《数据库系统实验》

实验报告

|  |  |
| --- | --- |
| **题目** | 实验6 |
| **姓名** | 郝裕玮 |
| **学号** | 18329015 |
| **班级** | 计科1班 |

一、实验环境

操作系统：Windows 10

DBMS：MySQL Workbench 8.0 CE

编程平台：Visual Studio 2019

二、实验内容与完成情况

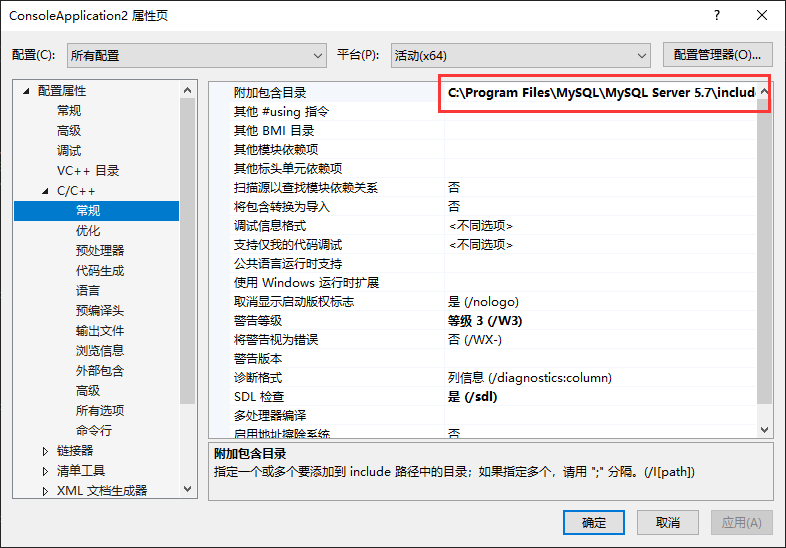
**6.1** **模拟create\_student\_table( )实现创建SC表或Course表。即实现create\_sc\_table或create\_course\_table( )子程序的功能**

**6.2模拟insert\_rows\_into\_student\_table( )实现对SC表或Course表的记录添加。即实现insert\_rows\_into\_sc\_table( )或 insert\_rows\_into\_course\_table()子程序的功能。**

