

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** | **2017** |
| **学 号** | **1715321032** |
| **姓 名** | **周邵嘉** |
| **指导老师** | **黄彩虹、李平** |
|  | |
| **2021年 1 月8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Python以及跨平台界面设计软件Qt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。使用QtDesigner完成基本界面的构建，使用Python对来处理界面和数据库的接口，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，SQL Server，Python

目录

[引言 1](#_Toc61013279)

[第一章 绪论 2](#_Toc61013280)

[1.1设计背景及意义 2](#_Toc61013281)

[1.2数据库设计意义 2](#_Toc61013282)

[1.3数据库设计内容 2](#_Toc61013283)

[第二章 KTV点歌系统的设计 4](#_Toc61013284)

[2.1需求分析 4](#_Toc61013285)

[2.1.1信息要求分析 4](#_Toc61013286)

[2.1.2处理要求分析 4](#_Toc61013287)

[2.2概念模式设计 5](#_Toc61013288)

[2.2.1 E-R模型的基本概念 6](#_Toc61013289)

[2.2.2局部E-R图设计 6](#_Toc61013290)

[2.3逻辑模式设计 9](#_Toc61013291)

[2.3.1 E-R图到关系模型中的转换 10](#_Toc61013292)

[第三章 数据库设计 12](#_Toc61013293)

[3.1表的建立 12](#_Toc61013294)

[3.1.1歌曲信息表 12](#_Toc61013295)

[3.1.2歌手信息表 13](#_Toc61013296)

[3.1.3管理员信息表 13](#_Toc61013297)

[3.1.4会员信息表 14](#_Toc61013298)

[第四章 Python-Qt界面设计 15](#_Toc61013299)

[4.1 界面实现逻辑 15](#_Toc61013300)

[4.2模块剖析 15](#_Toc61013301)

[4.2.1 SQL连接方法 15](#_Toc61013302)

[4.2.2 登录界面以及处理函数 16](#_Toc61013303)

[4.2.3 用户初始界面 18](#_Toc61013304)

[4.2.4 点歌界面 19](#_Toc61013305)

[4.2.5 个人信息和排行榜界面 19](#_Toc61013306)

[4.2.6 管理员管理界面 21](#_Toc61013307)

[4.2.7 修改密码 21](#_Toc61013308)

[4.2.8 管理界面的逻辑处理 22](#_Toc61013309)

[4.2.9 Qt自动生成窗体类以及类的继承 24](#_Toc61013310)

[第五章 课设总结 26](#_Toc61013311)

[参考文献 27](#_Toc61013312)

# 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it，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，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。

# 第一章 绪论

## 1.1设计背景及意义

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，越来越多的人们选择了去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-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. 管理员信息管理(增加、删除、更新、修改管理员信息)

5. 管理员密码修改

# 第二章 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、概念模式设计、逻辑模式设计、数据库实施、数据库运行和维护，如图2-1所示。

## 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:

歌手信息：歌手编号、歌手姓名、歌手性别、国籍、拼音缩写。

歌曲信息：歌曲编号、歌曲名、歌手名、语种、歌曲类型、文件路径、歌曲拼音缩写、歌曲时长、点击量。

会员信息：会员编号、会员姓名、会员性别、出生日期、身份证号、注册时间、积分、会员密码。

管理员信息：管理员编号、管理员用户名、管理员密码。

### 2.1.2处理要求分析

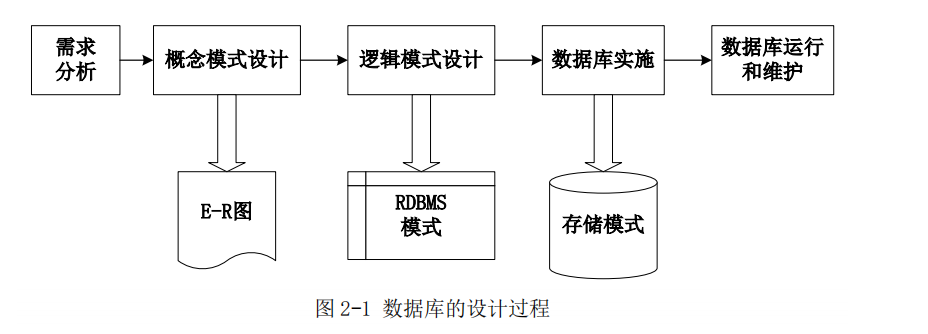
根据以表的属性可以得出，设计出来的数据库应面向用户和管理员。

图2-1 数据库设计流程

对于用户，用户可以使用会员编号以及密码进入前台点歌系统，进入后可以查询个人相关信息并可以修改密码。还可以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、歌曲名称的每一个汉字拼音打头的字母、歌曲名称来进行选歌。同时还可以查看歌曲排行榜信息。

对于管理员，管理员可以使用管理员编号以及管理员密码进入后台的操作系统，在后台操作系统可以对歌手信息、歌曲信息以及会员信息进行增加、删除、更新、修改等操作，同时也可以修改管理员信息。

