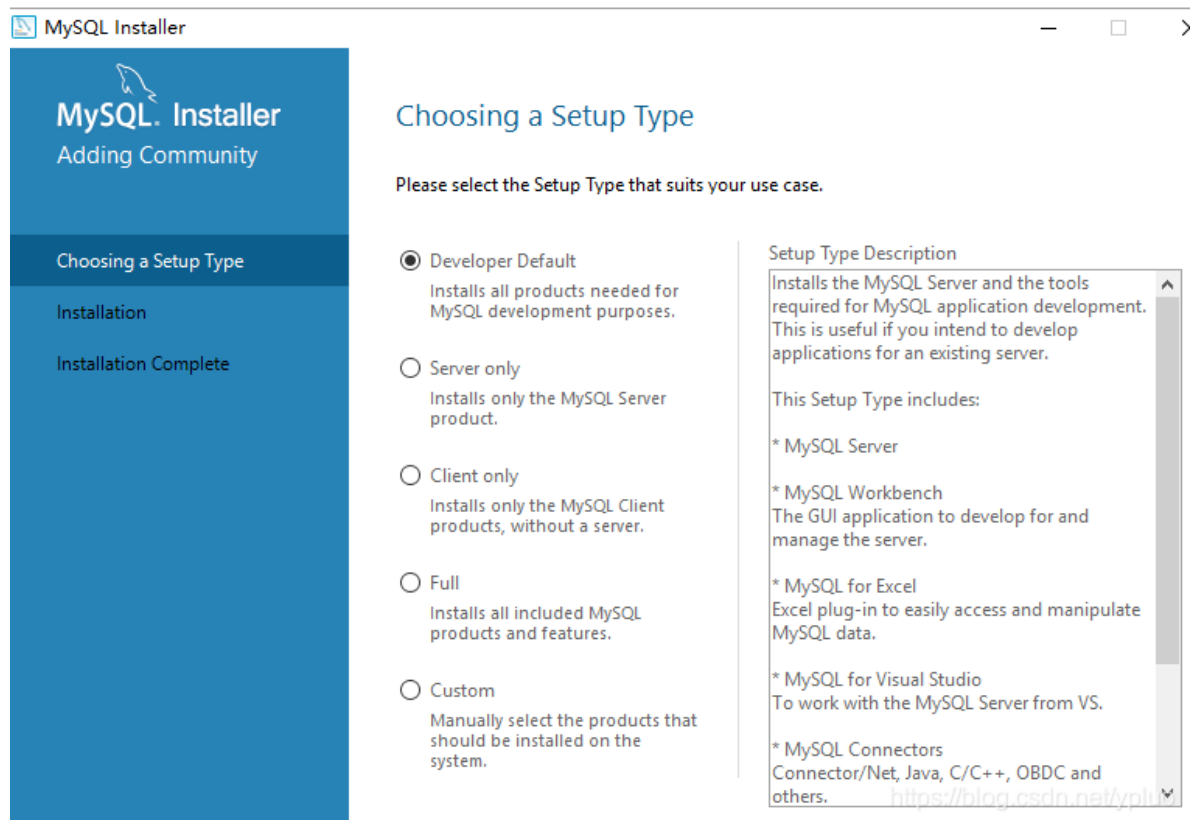
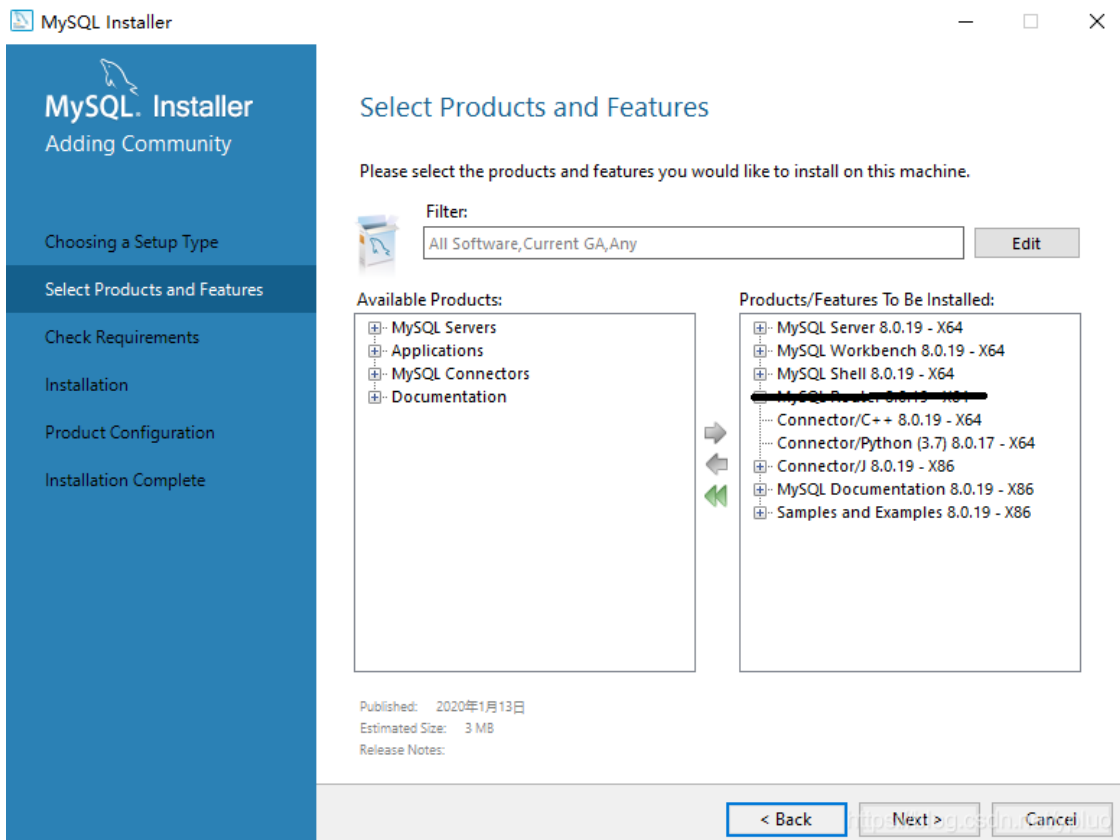


