歌手信息表")
```

```
Me.DataGridView1.DataSource = mydataset1.Tables("歌手信息表")
Else
    ' 定义SqlCommand对象
    Dim mycmd As New SqlCommand
    If 歌手编号.Text <> "" Then
```



华侨大学
HUAQIAO UNIVERSITY



華僑大學
HUAQIAO UNIVERSITY


```

歌手编号.Text = "" : 姓名.Text = ""
拼音缩写.Text = "" : 性别.Text = ""
出生日期.Text = "" : 国籍.Text = ""
End Sub

```

3.4.3 管理员密码修改

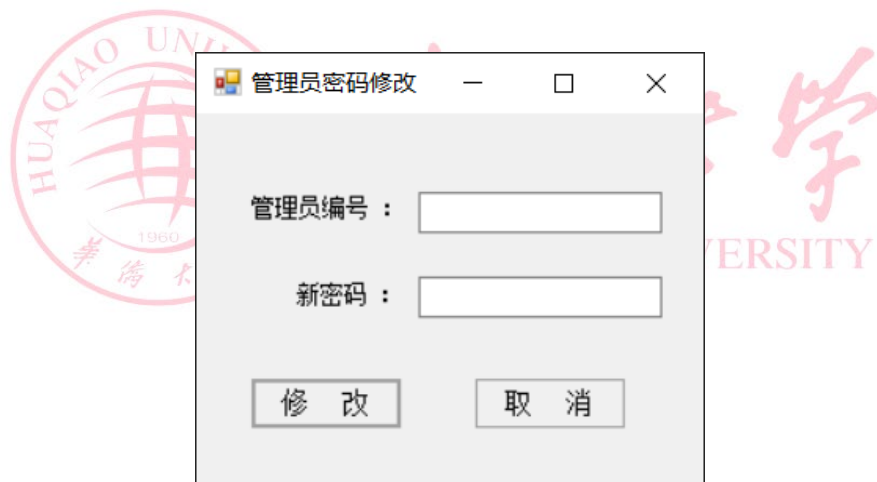


图 3-32 管理员密码修改

```

' 定义SqlConnection对象/windows身份验证模式
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source
= .;Integrated Security = True;")
If 管理员编号.Text = "" Then
    MsgBox("先填入管理员编号!", MsgBoxStyle.OkOnly, "信息提示")
Else
    ' 定义DataSet对象, 用mysql命令串执行的结果填充数据集mydataset
    Dim mydataset As New DataSet
    ' 定义要执行的SQL指令

```

```

End If
End If

```

3.4.4 超级管理员

1) 超级管理员登录页面

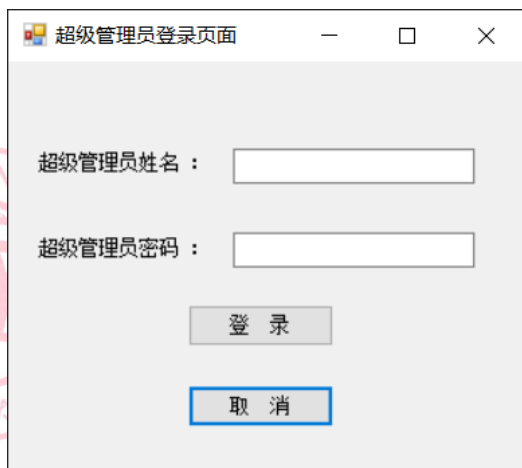


图 3-33 超级管理员登录页面

```

' 定义SqlConnection对象/windows身份验证模式
Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source
= .;Integrated Security = True;")
Dim n As Integer = 0
' 定义SqlDataAdapter对象
Dim mysql As String = "SELECT * FROM
                        "
Dim myadapter As New SqlDataAdapter(mysql, myConnection)
' 定义DataSet对象, 用mysql命令串执行的结果填充数据集mydataset
Dim mydataset As New DataSet
myadapter.Fill(mydataset, "超级管理员信息表")
If mydataset.Tables("超级管理员信息表").Rows.Count = 0 Then
    n += 1
    If n < 2 Then
        MsgBox("超级管理员姓名或密码错误!", MsgBoxStyle.OkOnly +
MsgBoxStyle.Exclamation, "信息提示")

```

```

        姓名.Text = ""
        密码.Text = ""
        姓名.Focus() '设置输入焦点
    Else
        MsgBox("已登入失败2次！退出系统", MsgBoxStyle.OkOnly +
MsgBoxStyle.Exclamation, "信息提示")
        Me.Close()
    End If
Else
    MsgBox("登录成功！")
    管理员信息管理.Show()
    Me.Close()
End If

```

2) 管理员信息管理操作界面



图 3-34 管理员信息管理操作界面

' 普通管理员信息查询

' 定义SqlConnection对象/windows身份验证模式

