

中期报告

项目管理

项目计划安排

初次讨论：

- 1, 对于 SQL 语句采取正则化匹配的形式
- 2, 数据库中数据用文件来保存，统一放在 data 当中
分支下，一个数据库一个文件夹
- 3, 考虑加入权限的问题，不同的用户
授予权限即为该数据库有没有操作权限

前期：

hhq 同学开始着手表的操作，先单独探索一会

——3.31

yw 正则化匹配，数据库数据的创建和删除

——3.31 完成部分

对任务的评估：

- 1, 以工作量大为鲜明特征
2. 没想清楚数据的修改与存储

任务感受：

yw:

正则化还是值得一学

其次数据库的创建，对于文件的创建，写入修改等等内容重新回顾了一下

第二次开会：

T

分工:

王 hx+杨涛:

1.2.2 表的创建、删除 5 分 ; 1.2.3 表的修改: 字段添加、删除、修改 5 分)

c) DML 功能实现 15 分 (1.3.1 完成记录插入 5 ;1.3.2 记录删除 5 ;1.3.3 记录修改 5)

yw:

对用户界面等等进行优化

hhq:

DQL 功能实现 (简单条件) 20 分 (1.4.1 所有记录 5 分 ; 1.4.2 指定字段的所有记录 5 分 ;1.4.3 指定字段和 where 条件的记录 10 分)(不要求完成但可以试着写完成部分逻辑等待上面的完成进行实现)

任务预期:

分工的内容希望是本周内完成, 如果过程中遇到问题则一起商讨解决

商讨内容

1. 界面与优化问题
2. 文件存储与读取问题
3. 带领大家一起理解目前整体思路, 和已实现功能的代码部分

——4.1

hhq 写:

这周我对于增删改查等基本操作做了一些尝试, 我试图将数据库的表写成一个数据结构, 对

于表的增删改查操作都是基于这个数据结构。接下来简要介绍一下这个数据结构的设计思路。

一张表显然是由表模式（表头）以及表的内容组成的。表头记录了每一列数据的字段名以及数据类型。我尝试将表头的每一个 cell 封装为一个结构体，叫做 tableSchemeElement。显然，这个结构体储存了相应列的字段名以及数据类型。

接下来考虑表的记录，也就是插入的元组。将每一行和每一列的交点所对应的 cell 封装为一个结构体，命名为 tableElement。

这样做有一些好处：所有的 cell 类型是一样的，这会屏蔽掉不同 cell 所具有的不同数据类型，使得之后的增删改查更加方便。

基于这两种基本的结构体，我们得以创建一个表。表的本质就是一个由这些基本元素构成的二维向量，增删改查等操作都可以看作是对于此二维向量的操作。

增删改查等基本功能得到了实现，但是遗憾的是我没有考虑数据库需要储存大量的数据这一特点，导致只能将一个表的信息储存在一个二维向量中，无法长期储存。也正是因为这种设计方式没有文件操作，导致不得不该用新的方法。

但是这些探索也为我们接下来的任务做了更好的铺垫，或许也算是一次试错。

第三次开会

这周主要是各自推进自己的任务，不开会

登录界面的修改，用户注册登录添加。

中间发生过一个小插曲，组员用了 Merge 操作后，master 这边下载后不能正常打开了，学习对历史数据进行还原，将项目还原到上一个版本了。

——yw

DML 的初代代码已经完成，包括插入 1.1，删除 2.1，修改 3.1 版本。后续将考虑复杂表的结构，高级约束检查持续迭代更新。同时将引入数据库日志功能，便于检查数据库异常。

-yt

项目追踪

平台管理总览

DBMS

免费版

项目 进展 知识库 日历 汇报 动态 团队 我自己

搜索

+

?