以下为KTV点歌系统结构图：

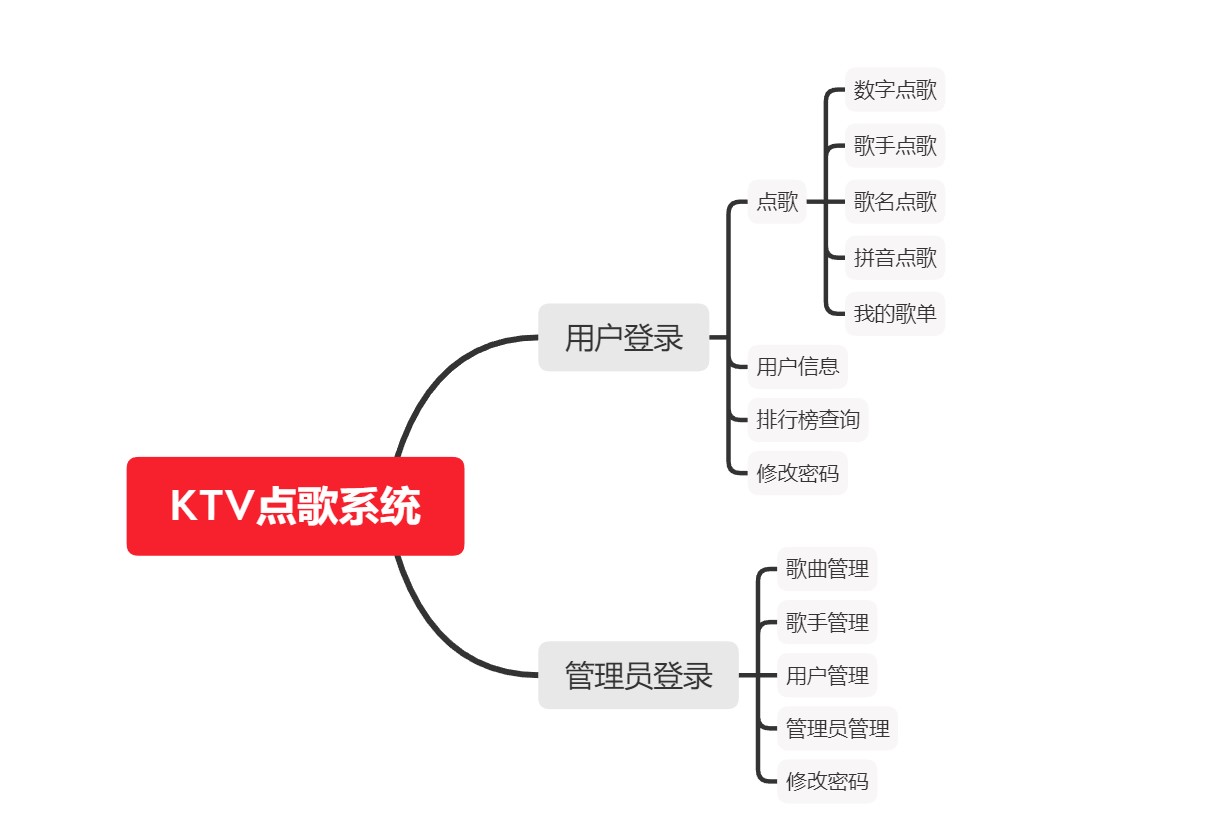


图2-2 KTV设计系统结构图

## 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模型，如关系数据模型。

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- - -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，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- -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

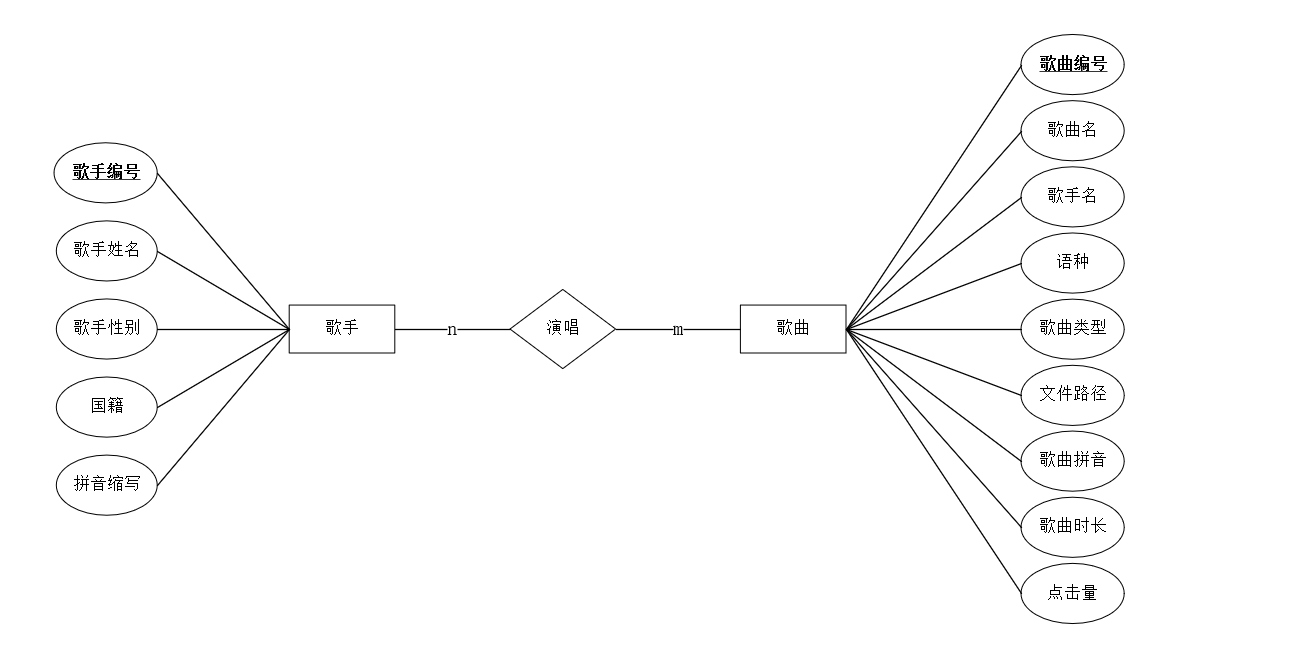
一首歌曲可由多位不同的歌手演唱，一位歌手可演唱多首歌。二者是多对多的关系，即：m:n的联系。其中歌手主键：歌手编号；歌曲主键：歌曲编号。