## 安装mysql (windows)

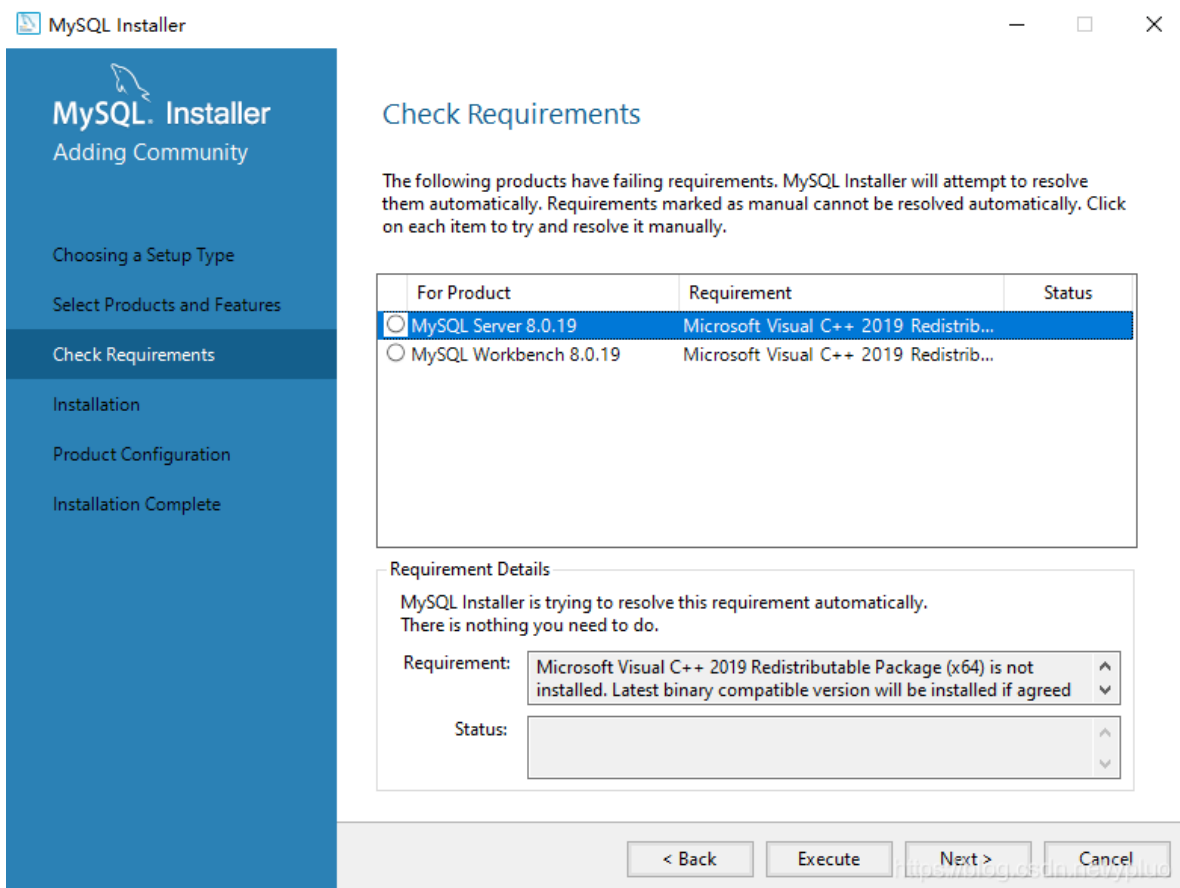
- 下载安装资源[mysql80](#)和[VC\\_redist\\_x64](#)
- 双击mysql80安装（可选择Developer Default或者Custom）



