

# 1. 技术方案设计

## C++

本项目选择 C++ 作为主要开发语言，这一决策是出于对 C++ 多方面优势的充分考虑。C++ 继承了 C 语言的传统，既支持经典的过程化程序设计，又引入了基于对象的程序设计理念。这使得我们在处理问题时既能使用传统的过程化方法，又能充分利用面向对象的特性，如封装、继承、多态等。

1. 语言简洁灵活： C++ 的语法简洁而灵活，提供了丰富的运算符，使得编码变得更加方便快捷。
2. 高效的目标代码生成： C++ 生成的目标代码质量高，执行效率优越，而且具备良好的可移植性，使得我们的程序能够轻松在不同平台上运行。
3. 适用于大规模项目： C++ 在可重用性、可扩充性、可维护性和可靠性等方面表现出众，非常适合开发大中型系统软件和应用程序。
4. 支持面向对象编程： C++ 提供了强大的面向对象编程机制，包括信息隐藏、封装函数、抽象数据类型、继承、多态、函数重载、运算符重载、泛型编程（模板）等特性。这使得团队开发更为简单，同时提高了代码的可维护性和可拓展性。



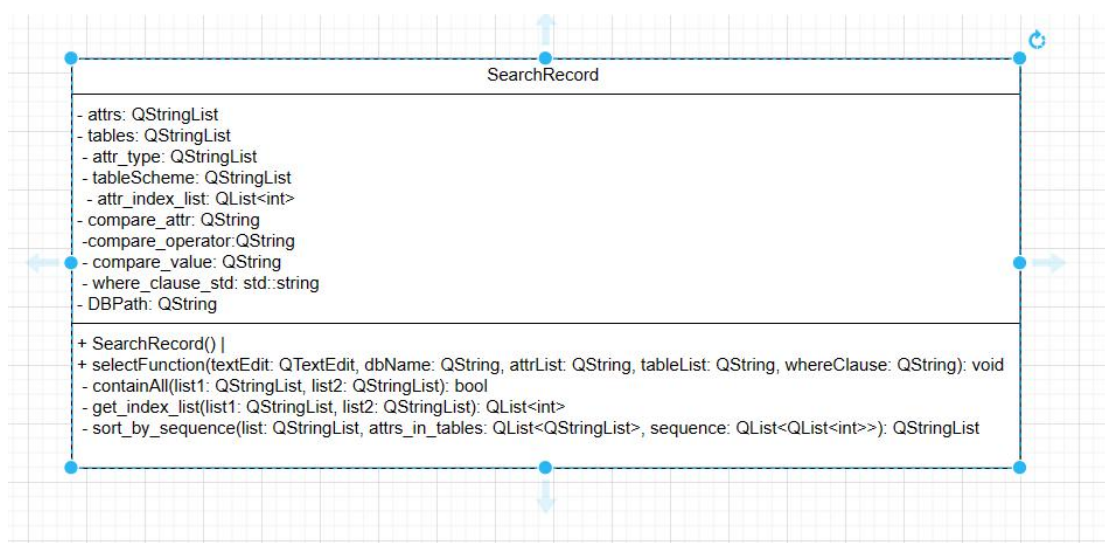
## QT



本项目选择 Qt 作为图形用户界面（GUI）开发框架。Qt 是一种跨平台的 C++ 框架，专注于为应用程序提供创建艺术级图形界面所需的所有功能。它是一个完全面向对象的框架，易于扩展，并允许真正的组件编程。Qt 在商业领域取得了巨大成功，成为全球数千种应用程序的基础，并且是 Linux 桌面环境 KDE 的基础。

## 关键算法实现描述-数据结构及算法流程图/类图等

### UML 类图



Searchrecord 是单表查询和多表查询

Altertable 实现了字段的修改

modifyrecord 实现了对已有记录的修改

Smtip 实现了邮箱发送验证码的功能

altertable
- filename: string - columns: vector<pair<string, string>>
+ alter(filename: string, operation: string, oldColumnName: string, newColumnName: string, newColumnType: string)

modifyrecord
- oldcurrentrow : int - newcurrentrow : int
+ valueExists(filePath : QString, targetKey1 : QString, currentrow : int&) : bool + modify(dbName : QString, tableName : QString, setClause : QString, whereClause : QString) : void

Smtip
- username: QByteArray - password: QByteArray - clientsocket: QTcpSocket* - recvaddr: QByteArray - subject: QString - content: QString - mailfrom: QByteArray - rcptto: QByteArray - prefrom: QByteArray - preto: QByteArray - presubject: QByteArray
+ Smtip(username: QByteArray, password: QByteArray) + send(recvaddr: QByteArray, subject: QString, content: QString): void + ~Smtip(): void

## 关键算法

### 1.单表查询

格式：select \* from table

从将要查询表对应的.txt 文件中读取该表的全部字段，输出到表格的第一行中，之后从要查询的表所对应的\_record.txt 中逐行读取记录，每读取一行，将数据分割成各个属性对应的记录后，插入到表格中新的一行的各个空格中

格式：select \* from table where condition

从将要查询表对应的.txt 文件中读取该表的全部字段，输出到表格的第一行中，之后从要查询的表所对应的\_record.txt 中逐行读取记录，每读取一行，将数据分割成各个属性对应的记录。在所读取的这一行中，找到 where 从句中要进行比较的属性所对应的字段值，如果符合比较条件，则将此行的字段插入到表格中新的一行的各个空格中

格式：select attr1, attr2, ..., attrn from table

首先从将要查询表对应的.txt 文件中读取该表的全部字段，同时将 select 从句后的字段记录下来，这表示最终要输出的字段。

获取每个要输出的字段在该表所有字段中的索引，构成一个列表，这个列表的次序与 select 之后的字段次序相对应。

将字段名输出到表格的第一行中，之后从要查询的表所对应的\_record.txt 中逐行读取记录，每读取一行，将数据分割成各个属性对应的记录后，按照之前获取的索引列表将字段记录依次插入到表格中新的一行的各个空格中。