首先需要对Visual Studio下的代码进行环境配置：

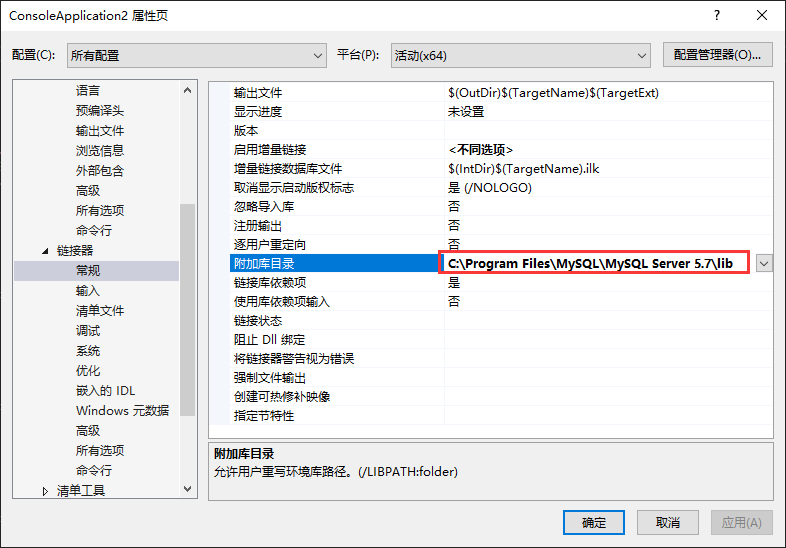
(1) 在“项目—属性—C/C++—常规—附加包含目录”中，添加如下路径：

C:\Program Files\MySQL\MySQL Server 5.7\include（见下图）

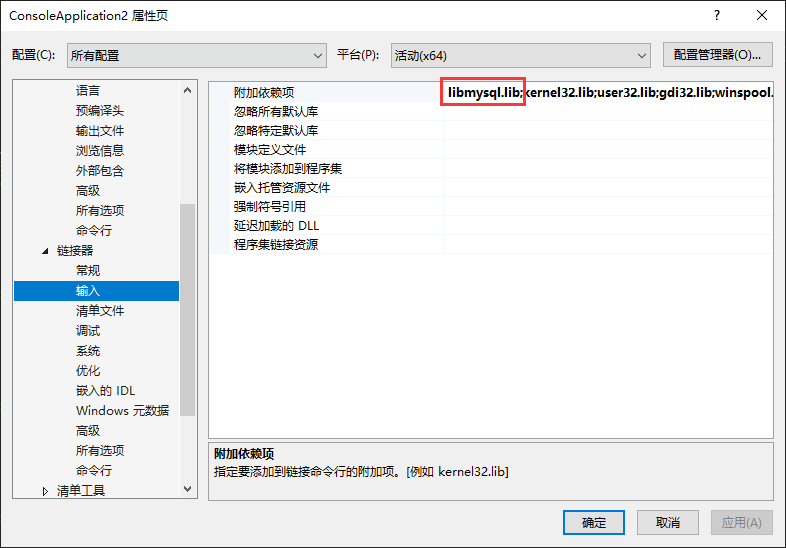


(2) 在“项目—属性—链接器—常规—附加库目录” 中，添加如下路径：

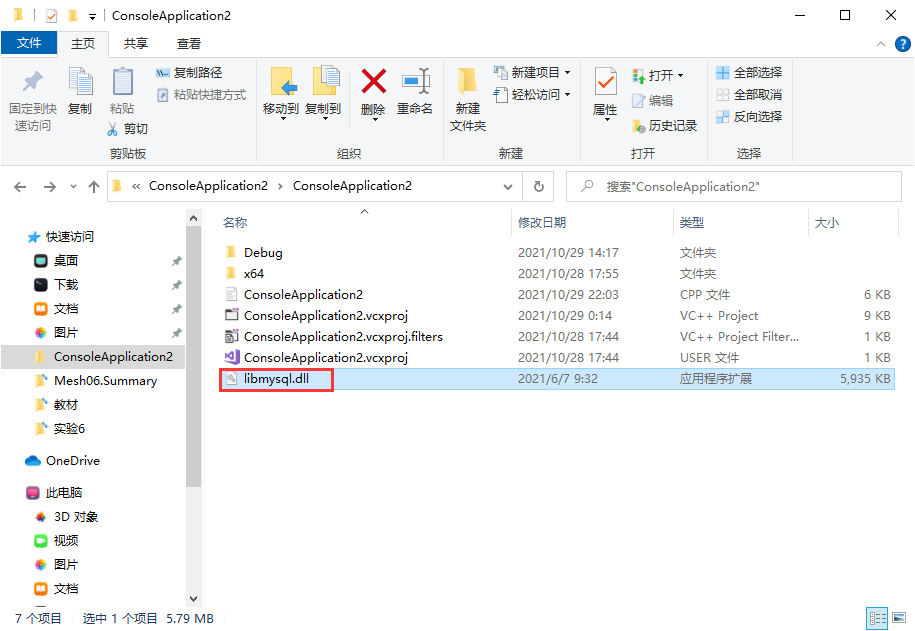
C:\Program Files\MySQL\MySQL Server 5.7\lib



(3) 在“项目—属性—链接器—输入—附加依赖项” 中，添加libmysql.lib（见下图）：



(4) 将动态链接库libmysql.dll复制到路径C:\Users\93508\source\repos\ConsoleApplication2\ConsoleApplication2下



(5)创建数据库lab6，代码如下图所示：

create database lab6

之后运行以下代码即可创建course表并向其中添加信息（代码具体思路及分析均已放在代码注释中）：

#define \_CRT\_SECURE\_NO\_WARNINGS //不添加此行会报错

#include<iostream>

#include<mysql.h>//需要调用各种mysql内置函数

#include<string.h>

#include<winsock.h>

using namespace std;

MYSQL mysql;//声明一个MYSQL类型的全局变量

void create\_course\_table() {//创建课程表course

    char yn[2];//用于存储输入的命令：Y/y or N/n

    if (mysql\_list\_tables(&mysql, "course")) {//先查看是否已经创建过course表

        //若值不为0则说明存在，打印提示信息

        cout << "course表已经存在，您是选择删除重建还是继续使用该course表？" << endl<<endl;

        cout << "1--删除重建   2--继续使用该course表"<<endl<<endl;

        cout << "请输入相关指令：";

        cin >> yn;

        cout << endl;

        if (yn[0] == '1') {

            if (mysql\_query(&mysql, "drop table course;") == 0) {

                //若输入1则执行"drop table course;"删除course表

                cout << "删除成功！" << endl << endl;

            }

            else {

                cout << "删除失败！" << endl << endl;

            }

        }

        else if (yn[0] == '2') {//若不创建表则直接返回

            return;

        }

    }

    if (mysql\_query(&mysql, "create table course(cno char(20),cname char(20),cpno char(20),ccredit char(20),primary key(cno)) engine=innodb;") == 0) {

        //输入创建course表的SQL语句，其中engine=innodb表示存储引擎是innodb，若需要使用外键则需要设置该引擎

        cout << "新的course表创建成功！" << endl << endl << endl;