🔔

🌐

DBMS

列表 看板 日历 时间线 文件 文档 进展 回收站

添加成员 设置

添加任务

☑ 未完成任务

🔍 筛选

⬆️ 排序

🗑 按清除分组

📄 任务字段

默认

⋮

任务标题	所属项目	优先级	截止时间	负责人	
<input type="checkbox"/> 完成开题答辩相关准备		最高	3月23日	旺旺	
<input checked="" type="checkbox"/> 数据库整体搭建	...	较高	3月23日	旺旺	
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	Baba Yetu	
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	旺旺	
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	Y_	○
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	麦秆	
<input type="checkbox"/> 完成登录界面，用户界面，数据库创建功能		较高	4月07日	旺旺	
<input type="checkbox"/> DQL探索		普通	4月07日	麦秆	○
<input type="checkbox"/> DDL/DML		较高	4月14日	Y_	○
<input type="checkbox"/> DDL/DML		较高	4月14日	Baba Yetu	○
<input type="checkbox"/> DDL/DML		较高	4月14日	Baba Yetu	○
<input type="checkbox"/> DQL开始实现		较高	4月14日	麦秆	○

🔊

开题

任务标题	所属项目	优先级	截止时间	负责人	
<input type="checkbox"/> 完成开题答辩相关准备		最高	3月23日	旺旺	
<input checked="" type="checkbox"/> 数据库整体搭建	...	较高	3月23日	旺旺	
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	Baba Yetu	

第五周：

<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	Baba Yetu	
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	旺旺	
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	Y_	○
<input type="checkbox"/> 熟悉各自的任务		普通	3月31日	麦秆	

第六周：

<input type="checkbox"/> 完成登录界面，用户界面，数据库创建功能		较高	4月07日	旺旺	
<input type="checkbox"/> DQL探索		普通	4月07日	麦秆	○

第七周/八周：中期答辩

<input type="checkbox"/> DDL/DML		较高	4月14日	Y_ <input type="radio"/>
<input type="checkbox"/> DDL/DML		较高	4月14日	Baba Yetu <input type="radio"/>
<input type="checkbox"/> DDL/DML	***	较高	4月14日	Baba Yetu <input type="radio"/>
<input type="checkbox"/> DQL开始实现		较高	4月14日	麦秆 <input type="radio"/>
<input type="checkbox"/> 中期答辩		最高	本周五	麦秆 <input type="radio"/>
<input type="checkbox"/> 界面修改问题		普通	本周日	余旺 <input type="radio"/>

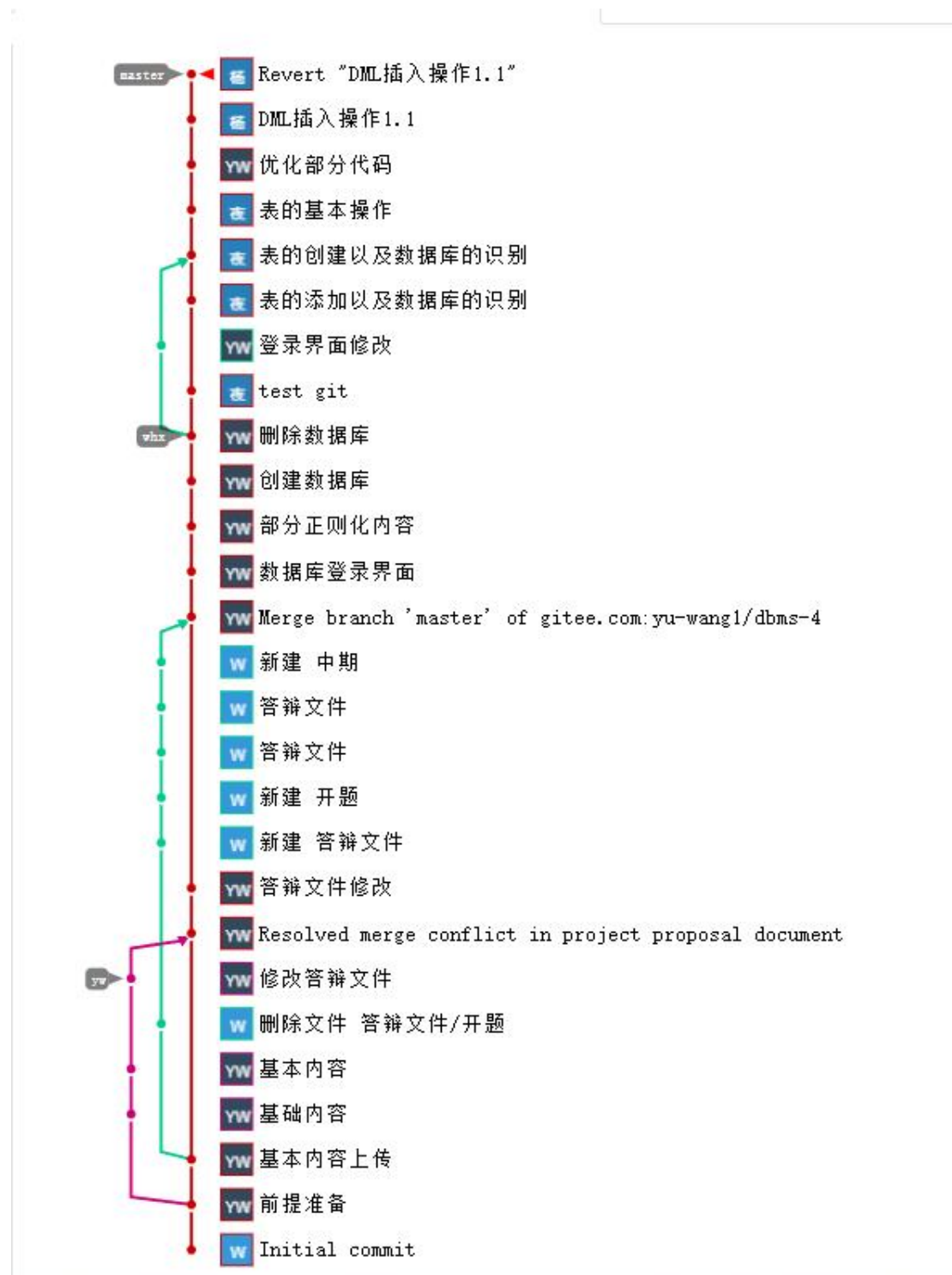
<input type="checkbox"/> 中期答辩			最高	本周五	 麦秆 <input type="radio"/>
<input type="checkbox"/> 中期答辩	...	添加	最高	本周五	 余旺
<input type="checkbox"/> 界面修改问题			普通	本周日	 余旺