格式：select attr1, attr2, ..., attrn from table where condition

首先从将要查询表对应的.txt 文件中读取该表的全部字段，同时将 select 从句后的字段记录下来，这表示最终要输出的字段。

获取每个要输出的字段在该表所有字段中的索引，构成一个列表，这个列表的次序与 select 之后的字段次序相对应。

将字段名输出到表格的第一行中，之后从要查询的表所对应的\_record.txt 中逐行读取记录，每读取一行，将数据分割成各个属性对应的记录。在所读取的这一行中，找到 where 从句中要进行比较的属性所对应的字段值，如果符合比较条件，则按照之前获取的索引列表将字

段记录依次插入到表格中新的一行的各个空格中。

## 2、多表查询

格式：select \* from table1, table2, ..., tablen

从将要查询的每一个表对应的.txt 文件中依次读取该表的全部字段，输出到表格的第一行中。之后从每个要查询的表所对应的\_record.txt 中逐行读取记录。

创建一个列表 table 储存最初所有要查询的表。从列表中弹出前两个表，将这两个表的记录按照行与行的笛卡尔积拼接成一个新的临时表。最后再将临时表插入到列表的头部。依次循环直到得到一个最终的文件，表示我们要展示的查询结果。

使用单表查询中 格式：select \* from table 的算法输出最终所得的临时查询文件的内容到表格中，之后删除所有临时文件。

格式：select \* from table1, table2, ..., tablen where conditon

从将要查询的每一个表对应的.txt 文件中依次读取该表的全部字段，输出到表格的第一行中。之后从每个要查询的表所对应的\_record.txt 中逐行读取记录。

创建一个列表 table 储存最初所有要查询的表。从列表中弹出前两个表，将这两个表的记录按照行与行的笛卡尔积拼接成一个新的临时表。如果 where 从句中要进行比较的属性所在的表是这两个表中的某一个，那么在做笛卡尔积时将不符合 where 条件的行过滤掉。之后再临时表插入到列表的头部。依次循环直到得到一个最终的文件，表示我们要展示的查询结果。

使用单表查询中 格式：select \* from table 的算法输出最终所得的临时查询文件的内容到表格中，之后删除所有临时文件。

格式：select attr1, attr2, ..., attrn from table1, table2, ..., tablen

从将要查询的每一个表对应的.txt 文件中依次读取该表的全部字段，输出到表格的第一行中。之后从每个要查询的表所对应的\_record.txt 中逐行读取记录。

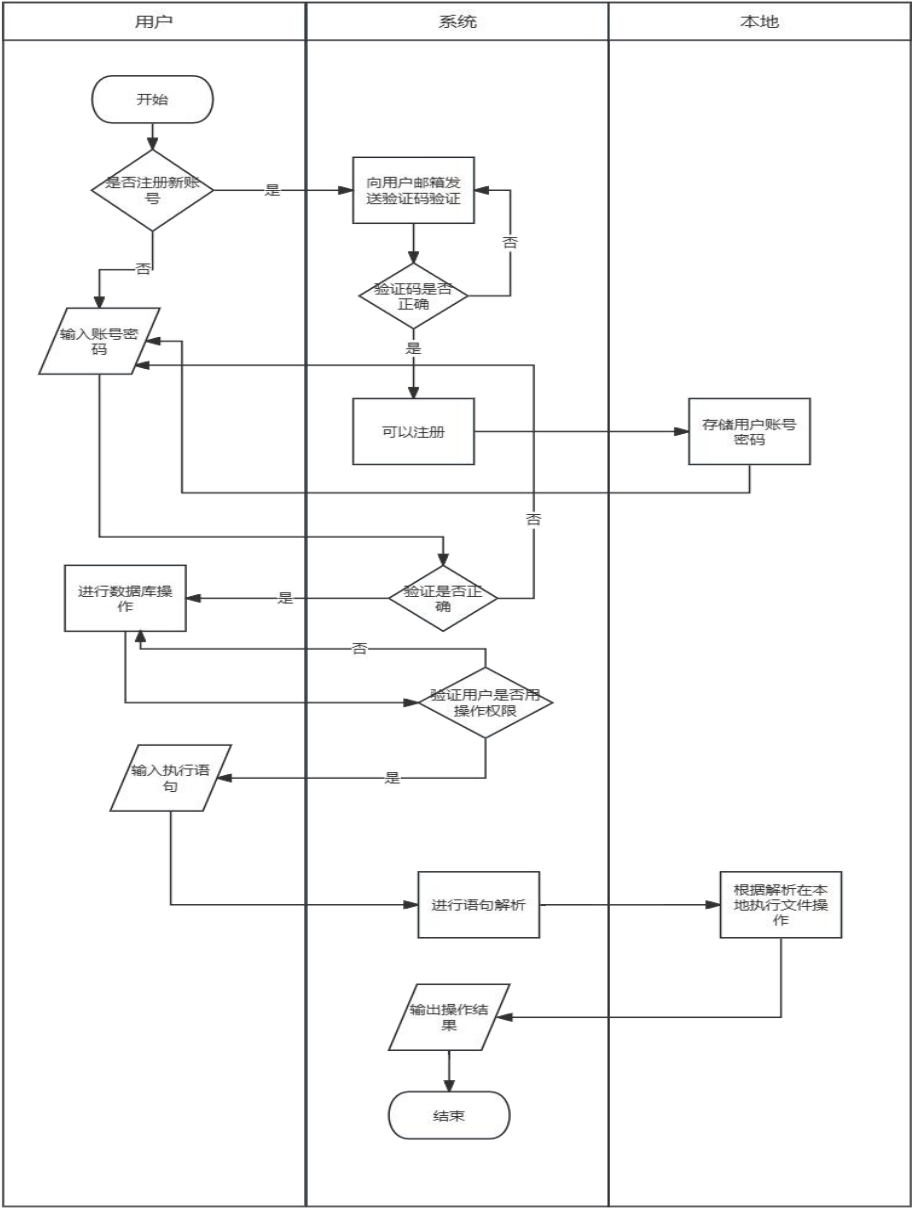
获取每个表中要查询的属性, 并且记录该属性在相应表中的索引 (方法同单表查询中 格式:  
select attr1, attr2, ..., attrn from table 相应的方法一致)

