《数据库系统实验》

实验报告

|  |  |
| --- | --- |
| **题目** | 实验六 |
| **姓名** | 卜海涛 |
| **学号** | 22336016 |
| **班级** | 22级计算机科学与技术（人工智能与大数据） |

1. 实验环境：

Visual Studio Community 2022

1. 实验内容与完成情况：
   * + 1. 创建一个MySQL类型的对象，使用MYSQL\_RES命名一个的指向结果集的指针，使用MYSQL\_ROW建立row变量存放结果集的行

MYSQL mysql;

MYSQL\_RES\* result;

MYSQL\_ROW row;

//声明为全局变量，待会在主函数，功能函数中都能对它们访问

* + - 1. 开始编写main函数，使用mysql\_init()在内存中建立一个对象，用于同MySQL服务器进行交互，创建失败则终止

if(!mysql\_init(&mysql))

{

printf("cannot initialize MySQL!");

return 1;

}

* + - 1. 使用mysql\_real\_connect()与指定数据库建立连接，连接失败则终止

if(!mysql\_real\_connect(&mysql, "localhost", "root", "bht123456", "jxgl", 3306, 0, 0))

{

printf("cannot connect to database!");

return 1;

}

* + - 1. 创建系统的交互界面，根据用户的输入确定需要调用哪个函数

for(;;)

{

printf("Sample Embedded SQL for C application\n");

printf("Please select one function to execute:\n\n");

printf(" 0--exit.\n");

printf(" 1--创建学生表 6--添加成绩记录 b--删除课程记录 h--学生课程成绩表\n");

printf(" 2--创建课程表 7--修改学生记录 c--删除成绩记录 j--学生成绩统计表\n");

printf(" 3--创建成绩表 8--修改课程记录 e--显示学生记录 k--课程成绩统计表\n");

printf(" 4--添加学生记录 9--修改成绩记录 f--显示课程记录 m--数据库表名\n");

printf(" 5--添加课程记录 a--删除学生记录 g--显示成绩记录\n");

printf("\n");

fu[0] = '0';

scanf("%s",&fu);

if(fu[0]=='0') exit(0);

if(fu[0]=='1') create\_student\_table();

if(fu[0]=='2') create\_course\_table();

if(fu[0]=='3') create\_sc\_table();

if(fu[0]=='4') insert\_rows\_into\_student\_table();

if(fu[0]=='5') insert\_rows\_into\_course\_table();

if(fu[0]=='6') insert\_rows\_into\_sc\_table();

if(fu[0]=='7') current\_of\_update\_for\_student();

if(fu[0]=='8') current\_of\_update\_for\_course();

if(fu[0]=='9') current\_of\_update\_for\_sc();

if(fu[0]=='a') current\_of\_delete\_for\_student();

if(fu[0]=='b') current\_of\_delete\_for\_course();

if(fu[0]=='c') current\_of\_delete\_for\_sc();

if(fu[0]=='e') using\_cursor\_to\_list\_student();

if(fu[0]=='f') using\_cursor\_to\_list\_course();

if(fu[0]=='g') using\_cursor\_to\_list\_sc();

if(fu[0]=='h') using\_cursor\_to\_list\_s\_sc\_c();

if(fu[0]=='j') using\_cursor\_to\_total\_s\_sc();