图2-3 歌手歌曲E-R图

（2）歌曲和会员的局部E-R图

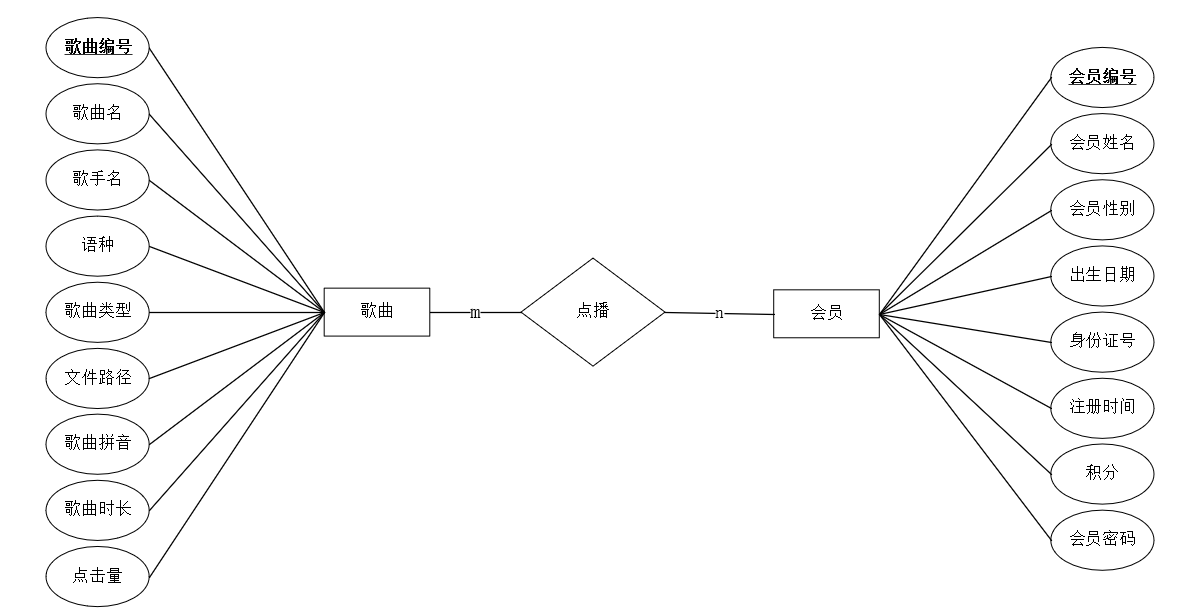
 一首歌曲可以被多名会员点播，一名会员可以点播多首歌曲。可见二者是多对多的关系，即m：n的联系。其中歌曲主键：歌曲编号；会员主键：会员编号。

图2-4 歌曲会员 E-R 图

（3）歌曲和管理员的局部E-R图

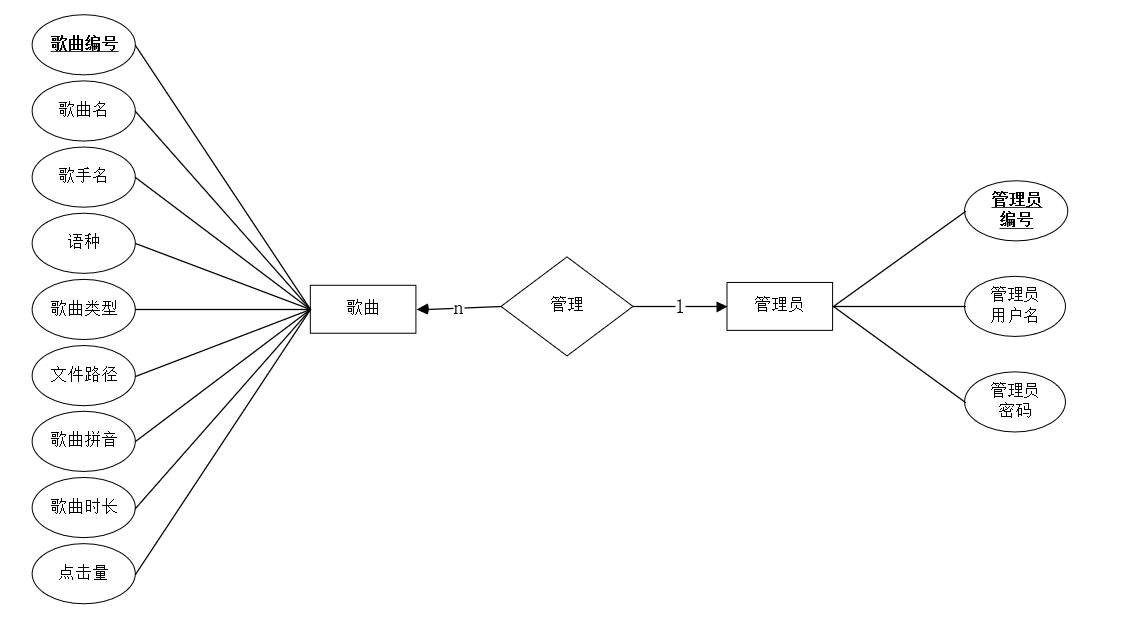
 一首歌曲由一名管理员管理，一名管理员管理所有歌曲。可见二者是多对一的关系，即n：1的联系。其中管理员的主键：管理员编号；歌曲的主键：歌曲编号。