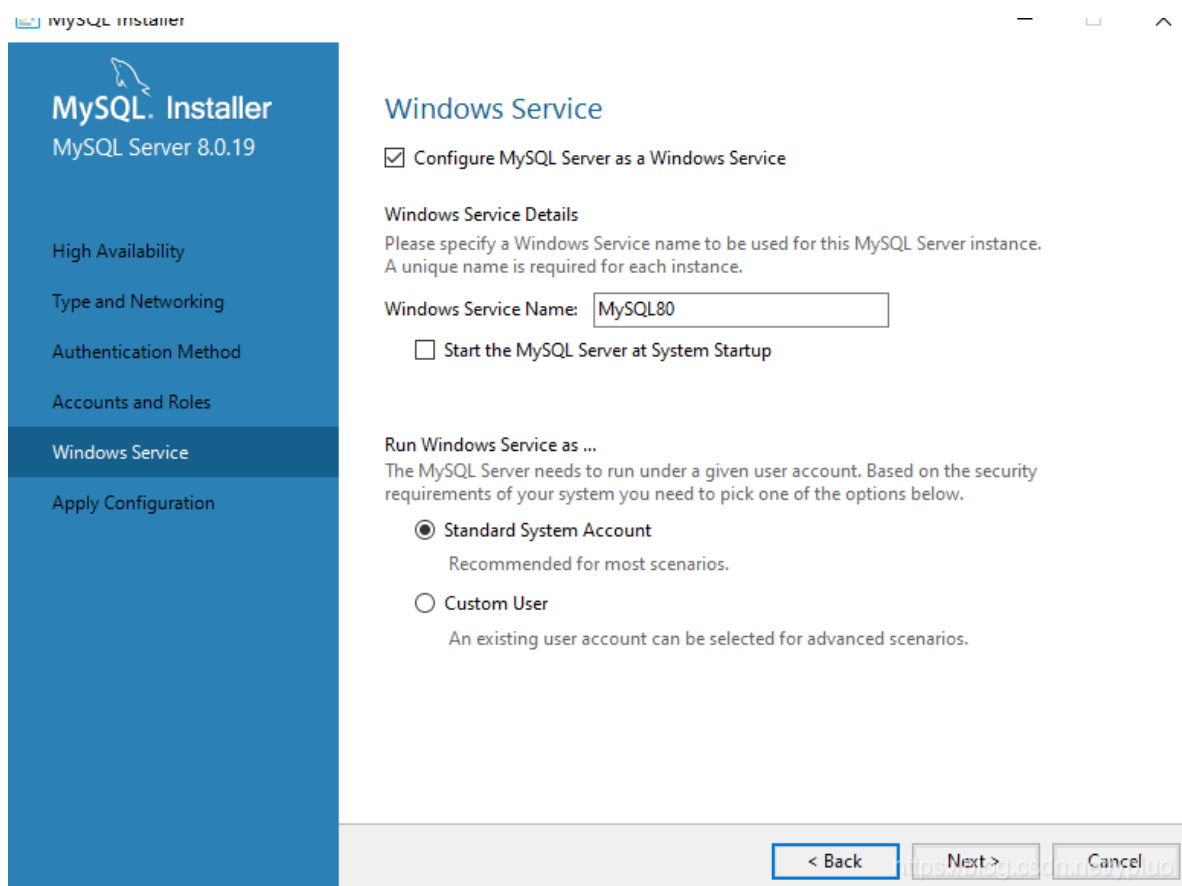
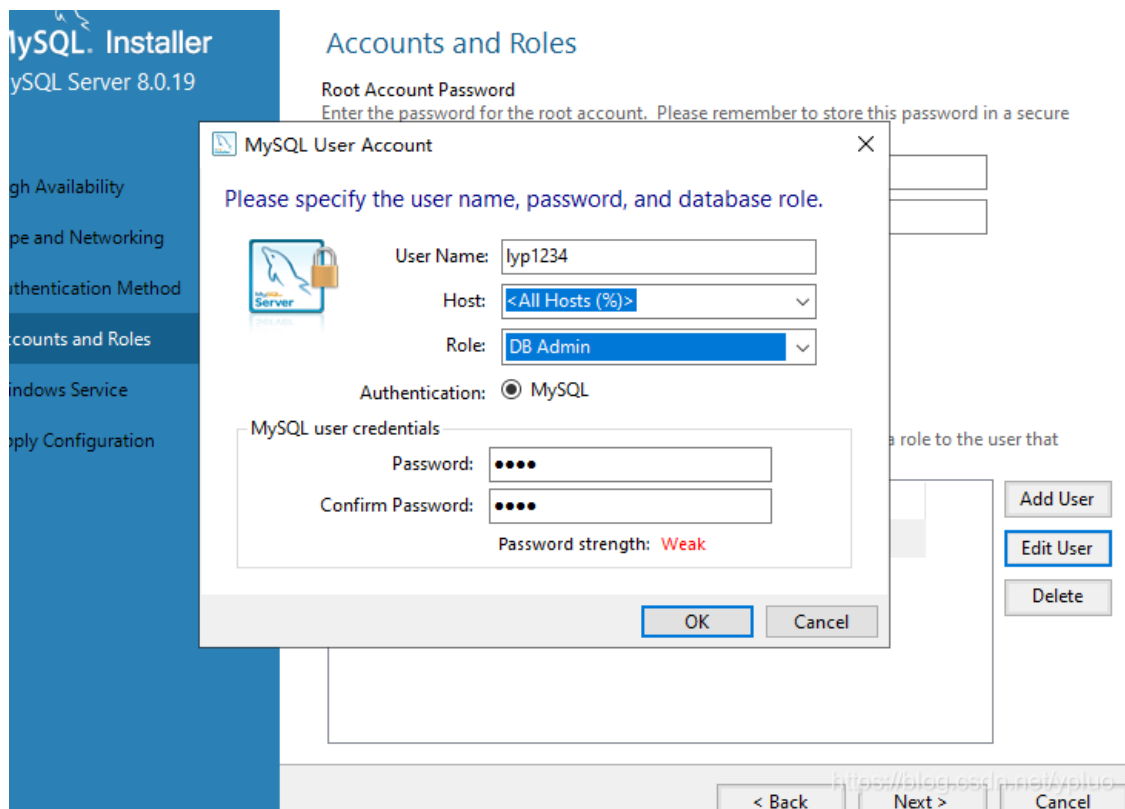
- 如果选择Custom安装，则可以选择想要安装的产品，下图中我们选择了【server（必须），workbench（必须），shell（可选）），Doc（可选）和Exmaple（可选）以及三个连接器用于从代码访问mysql数据库（ODBC，python，java，c++等，可选）。**注意：没列出的组件均可随喜好安装，安装时选择版本最好统一。**