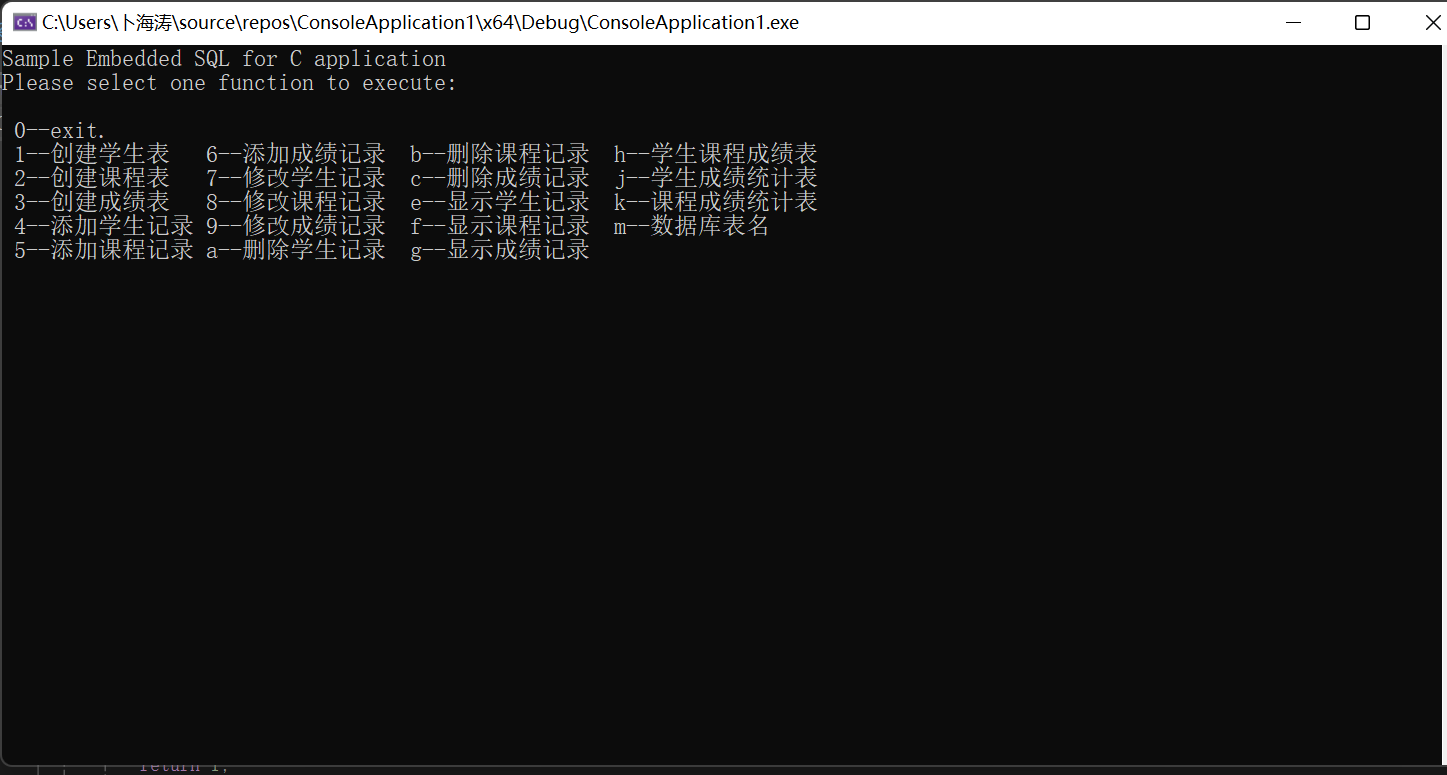
if(fu[0]=='k') using\_cursor\_to\_total\_c\_sc();

if(fu[0]=='m') using\_cursor\_to\_list\_table\_names();

system("pause");

}

交互界面视图



* + - 1. 程序最后要使用mysql\_close()结束MySQL会话，并断开与数据库的连接

mysql\_close(&mysql);

//访问完毕，关闭mysql

* + - 1. 编写创建course表的函数create\_course\_table()

int create\_course\_table()

{

char yn[2];

result = mysql\_list\_tables(&mysql, "course");

unsigned int rows = mysql\_num\_rows(result);

mysql\_free\_result(result);

if (rows>0)

{

//rows>0表示course表已经存在，考虑是否要删去原有的表

printf("The course table already exists,Do you want to delete it?\n");

printf("Delete the table? (y--yes,n--no):");

scanf("%s", &yn);

//如果选择删除,则执行MySQL语句

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

{

if (!mysql\_query(&mysql, "drop table course;"))

printf("Drop table course successfully!\n\n");

//如果Mysql语句执行失败，则报错

else

{

printf("ERROR: drop table course\n\n");

return 1;

}

}

//选择保留则继续使用原有的course表

else

return 0;

