

MySql5.7.17下载安装

author: 岳石磊 copyright: 科林明伦 内部资料禁止外泄

1. 下载

进入MYSQL官网下载: <https://dev.mysql.com/downloads/mysql/>
如果选择最新版直接点击【Go to Download Page】, 下载离线安装包。

[General Availability \(GA\) Releases](#)[Archives](#)

MySQL Community Server 8.0.29

Select Operating System:

Microsoft Windows

[Looking for previous GA versions?](#)

Recommended Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms.
In One Package.



Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

Other Downloads:

Windows (x86, 64-bit), ZIP Archive

8.0.29

214.0M

[Download](#)

(mysql-8.0.29-winx64.zip)

MD5: 1d0a93f5e6cd89f252693963ef1ab5da | [Signature](#)

Windows (x86, 64-bit), ZIP Archive

8.0.29

538.6M

[Download](#)

Debug Binaries & Test Suite

(mysql-8.0.29-winx64-debug-test.zip)

MD5: cdbdb54f9332255ca10e659d694a6c25 | [Signature](#)



We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

[General Availability \(GA\) Releases](#)[Archives](#)

MySQL Installer 8.0.29

Select Operating System:

Microsoft Windows

[Looking for previous GA versions?](#)

Windows (x86, 32-bit), MSI Installer

8.0.29

2.3M

[Download](#)

(mysql-installer-web-community-8.0.29.0.msi)

MD5: 4f735569267527de028d9e8d977f33d1 | [Signature](#)

Windows (x86, 32-bit), MSI Installer

8.0.29

439.6M

[Download](#)

(mysql-installer-community-8.0.29.0.msi)

MD5: 3f4def7aef5e2e030e2dd62e784f246 | [Signature](#)



We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

如果想选择其他版本下载，点击【Archives】，选择对应的版本进行下载。

MySQL Community Downloads

MySQL Installer

General Availability (GA) Releases **Archives** ⓘ

MySQL Installer 8.0.29

Select Operating System:
Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.29.0.msi)	8.0.29	2.3M	Download
MD5: 4f735569267527dec28d9e8d977f33d1 Signature			
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.29.0.msi)	8.0.29	439.6M	Download
MD5: 3f4def7aef5e2e030e2dd62e784f246 Signature			

MySQL Installer (Archived Versions)



Please note that these are old versions. New releases will have recent bug fixes and features!

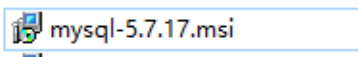
To download the latest release of MySQL Installer, please visit [MySQL Downloads](#).

Product Version: 5.7.17

Operating System: Microsoft Windows

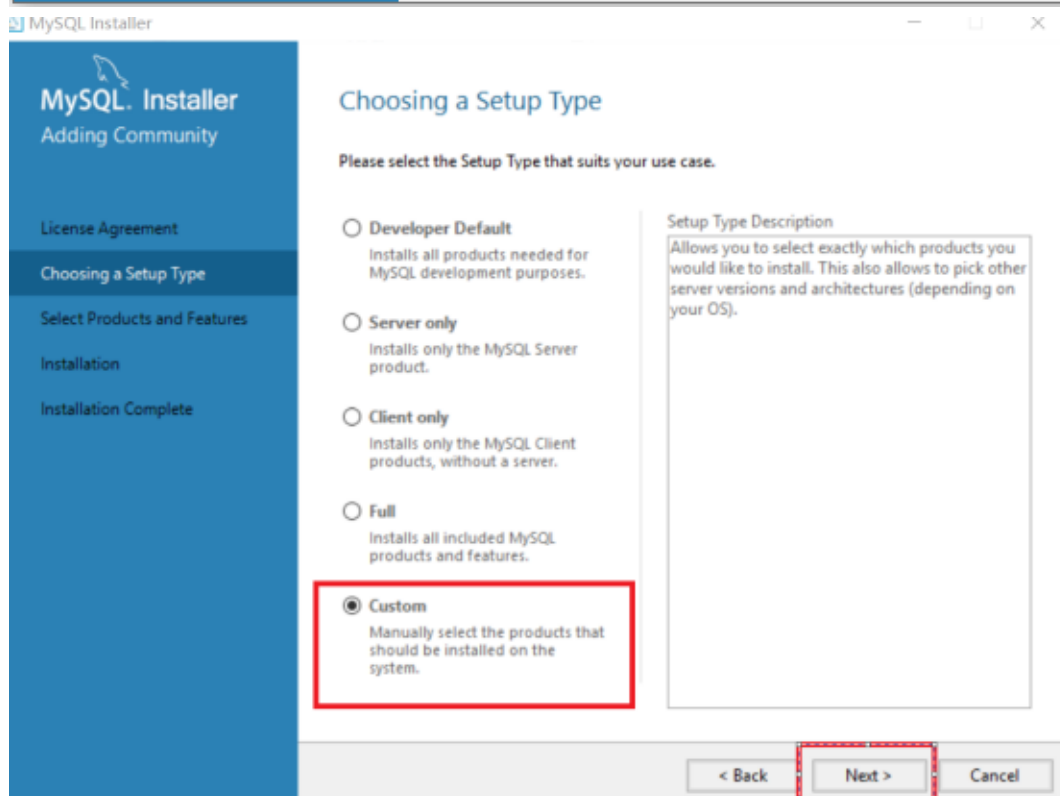
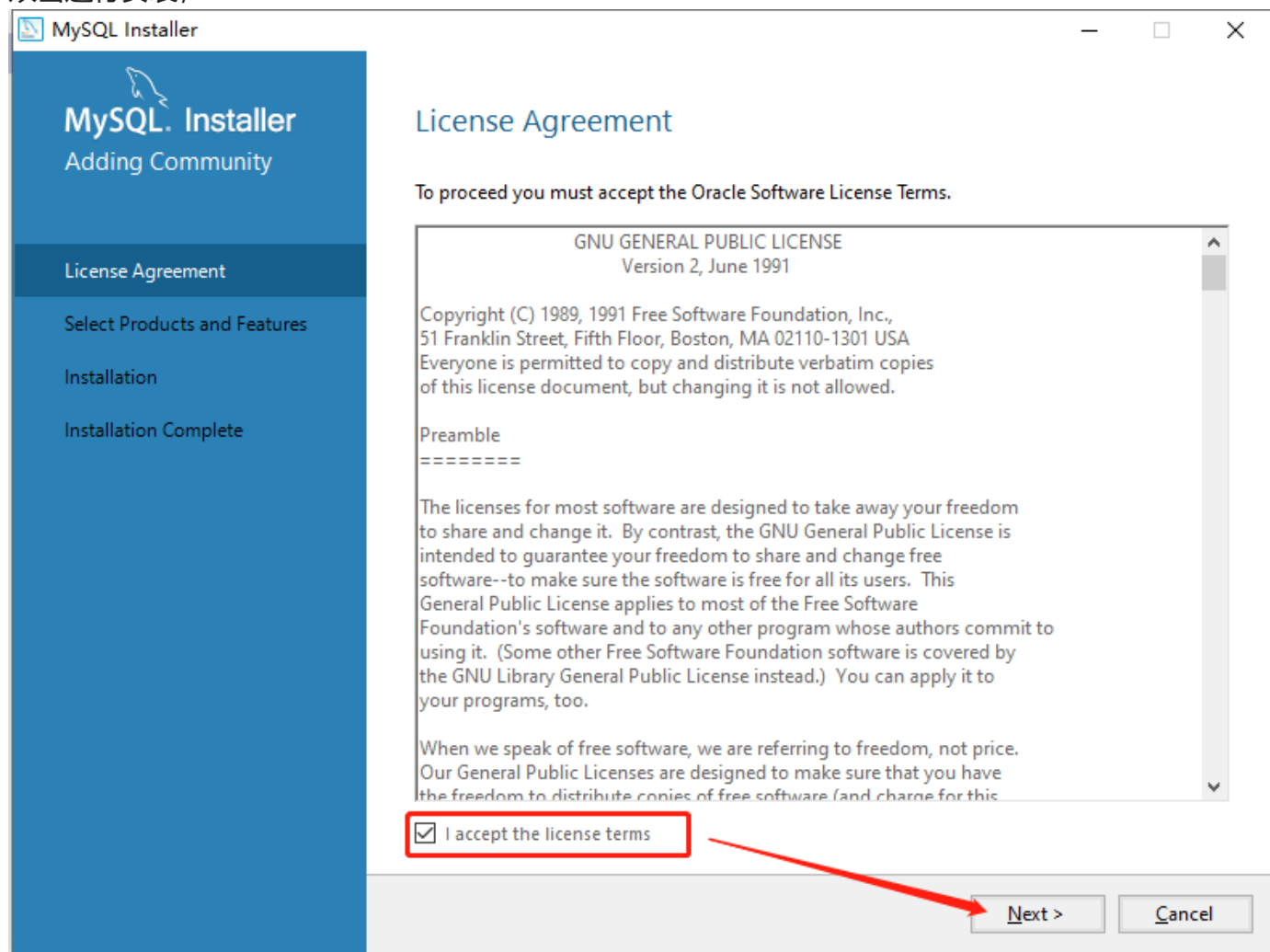
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-5.7.17.0.msi)	Nov 29, 2016	1.7M	Download
MD5: df80081cd386da03240c4fb4bae37758 Signature			
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-5.7.17.0.msi)	Nov 29, 2016	386.6M	Download
MD5: e03723eb6c6ba0271a848bd9031ea859 Signature			