Git、gittee 管理

总体提交量



提交



项目进度

小组任务估计：

- 1，总体任务基础功能基本完成，DQL 的功能完成大部分。
- 2，GUI，对于用户执行 sql 语句的界面还需要去修改，整体美工有待后期优化。
- 3，拓展功能还未开始，但拟定，日志管理，复杂 SQL，用户权限管理。

功能展示：

用户：

对于用户会在 User 文件夹下进行检查，添加用户等，对密码进行记录。

SQL 解析：

```
// 通过正则表达式读取内容
//正则化的所有语句
//1,创建数据库
std::regex createdatabase("CREATE\\s+DATABASE\\s+(\\w+)", std::regex_constants::icase);
//2,删除数据库
std::regex dropdatabase("DROP\\s+DATABASE\\s+(\\w+)", std::regex_constants::icase);

//3, 创建表
std::regex createTable("CREATE\\s+TABLE\\s+(\\w+)\\s*(([^\n])+)", regex_constants::icase);
//4 识别数据库
std::regex useDatabaseRegex("CREATE\\s+DATABASE\\s+(\\w+)", regex_constants::icase);

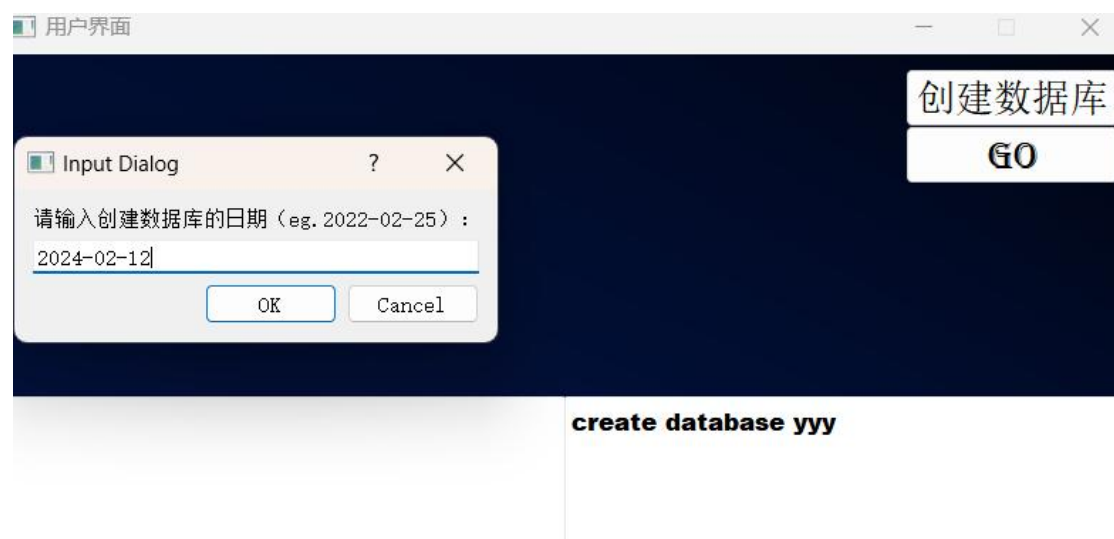
// 待匹配的字符串
//string query = "CREATE DATABASE my_database";

//5, 删除表
std::regex dropTableRegex("DROP\\s+TABLE\\s+(\\w+)", regex_constants::icase);
//6,修改表
std::regex alterTableRegex("ALTER\\s+TABLE\\s+(\\w+)\\s+(ADD|ALTER|DROP)\\s+(\\w+)(?:\\s+(\\w+(?:\\s+(\\d+\\s+)))?)?(?:\\s+(\\w+(?:\\s+(\\d+\\s+)))?)");
```