}

//执行创建course表的mysql语句

if (!mysql\_query(&mysql, "create table course(cno varchar(4) primary key,cname varchar(20),cpno varchar(4),ccredit smallint);"))

printf("create table course successfully!\n\n");

//执行失败则报错

else

{

printf("ERROR: create table course\n\n");

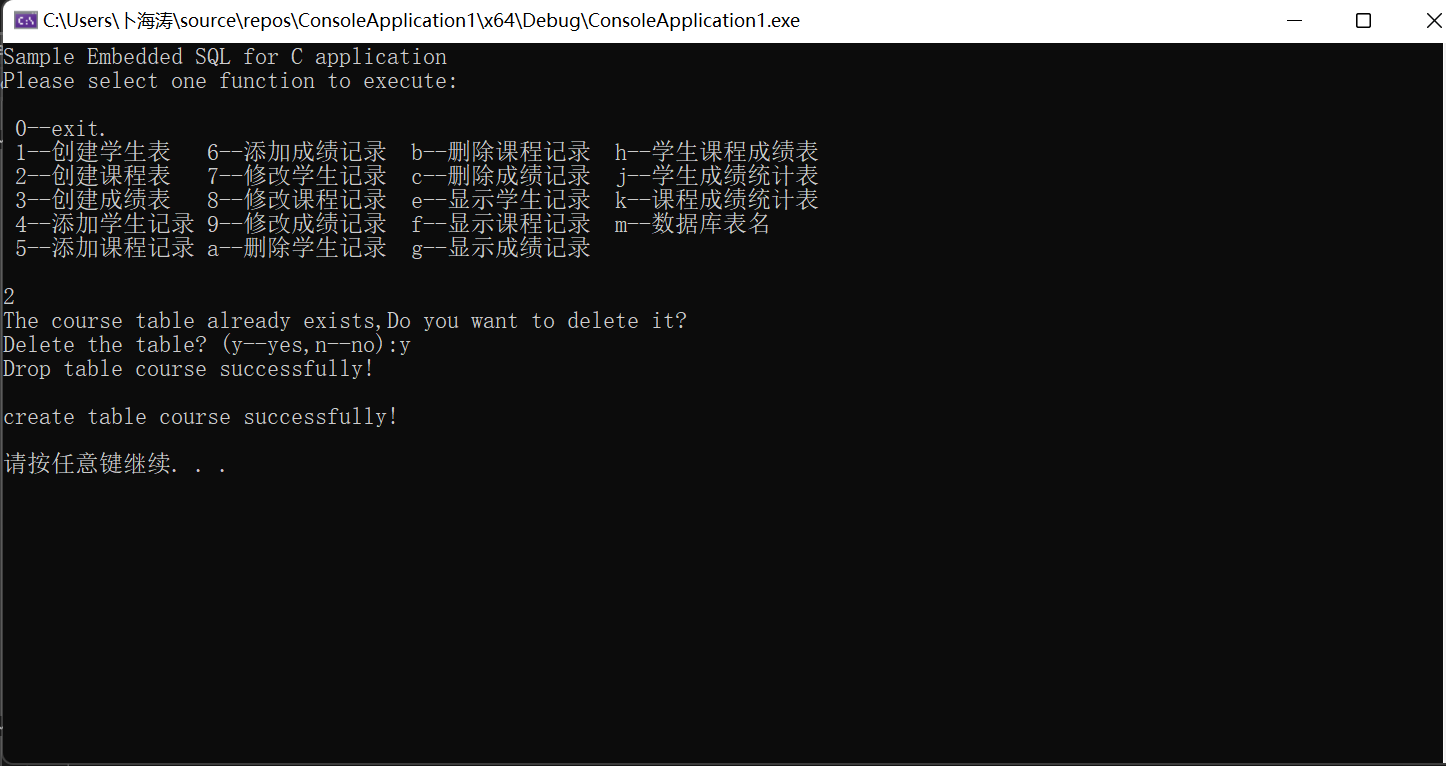
return 1;

}

return 0;