创建一个列表 table 储存最初所有要查询的表。

再将 table 列表中每个表要查询的字段对应的记录摘录到一个新的临时表中, 将这些临时表组成一个新的列表 temptable

从 temptable 列表中弹出前两个表, 将这两个表的记录按照行与行的笛卡尔积拼接成一个新的临时表。

# 流程图



# 用户手册

## 一、前言

本手册旨在为用户提供关于我们 DBMS 的详细指南。通过本手册,用户可以了解如何安装、使用该 DBMS。请确保在使用 DBMS 之前,您已阅读并理解了本手册中的所有内容。

## 二、系统概述

我们完成了基本的 DDL,DML,DQL 操作,额外添加了脚本执行的操作,方便用户批量执行代码,对用户的操作权限进行了授予与回收,对用户的操作进行日志管理,本项目还额外增加了多表查询功能。

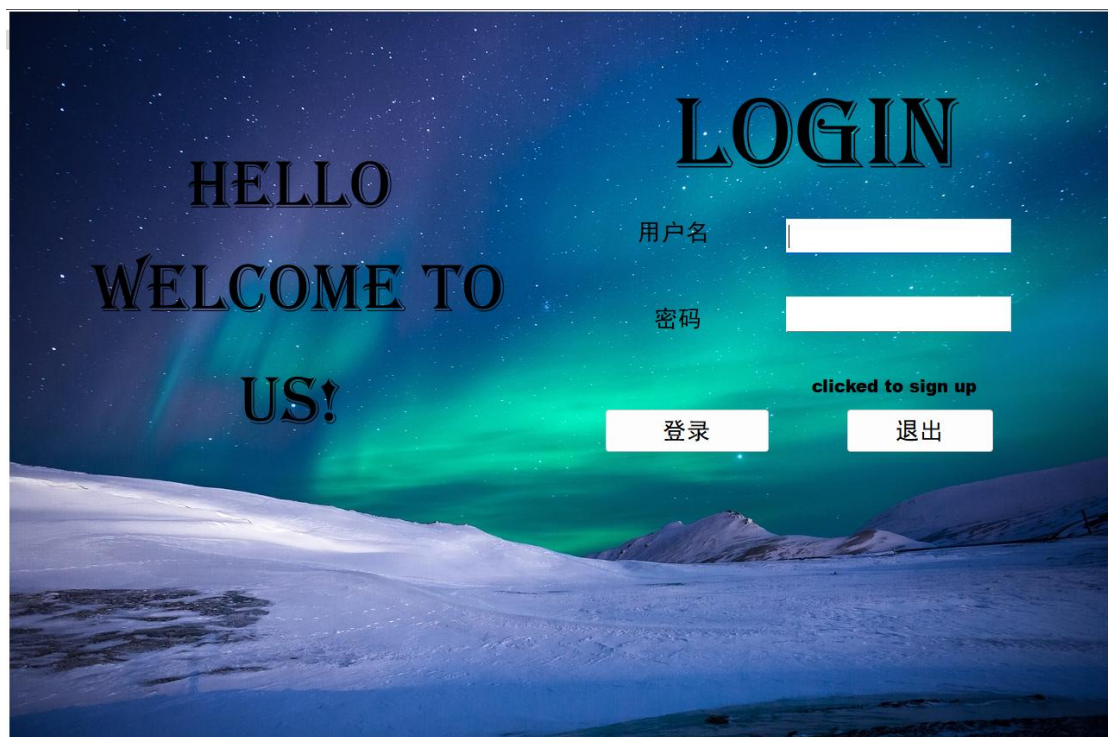
## 三、安装

在 gittee 上下载我们的项目 dbms-4,我们所使用的 qt 版本为 5.14.2,在 qt 上使用该版本能够成功运行

## 四、使用指南

### 4.1 登录与退出

请求用户输入用户名和密码,点击登录会去检验该用户是否已经创建过,点击退出,退出 DBMS







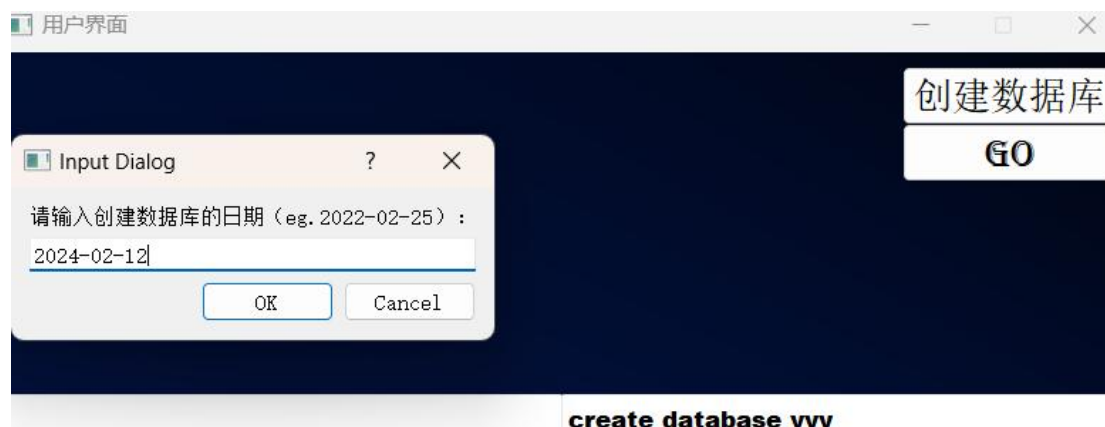
如果用户并不存在，可以点击'clicked to sign up',在注册界面，用户需要输入用户名，密码，确定密码。邮箱用来验证是否为人机注册，在邮箱输入可以使用的邮箱，点击“发送邮箱验证码”，获取邮箱验证码需要输入，点击注册会去验证用户的存在，密码输入是否一致，邮箱验证码是否正确