笔者选择了版本【5.7.17】



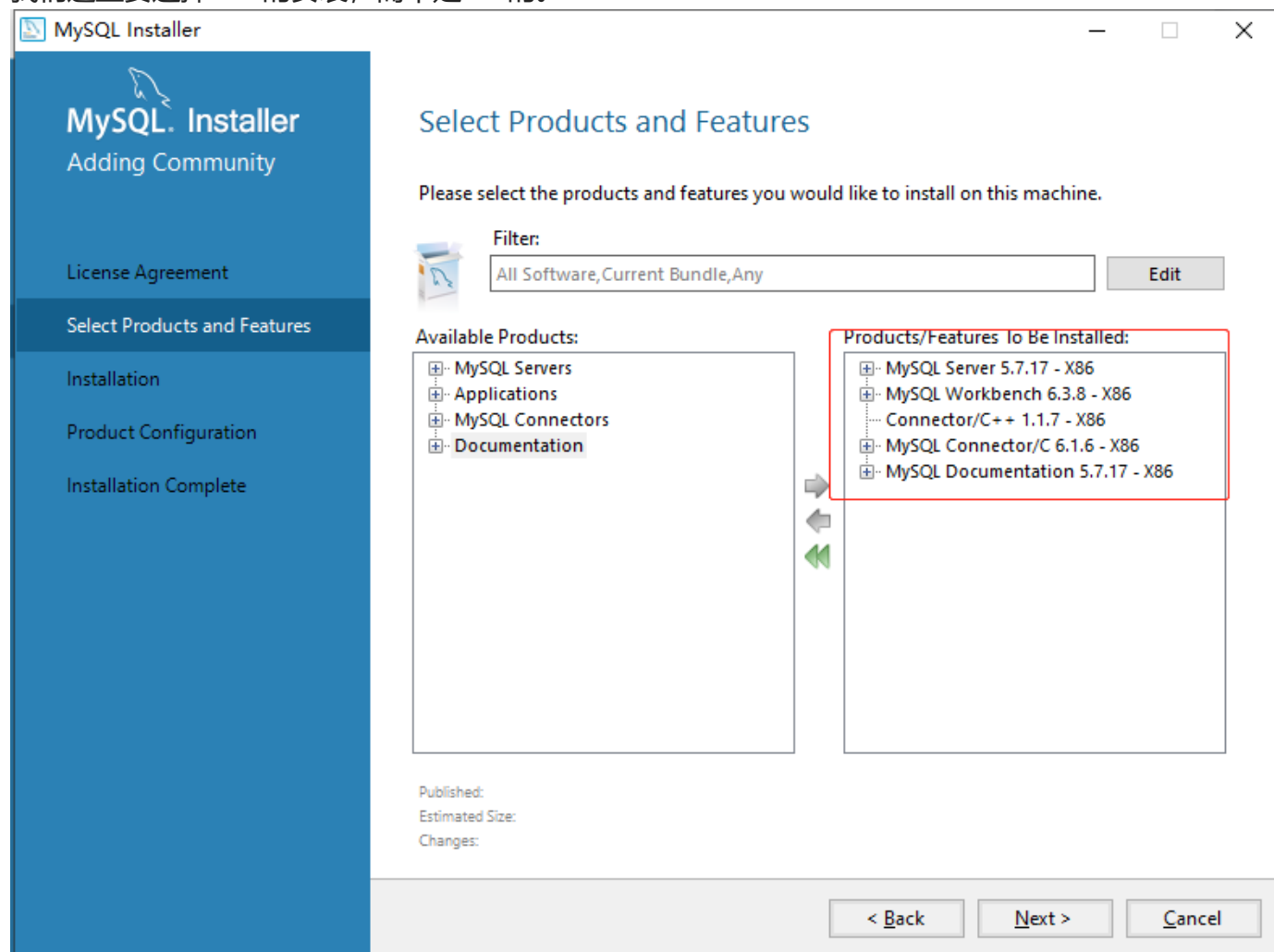
2. 安装

双击进行安装,

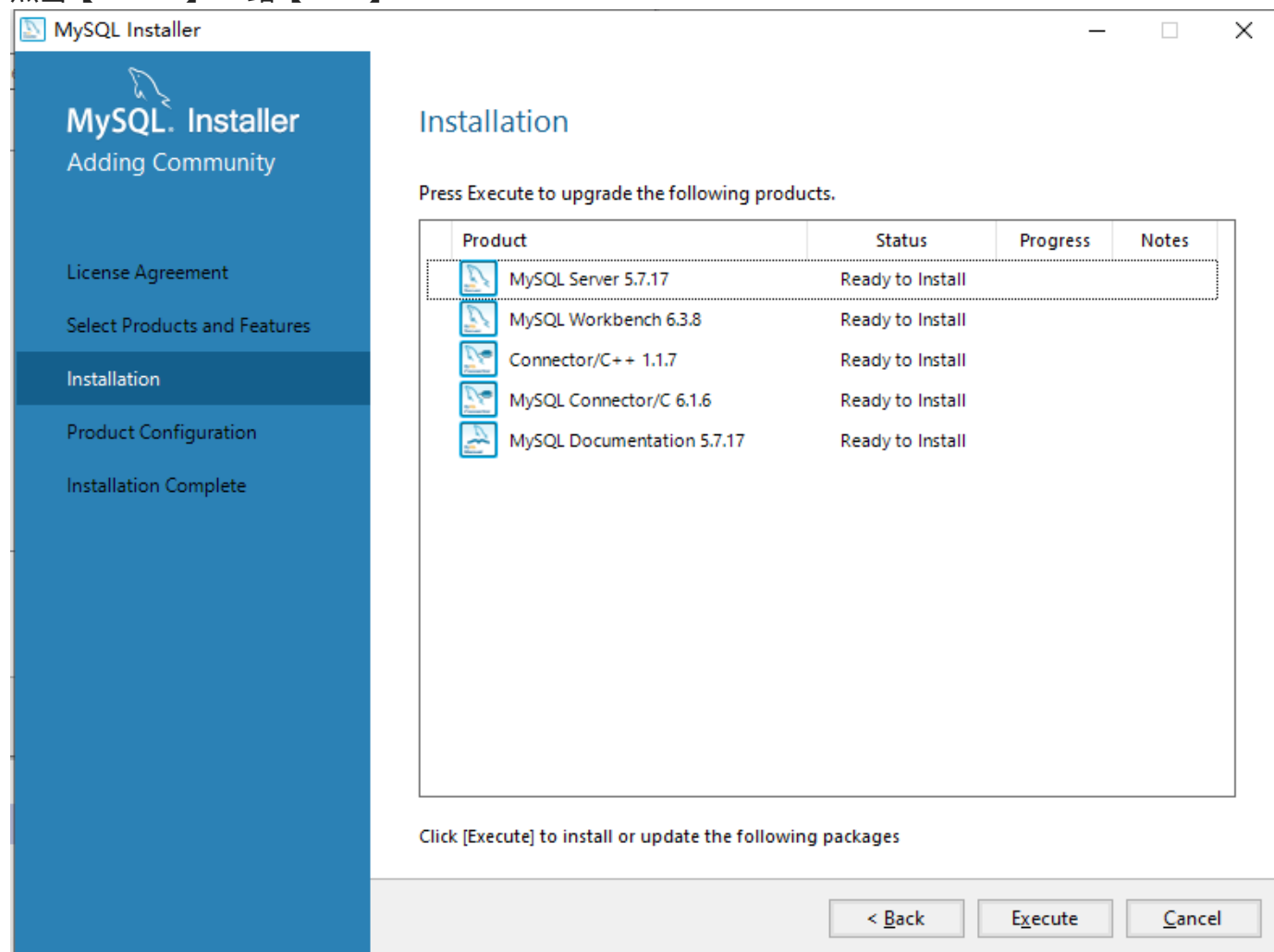


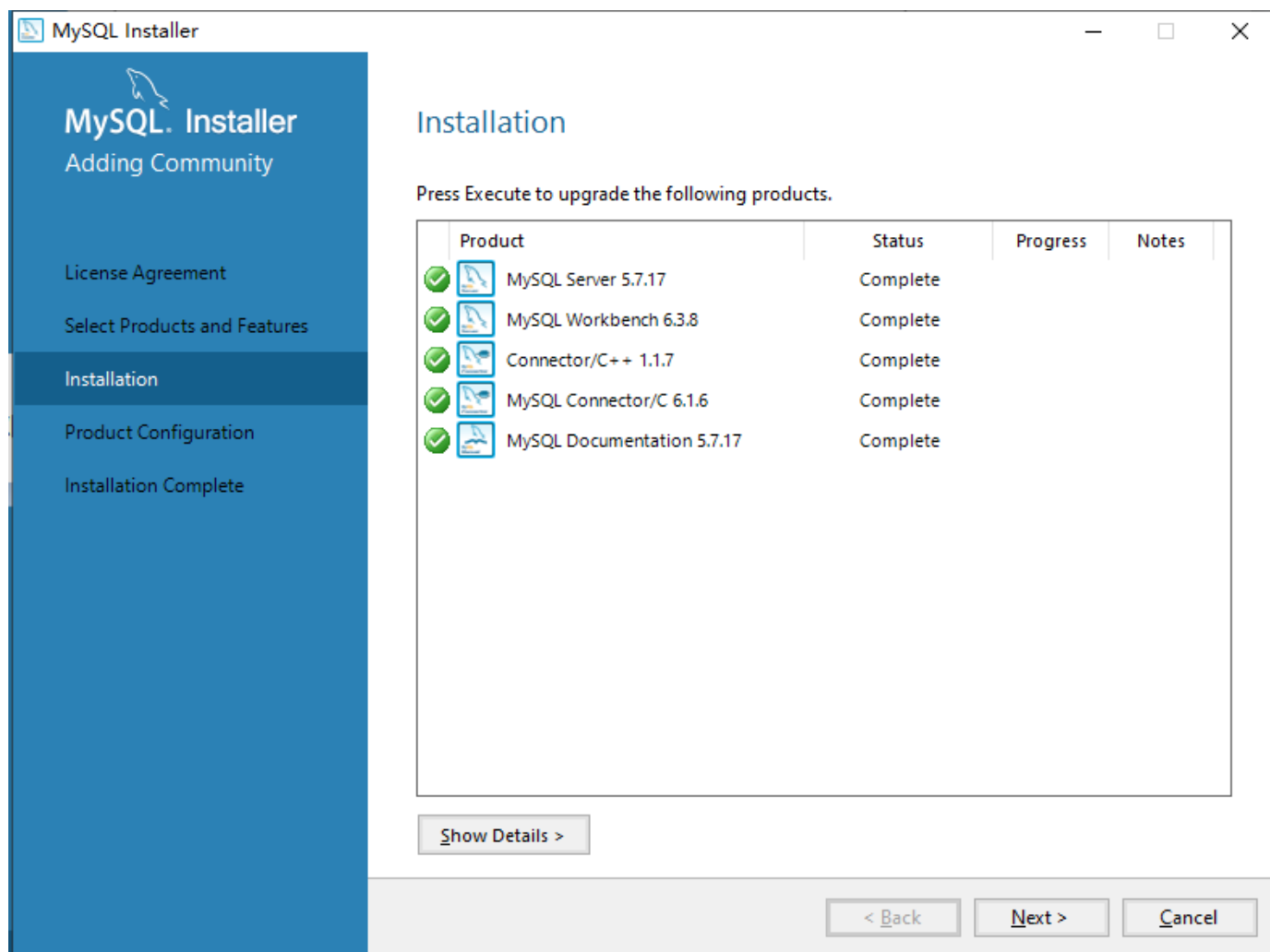
可获得的产品选择如下：MySQL Server 5.7.17 是一定要安装的，Workbench 为其客户端，可根据自己的习惯来，其他的仅参考。

