VC++ C++ 数据库操作类API以及主要流程

# 主要流程

## 创建数据库操作结构体对象并初始化

数据库操作类结构体定义：

typedef struct MYSQL {

NET net; /\* Communication parameters \*/

unsigned char \*connector\_fd; /\* ConnectorFd for SSL \*/

char \*host, \*user, \*passwd, \*unix\_socket, \*server\_version, \*host\_info;

char \*info, \*db;

struct CHARSET\_INFO \*charset;

MYSQL\_FIELD \*fields;

struct MEM\_ROOT \*field\_alloc;

uint64\_t affected\_rows;

uint64\_t insert\_id; /\* id if insert on table with NEXTNR \*/

uint64\_t extra\_info; /\* Not used \*/

unsigned long thread\_id; /\* Id for connection in server \*/

unsigned long packet\_length;

unsigned int port;

unsigned long client\_flag, server\_capabilities;

unsigned int protocol\_version;

unsigned int field\_count;

unsigned int server\_status;

unsigned int server\_language;

unsigned int warning\_count;

struct st\_mysql\_options options;

enum mysql\_status status;

enum enum\_resultset\_metadata resultset\_metadata;

bool free\_me; /\* If free in mysql\_close \*/

bool reconnect; /\* set to 1 if automatic reconnect \*/

/\* session-wide random string \*/

char scramble[SCRAMBLE\_LENGTH + 1];

LIST \*stmts; /\* list of all statements \*/

const struct MYSQL\_METHODS \*methods;

void \*thd;

/\*

Points to boolean flag in MYSQL\_RES or MYSQL\_STMT. We set this flag

from mysql\_stmt\_close if close had to cancel result set of this object.

\*/

bool \*unbuffered\_fetch\_owner;

void \*extension;

} MYSQL;

列子：

MYSQL mSQL;//数据库句柄

mysql\_init(&mSQL);//初始化数据库操作结构体对象

初始化函数：mysql\_init(MYSQL \*mysql)

参数：数据库操作结构体对象指针

作用：通过数据库操作结构体对象指针初始化传入的数据库操作结构体对象

## 2 连接数据库并关闭

例子：

MYSQL mysql;

mysql\_init(&mysql);

MYSQL\* t = mysql\_real\_connect(&mysql,"127.0.0.1", "root", "123456", "mysql", 3306, NULL, 0);

if (t == NULL)

{

//返回空指针则连接不成功

}

else

{

//连接成功

DialogTest dialog\_test;

dialog\_test.DoModal();

}

mysql\_close(&mysql);

连接函数：

MYSQL \*mysql\_real\_connect(

MYSQL \*mysql, //1、数据库操作结构体指针

Const char\* host,//2、主机ip

Const char\* user, //3、用户名

Const char\* password,//4、用户密码

Const char\* database, //5、数据库名

Unsigned int port, //6、端口号

Const char\* unix\_socket, //7、unix连接方式。

Unsigned long clientflag //8、Mysql运行为ODBC数据库的标记，一般取0

）

如果unix\_socket不是NULL，该字符串描述了应使用的套接字或命名管道。注意，“host”参数决定了连接的类型。

关闭连接函数：

Void mysql\_close(MYSQL \*MySQL)

## 3、常见的数据库操作”增删改查”

查询函数：

Int mysql\_querry(MYSQL \*mysql,const char\* sql)

参数：

1. 数据库操作结构体对象指针
2. 数据库操作语句支持增删查改

储存查询返回的结果集函数：

MYSQL\_RES\* mysql\_store\_result(MYSQL \*mysql)

参数：

1、数据库操作结构体对象指针

返回值：MSQL\_RES \* 即指向保存查询结果的指针

释放查询结果空间函数：

Void mysql\_free\_result(MYSQL\_RES \*mysql\_res)

//关闭连接前释放

获取元组即行对象函数：

MYSQL\_ROW mysql\_fetch\_row(MYSQL\_RES \* mysql\_res)