    }

    else {

        cout << "新的course表创建失败！" << endl << endl << endl;

    }

}

void insert\_rows\_into\_course\_table() {

    while (1) {

        char cno[20];//课程号

        char cname[20];//课程名

        char cpno[20];//先修课程号

        char ccredit[20];//课程学分

        char yn[2]; //用来判断是否继续插入

        char strquery[100] = "insert into course(cno,cname,cpno,ccredit) values('";

        //先设置好前面一部分要插入的SQL语句，后续逐步用strcat连接扩充该SQL语句

        //当前语句为insert into course(cno,cname,cpno,ccredit) values('

        cout << "请输入课程号cno：";

        cin >> cno;

        cout << endl;

        strcat(strquery, cno);//将课程号连接到之前的语句后面

        strcat(strquery, "','");//这里补充的是','

        //连接后目前语句为：insert into course(cno,cname,cpno,ccredit) values('xxx','

        cout << "请输入课程名cname：";

        cin >> cname;

        cout << endl;

        strcat(strquery, cname);//将课程名连接到之前的语句后面

        strcat(strquery, "','");//这里补充的是','

        //连接后目前语句为：insert into course(cno,cname,cpno,ccredit) values('xxx','xxx','

        cout << "请输入先修课号cpno：";

        cin >> cpno;

        cout << endl;

        strcat(strquery, cpno);//将先修课号连接到之前的语句后面

        strcat(strquery, "','");//这里补充的是','

        //连接后目前语句为：insert into course(cno,cname,cpno,ccredit) values('xxx','xxx','xxx','

        cout << "请输入课程学分ccredit：";

        cin >> ccredit;

        cout << endl;

        strcat(strquery, ccredit);//将课程学分连接到之前的语句后面

        strcat(strquery, "');");//这里补充的是');

        //连接后目前语句为：insert into course(cno,cname,cpno,ccredit) values('xxx','xxx','xxx','xxx');

        cout << strquery << endl;//输出刚才的插入语句，检验是否有语法错误或插入信息错误

        if (mysql\_query(&mysql, strquery) == 0) {//若返回值为0则插入成功

            cout << "数据插入成功！" << endl << endl << endl;

        }

        else {

            cout << "数据插入失败！" << endl;

            cout << "失败原因：" << mysql\_error(&mysql) << endl << endl << endl;

        }

        //判断是否需要继续插入记录

        cout << "是否需要继续插入课程信息？y代表是，n代表否（不区分大小写）：";

        cin >> yn;

        cout << endl << endl;

        if (yn[0] == 'y' || yn[0] == 'Y') {

            continue;//若为Y/y则继续该while循环，继续插入课程信息

        }

        else {

            break;

        }

    }

}

int main() {

    char order[2];//存储要执行的操作1/2/3

    mysql\_init(&mysql); //初始化一个MYSQL结构

    mysql\_options(&mysql, MYSQL\_SET\_CHARSET\_NAME, "gbk");//插入这句话才能使得SQL语句可以插入中文

    if (mysql\_real\_connect(&mysql, "127.0.0.1", "root", "hyw19991201", "Lab6", 3306, 0, 0)) {

        //127.0.0.1为服务器地址，root为连接用户名，hyw19991201为密码，Lab6为数据库名，3306为连接端口号

        while (1) {

            cout << "这里是C++嵌入式SQL！" << endl << endl;

            cout << "1--创建课程表course  2--添加课程记录   3--退出系统" << endl << endl;

            cout << "请输入相关指令：";

            cin >> order;//输入命令

            cout << endl;

            if (order[0] == '1') {//为1则创建新课程表course

                create\_course\_table();

            }

            else if (order[0] == '2') {//为2则添加课程记录

                insert\_rows\_into\_course\_table();

            }

            else if (order[0] == '3') {//为3退出系统

                break;

            }

        }

    }

    else {//如果连接失败证明该数据库不存在/尚未创建/名称错误

        cout << "该数据库不存在！请检查是否已经创建该数据库！" << endl;

    }

    mysql\_close(&mysql);//访问结束后关闭SQL

    system("pause");

}

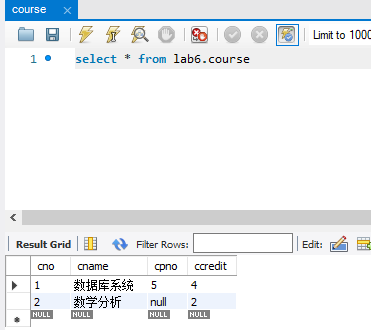
运行结果如下所示：





此时回到MySQL Workbench 8.0 CE执行下列语句查看是否创建和插入成功：

select \* from lab6.course



至此，本次实验圆满完成！