<https://www.cnblogs.com/uolo/p/6115448.html>

c语言操作Mysql数据库，主要就是为了实现对数据库的增、删、改、查等操作，操作之前，得先连接数据库啊，而连接数据库主要有两种方法。一、使用mysql本身提供的API,在mysql的安装目录中可可以看到大量的头文件、lib文件、dll文件，这说明mysql原生就支持了c语言，操作起来相当简单。二、使用win32 api(这里只探讨windows平台),主要是ODBC。

　　ODBC API 接口是(Open Database Connectivity)开放式数据库接口，它建立了一组规范，并提供了一组对数据库访问的标准API，这些API利用SQL来完成其大部分任务。ODBC本身也提供了对SQL语言的支持，用户可以直接将SQL语句送给ODBC。这个基于ODBC的应用程序对数据库的操作不依赖任何DBMS，不直接与DBMS打交道，所有的数据库操作由对应的DBMS的ODBC驱动程序完成。也就是说，不论是FoxPro、Access还是Oracle数据库，均可用ODBC API进行访问。由此可见，ODBC的最大优点是能以统一的方式处理所有的数据库。

　　这里先从第一种方法开始，先认识基本的几个api函数。

　　1　　MYSQL \*mysql\_init(MYSQL \*mysql)

　　　　功能: 分配或初始化MYSQL对象。

　　　　参数:mysql

　　　　待初始化的MYSQ对象，将对象地址传入，NULL指针，该函数将分配、初始化、并返回新对象。否则，将初始化对象，并返回对象的地址。

　　　　用法实例:

　　　　MYSQL 　mysql;

　　　　mysql\_init(&mysql); // 或者 mysql = mysql\_init(NULL);

　　2　　MYSQL \*mysql\_real\_connect(MYSQL \*mysql, const char \*host, const char \*user, const char \*passwd, const char \*db, unsigned int port, const char \*unix\_socket, unsigned long client\_flag)

　　　　功能:尝试与运行在主机上的MySQL数据库引擎建立连接，这个函数参数很多啊。看名字就可以是什么意思了

　　　　参数：mysql 前面一个函数的返回的mysql实例句柄。host 要连接的数据库的主机，可以是ip地址或主机名。user 表示登录数据库的用户名

passwd 登录的密码。db 就是访问的数据库。port mysql的tcp/ip端口默认是3306.unix\_socket 表示连接类型。client\_flag 暂时为0即可。

　　3　　int mysql\_query(MYSQL \*mysql,const char \*query)

　　　　　功能:根据query查询语句执行查询数据库

　　　　　参数: mysql mysql的实例。query 查询语句字符串

　　　　返回值: 成功返回0，失败返回非0

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

　　　　功能:得到查询的结果集,对于成功检索了数据的每个查询（SELECT、SHOW、DESCRIBE、EXPLAIN、CHECK TABLE等），必须调用mysql\_store\_result()或mysql\_use\_result()

　　　　参数：mysql前面的mysql实例

　　　　返回值：成功返回MYSQL\_RES结构体，该结构体中保存查询的结果

　　5　　MYSQL\_ROW mysql\_fetch\_row(MYSQL\_RES \*result)

　　　　功能:mysql\_store\_result得到的结果结构中提取一行，并把它放到一个行结构中。

　　　　返回值:下一行的一个MYSQL\_ROW结构当数据用完或发生错误时返回null

下面是简单实例代码

#include <stdio.h>   
#include <stdlib.h>   
#include<Windows.h>  
#include "C:\Program Files\MySQL\MySQL Server 5.7\include\mysql.h" //包含MySQL所需要的头文件   
#pragma comment(lib, "libmysql.lib")

int main(void)   
{   
　　MYSQL mysql, \*sock; //声明MySQL的句柄   
　　const char \* host = "127.0.0.1"; //因为是作为本机测试，所以填写的是本地IP   
　　const char \* user = "root"; //这里改为你的用户名，即连接MySQL的用户名   
　　const char \* passwd = "root"; //这里改为你的用户密码   
　　const char \* db = "test"; //这里改为你要连接的数据库的名字,一个数据可能有几张表  
　　unsigned int port = 3306; //这是MySQL的服务器的端口，如果你没有修改过的话就是3306。   
　　const char \* unix\_socket = NULL; //unix\_socket这是unix下的，我在Windows下，所以就把它设置为NULL   
　　unsigned long client\_flag = 0; //这个参数一般为0   
  
　　const char \* i\_query = "select \* from person"; //查询语句，从那个表中查询,这里后面没有;  
  
　　MYSQL\_RES \* result; //保存结果集的  
　　MYSQL\_ROW row; //代表的是结果集中的一行

　　//my\_ulonglong row;  
  
　　mysql\_init(&mysql); //连接之前必须使用这个函数来初始化   
　　if ( (sock = mysql\_real\_connect(&mysql, host, user, passwd, db, port, unix\_socket, client\_flag) ) == NULL ) //连接MySQL   
　　{   
　　　　printf("fail to connect mysql \n");   
　　　　fprintf(stderr, " %s\n", mysql\_error(&mysql));   
　　　　exit(1);   
　　}   
　　else   
　　{   
　　　　fprintf(stderr, "connect ok!!\n");   
　　}   
  
　　if ( mysql\_query(&mysql, i\_query) != 0 ) //如果连接成功，则开始查询 .成功返回0  
　　{   
　　　　fprintf(stderr, "fail to query!\n");   
　　　　exit(1);   
　　}   
　　else   
　　{   
　　　　if ( (result = mysql\_store\_result(&mysql)) == NULL ) //保存查询的结果   
　　　　{   
　　　　　　fprintf(stderr, "fail to store result!\n");   
　　　　　　exit(1);   
　　　　}   
　　　　else   
　　　　{   
　　　　　　　　while ( (row = mysql\_fetch\_row(result)) != NULL ) //读取结果集中的数据，返回的是下一行。因为保存结果集时，当前的游标在第一行【之前】   
　　　　　　　　{   
　　　　　　　　　　printf("name is %s\t", row[0]); //打印当前行的第一列的数据   
　　　　　　　　　　printf("age is %s\t\n", row[1]); //打印当前行的第二列的数据

　　　　　　　　　　//row = mysql\_num\_row(result);  
　　　　　　　　　　//printf("%lu\n", mysql\_num\_row(result));  
　　　　　　　　}   
　　　　}   
  
　　}

　　mysql\_free\_result(result); //释放结果集   
　　mysql\_close(sock); //关闭连接   
　　system("pause");   
　　exit(EXIT\_SUCCESS);   
  
}