图2-5 管理员歌曲E-R图

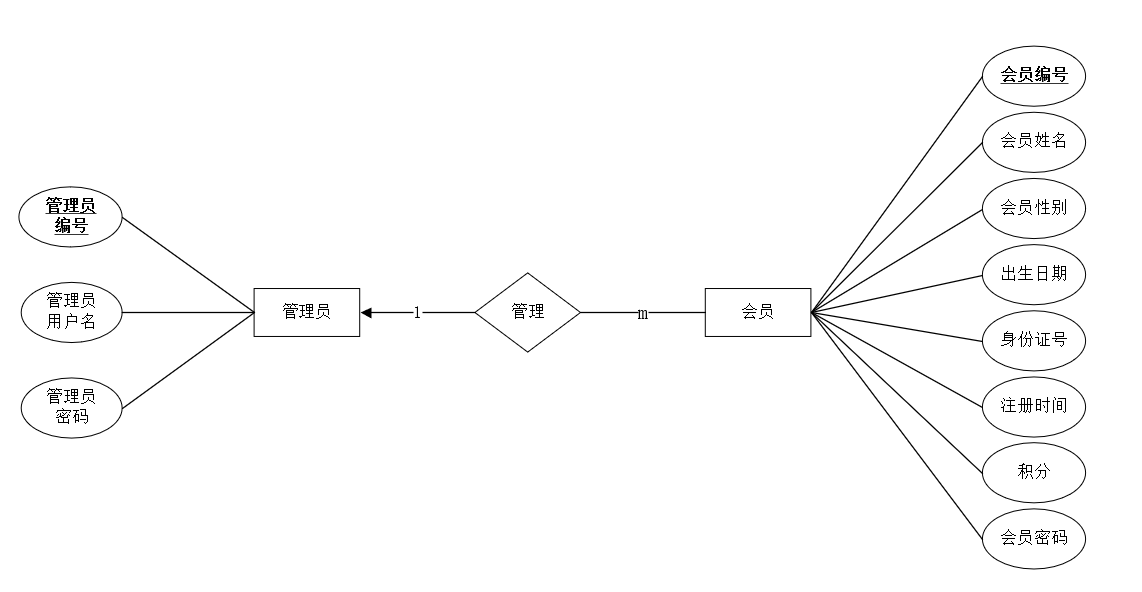
（4）会员和管理员的局部E-R图

图2-6 管理员会员E-R图

一名会员由一位管理员管理，一位管理员管理所有会员。所以二者是多对一的关系，即n：1的联系。其中会员的主键：会员编号；管理员的主键：管理员编号。

（5）歌手和管理员的局部E-R图

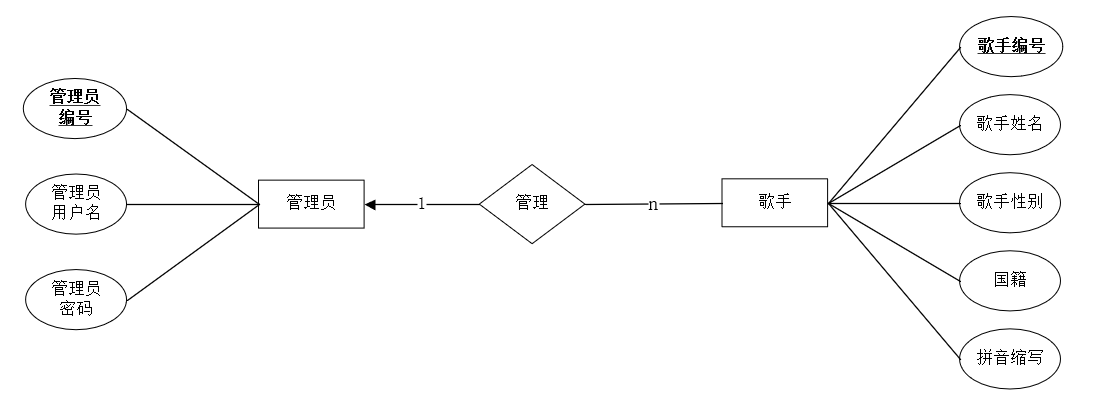
一名歌手由一名管理员管理，一名管理员管理所有歌手。所以二者是多对多的关系，即n：1的联系。其中歌手的主键：歌手编号；管理员的主键：管理员编号。

图2-7 歌手管理员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。下图2-9即为KTV点歌系统的全局E-R图。通过下图可以清晰的看出各个实体之间的关系。

## 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。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--定的规范化要求。规范化的主要目的就是要尽可能的消除关系操作过程中的异常情况。

### 2.3.1 E-R图到关系模型中的转换

图2-9 全局E-R图

关系模型的逻辑结构是- -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):

歌手（歌手编号、歌手姓名、歌手性别、国籍、拼音缩写）

2）歌曲（歌曲编号、歌曲名、歌手名、语种、歌曲类型、文件路径、歌曲拼音缩写、歌曲时长）

3）会员（会员编号、会员姓名、会员性别、出生日期、身份证号、注册时间、积分）

4）管理员（管理员编号、管理员用户名、管理员密码）

# 第三章 数据库设计

创建KTV点歌系统数据库。在其中建立5张表，分别为歌曲信息表、歌手信息表、会员信息表、管理员信息表。

3.1表的建立

3.1.1歌曲信息表

图3-1 歌曲信息表设计

图3-2 歌曲信息表数据

### 3.1.2歌手信息表



图3-3 歌手信息表设计



图3-4 歌手信息表数据

### 3.1.3管理员信息表

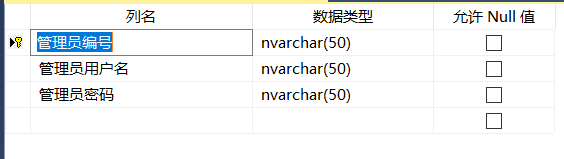
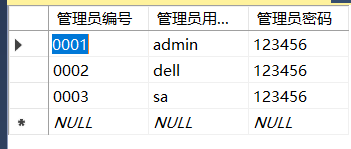


图3-6 管理员信息表数据

图3-5 管理员信息表设计

### 3.1.4会员信息表

图3-8 会员信息表数据

图3-7 会员信息表设计

# 第四章 Python-Qt界面设计

## 4.1 界面实现逻辑

本次数据库课程设计采用了界面与业务处理逻辑相分离的方法，先利用Qtdesigner软件绘制界面所需的基本控件，然后将生产的Python脚本代码作为单独模块引入到主程序当中。最后在主程序里利用super方法继承窗体类，构造属于自己的类和类方法，初始化好控件的信号和槽函数连接。接下来通过范例模块分析对界面的构造进行逐步分解。