}

运行结果



* + - 1. 编写向course表中添加记录的函数insert\_rows\_into\_course\_table()

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

{

//设置数据库对于客户端发送信息的编码和解码方式

if (!mysql\_query(&mysql, "set names \'GBK\'"))

{

printf("set successfully!\n\n");

}

else

{

printf("ERROR: set\n");

return 1;

}

//初始化各个属性值

char cno[]="0000";

char cname[]="操作系统原理";

char cpno[]="0000";

char ccredit[]="0000";

//strquery字符数组用于存放查询语句

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

char strquery[1000];

strcpy(strquery, temp);

char yn[2];

while (1)

{

//用户输入要添加的属性值，然后使用strcat函数拼凑出完整的查询语句

printf("Please input cno:");

scanf("%s", cno);

strcat(strquery, cno);

strcat(strquery, "','");

printf("Please input cname:");

scanf("%s",cname);

strcat(strquery, cname);

strcat(strquery, "','");

printf("Please input cpno:");

scanf("%s", cpno);

strcat(strquery, cpno);

strcat(strquery, "',");

printf("Please input ccredit:");

scanf("%s", ccredit);

strcat(strquery, ccredit);

strcat(strquery, ");");

int test = mysql\_query(&mysql, strquery);

if (test==0)

printf("execute successfully!\n\n");

//查询语句执行失败，报错

else

{

printf("ERROR: execute\n");

return 1;

}

//是否要继续插入数据

printf("Insert again? (y--yes,n--no):");

scanf("%s", &yn);

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

break;

//若要继续插入，strquery字符数组要还原

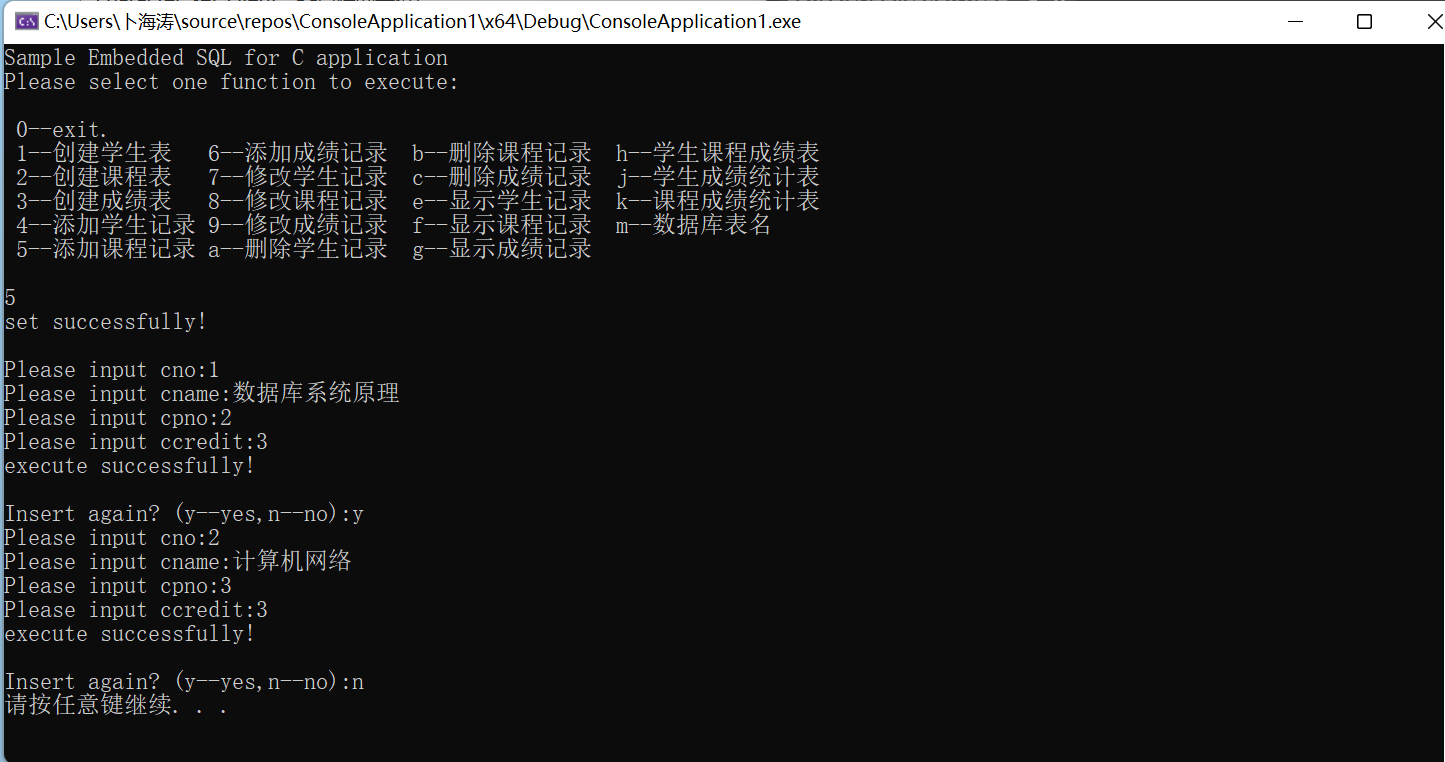
strcpy(strquery, temp);