参数：

1. MSQL\_RES \* 即指向保存查询结果的指针

作用：

取得一个元组并将游标向下移动一行

返回值：

若超出表的行数则返回null

获取结果集元组列数函数：

Unsigned int mysql\_num\_filds(MYSQL\_RES \*mysql\_res)

查询：

Select \* from TableName where

例子：

MYSQL mysql;

MYSQL\_RES\* res;//查询结果集指针

MYSQL\_ROW row; //元组实例

mysql\_init(&mysql);//初始化

int temp = 0;

int times = 0;

char\* p = NULL;

const char\* search\_sql = "select \* from mysqltest";

MYSQL\* t = mysql\_real\_connect(&mysql,"127.0.0.1", "root", "123456", "test", 3306,NULL,0);//连接数据库

if (t == NULL)

{

//DialogTest dialog\_test;

//dialog\_test.DoModal();

const char\* str= mysql\_error(&mysql);

OutputDebugString((LPCWSTR)str);

}

else

{

mysql\_query(&mysql, "set names gbk");

temp = mysql\_query(&mysql, search\_sql);//执行数据库语句

res = mysql\_store\_result(&mysql);//储存查询结果

while ((row = mysql\_fetch\_row(res)))//遍历结果集

{

times++;//统计结果集行数

int mun = mysql\_num\_fields(res);//获取结果集列数

for (int i = 1; i < mun; i++)

{

p=row[1];

}

}

mysql\_free\_result(res);//释放结果集

}

mysql\_close(&mysql);

}

插入语句

Insert into table values()

例子：

// TODO: 在此添加命令处理程序代码

MYSQL mysql;

MYSQL\_RES\* res;//查询结果集指针

MYSQL\_ROW row; //元组实例

mysql\_init(&mysql);//初始化

int temp = 0;

int times = 0;

char\* p = NULL;

char\* insert\_sql;

CString str(L"insert into mysqltest values(4,'黄老爷',45");

const char\* search\_sql = "insert into mysqltest values(4,'黄老爷',45)";

MYSQL\* t = mysql\_real\_connect(&mysql,"127.0.0.1", "root", "123456", "test", 3306,NULL,0);//连接数据库

if (t == NULL)

{

//DialogTest dialog\_test;

//dialog\_test.DoModal();

const char\* str= mysql\_error(&mysql);

OutputDebugString((LPCWSTR)str);

}

else

{

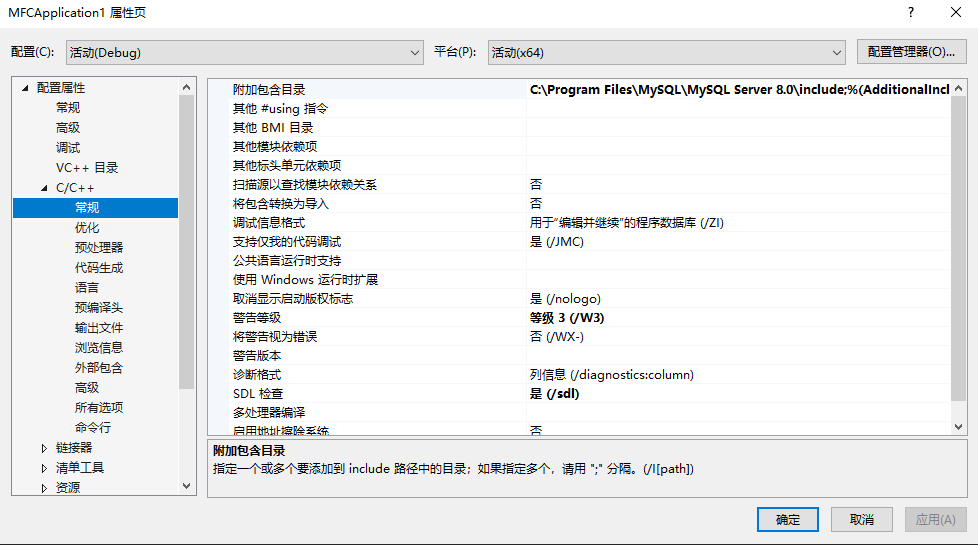
mysql\_query(&mysql, "set names gbk");