## 4.2 DDL

### 1.数据库的创建和删除

#### create database (数据库名)

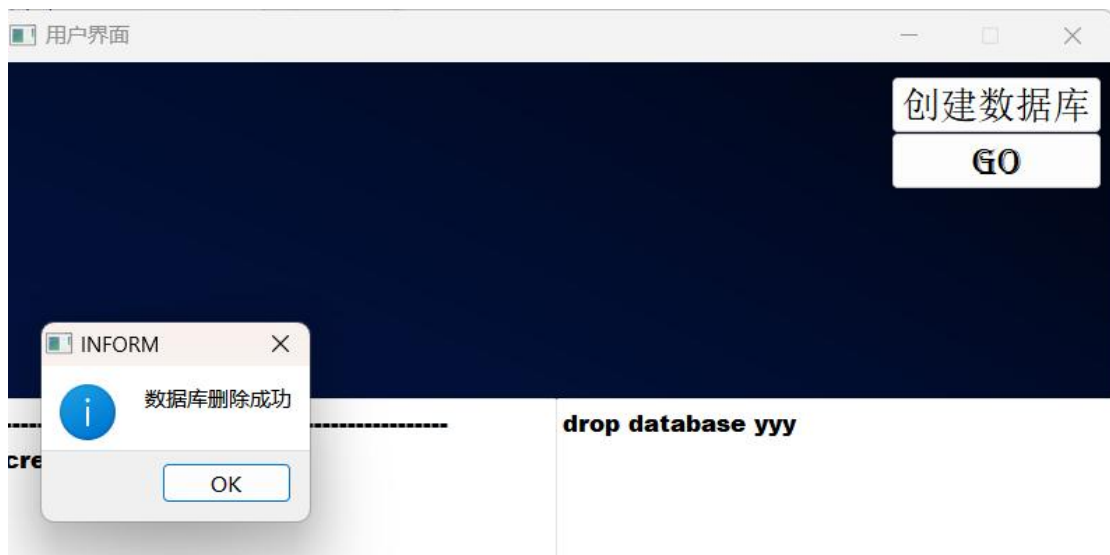
当用户创建数据库时，会去 Data 文件夹下，Ruanko 中判断是否已经创建过数据库，如果已经创建过，则提示，否则创建数据库，提示写入创建日期。



**create database yyy**

同样，当删除数据库时，会去判断数据库是否存在，存在则删除，对于存储数据库信息的 Ruanko 不允许删除。

#### drop database (数据库名)



## 2.表的创建和删除

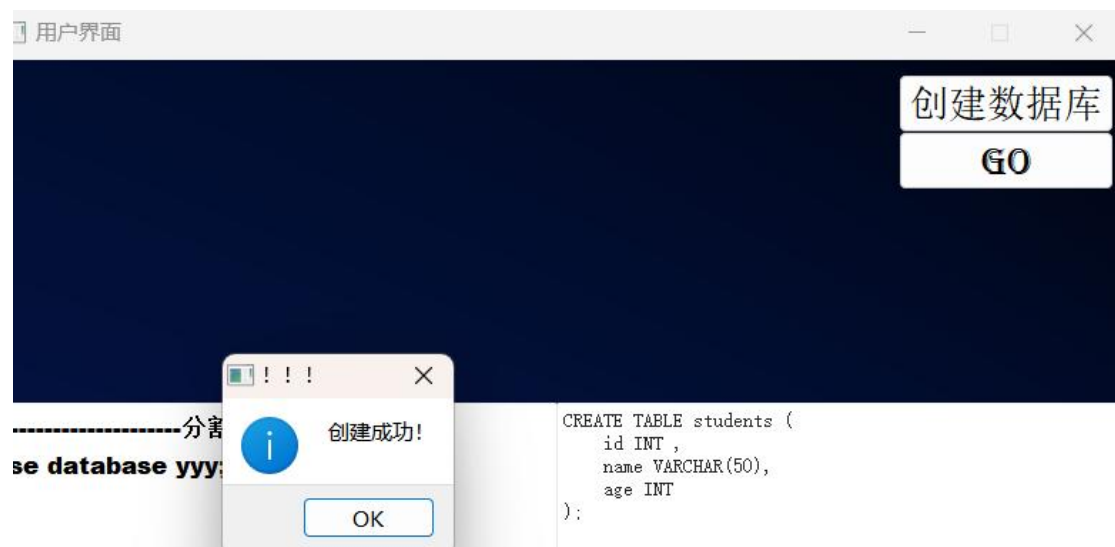
用户需要首先指定 db:

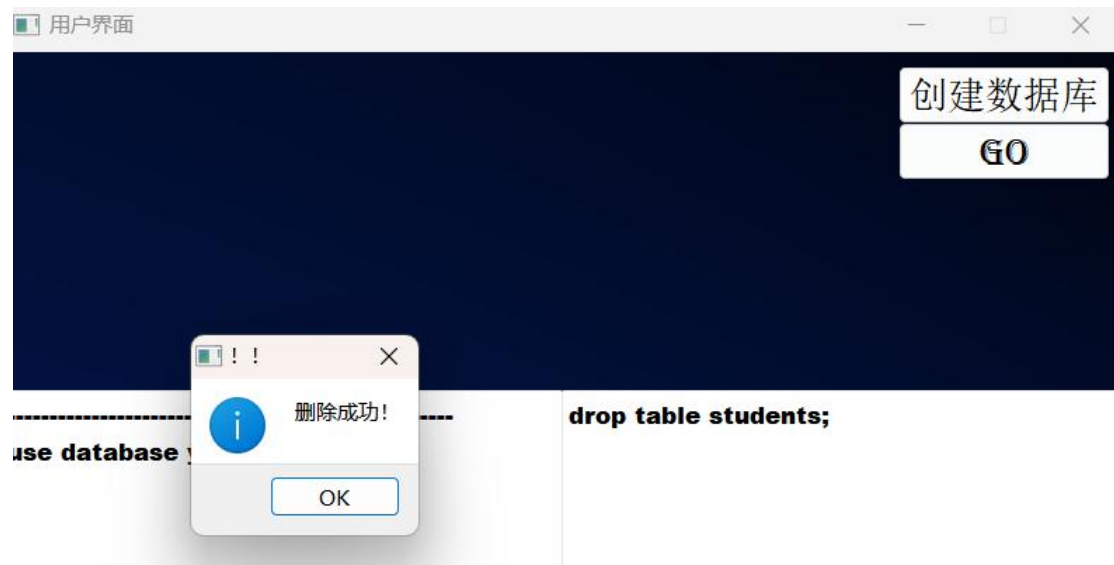
**use database xxx;**

表的创建和删除信息都存储在 data 对应的数据库文件夹继续创建 table 表，表内放记录和表的属性等两个 txt，目前只完成基本的创建和删除表，只包含字段名和字段类型。成功与失败都会显示信息。