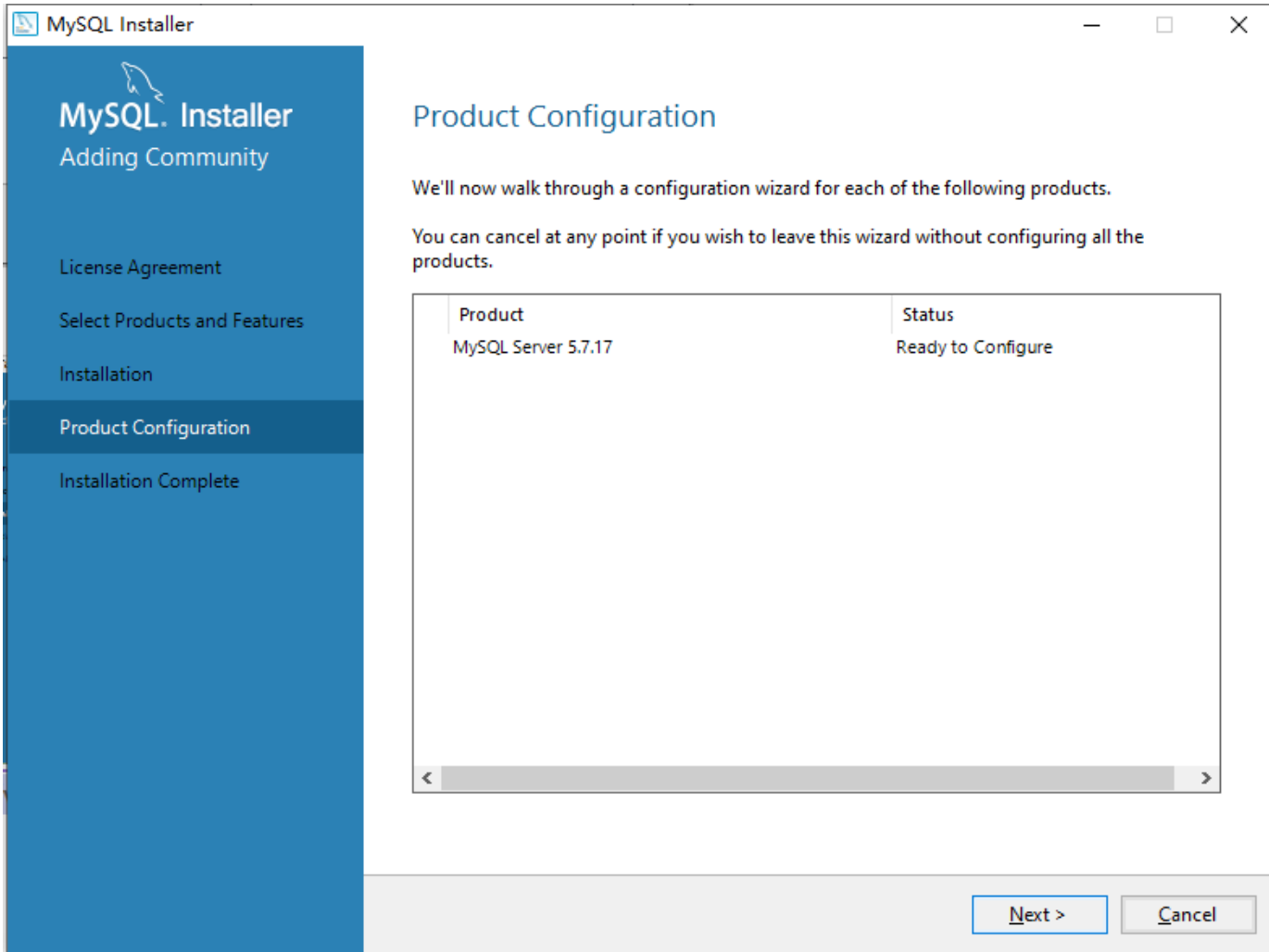
我们这里要选择 x86的安装，而不是x64的。



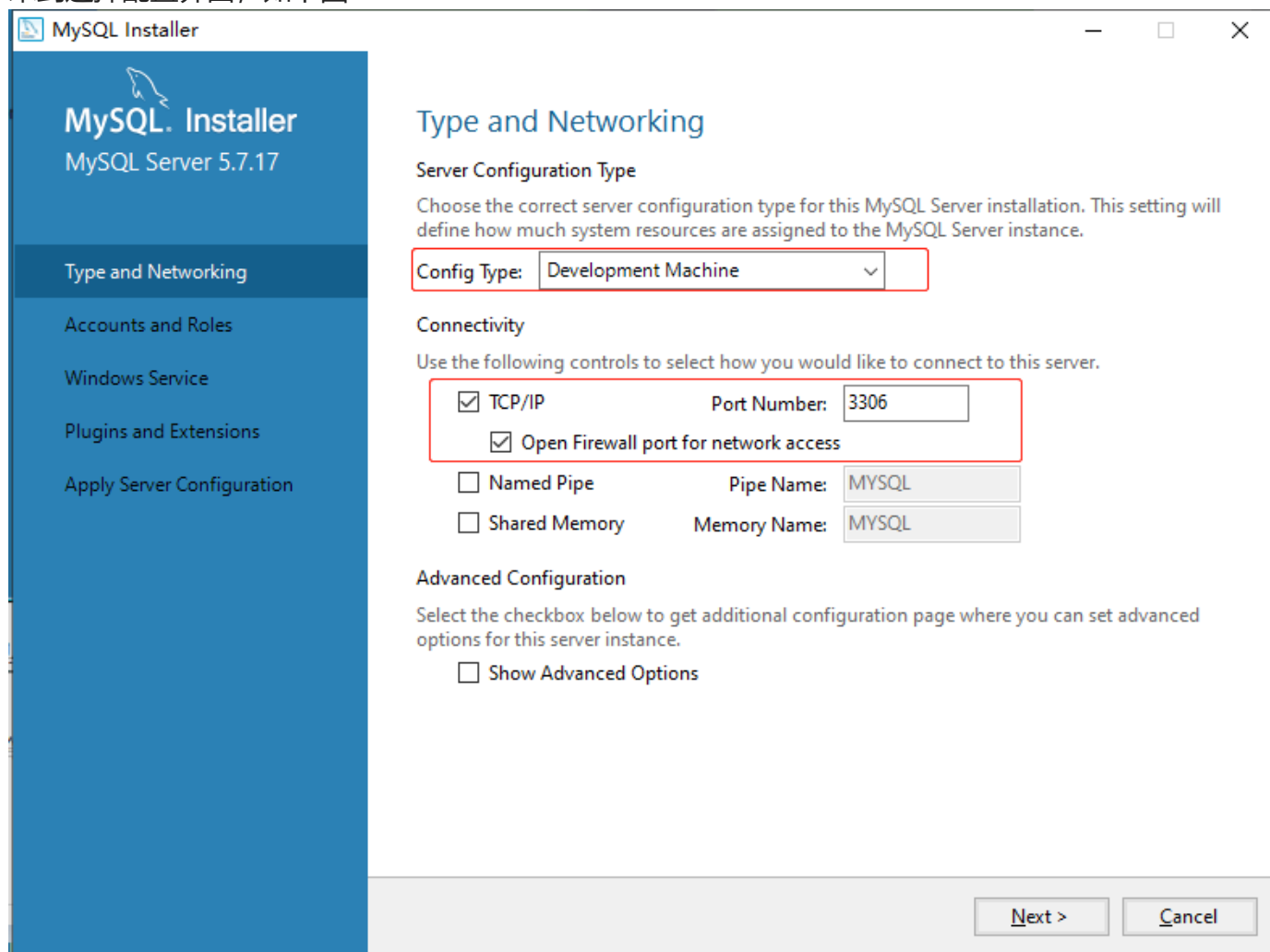
点击【Execute】一路【Next】



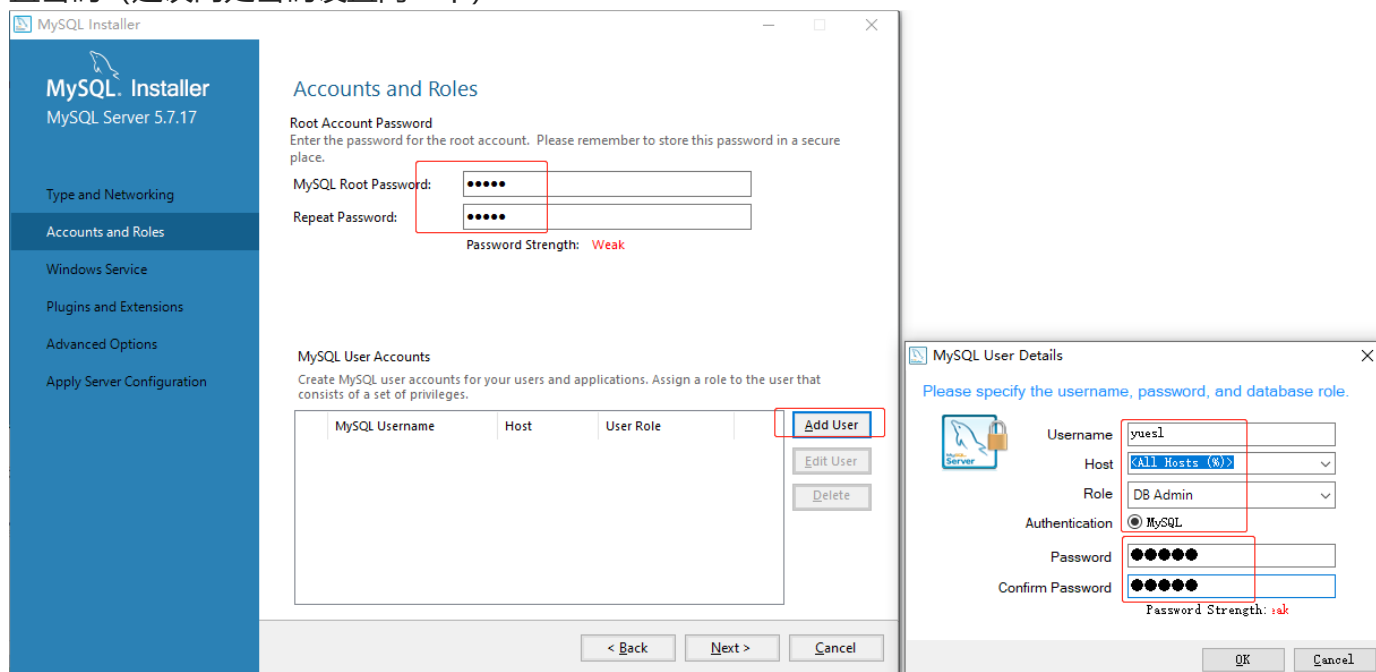




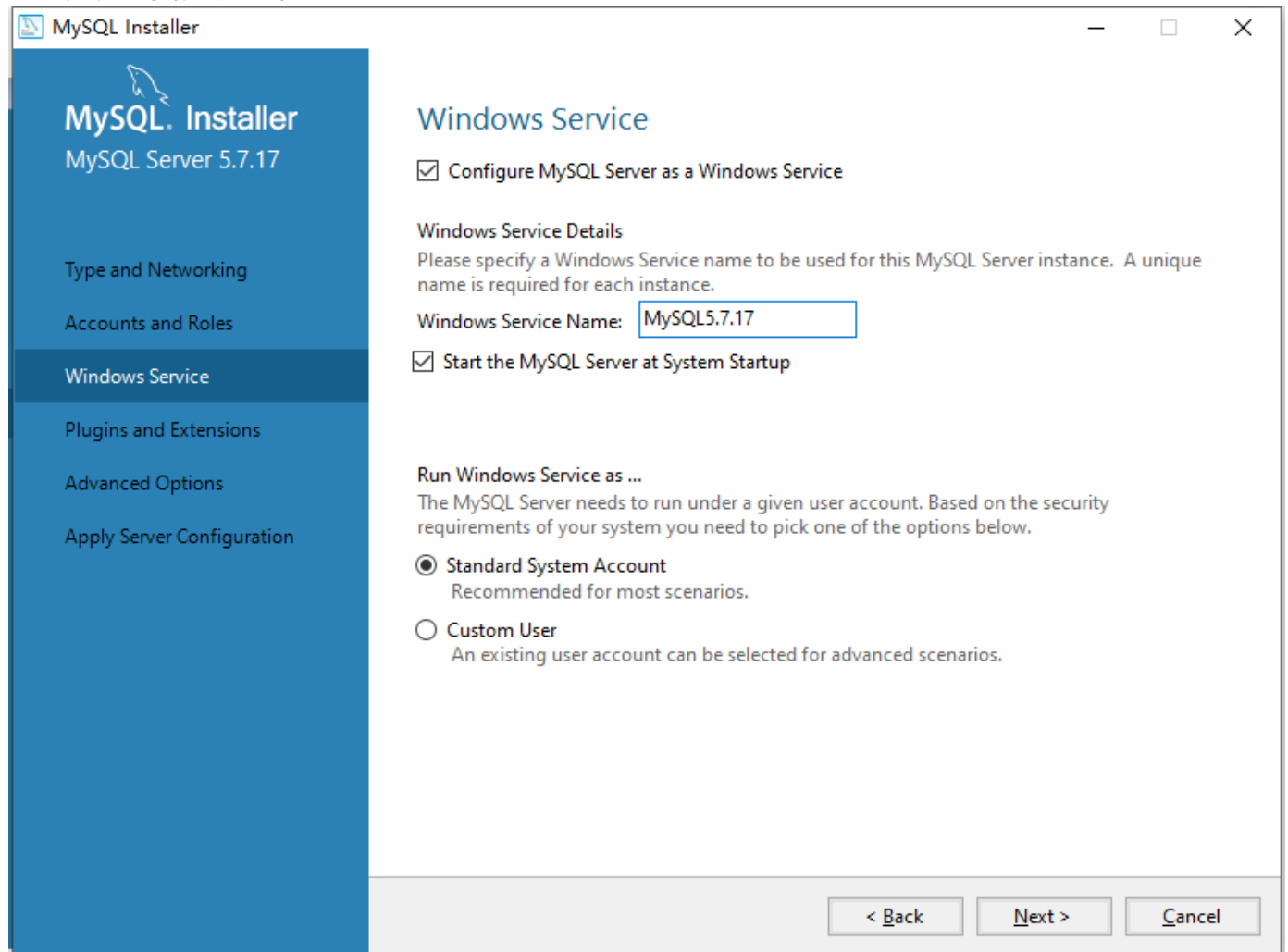
来到选择配置界面，如下图：

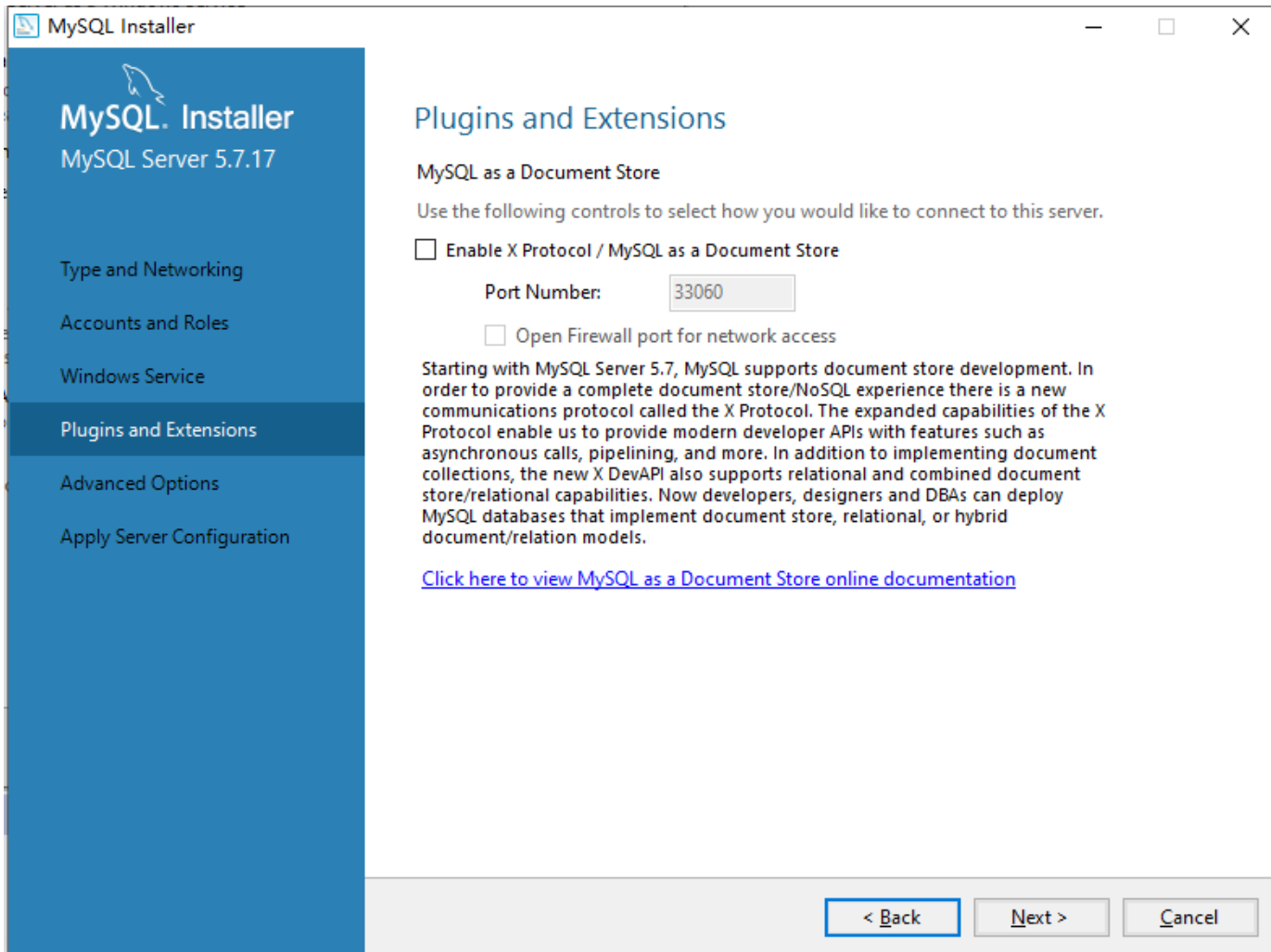


设置MySQL root 用户密码，这个**密码一定要牢记**，**密码一定要牢记**，**密码一定要牢记**。添加用户并设置密码（建议两处密码设置同一个）



下一步，如图，默认即可





日志配置根据自己的需求来。

MySQL Installer

MySQL Server 5.7.17

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

Advanced Options

Apply Server Configuration

Advanced Options

Logging Options

Please select the logs you want to activate for this server in addition to the Error Log. On production machines it can be beneficial to separate the log files from the data.