搭建环境：Anaconda下的虚拟Python环境，python3.6、pymssql、PyQt5、PyQt5-tools。编辑器 PyCharm

## 4.2模块剖析

### 4.2.1 SQL连接方法

图4-1 自定sql类继承与重写新方法

利用pymssql库提供的方法包装自己的类，通过sql认证方式连接本地数据库。在初始化时创建数据库连接，在对象销毁时自动关闭连接。

### 4.2.2 登录界面以及处理函数

图4-2 登录界面

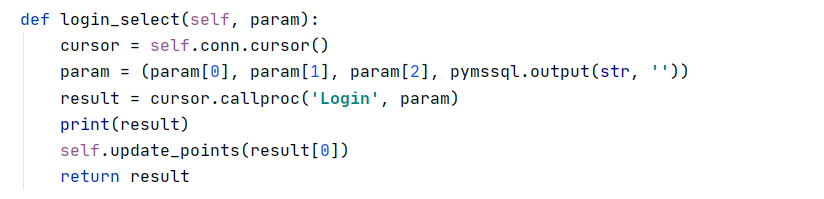
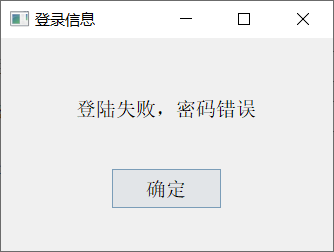
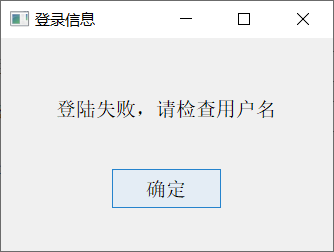
登录界面提供了一个RadioButton对登录身份进行选择，这个选择是互斥的，默认情况下自动选择了普通用户登录。

图4-3 登录界面sql处理函数

这是sql自定义类中对于登录的处理关系，为了保障数据库登陆的安全，我们采用了存储方法的过程，python语句通过调用数据库存储方法，将用户名密码登录角色和登录状态码，作为变量，前三个变量输入到存储过程中，最后一个变量作为登录码返回到主程序，我们可以通过不同登录码来区分登录状态，登陆成功，密码错误还是用户名错误。

图4-4 登录提示信息



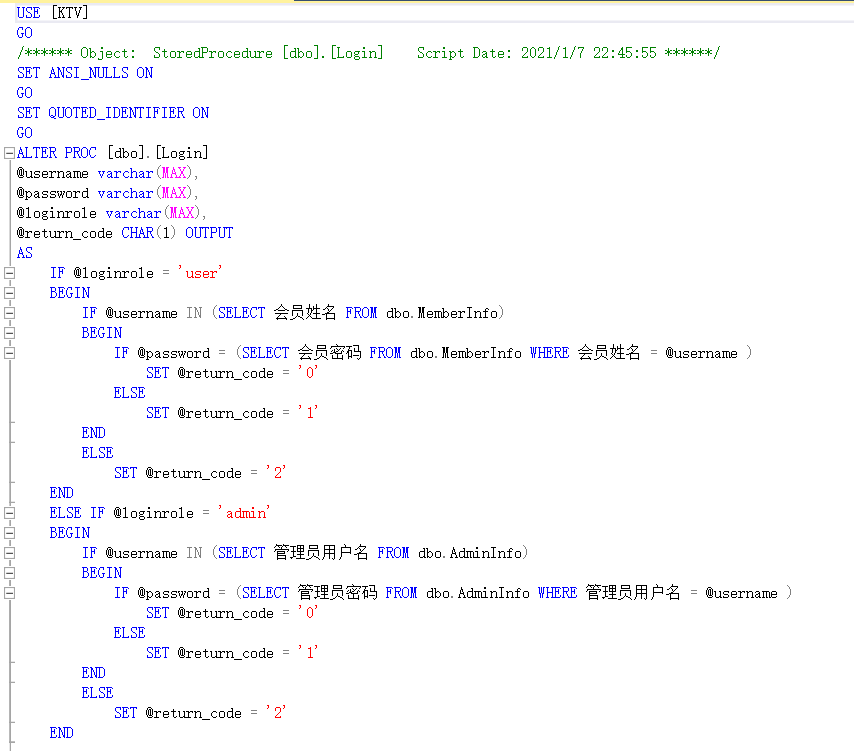


图4-6 sql登录的存储过程

图4-5 登录信息处理函数

### 4.2.3 用户初始界面

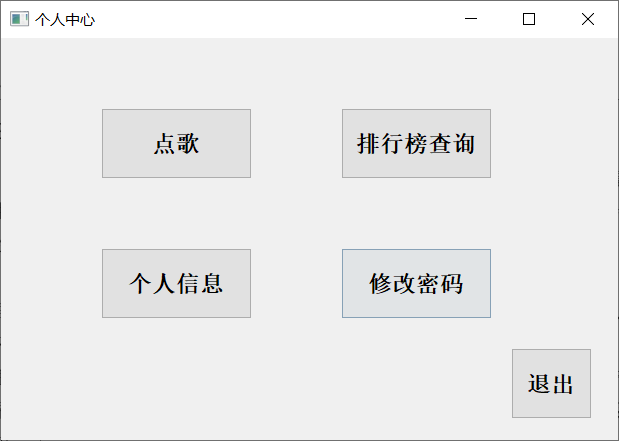
该界面下只有几个按钮链接到其他不同的页面，有一系列窗口的处理槽函数，利用show方法打开其他窗口，我们在init初始化方法当中将按钮单击信号链接到以上的槽函数中，从而实现窗口跳转。

图4-8 个人中心界面初始化

图4-7 个人中心界面

### 4.2.4 点歌界面

图4-9 点歌界面

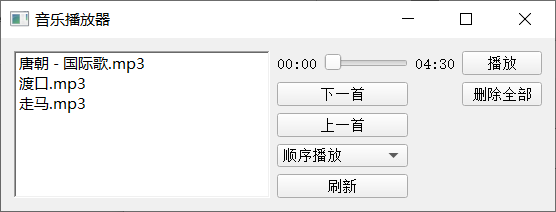
点歌界面如上图所示，我们在啊关键词一栏输入，然后点击不同的点歌方式，会自动通过关键词对歌曲库进行检索，如果不输入任何内容的话，默认打印歌曲库的全部歌曲。检索歌曲之后我们将光标移至自己喜欢的歌曲的歌曲编号点击点歌。点歌之后打开我的歌单，就可以看到自己点的歌曲。