}

return 0;

}

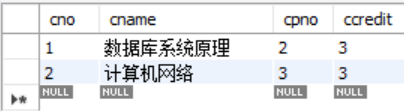
运行结果



运行前



运行后



三、实验源代码：

#include<mysql.h>

#include<stdio.h>

#include<stdlib.h>

#include<winsock.h>

#include<iostream>

#pragma warning(disable:4996)

using namespace std;

MYSQL mysql;

MYSQL\_RES\* result;

MYSQL\_ROW row;

//声明为全局变量，待会在主函数，功能函数中都能对它们访问

int create\_student\_table() { return 0; }

int create\_course\_table()

{

char yn[2];

result = mysql\_list\_tables(&mysql, "course");

unsigned int rows = mysql\_num\_rows(result);

mysql\_free\_result(result);

if (rows>0)

{

//rows>0表示course表已经存在，考虑是否要删去原有的表

printf("The course table already exists,Do you want to delete it?\n");

printf("Delete the table? (y--yes,n--no):");

scanf("%s", &yn);

//如果选择删除,则执行MySQL语句

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

{

if (!mysql\_query(&mysql, "drop table course;"))

printf("Drop table course successfully!\n\n");

//如果Mysql语句执行失败，则报错

else

{

printf("ERROR: drop table course\n\n");

return 1;

}

}

//选择保留则继续使用原有的course表

else

return 0;

}

//执行创建course表的mysql语句

if (!mysql\_query(&mysql, "create table course(cno varchar(4) primary key,cname varchar(20),cpno varchar(4),ccredit smallint);"))

printf("create table course successfully!\n\n");

//执行失败则报错

else

{

printf("ERROR: create table course\n\n");

return 1;

}

return 0;

}

int create\_sc\_table() { return 0; }

int insert\_rows\_into\_student\_table() { return 0; }

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

{

//设置数据库对于客户端发送信息的编码和解码方式

if (!mysql\_query(&mysql, "set names \'GBK\'"))

{

printf("set successfully!\n\n");

}

else

{

printf("ERROR: set\n");

return 1;

}

//初始化各个属性值

char cno[]="0000";

char cname[]="操作系统原理";

char cpno[]="0000";

char ccredit[]="0000";

//strquery字符数组用于存放查询语句

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

char strquery[1000];

strcpy(strquery, temp);

char yn[2];

while (1)

{

//用户输入要添加的属性值，然后使用strcat函数拼凑出完整的查询语句

printf("Please input cno:");

scanf("%s", cno);

strcat(strquery, cno);

strcat(strquery, "','");

printf("Please input cname:");

scanf("%s",cname);

strcat(strquery, cname);

strcat(strquery, "','");

printf("Please input cpno:");

scanf("%s", cpno);

strcat(strquery, cpno);

strcat(strquery, "',");

printf("Please input ccredit:");

scanf("%s", ccredit);

strcat(strquery, ccredit);

strcat(strquery, ");");

int test = mysql\_query(&mysql, strquery);

if (test==0)

printf("execute successfully!\n\n");