删除指令：

**create table xx(** **drop table xx;**  
     **字段名 类型,**  
     **字段名 类型**  
**)**





### 3.字段的添加, 删除, 修改

DDL 的 add ,drop,modify 关键词大小写无影响, 但是格式必须严格遵照如下格式

`alter table (tablename) drop (字段名);`

`alter table (tablename) modify 旧的列名 新的列名 新的数据类型;`

`alter table (tablename) add 字段名 字段类型;`





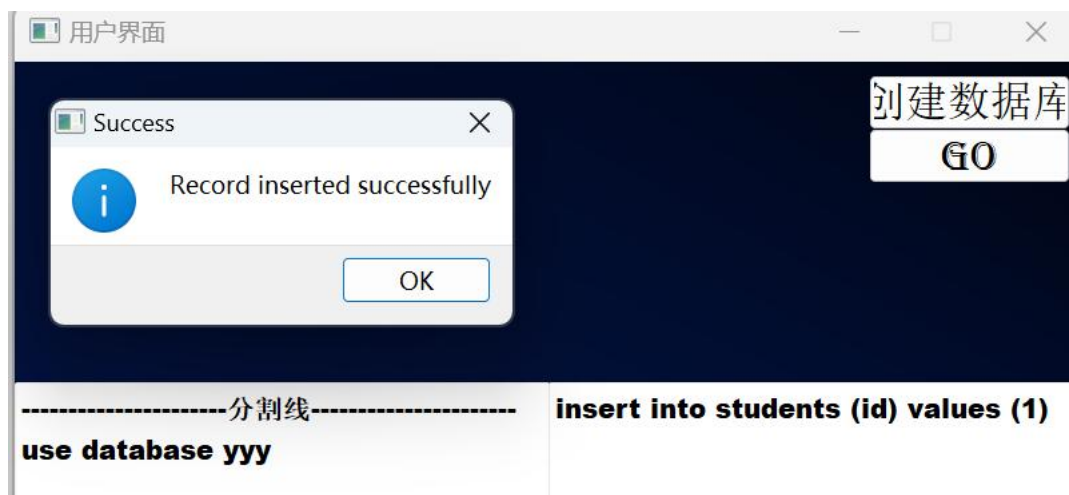
## 4.3 DML

会在表下新建一个 `tablename_record.txt` 文件，文件内放表的记录

### 1:完成记录插入

(插入的必须指定列，按照如下的格式，不支持 `insert into tablename values();`

**`insert into (tablename) (列名,...) values (值,..);`**

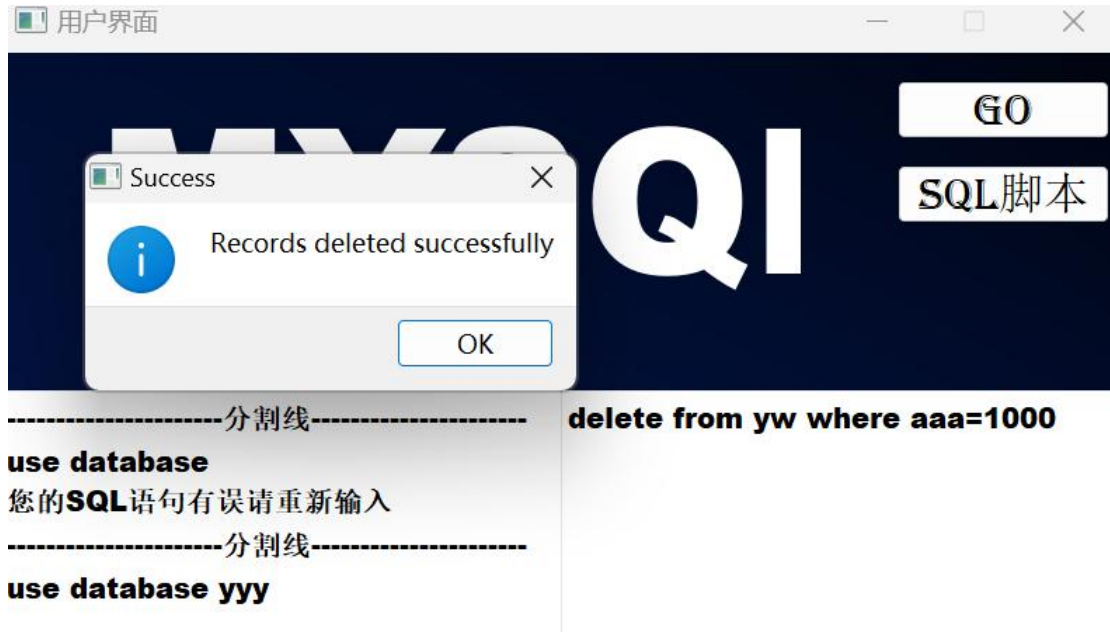


### 2:记录删除

**`delete from (tablename) where 字段名=xx;`**

2.0 版本：在 1.0 单列版本上变为多列表，操作如下。

具体逻辑为，先判断在 `yw.txt` 信息表中是否有 'aaa' 字段，若存在字段，则进行值匹配，若相等则删除该行记录