- 检查依赖，通过安装界面提示安装所需库，我这边安装显示缺少库[VC redist x64](#)，下载安装即可。



- 下载安装组件，请保持网络连接。
- 选择服务器配置类型：默认设置。
- 认证方法：随意。
- 设置root用户密码和管理员用户名，管理员用户密码。此处建议使用<name: lyp1234, password: 1234>这种



- 完成安装。注意安装后可以使用安装器增加额外的组件，单个组件可以在[控制面板->程序]中手动删除。

## 安装mysql (linux)

Ubuntu:

```
sudo apt-get install mysql-server
```

详情见 <https://blog.csdn.net/weixx3/article/details/80782479>

CentOS:

```
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
rpm -ivh mysql-community-release-el7-5.noarch.rpm
yum update
yum install mysql-server
```

详情见 <https://www.runoob.com/mysql/mysql-install.html>

其它linux:自行查找资料

安装后密码为空, 可使用命令 `sudo mysqladmin -u root password";` 修改mysql root用户密码。

## 安装mysql (MacOS)

本人买不起MAC, [MAC安装mysql教程](#), 大致和windows安装流程相似。

## 启动mysql服务

Windows:

- 将mysql可执行命令添加到环境变量。mysql默认安装位置默认为 `C:\Program Files\MySQL\MySQL Server 8.0\bin`, 将其[添加到环境变量](#)中去即可在cmdline直接调用mysql。
- 添加环境变量后, 打开一个具有管理员权限的cmd, 输入命令 `net start mysql180` 启动mysql服务。

linux:

- 在命令行使用 `service mysql start/stop` 打开或关闭mysql服务

MacOS:

- 在命令行使用 `sudo /usr/local/mysql/support-files/mysql.server start/stop` 打开或关闭mysql服务

## 从命令行连接mysql

- 启动mysql服务后, 可以在一般权限的命令行中使用命令 `mysql -u <your user name> -p` 并输入密码启动mysql命令行。
- 以root用户在Linux系统登录需要sudo权限。

```

PS C:\WINDOWS\system32> mysql -u lyp1234 -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.19 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use test
Database changed
mysql> show tables
+-----+
| Tables_in_test |
+-----+
| tabl            |
+-----+
1 row in set (0.01 sec)

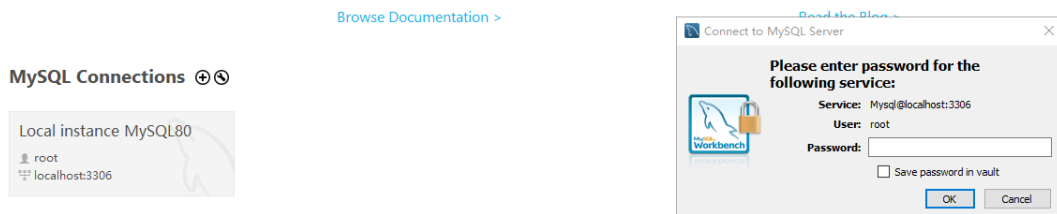
```

<https://blog.csdn.net/ypluo>

- 使用mysql命令行：
  - 创建数据库： `create database test`
  - 启动test数据库： `use test`
  - 在数据库下执行基本的sql命令。这里是[基本的sql命令](#)

## 使用workbench连接mysql (windows/MacOS only)

- 登录root用户，点击local instance MySQL80即可建立数据库连接



- 登录我的用户：点击上图中+号新建一个连接。

<https://blog.csdn.net/ypluo>

Setup New Connection

Connection Name:  Type a name for the connection

Connection Method:  Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname:  Port:  Name or IP address of the server host - and TCP/IP port.