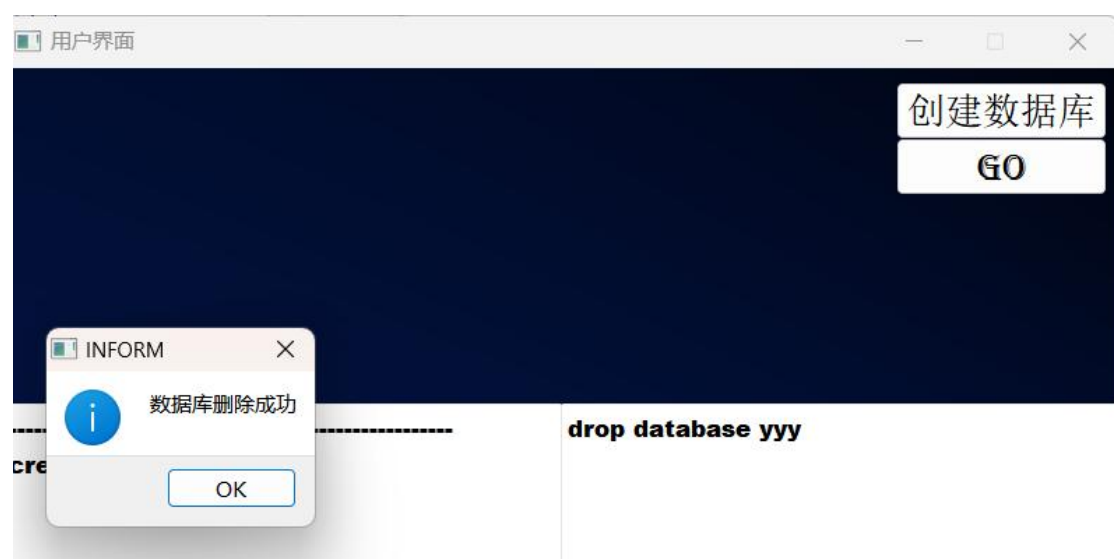
DDL:

1, 数据库的创建和删除

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。



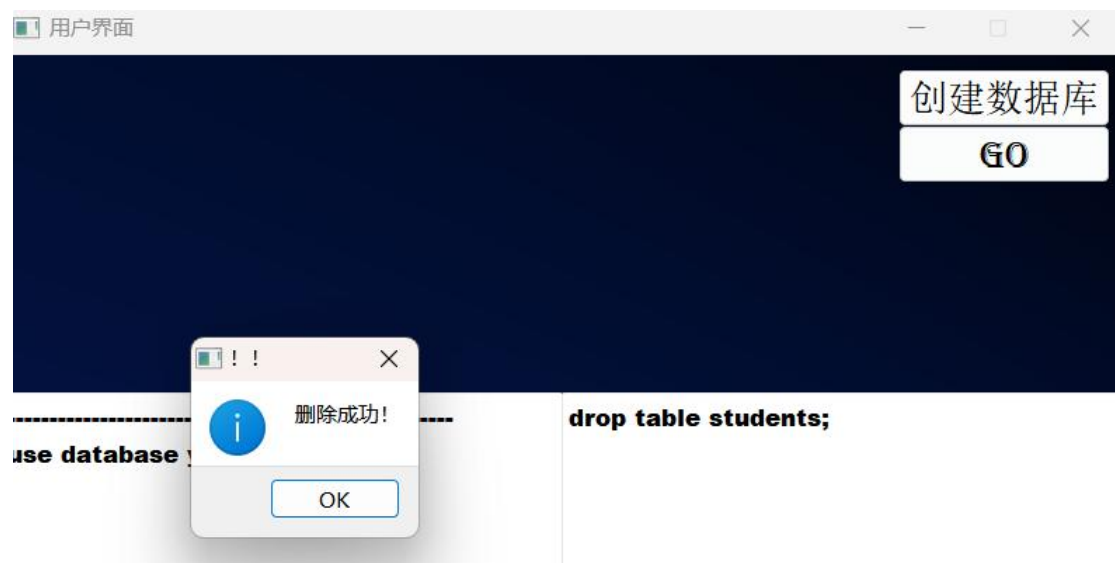
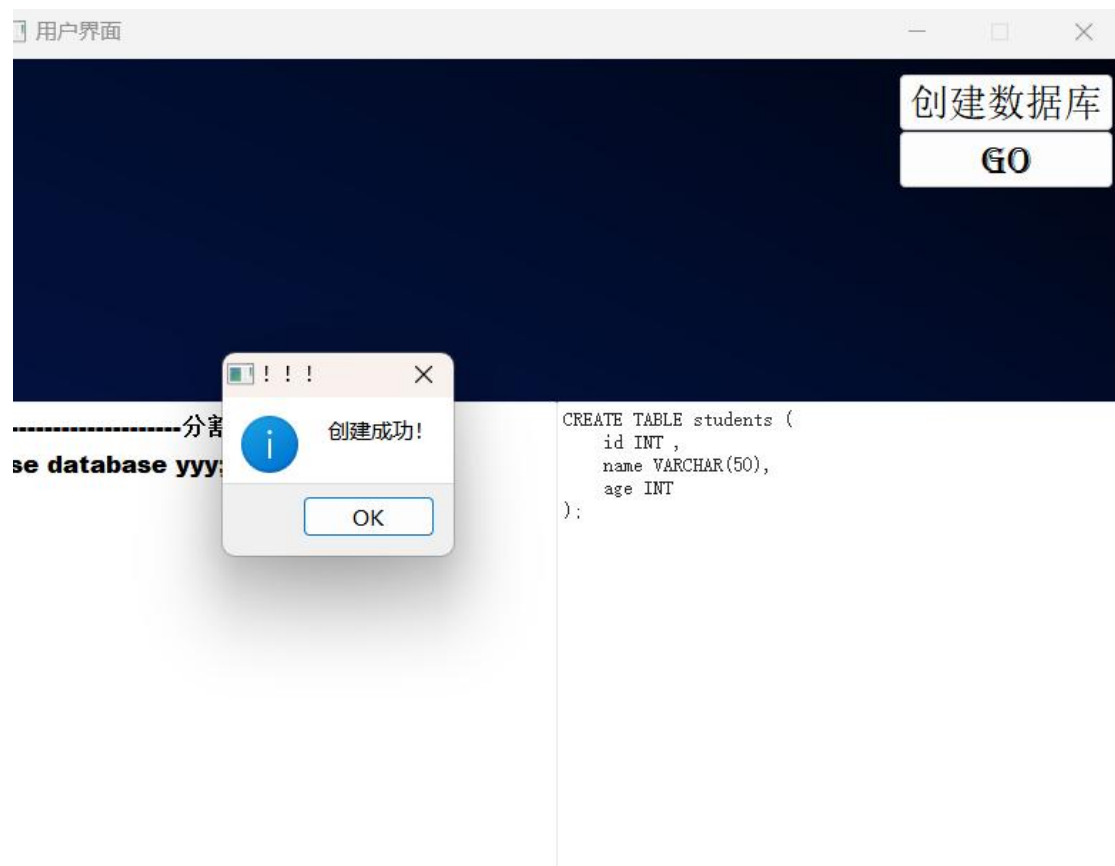
同样，当删除数据库时，会去判断数据库是否存在，存在则删除，对于存储数据库信息的 **Ruanko** 不允许删除。