```

Dim myConnection As New SqlConnection("Initial Catalog = KTV; Data Source
= .;Integrated Security = True;")

```

```

If 管理员编号.Text = "" And 管理员姓名.Text = "" Then

```

' 定义SqlDataAdapter对象

```

Dim mysql1 As String = "SELECT * FROM 管理员信息表"

```

```

Dim myadapter1 As New SqlDataAdapter(mysql1, myConnection)

```

' 定义DataSet对象，用mysql命令串执行的结果填充数据集mydataset

```

Dim mydataset1 As New DataSet

```

```

myadapter1.Fill(mydataset1, "管理员信息表")

```

```

Me.DataGridView1.DataSource = mydataset1.Tables("管理员信息表")

```

Else

'定义SqlCommand对象

Dim mycmd As New SqlCommand



华侨大学
HUAQIAO UNIVERSITY



華僑大學
HUAQIAO UNIVERSITY



華僑大學
HUAQIAO UNIVERSITY



華僑大學
HUAQIAO UNIVERSITY

总结

通过四周的课程设计，基本完成了 KTV 点歌系统的设计。前台点歌可以实现按数字点歌、拼音点歌、歌星点歌和歌名点歌，可以实现歌曲的媒体播放功能，同时也可以查看歌曲排行榜。后台信息管理可以实现用户、歌曲、歌手和管理员的信息管理。但与此同时，此次的设计仍存在许多的不足，比如无法顺序播放歌单中的歌曲，需要将所有的歌曲信息导入歌曲排行榜才能完整实现该功能。

同时，通过这次课程设计我巩固了 SQL Server 数据库和 VB.NET 界面开发的知识，通过实践学习到很多课本上学不到的知识，学习到了很多解决实际问题的经验。也让我深刻的意识到在做数据库设计时，必须包含需求分析、概念模式设计、逻辑模式设计、数据库实施、数据库运行和维护 5 个阶段。

此次设计的系统很简洁也很微小，在设计过程中简化了非常多的步骤，在调试过程中，总会想到更多符合实际需求的条件去充实完善系统，也知道自己当初设计的时候考虑略欠周全，也明白了实际应用的系统是需要不断更新、维护，是需要许多人的智慧和努力。



参考文献

- [1] 龚沛曾, 袁科萍等编著. 数据库原理及应用[M]. 北京: 高等教育出版社
- [2] 梁玉英, 江涛等编著. SQL Server 数据库设计与项目实践[M]. 北京: 清华大学出版社
- [3] 李春葆, 曾平, 赵丙秀编著. 数据库系统开发教程-基于 SQL Server 2005+VB.NET 2005[M]. 北京: 清华大学出版社
- [4] 臧玉琴, 腾跃编著. Visual Basic 界面、多媒体操作系统程序设计[M]. 北京: 人民邮电出版社
- [5] 蔡宏编著. Visual Basic.NET 案例开发集锦[M]. 北京: 电子工业出版社
- [6] 冯小燕, 刘斌等编著. Visual Basic 项目开发案例精粹[M]. 北京: 电子工业出版社
- [7] 李丹, 赵占坤, 丁宏伟, 史彦芳. SQL Server 2005 数据库管理与开发实用教程[M]. 北京: 机械工业出版社, 2009.11
- [8] 姚永一, SQL Server 2008 数据库技术实用教程, 北京: 电子工业出版社, 2010
- [9] 壮志剑, 数据库原理与 SQL Server, 北京: 高等教育出版社, 2008
- [10] 何玉洁, 梁琦, 数据库原理与应用(第二版), 北京: 机械工业出版社, 2011
- [11] 张志强, 孙福兆, 余健. SQL Server 2005 课程设计案例精编. 北京: 清华大学出版社, 2005
- [12] 张丽萍, 汤涌涛. Visual Basic 2010 从入门到精通[M]. 北京: 清华大学出版社
- [13] 王珊, 萨师煊. 数据库系统概论(第四版)[M]. 北京: 高等教育出版社
- [14] 郑阿奇. Visual Basic 教程[M]. 北京: 清华大学出版社, 2010
- [15] 段兴. Visual Basic 数据库实用程序设计 100 例[M]. 北京: 人民邮电出版社
- [16] 王运坚. Visual Basic 6.0 应用指南[M]. 北京: 人民邮电出版社
- [17] 曹军生. SQL Server 2000 实用教程[M]. 北京: 北京理工大学出版社