图4-10 播放列表界面

在该界面我们可以播放和换歌，如果没有刷新出歌曲的话我们可以手动点击刷新来刷新歌曲。

### 4.2.5 个人信息和排行榜界面

歌曲排行榜界面会按点击量加载歌曲信息，个人信息按钮会将当前登录的用

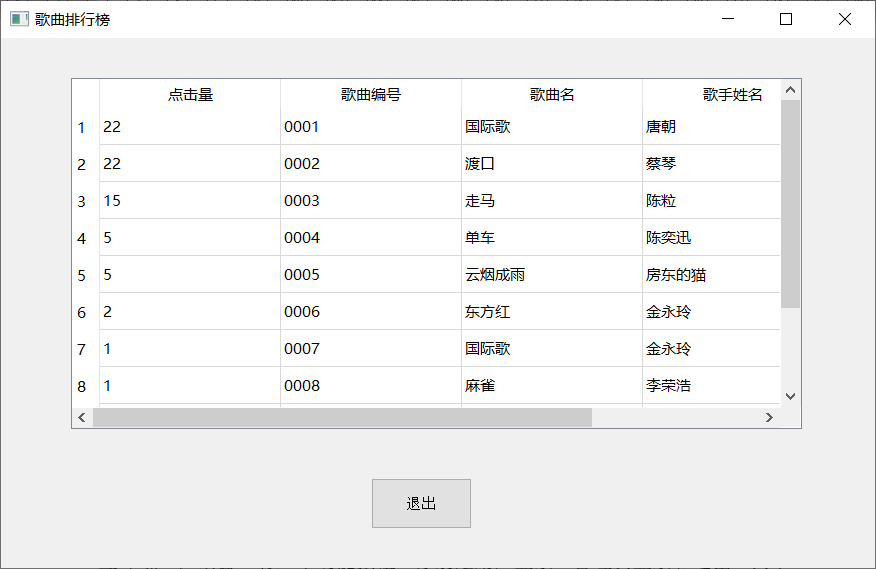
户的个人信息给显示出来。会员积分每次登录都会自动+10。

图4-11 歌曲排行榜界面



图4-12 会员信息界面

### 4.2.6 管理员管理界面

图4-13 会员管理界面

管理员对歌手歌曲会员和管理员的管理界面使用了同一种处理逻辑，上图是会员管理的界面，确定按钮是通过对会员编号进行检索的，左边有添加删除和更新数据三个功能，单击添加之后表格会多出一行空白行，我们在填写好信息后点击确定即可更新到表内，鼠标点击一个会员编号在点击删除用户即可删除该用户。如果需要修改信息，我们只需要将改修改的信息在表内修改完毕，然后点击更新数据，数据便会自动同步到数据表内。

### 4.2.7 修改密码

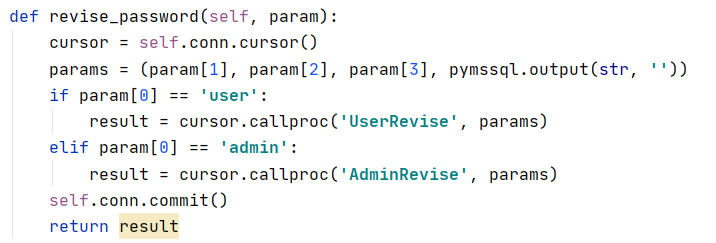
管理员和用户都可以修改自己的密码，他们采用了同样的逻辑处理方式，只要提供用户编号和旧密码即可修改密码。

图4-14 sql修改密码处理函数

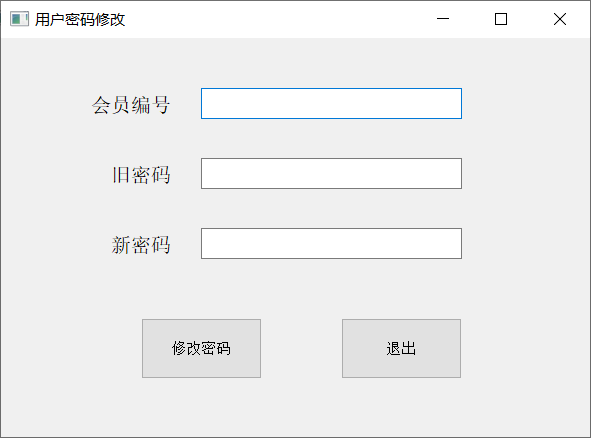


图4-15 修改密码窗口

### 4.2.8 管理界面的逻辑处理

管理界面采用了统一的界面，在处理方面上逻辑也大致相同，下面将具体处理代码进行剖析

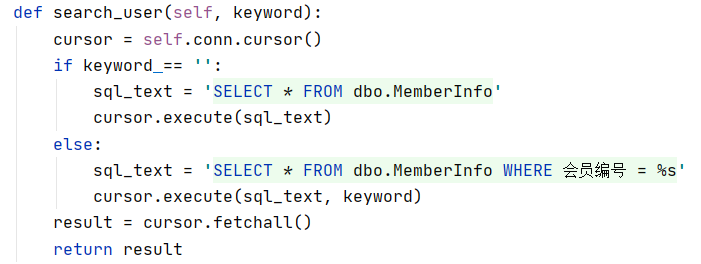
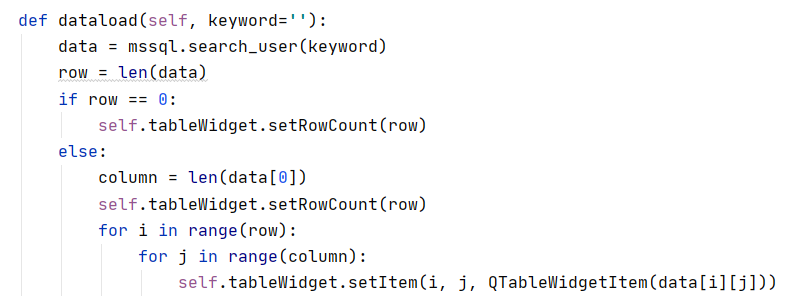
数据表加载，我们在进入界面和关键词检索之后需要加载从数据库中select出的结果，就需要用到该函数