2, 表的创建和删除

首先需要先 `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底下，
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。

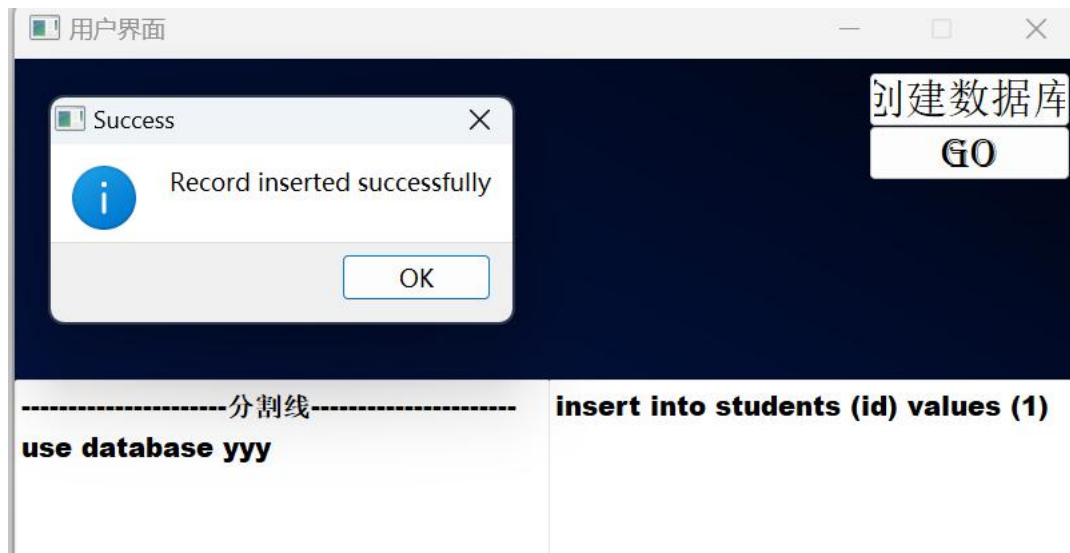


3, 字段的添加, 删除, 修改