Error Log: ...

☒ General Log

The general query log is a general record of what the MySQL Server is doing. It should only be used to track down issues.

File Path: ...

☒ Slow Query Log

The slow query log consists of SQL statements that took more than the given value of seconds to execute. It is recommended to turn this log on.

File Path: ...

Seconds:

☐ Bin Log

The binary log contains all database events and is used for replication and data recovery operations. It has a performance impact on the server. Only a filename base should be given as the server will append an appropriate extension

File Path: ...

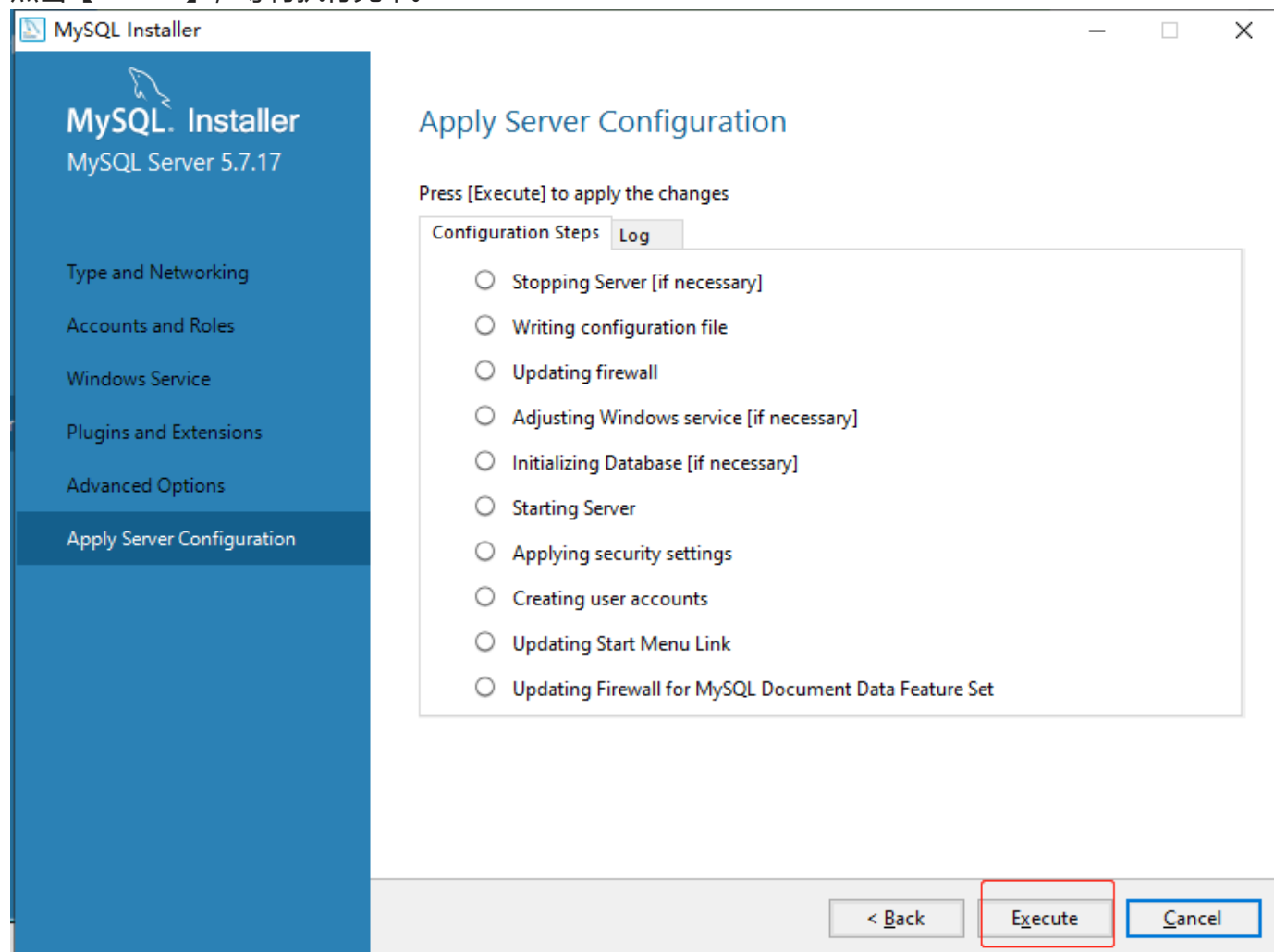
Server Id:

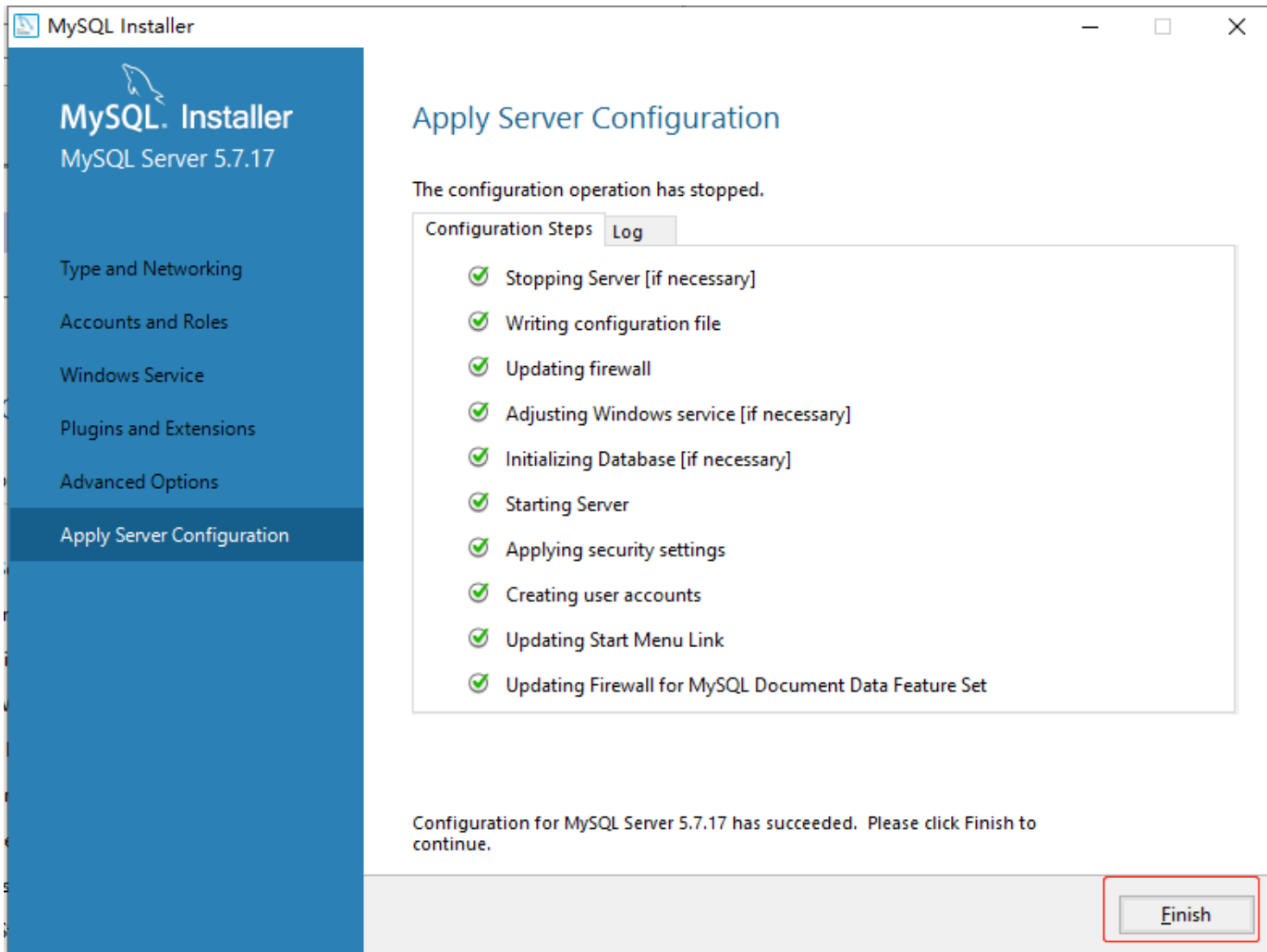
< Back

Next >

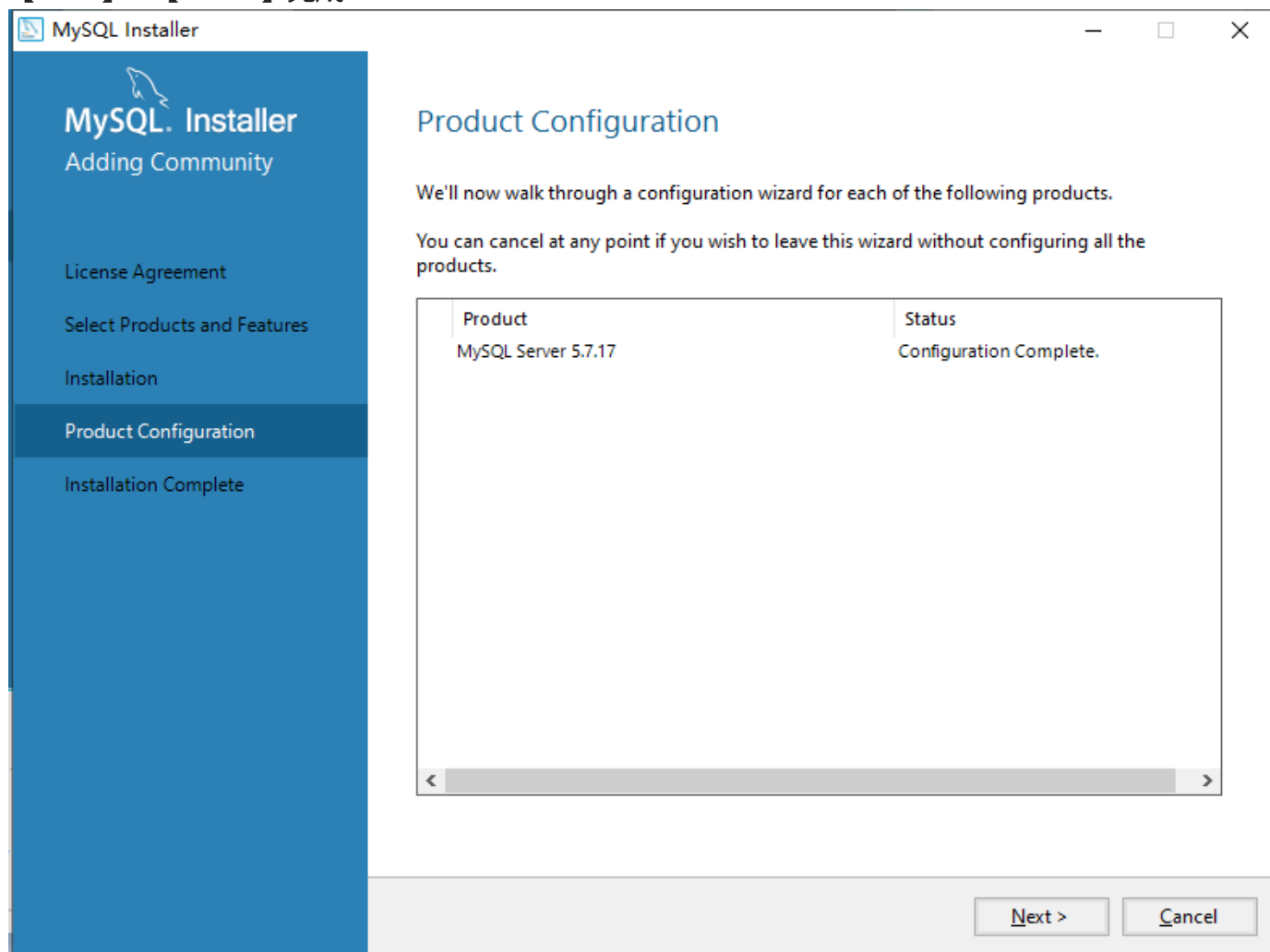
Cancel

点击【Excute】，等待执行完毕。





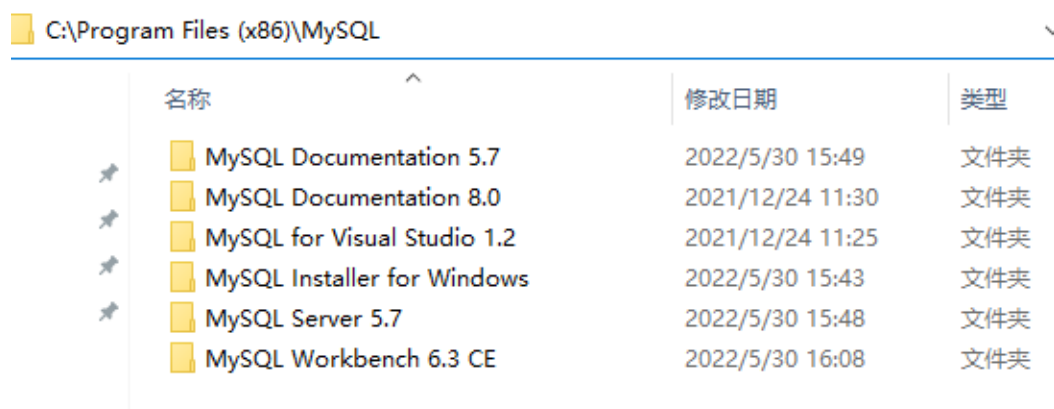
【Next】->【Finish】完成



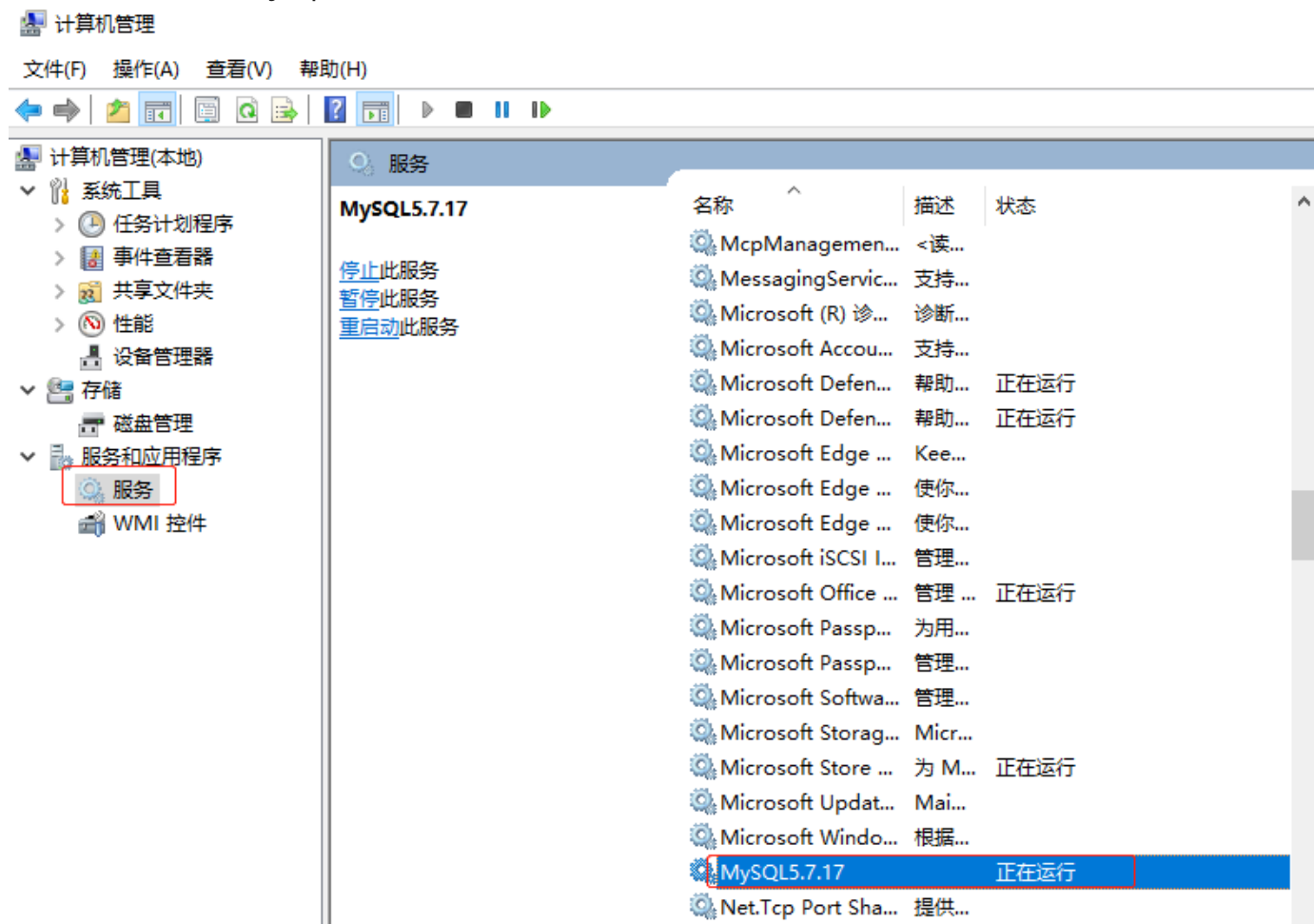
安装完成。

默认的安装路径为：

C:\Program Files (x86)\MySQL



每次开机后要使用MySQL，确保其服务已经被启动（状态为：【正在运行】）。

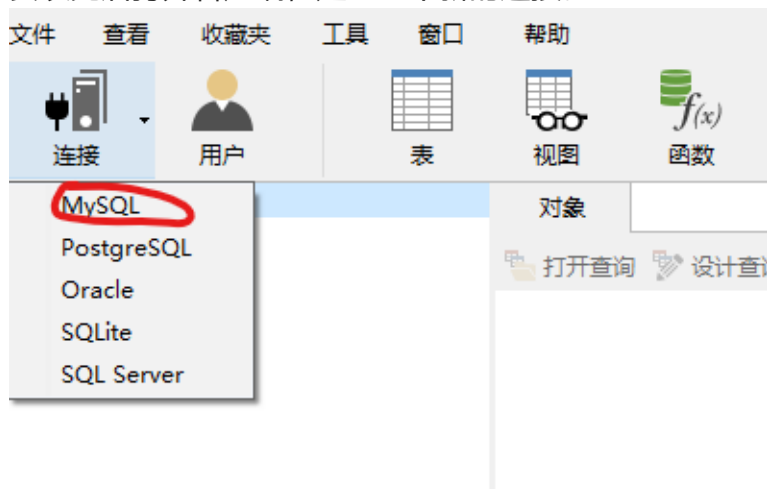


3. 客户端连接

安装【Navicat】，双击运行安装即可（无需破解）。

Navicat_Premium_11.0.10.exe

安装完后打开客户端，建立一个新的连接。



MySQL - 新建连接

常规 高级 SSL SSH HTTP

连接名: MySql

主机名或 IP 地址: localhost

端口: 3306

用户名: root

密码: ●●●●

☒ 保存密码

连接测试 确定 取消

如果出现这个错误，说明密码写错了。

MySQL - 新建连接

常规 高级 SSL SSH HTTP

连接名: MySql

主机名或 IP 地址: localhost

端口: 3306

用户名: root

密码: ●●●●

☒ 保存密码

1045 - Access denied for user 'root'@'localhost' (using password: YES)