Username:  Name of the user to connect with.

Password:   The user's password. Will be requested later if it's not set.

Default Schema:  The schema to use as default schema. Leave blank to select it later.

- 使用sql编辑器测试mysql, 点击闪电按钮可执行选中语句

```
create database test;
use test;

#drop table tab1;

create table tab1 (
  ID int not null,
  name varchar(10)
);

insert into tab1 (ID, name) values (1, 'aaa');
insert into tab1 (ID, name) values (2, 'bbb');
insert into tab1 (ID, name) values (3, 'ccc');

select * from tab1;
```

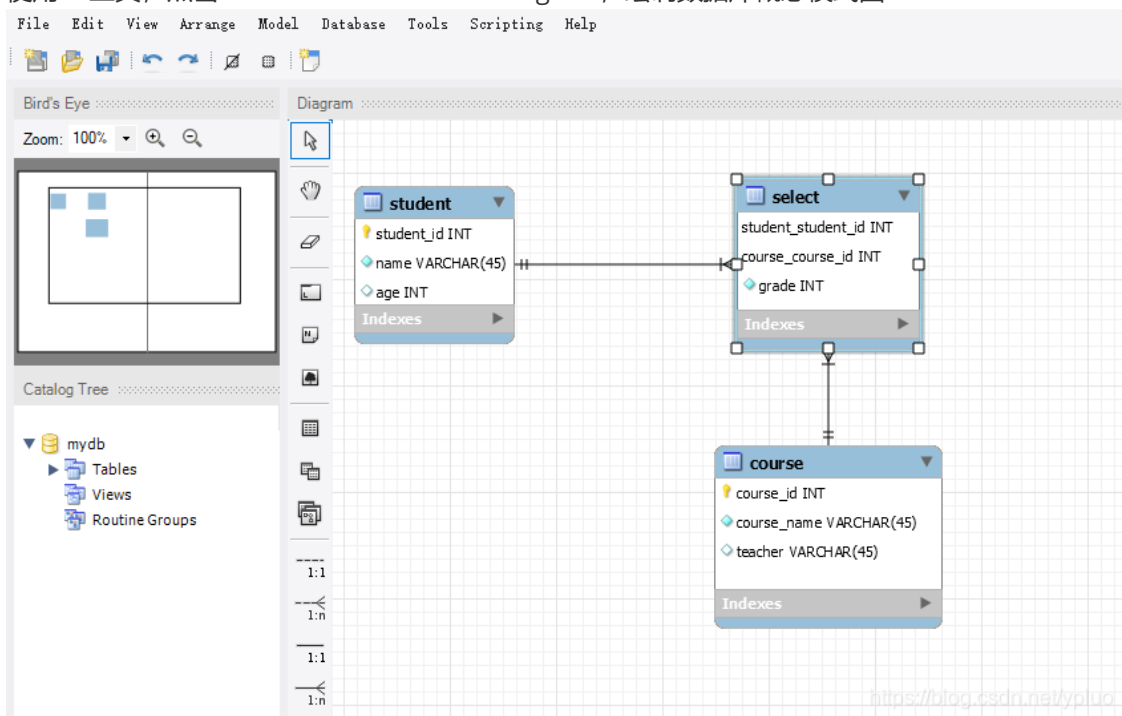
```

1 • create database test;
2
3 • use test;
4
5 • drop table tab1;
6
7 • create table tab1 (
8     ID int not null,
9     name varchar(10)
10 );
11
12 • insert into tab1 (ID, name) values (1, 'aaa');
13 • insert into tab1 (ID, name) values (2, 'bbb');
14 • insert into tab1 (ID, name) values (3, 'ccc');
15
16 • select * from tab1;

```

<https://blog.csdn.net/ypluo>

- 使用ER工具，点击File->New Model->Add diagram，绘制数据库概念模式图



<https://blog.csdn.net/ypluo>

## 使用python连接mysql

- 安装了python connector
- 使用 `pip install mysql` 安装python库MySQLdb
- 测试连接

```

import MySQLdb

# 打开数据库连接 url,username,password,database
db = MySQLdb.connect("localhost","lpy1234","1234","test" )

# 使用cursor()方法获取操作游标
cursor = db.cursor()