//查询语句执行失败，报错

else

{

printf("ERROR: execute\n");

return 1;

}

//是否要继续插入数据

printf("Insert again? (y--yes,n--no):");

scanf("%s", &yn);

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

break;

//若要继续插入，strquery字符数组要还原

strcpy(strquery, temp);

}

return 0;

}

int insert\_rows\_into\_sc\_table() { return 0; }

int current\_of\_update\_for\_student() { return 0; }

int current\_of\_update\_for\_course() { return 0; }

int current\_of\_update\_for\_sc() { return 0; }

int current\_of\_delete\_for\_student() { return 0; }

int current\_of\_delete\_for\_course() { return 0; }

int current\_of\_delete\_for\_sc() { return 0; }

int using\_cursor\_to\_list\_student() { return 0; }

int using\_cursor\_to\_list\_course() { return 0; }

int using\_cursor\_to\_list\_sc() { return 0; }

int using\_cursor\_to\_list\_s\_sc\_c() { return 0; }

int using\_cursor\_to\_total\_s\_sc() { return 0; }

int using\_cursor\_to\_total\_c\_sc() { return 0; }

int using\_cursor\_to\_list\_table\_names() { return 0; }

int main(int argc, char\*\* argv, char\*\* envp)

{

//SetConsoleOutputCP(CP\_UTF8);

char fu[2];

if(!mysql\_init(&mysql))

{

printf("cannot initialize MySQL!");

return 1;

}

//获得或初始化一个MYSQL结构

if(!mysql\_real\_connect(&mysql, "localhost", "root", "bht123456", "jxgl", 3306, 0, 0))

{

printf("cannot connect to database!");

return 1;

}

for(;;)

{

printf("Sample Embedded SQL for C application\n");

printf("Please select one function to execute:\n\n");

printf(" 0--exit.\n");

printf(" 1--创建学生表 6--添加成绩记录 b--删除课程记录 h--学生课程成绩表\n");

printf(" 2--创建课程表 7--修改学生记录 c--删除成绩记录 j--学生成绩统计表\n");

printf(" 3--创建成绩表 8--修改课程记录 e--显示学生记录 k--课程成绩统计表\n");

printf(" 4--添加学生记录 9--修改成绩记录 f--显示课程记录 m--数据库表名\n");

printf(" 5--添加课程记录 a--删除学生记录 g--显示成绩记录\n");

printf("\n");

fu[0] = '0';

scanf("%s",&fu);

if(fu[0]=='0') exit(0);

if(fu[0]=='1') create\_student\_table();

if(fu[0]=='2') create\_course\_table();

if(fu[0]=='3') create\_sc\_table();

if(fu[0]=='4') insert\_rows\_into\_student\_table();

if(fu[0]=='5') insert\_rows\_into\_course\_table();

if(fu[0]=='6') insert\_rows\_into\_sc\_table();

if(fu[0]=='7') current\_of\_update\_for\_student();

if(fu[0]=='8') current\_of\_update\_for\_course();

if(fu[0]=='9') current\_of\_update\_for\_sc();

if(fu[0]=='a') current\_of\_delete\_for\_student();

if(fu[0]=='b') current\_of\_delete\_for\_course();

if(fu[0]=='c') current\_of\_delete\_for\_sc();

if(fu[0]=='e') using\_cursor\_to\_list\_student();

if(fu[0]=='f') using\_cursor\_to\_list\_course();

if(fu[0]=='g') using\_cursor\_to\_list\_sc();

if(fu[0]=='h') using\_cursor\_to\_list\_s\_sc\_c();

if(fu[0]=='j') using\_cursor\_to\_total\_s\_sc();

if(fu[0]=='k') using\_cursor\_to\_total\_c\_sc();

if(fu[0]=='m') using\_cursor\_to\_list\_table\_names();

system("pause");

}

mysql\_close(&mysql);

//访问完毕，关闭mysql

return 0;

}