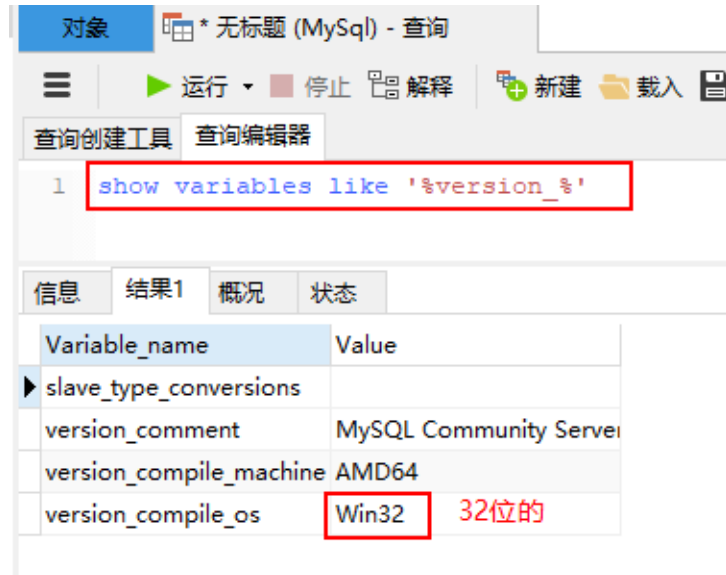
确定

点击连接信息，出现绿色，代表连接成功了。

查看是32位还是64位的，如下：



`show variables like '%version_%'`



或者是使用 status 命令：

32位：

```
mysql> status;
-----
mysql Ver 14.14 Distrib 5.7.17, for Win32 (AMD64)

Connection id:          1009
Current database:
Current user:            root@localhost
SSL:                     Not in use
Using delimiter:        ;
Server version:          5.7.17-log MySQL Community Server (GPL)
Protocol version:        10
Connection:              127.0.0.1 via TCP/IP
Server characterset:     utf8
Db characterset:         utf8
Client characterset:     gbk
Conn. characterset:      gbk
TCP port:                3306
Uptime:                  2 days 17 hours 53 min 4 sec

Threads: 1  Questions: 3021  Slow queries: 0  Opens: 179  Flush tables: 1  Open tables: 172  Queries per second avg: 0.12
-----
```

64位：

```
mysql> status
-----
C:\Program Files\MySQL\MySQL Server 5.7\bin\mysql.exe Ver 14.14 Distrib 5.7.17, for Win64 (x86_64)

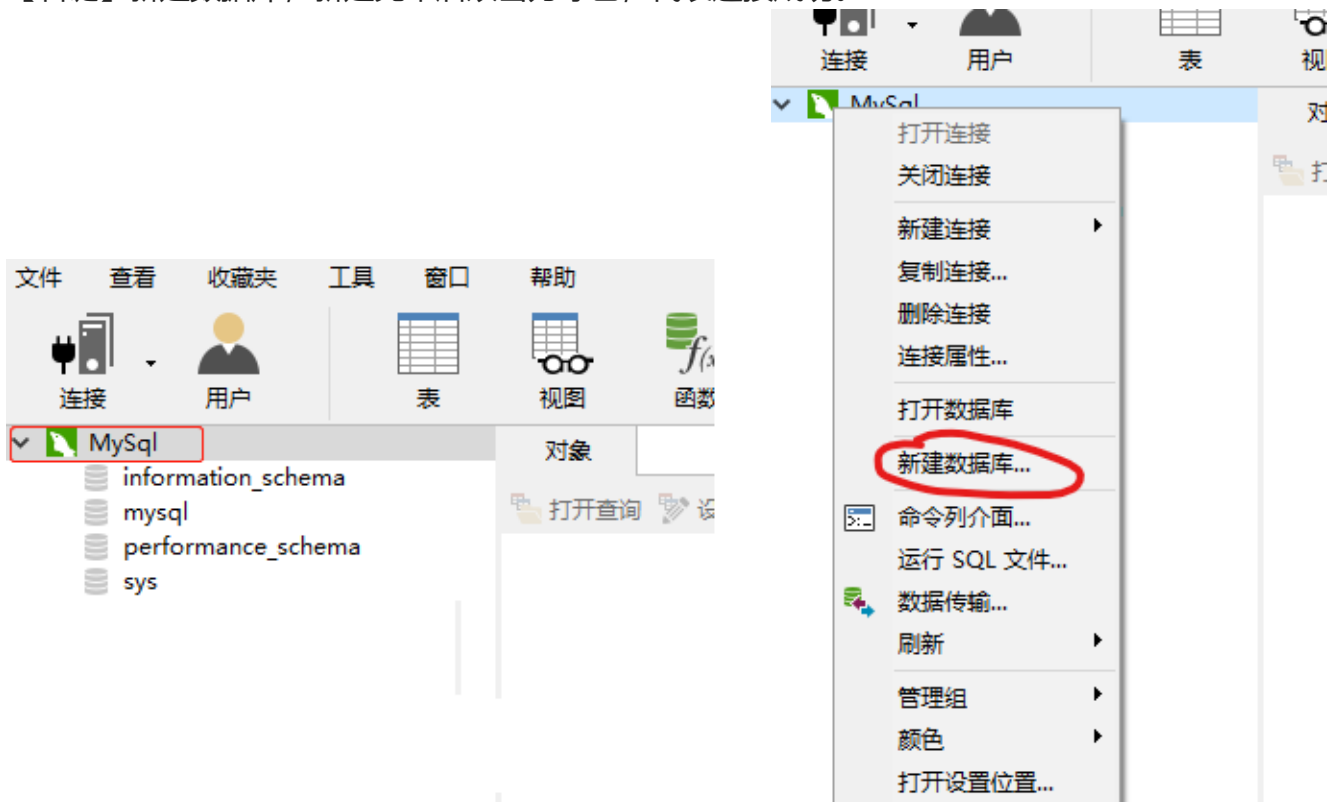
Connection id:          3
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.7.17-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:     utf8
Db characterset:         utf8
Client characterset:     utf8
Conn. characterset:      utf8
TCP port:               3306
Uptime:                 3 hours 33 min 25 sec

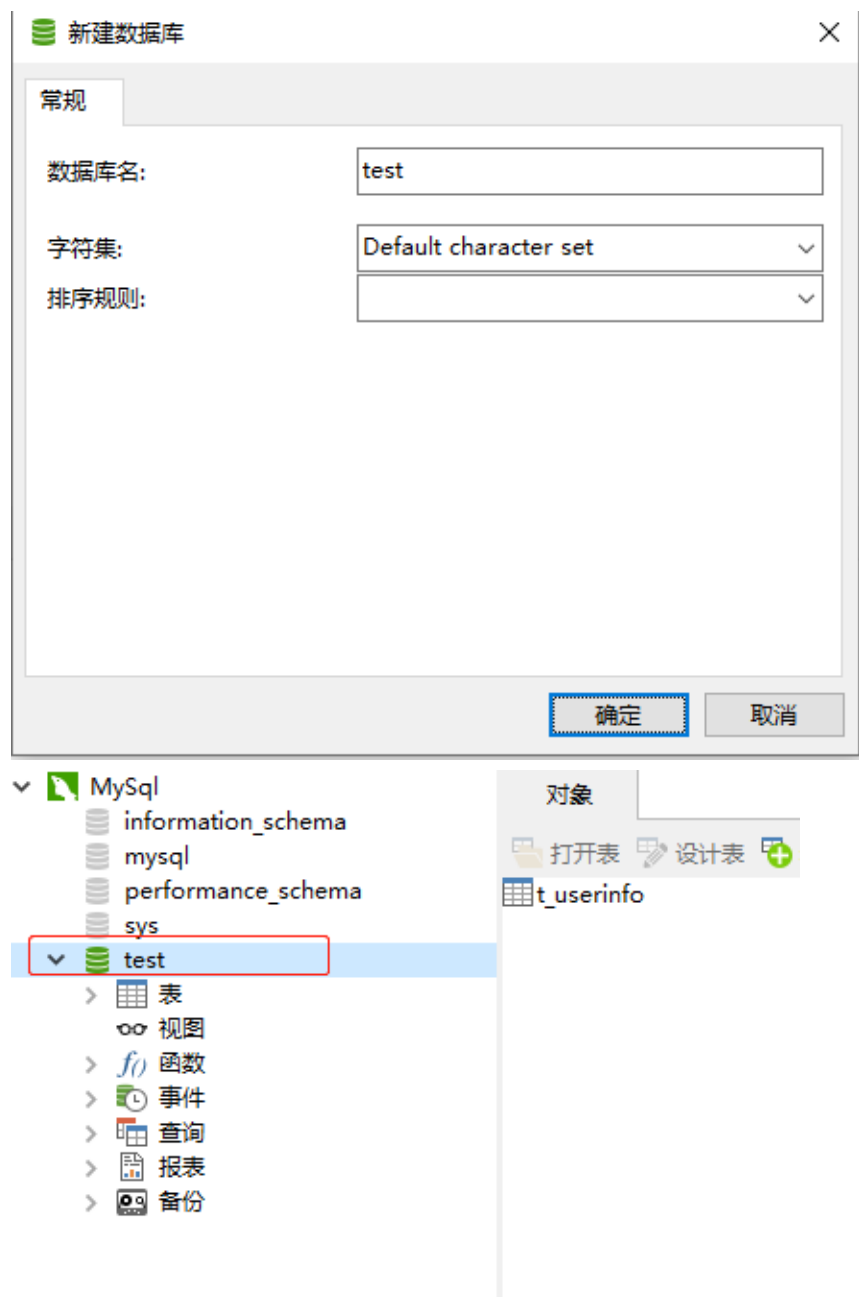
Threads: 1  Questions: 6  Slow queries: 0  Opens: 108  Flush tables: 1  Open tables: 101  Queries per second avg: 0.000
-----
```

如果是64位的，就麻烦小伙伴重新安装32位的吧。

//=====安装到此结束=====

【右键】新建数据库，新建完毕后双击为绿色，代表连接成功。

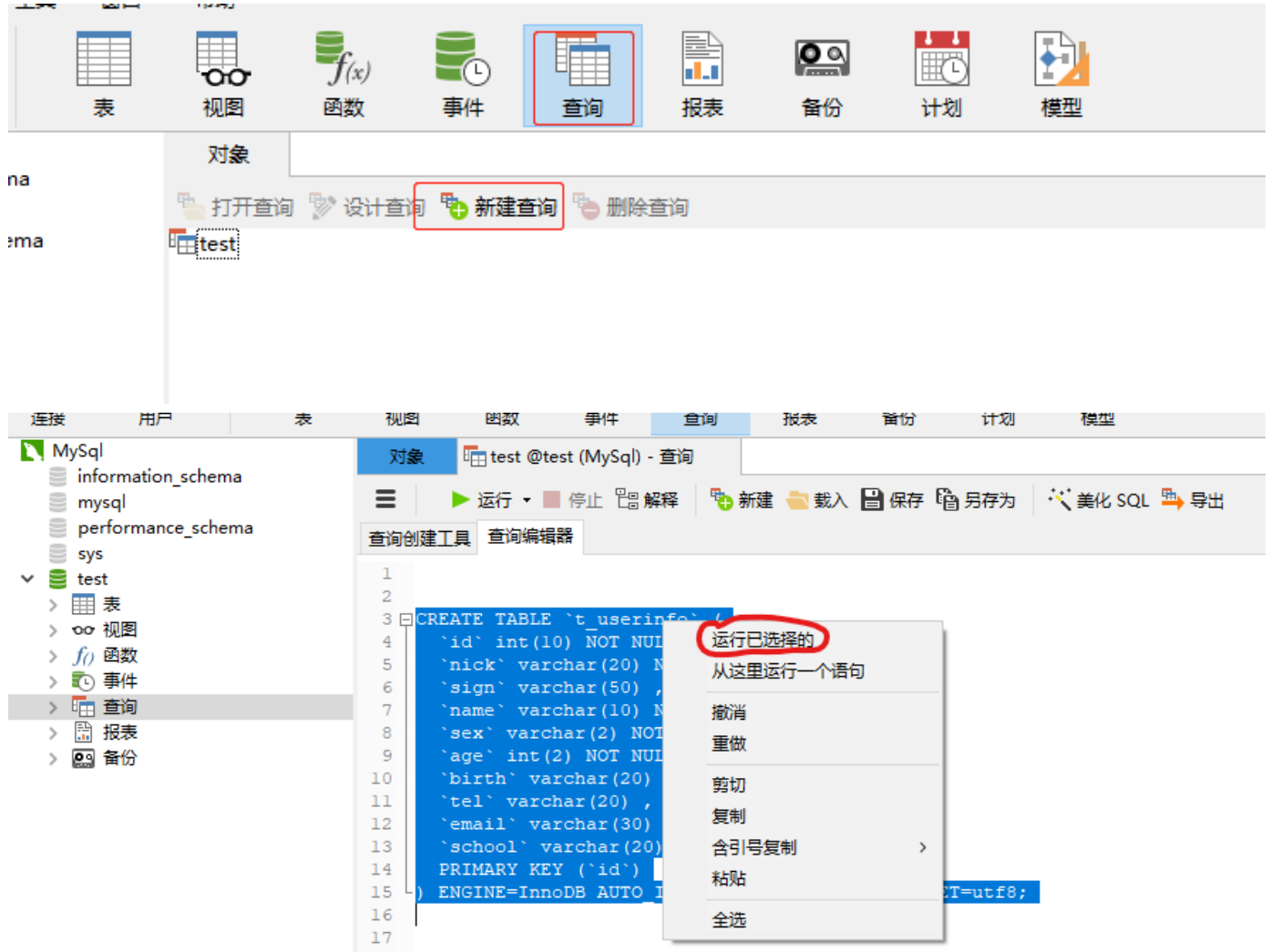




使用sql语句新建表：建立一张用户信息表，其中 `t_userinfo` 为表名 `id` 为主键且设置为自动递增。

```
1 CREATE TABLE `t_userinfo` (  
2     `id` int(10) NOT NULL AUTO_INCREMENT,  
3     `nick` varchar(20) NOT NULL,  
4     `sign` varchar(50) ,  
5     `name` varchar(10) NOT NULL,  
6     `sex` varchar(2) NOT NULL,  
7     `age` int(2) NOT NULL,  
8     `birth` varchar(20) ,  
9     `tel` varchar(20) ,  
10    `email` varchar(30) ,  
11    `school` varchar(20) ,  
12    PRIMARY KEY (`id`)  
13 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

【查询】->【新建查询】，将创建表的语句粘贴过来，运行一下。



创建表成功