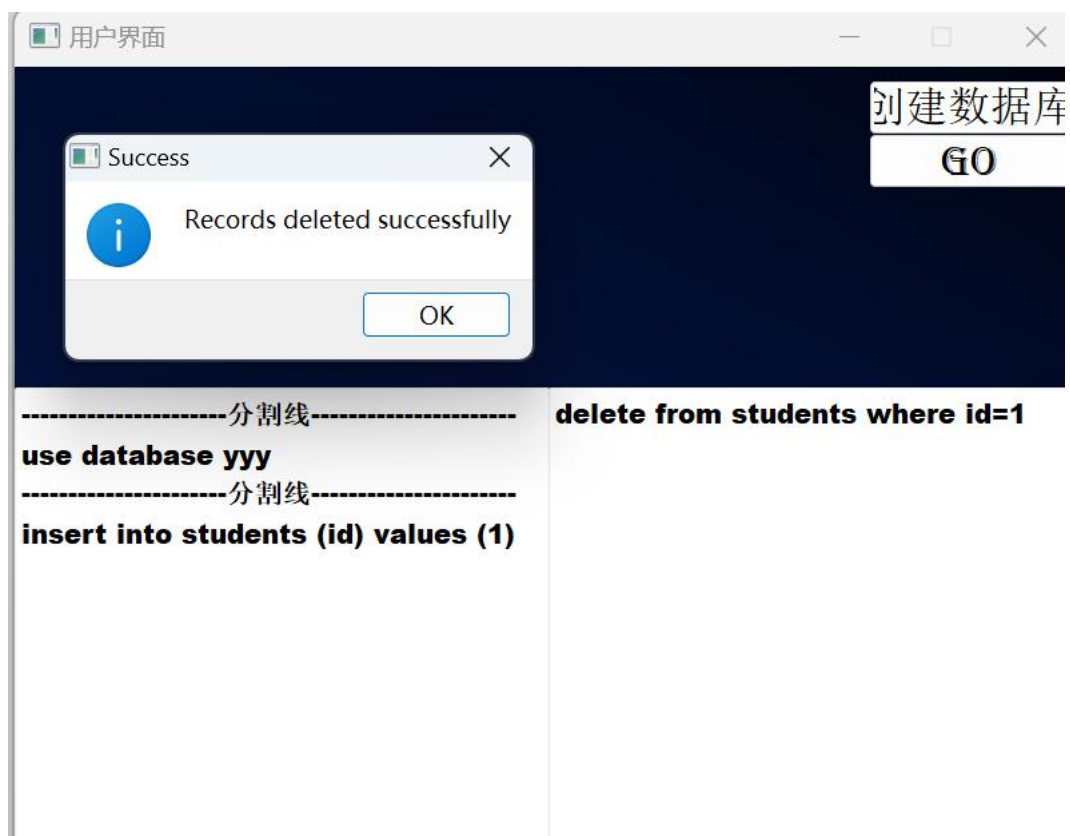


DML:

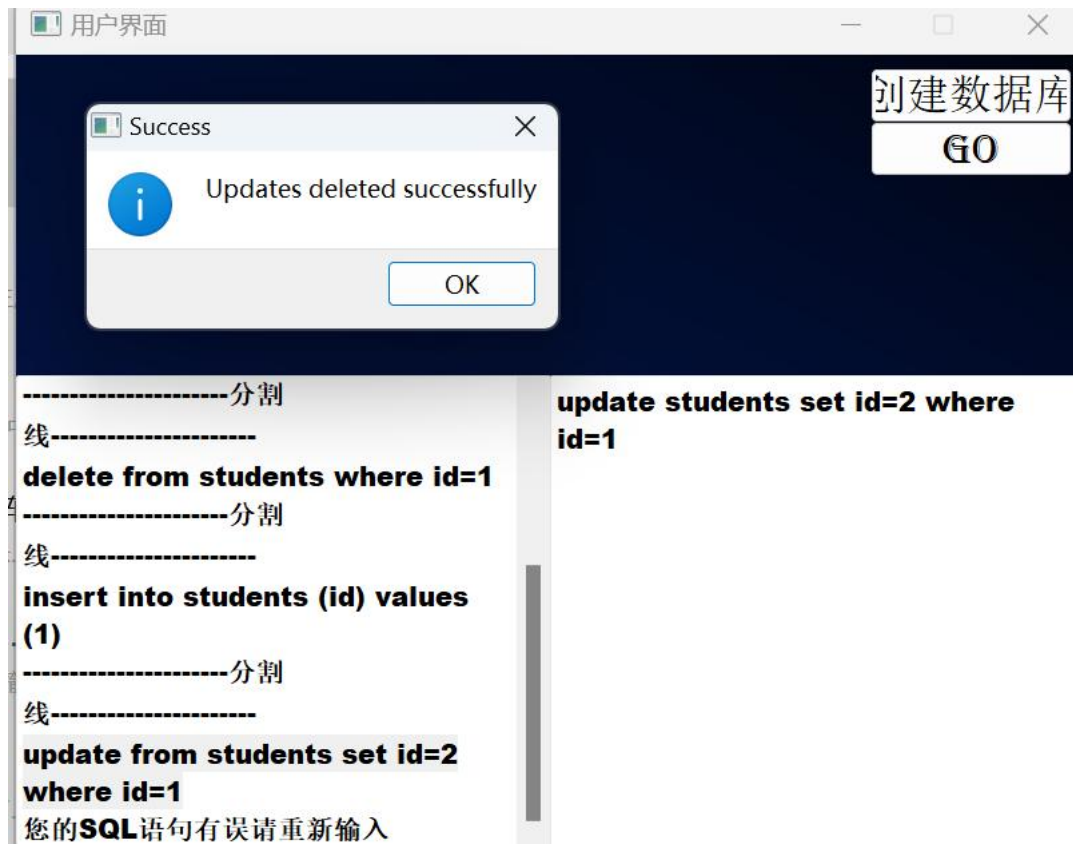
1, 完成记录插入



2, 记录删除



3, 记录修改



DQL 功能实现（简单条件）

还在实现中

- 1, 所有记录
- 2, 指定字段的所有记录
- 3, 指定字段和 where 条件的记录