```

```
# 使用execute方法执行SQL语句
cursor.execute("SELECT * from tab1")

# 使用 fetchone() 方法获取一条数据
data = cursor.fetchall()

for d in data:
    print(d)

# 关闭数据库连接
db.close()
```

- 测试结果:

 Windows PowerShell

```
PS C:\Users\lyp\Desktop> python .\conect.py
(1, 'aaa')
(2, 'bbb')
(3, 'ccc')
PS C:\Users\lyp\Desktop>
```

## 使用C++连接mysql

- 安装了c++ connector
- 使用网上的[测试代码](#)，连接mysql数据库执行 `select * from tab1` 命令

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "mysql.h" //所需头文件

using namespace std;

int main()
{
    const char host[] = "localhost"; //MySQL服务器IP地址；若是本地可填写
    "localhost"或127.0.0.1
    const char user[] = "lyp1234"; //MySQL的用户名
    const char pswd[] = "1234"; //密码
    const char table[] = "test"; //数据库名称
    unsigned int port = 3306; //MySQL服务端口号，默认是3306
    MYSQL myCont; //创建MYSQL对象，定义数据库连接句柄
    MYSQL_RES *result; //查询结果集，存放查询结果
    MYSQL_ROW sql_row; //存放一行查询结果的字符串数组
    MYSQL_FIELD *fd; //包含字段信息的结构
    char column[32][32];
    int res;
    mysql_library_init(0, NULL, NULL); //初始化MySQL库

    mysql_init(&myCont); //初始化连接处理程序
    if(mysql_real_connect(&myCont, host, user, pswd, table, port, NULL, 0))
```



```

    { //通过调用mysql_real_connect()连接到服务器
        cout<<"connect succeed!"<<endl;
        mysql_query(&myCont, "SET NAMES GBK"); //设置编码格式,否则在cmd下无法显
示中文
        res=mysql_query(&myCont,"select * from tab1");//执行查询语句,
mysql_query如果查询成功, 零; 如果出现一个错误, 非零。
        if(!res)
        {
            result=mysql_store_result(&myCont); //保存查询到的数据到result
            if(result)
            {
                int i,j;
                cout<<"number of result: "<<(unsigned
long)mysql_num_rows(result)<<endl;
                for(i=0;fd=mysql_fetch_field(result);i++)//获取列名
                {
                    strcpy(column[i],fd->name);
                }
                j=mysql_num_fields(result);
                for(i=0;i<j;i++) {
                    printf("%s\t",column[i]);
                }
                printf("\n");
                while(sql_row=mysql_fetch_row(result))//获取具体的数据
                {
                    for(i=0;i<j;i++)
                    {
                        printf("%s\t",sql_row[i]);
                    }
                    printf("\n");
                }
            }
        } else {
            cout<<"query sql failed!"<<endl;
        }
    } else {
        cout<<"connect failed!"<<endl;
    }
}

//注意用完数据库要及时回收资源
if(result!=NULL) mysql_free_result(result); //释放结果资源
mysql_close(&myCont); //关闭MySQL连接
mysql_library_end(); //关闭MySQL库
return 0;
}

```

- 从mysql安装目录（我的是 C:\Program Files\MySQL\MySQL Server 8.0）中拷贝libmysql.dll和libmysql.lib到工程目录下
- gcc编译, 使用命令 g++ -I 'C:\Program Files\MySQL\MySQL Server 8.0\include' -L 'C:\Program Files\MySQL\MySQL Server 8.0\lib' test.cc -lmysql -o test 编译, 命令中指定了mysql 安装目录下的lib和include 文件, 注意填写你对应的文件夹。

- 执行结果

```
PS C:\Users\lyp\Desktop\test> .\test.exe
connect succeed!
number of result: 3
ID      name
1       aaa
2       bbb
3       ccc
```

- linux系统下应该就是mysql安装目录不同，编译时注意修正即可。