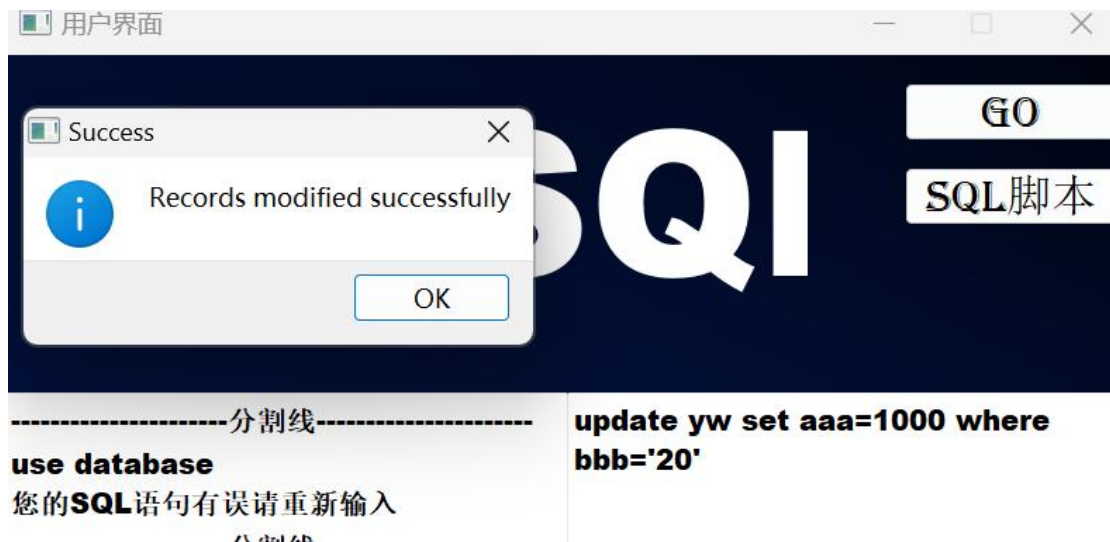
### 3:记录修改

**update (tablename) set 字段名='xx' where 字段名='xx'**

2.0 版本

对于单列表的基础下拓展为多列表。具体实现逻辑为，先校验是否存在字段，如存在字段则进行 where 条件判断和值的修改操作。

注意：where 条件和 set 仅支持单列



## 4.4 DQL

### 1:所有记录

**select \*(必须有空格)from tablename;**



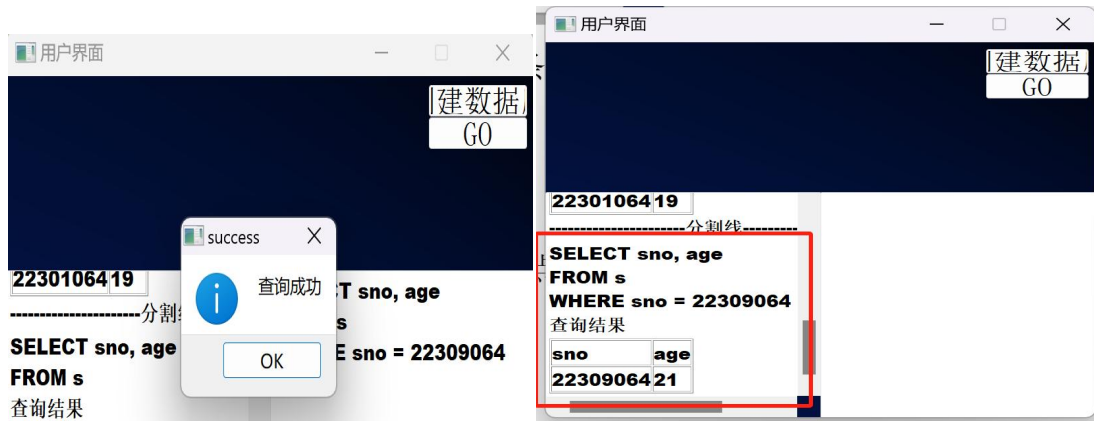
### 2:指定字段的所有记录

**select 字段名, 字段名... from (tablename) ;**



### 3:指定字段和 where 条件的记录

**select 字段名, 字段名... from (tablename) where 字段名='xx';**



## 4.5 附加功能

### 1: 脚本批量处理:

(写好的脚本在 texttest 文件当中)

脚本命令，在用户界面点击按钮“SQL 脚本”，选择一个脚本进行执行  
这里给出的测试样本是在 texttest 文件夹当中

### 2: 日志管理

每当输入一条 SQL 语句时候都会将记录存入到 Log 文件夹当中

### 3: 权限管理

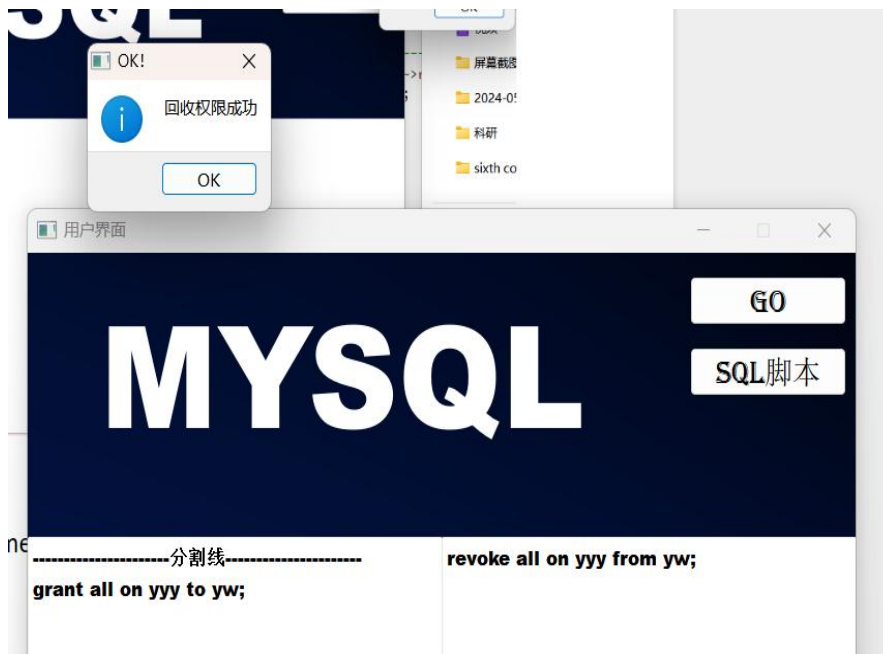
回收和授予权限的操作都是在 User/用户名/Permission/txt 中文件存放用户操作数据库的权限。