```
2
3 CREATE TABLE `t_userinfo` (
4   `id` int(10) NOT NULL AUTO_INCREMENT,
5   `nick` varchar(20) NOT NULL,
6   `sign` varchar(50) ,
7   `name` varchar(10) NOT NULL,
8   `sex` varchar(2) NOT NULL,
9   `age` int(2) NOT NULL,
10  `birth` varchar(20) ,
11  `tel` varchar(20) ,
12  `email` varchar(30) ,
13  `school` varchar(20) ,
14  PRIMARY KEY (`id`)
15 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
16
```

信息

概况

状态

```
[SQL]CREATE TABLE `t_userinfo` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `nick` varchar(20) NOT NULL,
    `sign` varchar(50) ,
    `name` varchar(10) NOT NULL,
  `sex` varchar(2) NOT NULL,
  `age` int(2) NOT NULL,
    `birth` varchar(20) ,
    `tel` varchar(20) ,
    `email` varchar(30) ,
    `school` varchar(20) ,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
受影响的行: 0
时间: 0.397s
```

在表中插入数据:

```
1  INSERT INTO `t_userinfo` VALUES ('1', '一心一意', '签名_一心一意', '小张',
2    , '男', '20', '2022-01-02', '15509457453', '123@qq.com', '哈理工');
3  INSERT INTO `t_userinfo` VALUES ('2', '二龙戏珠', '签名_二龙戏珠', '阿磊',
4    , '男', '22', '2022-07-26', '15512344321', '123@qq.com', '黑大');
5  INSERT INTO `t_userinfo` VALUES ('3', '三阳开泰', '签名_三阳开泰', '小明',
6    , '女', '12', '2022-07-26', '15565461657', '123@qq.com', '黑大');
7  INSERT INTO `t_userinfo` VALUES ('4', '四时吉庆', '签名_四时吉庆', '李四',
8    , '男', '12', '2022-07-26', '15512344321', '123@qq.com', '黑大');
9  INSERT INTO `t_userinfo` VALUES ('5', '五福临门', '签名_五福临门', '小王',
10    , '女', '20', '2022-01-02', '15594413498', '123@qq.com', '哈理工');
11  INSERT INTO `t_userinfo` VALUES ('6', '六六大顺', '签名_六六大顺', '吴某',
12    , '男', '11', '2022-08-01', '15564641354', '123@qq.com', '黑大');
13  INSERT INTO `t_userinfo` VALUES ('7', '七星高照', '签名_七星高照', '赵某',
14    , '男', '32', '2022-08-01', '15564641354', '123@qq.com', '黑大');
15  # 或 指定列名进行新增一条数据
16  INSERT INTO `t_userinfo` (nick,sign,name,sex,age,birth,tel,email,school)
```

```
VALUES ('八方进宝', '签名_八方进宝', '张某某', '女', '100', '2013-08-01', '15539457983', '123@qq.com', '黑大');
```

查询表:

```
1 | select * from t_userinfo;
```

查询结果如下:

18

INSERT INTO `t_userinfo` VALUES ('1', '一心一意', '签名_一心一意', '小张', '男', '20', '2022-01-02', '15509457453', '123@qq.com', '哈理工');

19

INSERT INTO `t_userinfo` VALUES ('2', '二龙戏珠', '签名_二龙戏珠', '阿磊', '男', '22', '2022-07-26', '15512344321', '123@qq.com', '黑大');

20

INSERT INTO `t_userinfo` VALUES ('3', '三阳开泰', '签名_三阳开泰', '小明', '女', '12', '2022-07-26', '15565461657', '123@qq.com', '黑大');

21

INSERT INTO `t_userinfo` VALUES ('4', '四时吉庆', '签名_四时吉庆', '李四', '男', '12', '2022-07-26', '15512344321', '123@qq.com', '黑大');

22

INSERT INTO `t_userinfo` VALUES ('5', '五福临门', '签名_五福临门', '小王', '女', '20', '2022-01-02', '15594413498', '123@qq.com', '哈理工');

23

INSERT INTO `t_userinfo` VALUES ('6', '六六大顺', '签名_六六大顺', '吴某', '男', '11', '2022-08-01', '15564641354', '123@qq.com', '黑大');

24

INSERT INTO `t_userinfo` VALUES ('7', '七星高照', '签名_七星高照', '赵某', '男', '32', '2022-08-01', '15564641354', '123@qq.com', '黑大');

25

INSERT INTO `t_userinfo` (nick,sign,name,sex,age,birth,tel,email,school) VALUES ('八方进宝', '签名_八方进宝', '张某某', '女', '100', '2013-08-01', '15539457983', '123@qq.com', '黑大');

26

27

28

29

select * from t_userinfo;

信息	结果1	概况	状态						
id	nick	sign	name	sex	age	birth	tel	email	school
1	一心一意	签名_一心一意	小张	男	20	2022-01-02	15509457453	123@qq.com	哈理工
2	二龙戏珠	签名_二龙戏珠	阿磊	男	22	2022-07-26	15512344321	123@qq.com	黑大
3	三阳开泰	签名_三阳开泰	小明	女	12	2022-07-26	15565461657	123@qq.com	黑大
4	四时吉庆	签名_四时吉庆	李四	男	12	2022-07-26	15512344321	123@qq.com	黑大
5	五福临门	签名_五福临门	小王	女	20	2022-01-02	15594413498	123@qq.com	哈理工
6	六六大顺	签名_六六大顺	吴某	男	11	2022-08-01	15564641354	123@qq.com	黑大
7	七星高照	签名_七星高照	赵某	男	32	2022-08-01	15564641354	123@qq.com	黑大
8	八方进宝	签名_八方进宝	张某某	女	100	2013-08-01	15539457983	123@qq.com	黑大

修改表数据: set 后为要修改的字段和修改后的值, 后面一定要加where条件, 否则都会修改。

```
1 | UPDATE t_userinfo SET `name`='小明明', `sign`='teacher' WHERE id=3;
```

修改前:

id	nick	sign	name	sex	age	birth	tel	email	school
1	一心一意	签名_一心一意	小张	男	20	2022-01-02	15509457453	123@qq.com	哈理工
2	二龙戏珠	签名_二龙戏珠	阿磊	男	22	2022-07-26	15512344321	123@qq.com	黑大
3	三阳开泰	签名_三阳开泰	小明	女	12	2022-07-26	15565461657	123@qq.com	黑大
4	四时吉庆	签名_四时吉庆	李四	男	12	2022-07-26	15512344321	123@qq.com	黑大
5	五福临门	签名_五福临门	小王	女	20	2022-01-02	15594413498	123@qq.com	哈理工
6	六六大顺	签名_六六大顺	吴某	男	11	2022-08-01	15564641354	123@qq.com	黑大
7	七星高照	签名_七星高照	赵某	男	32	2022-08-01	15564641354	123@qq.com	黑大
8	八方进宝	签名_八方进宝	张某某	女	100	2013-08-01	15539457983	123@qq.com	黑大

修改后:

信息	结果1	概况	状态						
id	nick	sign	name	sex	age	birth	tel	email	school
1	一心一意	签名_一心一意	小张	男	20	2022-01-02	15509457453	123@qq.com	哈理工
2	二龙戏珠	签名_二龙戏珠	阿磊	男	22	2022-07-26	15512344321	123@qq.com	黑大
3	三阳开泰	teacher	小明明	女	12	2022-07-26	15565461657	123@qq.com	黑大
4	四时吉庆	签名_四时吉庆	李四	男	12	2022-07-26	15512344321	123@qq.com	黑大
5	五福临门	签名_五福临门	小王	女	20	2022-01-02	15594413498	123@qq.com	哈理工
6	六六大顺	签名_六六大顺	吴某	男	11	2022-08-01	15564641354	123@qq.com	黑大
7	七星高照	签名_七星高照	赵某	男	32	2022-08-01	15564641354	123@qq.com	黑大
8	八方进宝	签名_八方进宝	张某某	女	100	2013-08-01	15539457983	123@qq.com	黑大

删除: 一定要加where限定条件, 否则全部删除, 相当于清空表, 造成无法挽回的后果。

```
1 | DELETE FROM t_userinfo WHERE id = 6;
```

4. qt-c++连接数据库

找到MySQL的安装目录，其中【include】为操作数据库所需要的各种头文件，lib下为所需要的库文件。


C:\Program Files (x86)\MySQL\MySQL Server 5.7\				
名称	修改日期	类型	大小	
bin	2022/5/30 15:48	文件夹		
include	2022/5/30 15:48	文件夹		
lib	2022/5/30 15:48	文件夹		
share	2022/5/30 15:48	文件夹		
COPYING	2016/11/28 14:32	文件	18 KB	
my-default.ini	2016/11/29 2:24	配置设置	2 KB	
README	2016/11/28 14:32	文件	3 KB	


C:\Program Files (x86)\MySQL\MySQL Server 5.7\lib				
名称	修改日期	类型	大小	
mecab	2022/5/30 15:48	文件夹		
plugin	2022/5/30 15:48	文件夹		
libmysql.dll	2016/11/29 2:28	应用程序扩展	4,913 KB	
libmysql.lib	2016/11/29 2:28	Object File Library	29 KB	
mysqlclient.lib	2016/11/29 2:28	Object File Library	22,682 KB	

新建Qt项目后，在工程文件【.pro】中添加如下配置：