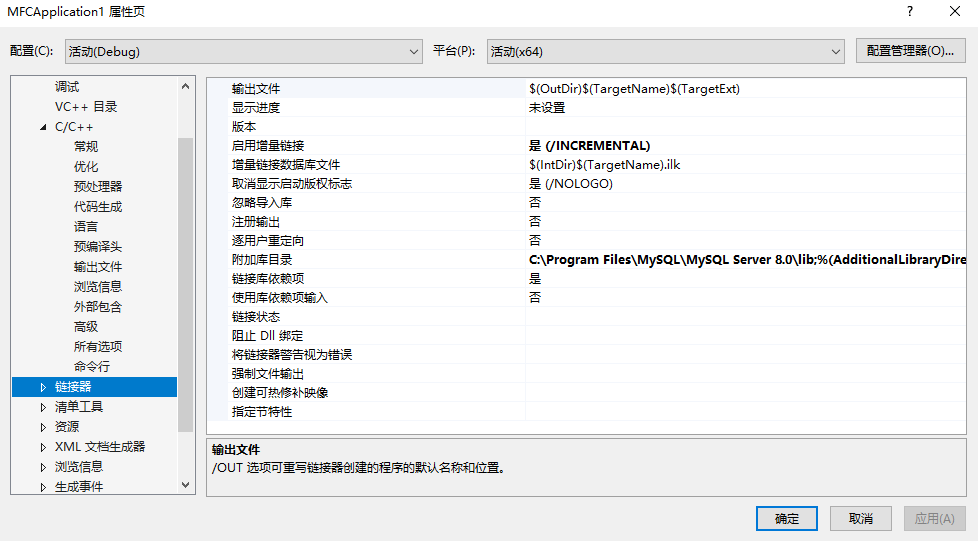
temp = mysql\_query(&mysql, search\_sql);//执行数据库语句

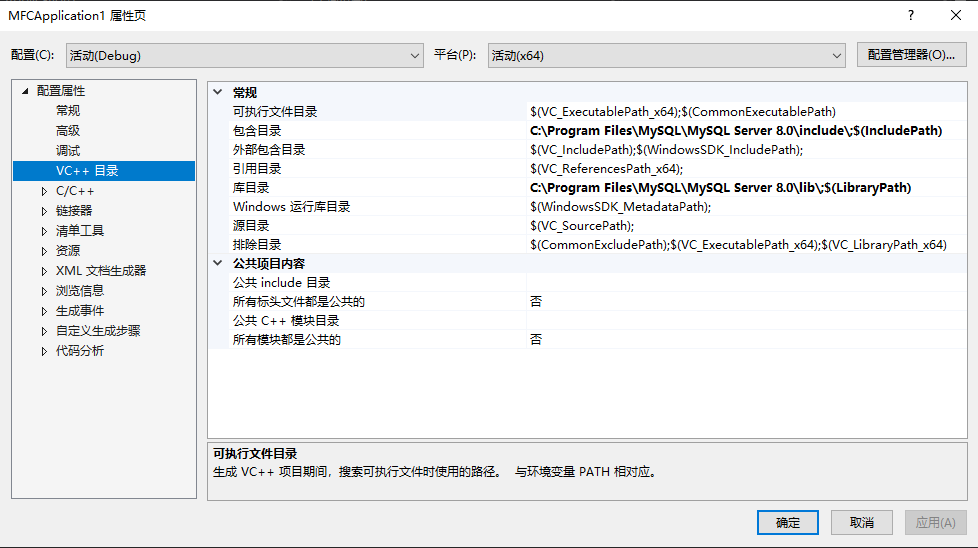
}

mysql\_close(&mysql);

# 二 visual studio2019 mysql环境配置







将mysql安装目录下的libmysql.lib、libmysql.dill复制到C:\Windows\System32\下

使用mysqlAPI前须先引入头文件和库文件

#include "mysql.h"

#pragma comment(lib,"libmysql.lib")

#pragma comment(lib,"wsock32.lib")