在 use database 时候会去检查该用户是否有操作指定数据库的权限

**grant all on dbName to user;**



revoke all on dbName from user;



#### 4：邮箱验证功能

该功能旨在在注册界面通过邮箱发验证码，验证是否是人机



The image shows a registration form titled "SIGN" in a large, stylized, black serif font. The background is a deep blue space scene with stars and a green aurora borealis. Below the title, there are five white input fields arranged vertically. To the right of each field is a label in Chinese: "用户名" (Username), "密码" (Password), "ENSURE" (likely a brand or company name), "邮箱" (Email), and "邮箱验证码" (Email verification code). At the bottom of the form, there are three white buttons: "注册" (Register), "退出" (Logout), and "发送邮箱验证码" (Send email verification code).

# SIGN

用户名

密码

ENSURE

邮箱

邮箱验证码

注册

退出

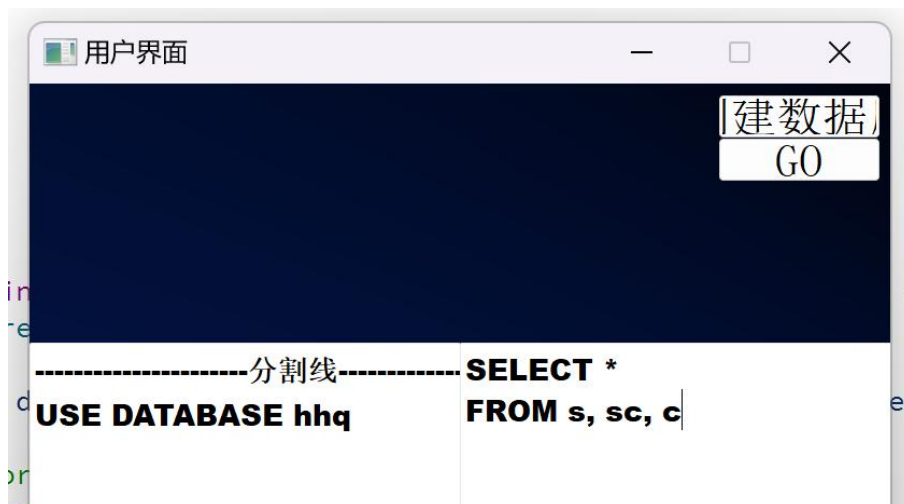
发送邮箱验证码

在创建用户的时候，会要求发送一封邮件到你已有的邮箱，需要正确的输入邮箱验证码方可注册成功

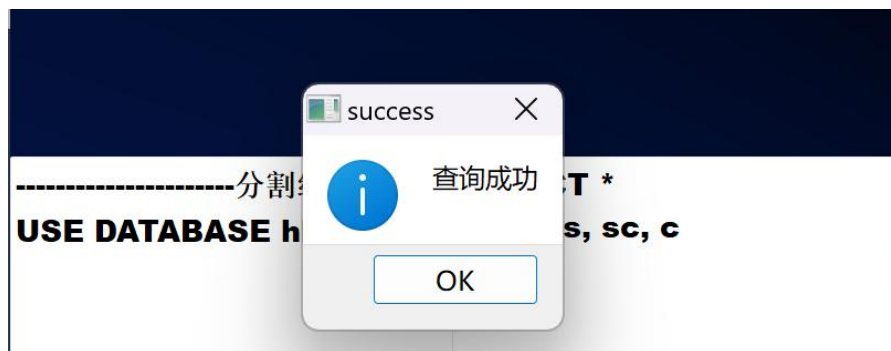
## 5: 多表查询功能

1) 查询全部字段（下文演示中由于显示区域大小有限，有时不能完全显示查询结果，实际应用中拖动滚动条即可看到整个表）

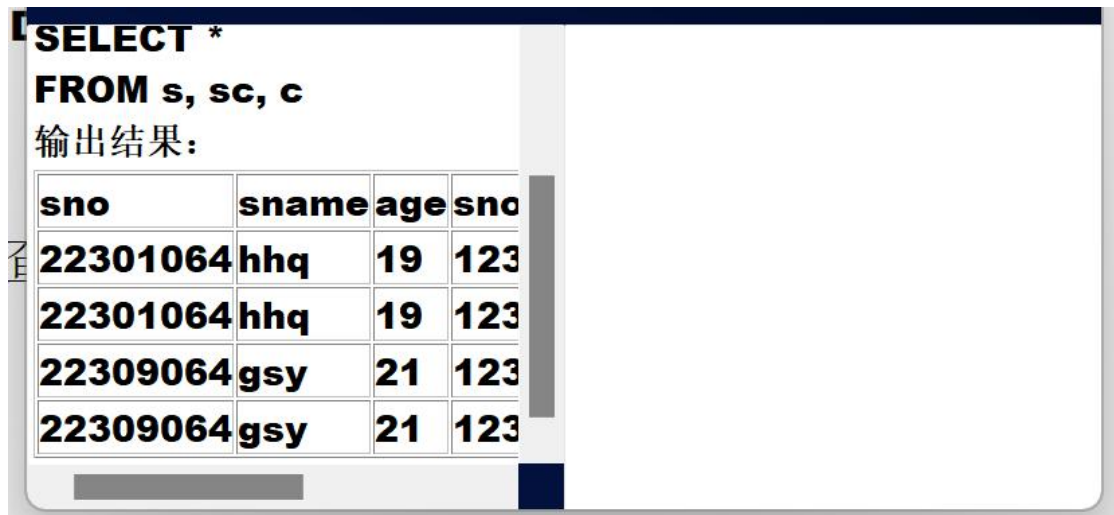
输入 SQL 语句



点击 GO



显示查询结果

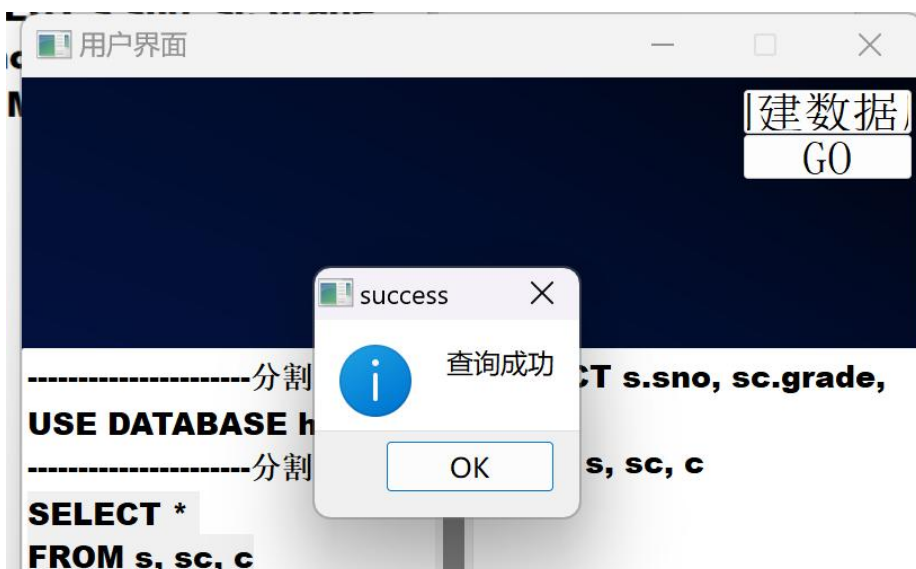


2) 查询指定字段

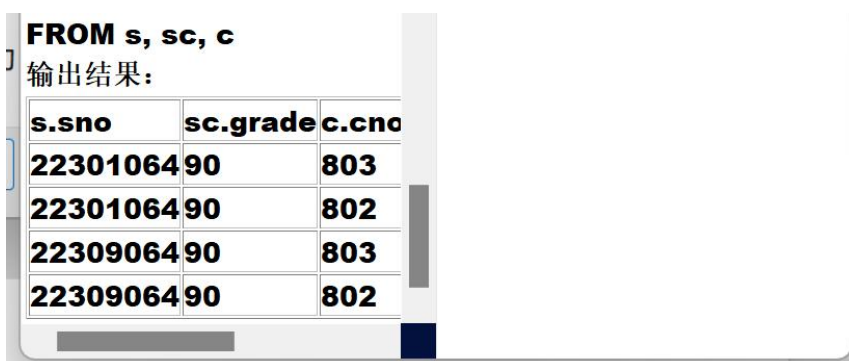
输入 SQL 语句



点击 GO



显示查询结果



3) 查询指定字段且有简单的 where 从句

输入 SQL 语句

**SELECT s.sno, sc.grade,**  
**FROM s, sc, c**  
 输出结果:
 

s.sno	sc.grade	c.cno
22301064	90	803
22301064	90	802
22309064	90	803
22309064	90	802

**SELECT s.sno, sc.grade,**  
**c.cno**  
**FROM s, sc, c**  
**WHERE s.sno = 22301064**


点击 GO

**SELECT s.sno**  
**FROM s, sc, c**  
 输出结果:
 

s.sno	sc.grade	c.cno
22301064	90	803
22301064	90	802
22309064	90	803
22309064	90	802

**SELECT s.sno, sc.grade,**  
**c.cno**  
**FROM s, sc, c**  
**WHERE s.sno = 22301064**

success

 查询成功

OK

显示查询结果

**WHERE s.sno = 2230106**  
 输出结果:
 

s.sno	sc.grade	c.cno
22301064	90	803
22301064	90	802

4)查询全部字段且有简单的 where 从句

输入 SQL 语句

分割线

SELECT s.sno, sc.grade,  
FROM s, sc, c  
WHERE s.sno = 2230106

输出结果:

s.sno	sc.grade	c.cno
22301064	90	803
22301064	90	802

SELECT \*  
FROM s, sc, c  
WHERE s.sno = 22301064

点击 GO

success

查询成功

OK

分割线

SELECT s.sno, sc.  
FROM s, sc, c  
WHERE s.sno = 2230106

输出结果:

s.sno	sc.grade	c.cno
22301064	90	803
22301064	90	802

T \*  
s, sc, c  
s.sno = 22301064

显示查询结果

SELECT \*  
FROM s, sc, c  
WHERE s.sno = 2230106

输出结果:

sno	sname	age	sno
22301064	hhq	19	123
22301064	hhq	19	123

# 项目总结

## 个人完成的任务总结

余旺：

负责开题任务，整个框架的搭建和用户界面，整体对 SQL 语句进行解析

基本的 create db 操作，提供一个整体的书写框架

对组员的代码 Merge,debug,完善功能,项目管理模块负责

完成 4 个附加功能，对于并不完善的授予回收权限交给杨涛同学完成

结题答辩审核和材料书写

杨涛：DML，授予权限功能完善

王宏翔：DDL 部分，负责最终答辩

黄浩琪：中期答辩，DQL 功能，附加项多表查询完成

## 工作量分配比

余旺：28%

杨涛：25%

王宏翔：22%

黄浩琪：25%