图4-16 数据加载函数

图4-17 sql数据加载处理函数

默认传入的关键词是空白，点击确定即可开始检索，他可以自动分出行列的长度从而添加到tablewidget控件当中。

这是调用sql类的方法，通过传递关键词进入函数当中使用歌手编号进行检索，之后返回一个列表，我们从列表中取出数据。sql语句的执行和取出结果需要用到以上游标的方法。

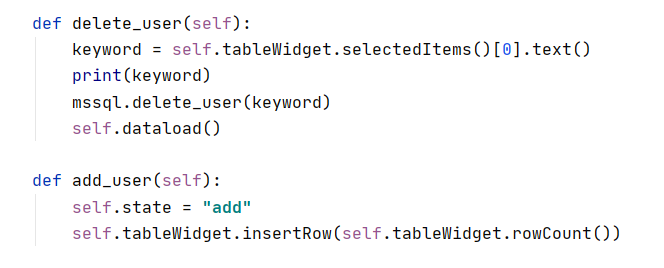
新建和删除

图4-18 用户新建和删除处理函数

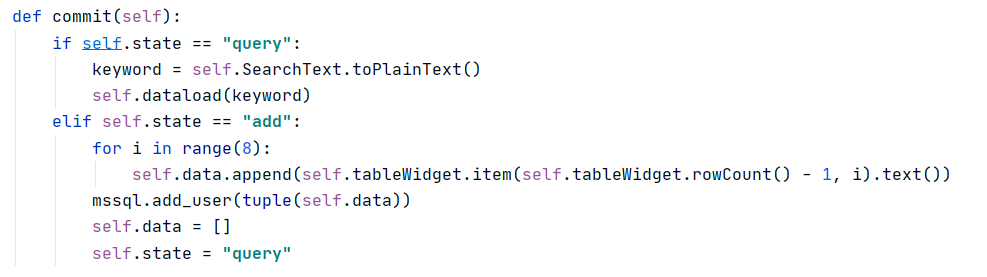
新建表的话会自动新建一个空行，输入好数据之后点击确定按钮进行数据处理，删除表需要点击指定要删除的编号，之后点击删除用户按钮即可。

图4-19 确定按钮处理函数

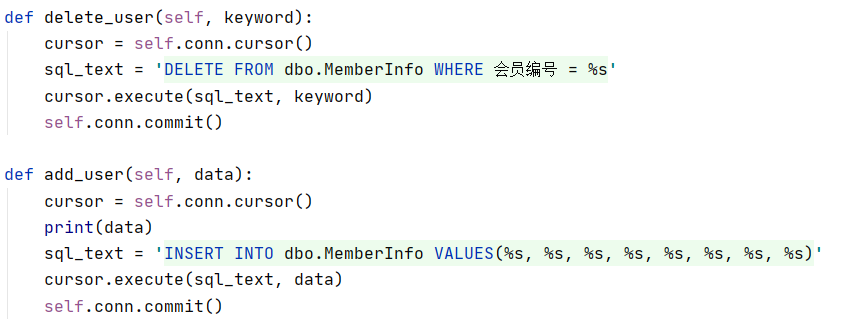
sql对应处理方法

图4-20 用户删除sql函数

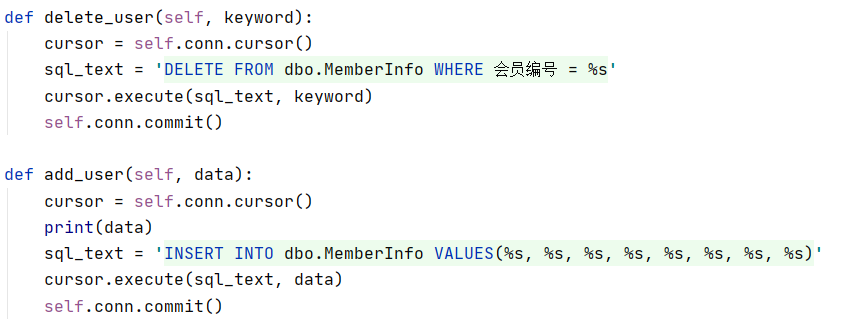
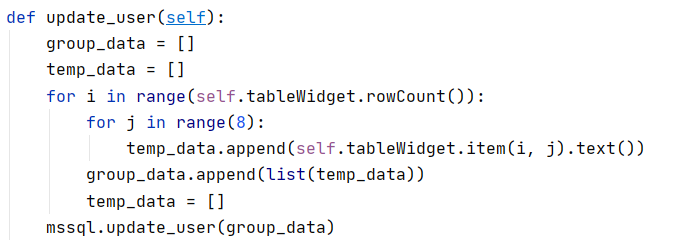
数据修改和更新

图4-22 用户信息更新处理函数

图4-21 用户添加sql处理函数

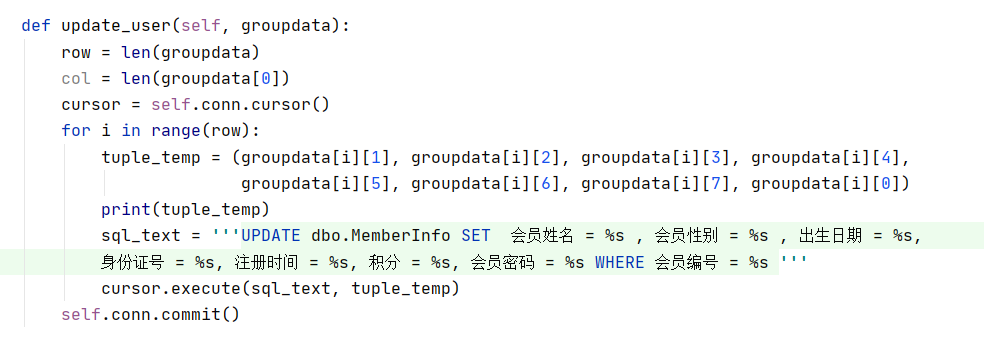
在点击更新数据之后，会将当前修改过的所有信息进行读取并保存在一个列表当中，之后将会调用sql方将列表内的全部数据更新到表内