```
1 | # 添加头文件包含的目录
2 | INCLUDEPATH += "C:\Program Files (x86)\MySQL\MySQL Server 5.7\include"
3 | # 添加库文件
4 | LIBS += "C:\Program Files (x86)\MySQL\MySQL Server 5.7\lib\libmysql.lib"
```

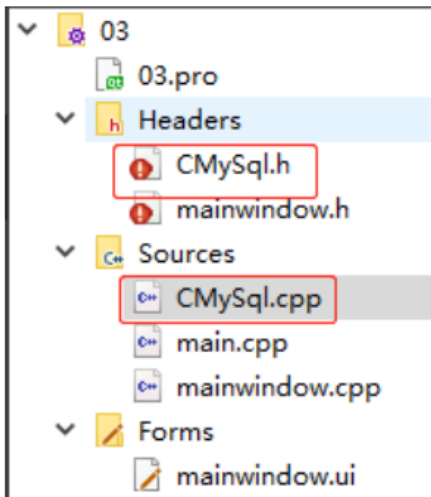
对于数据库的操作进行了二次封装。需要以下两个文件（文件代码见附录）。

 CMySql.cpp

 CMySql.h

将这两个文件拷贝到项目的代码目录中，并添加到项目中。

名称	修改日期	类型	大小
03.pro	2022/8/11 10:35	Qt Project file	1 KB
03.pro.user	2022/8/7 8:23	USER 文件	22 KB
CMySQL.cpp	2022/8/11 16:05	C++ Source file	6 KB
CMySQL.h	2022/8/6 8:21	C++ Header file	2 KB
main.cpp	2022/8/4 9:01	C++ Source file	1 KB
mainwindow.cpp	2022/8/11 12:20	C++ Source file	12 KB
mainwindow.h	2022/8/11 12:12	C++ Header file	1 KB
mainwindow.ui	2022/8/11 11:38	Qt UI file	9 KB



构建项目后将【libmysql.dll】拷贝到可执行文件所在的目录中。

E:\workspace\qtclass\0321\03\build-03-Desktop_Qt_5_12_11_MinGW_32_bit-Debug\debug				
名称	修改日期	类型	大小	
03.exe	2022/8/11 12:20	应用程序	2,297 KB	
CMySQL.o	2022/8/11 10:35	O 文件	581 KB	
libmysql.dll	2016/11/29 2:28	应用程序扩展	4,913 KB	

准备工作准备完毕。

连接数据库：

```

1  char perr[MAX_ERR_LEN] = {0};
2  //返回结果为true连接成功，否则失败
3  if(!m_mysql.ConnectMySQL("localhost"/*IP地址*/, "root"/*用户名*/, "yues1"
   /*密码*/, "test"/*要连接的数据库*/, 3306/*端口*/, perr/*连接失败返回的错误
   信息*/))
4  {
5      //连接失败
6  }
```

查询数据库：

```

1  list<vector<string>> lstVecData;
2  char perr[MAX_ERR_LEN] = {0};
3  //返回结果为true查询成功，否则失败
4  if(!m_mysql.SelectMySQL("select * from t_userinfo"/*要查询的sql 语句*/,1
   stVecData/*查询结果*/,perr/*查询失败返回错误信息*/))
5  {
6      //查询失败
7  }

```

新增、修改，删除

```

1  //返回结果为true操作成功，否则失败
2  if(!m_mysql.UpdateMySQL("新增、修改，删除 的sql语句")
3  {
4      //操作失败
5  }

```

附：

完整代码如下：

```

1  //CMySQL.h
2  #pragma once
3  #include <list>
4  #include <vector>          //add by yuesl
5  #include<string>
6  using namespace std;
7  #include "mysql.h"
8  #pragma comment(lib,"libmysql.lib")
9
10 #define MAX_ERR_LEN    256  //add by yuesl
11
12 class CMySQL
13 {
14 public:
15     //CMySQL(void); //update by yuesl
16     CMySQL(string charEncoding = "gb2312");
17     ~CMySQL(void);
18 public:          //ip,用户名,密码，数据库，端口号
19     bool  ConnectMySQL(char *host,char *user,char *pass,char *db,short
   nport = 3306);
20     void  Disconnect();

```

```

21     bool SelectMySQL(char* szSql,int nColumn ,list<string>& lstStr);
22     //获得数据库中的表
23     bool GetTables(char* szSql,list<string>& lstStr);
24     //更新：删除、插入、修改
25     bool UpdateMySQL(char* szSql);
26
27     //----add by yuesl begin-----
28 #define _in_
29 #define _out_
30 #define _in_out_
31     using C_CHAR = const char;
32     //连接数据库
33     bool ConnectMySQL(_in_ C_CHAR *host, _in_ C_CHAR *user,
34                      _in_ C_CHAR *pass, _in_ C_CHAR *db,
35                      _in_ const short nport,
36                      _out_ char (&pErrinfo)[MAX_ERR_LEN]); //如果连
接失败返回错误信息
37
38     //仅限查询数据库，将结果信息返回 lstVecStr 中，list 为数据条数，vector
为各个字段数据
39     bool SelectMySQL(_in_ C_CHAR* szSql, _out_ list<vector<string>>& ls
tVecStr,
40                      _out_ char (&pErrinfo)[MAX_ERR_LEN]); //查询失败返
回错误信息
41     //获取某一个表的列字段信息（返回的是不是表数据，而是列名字）
42     bool GetTableColumnName(_in_ string table, //只需要传表名，而不是s
ql语句
43                             _out_ vector<string>& vecColumn,
44                             _out_ char (&pErrinfo)[MAX_ERR_LEN]);
45
46     //----add by yuesl end-----
47 private:
48     MYSQL *m_sock;
49     MYSQL_RES *m_results;
50     MYSQL_ROW m_record;
51
52 };

```

```

1 //CMySQL.cpp
2 //#include "stdafx.h"
3 #include "CMySQL.h"
4 #include<QDebug>
5

```

```

6   CMySQL::CMySQL(string charEncoding)
7   {
8       /*这个函数用来分配或者初始化一个MYSQL对象，用于连接mysql服务端。
9       如果你传入的参数是NULL指针，它将自动为你分配一个MYSQL对象，
10      如果这个MYSQL对象是它自动分配的，那么在调用mysql_close的时候，会释放这个
      对象*/
11      m_sock = new MYSQL;
12      mysql_init(m_sock);
13      //mysql_set_character_set(m_sock, "gb2312"); //gb2312 中华人民共和国
      简体字标准
14
15      //add by yuesl
16      //mysql_set_character_set(m_sock, "utf8"); //Qt下默认的文件字符编
      码为uft-8 所以此处需要修改
17      mysql_set_character_set(m_sock, charEncoding.c_str()=="?"?"gb2312":c
      harEncoding.c_str());
18  }
19
20  CMySQL::~CMySQL(void)
21  {
22      if(m_sock) {
23          delete m_sock;
24          m_sock = NULL;
25      }
26  }
27
28  void CMySQL::Disconnect()
29  {
30      mysql_close(m_sock);
31  }
32
33  bool CMySQL::ConnectMySQL(char *host, char *user, char *pass, char *db,
      short nport)
34  {
35      if (!mysql_real_connect(m_sock, host, user, pass, db, nport, NULL,
      /*CLIENT_MULTI_STATEMENTS*/0)) {
36          //连接错误
37          return false;
38      }
39      return true;
40  }
41
42  bool CMySQL::GetTables(char* szSql, list<string>& lstStr)
43  {
44      if(mysql_query(m_sock, szSql)) {

```

```

44         return false;
45     }
46
47     m_results = mysql_store_result(m_sock);
48     if(NULL == m_results) {
49         return false;
50     }
51     while (m_record = mysql_fetch_row(m_results)) {
52         lstStr.push_back(m_record[0]);
53     }
54     return true;
55 }
56 bool CMySQL::SelectMySQL(char* szSql, int nColumn, list<string>& lstStr)
57 {
58     //mysql_query() 函数用于向 MySQL 发送并执行 SQL 语句
59     if(mysql_query(m_sock, szSql)) {
60         return false;
61     }
62
63     /*.mysql_store_result 对于成功检索了数据的每个查询(SELECT、SHOW、DESCRIBE、EXPLAIN、CHECK TABLE等)
64     返回值:
65     . CR_COMMANDS_OUT_OF_SYNC      以不恰当的顺序执行了命令。
66     . CR_OUT_OF_MEMORY            内存溢出。
67     . CR_SERVER_GONE_ERROR        MySQL服务器不可用。
68     . CR_SERVER_LOST              在查询过程中, 与服务器的连接丢失。
69     . CR_UNKNOWN_ERROR            出现未知错误。*/
70     m_results = mysql_store_result(m_sock);
71     if(NULL == m_results) return false;
72
73     //获取每行的列数:mysql_field_count
74     nColumn = nColumn==0?:mysql_field_count(m_sock):nColumn;
75
76
77     //遍历表中的下一行, 取出内容放入m_record 结果集
78     while (m_record = mysql_fetch_row(m_results)) {
79
80         for(int i = 0; i < nColumn; i++) {
81             if(!m_record[i]) {
82                 lstStr.push_back("");
83             } else {
84                 lstStr.push_back(m_record[i]);
85             }

```

```

86         }
87     }
88     return true;
89 }
90
91 bool CMySQL::UpdateMySQL(char* szSql)
92 {
93     if(!szSql) {
94         return false;
95     }
96     if(mysql_query(m_sock, szSql)) {
97         qDebug()<<mysql_error(m_sock);
98         return false;
99     }
100     return true;
101 }
102
103 //-----add by yuesl begin -----
104 -----
105
106 bool CMySQL::ConnectMySQL(C_CHAR *host, C_CHAR *user, C_CHAR *pass,
107 C_CHAR *db, const short nport, char (&pErrinfo)[MAX_ERR_LEN]){
108     if (!mysql_real_connect(m_sock, host, user, pass, db, nport, NULL,
109 /*CLIENT_MULTI_STATEMENTS*/0)) {
110         //连接错误
111         strncpy_s(pErrinfo,mysql_error(m_sock),MAX_ERR_LEN-1); //最大
112         截取MAX_ERR_LEN-1长度的错误信息
113         return false;
114     }
115     return true;
116 }
117
118 //查询功能
119 bool CMySQL::SelectMySQL(C_CHAR* szSql, list<vector<string>>& lstVecS
120 tr, char (&pErrinfo)[MAX_ERR_LEN])
121 {
122     if(mysql_query(m_sock, szSql)) { //mysql_query() 函数用于向 MySQL
123         发送并执行 SQL 语句
124         //执行失败，返回错误信息
125         strncpy_s(pErrinfo,mysql_error(m_sock),MAX_ERR_LEN-1);
126         return false;
127     }
128
129     /*.mysql_store_result 对于成功检索了数据的每个查询(SELECT、SHOW、DESC

```

RIBE、EXPLAIN、CHECK TABLE等)

返回值:

- CR_COMMANDS_OUT_OF_SYNC 以不恰当的顺序执行了命令。
- CR_OUT_OF_MEMORY 内存溢出。
- CR_SERVER_GONE_ERROR MySQL服务器不可用。
- CR_SERVER_LOST 在查询过程中,与服务器的连接丢失。
- CR_UNKNOWN_ERROR 出现未知错误。

*/

```
m_results = mysql_store_result(m_sock); //获取结果集
```

```
if(nullptr == m_results)return false;
```

```
//获取表数据的列数
```

```
size_t nColumn = mysql_field_count(m_sock);
```

```
//遍历表中的下一行,取出内容放入m_record 结果集
```

```
while (m_record = mysql_fetch_row(m_results)) {
```

```
    vector<string> oneRowData(nColumn); //存储每一行数据
```

```
    for(int i = 0; i < nColumn; i++) {
```

```
        oneRowData[i] = m_record[i];
```

```
    }
```

```
    lstVecStr.push_back(oneRowData); //添加每行数据
```

```
}
```

```
return true;
```

```
}
```

```
bool CMySql::GetTableName(string table,vector<string>& vecColumn  
,char (&pErrinfo)[MAX_ERR_LEN]){
```

```
    if(table == ""){
```

```
        //参数为空
```

```
        strncpy_s(pErrinfo,"表名为空",MAX_ERR_LEN-1);
```

```
        return false;
```

```
    }
```

```
//mysql_query: 查询成功返回0
```

```
return !::mysql_query(m_sock, (string("select *from ") + table).c_str()  
r())? [&,this]() ->bool{
```

```
    if ( MYSQL_RES* res = nullptr;res = mysql_store_result(m_sock)  
) { //获得sql语句结束后返回的结果集
```

```
        //获取字段数量
```

```
        size_t columnNum = ::mysql_field_count(m_sock);
```

```
        for (int i = 0; i < columnNum;i++){ //在已知字段数量的情况下
```

```
        获取字段名
```

```
164         vecColumn.push_back( mysql_fetch_field(res)->name);
165     }
166     return true;
167 }
168 strncpy_s(pErrinfo,mysql_error(m_sock),MAX_ERR_LEN-1);
169 return false;
170
171 }(): (strncpy_s(pErrinfo,mysql_error(m_sock),MAX_ERR_LEN-1),false)
172 ;
173 }
```