图4-23 sql用户信息更新处理函数

### 4.2.9 Qt自动生成窗体类以及类的继承

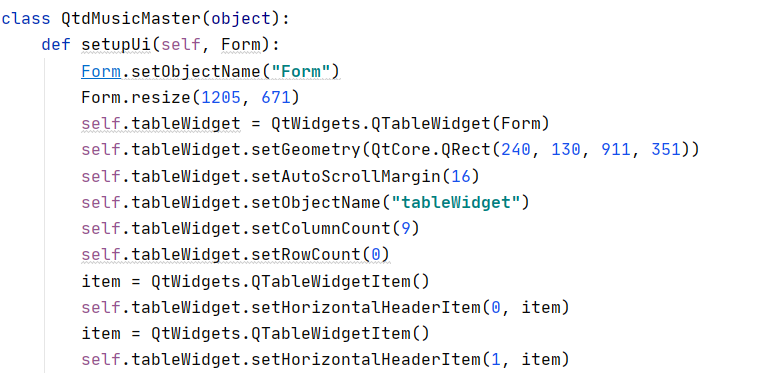
在本次课程设计中，所有的窗体控件初始化都是使用Qtdesigner绘制的，之后我们使用pyuic5命令将.ui文件转化为python脚本，在主程序中引用脚本中自动生成的类并将其继承。之后我们只需要在新继承的类中进行修改和重写，就能实现自己界面自定义的处理逻辑。

图4-24 QtDesigner自动生成窗体定义类

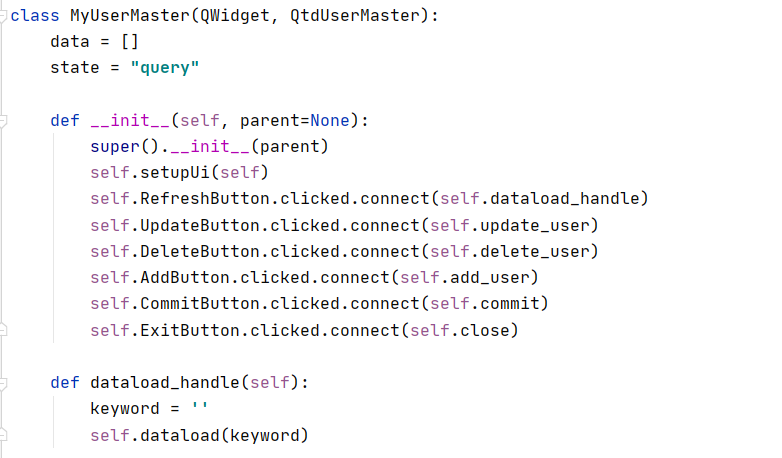
这个是qt自动生成的界面初始化的部分代码，主要是对控件类型位置大小的定义。

图4-25 窗体继承并添加自己方法

以上是对自动生成类的一个继承并添加自己要用到的方法，clicked.connect是按键单击时的信号，他会被对应的槽函数拦截并在槽函数中执行一个事件。

# 第五章 课设总结

本次课程设计制作了KTV点歌系统，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四种点歌方式，通过调用媒体播放器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也可以查看个人信息和修改密码。管理员也可以在后台信息管理中实现用户、歌曲、歌手和管理员的信息管理。

通过本次课程设计，我复习和了解到了SQL Server的一些基本知识和基本操作方法，同时也让巩固了我对Python语言的一些语法和方法的了解。学会了跨平台界面设计软件Qt的使用以及控件的调用，了解了界面同业务相分离的处理逻辑。提升了自己得代码阅读水平和编程能力。

本次数据库课设的开发过程中，发现了自己在很多方面存在不足，首先是数据库的设计方面不太完美，由于对课程知识的遗忘，自己一开始并没有设计好一个数据表，而是先对界面和接口进行了开发，最后在对接的过程中发现无法继续进行下去。第二点是自己的窗体开发仍然存在着较多的Bug，由于自身能力水平和时间问题，代码的健壮性很差，一些地方容易出现崩溃的情况。这是目前存在的最大问题，在之后的时间里，我会认真努力学习，不断提高自己的各项能力。

# 参考文献

[1]许谦,李元栋,王彧之.基于SQL Server的高校信息资源管理系统设计[J].现代电子技术,2020,43(20):115-118.

[2]温立辉.关系数据库触发器应用的研究[J].无线互联科技,2020,17(19):54-55.

[3]皇甫大双.SQL Server数据库中子查询的研究以及实例应用[J].电脑知识与技术,2020,16(28):46-47.

[4]陶文玲,侯冬青.PyQt5与Qt设计师在GUI开发中的应用[J].湖南邮电职业技术学院学报,2020,19(01):19-21.

[5]方炜.基于Python的图书采购查重系统的设计与实现[J].图书馆研究与工作,2019(12):36-41.

[6]张笑寒,曹菊英,王明亮,罗珩闻,姚武军,孙文.基于Python语言的学生考勤管理系统的设计与实现[J].信息技术与信息化,2019(09):156-159.

[7]于凯华,李华龙.基于QtDesigner插件机制的二次开发及应用[J].电脑知识与技术,2019,15(12):293-294.

[8]姜赵,谢晓伟.基于Python的数据可视化应用与研究[J].科学技术创新,2020(30):88-89.

[9]sql server数据库在实践教学中的改革[J]. 刘芳芳. 报刊荟萃. 2017(12)

[10]浅谈SQL Server数据库的特点和基本功能[J]. 闫旭. 价值工程. 2012(22)