

# 1.补充材料：高级语言操作 MySQL 数据库

## 1.1 C 语言操作 MySQL

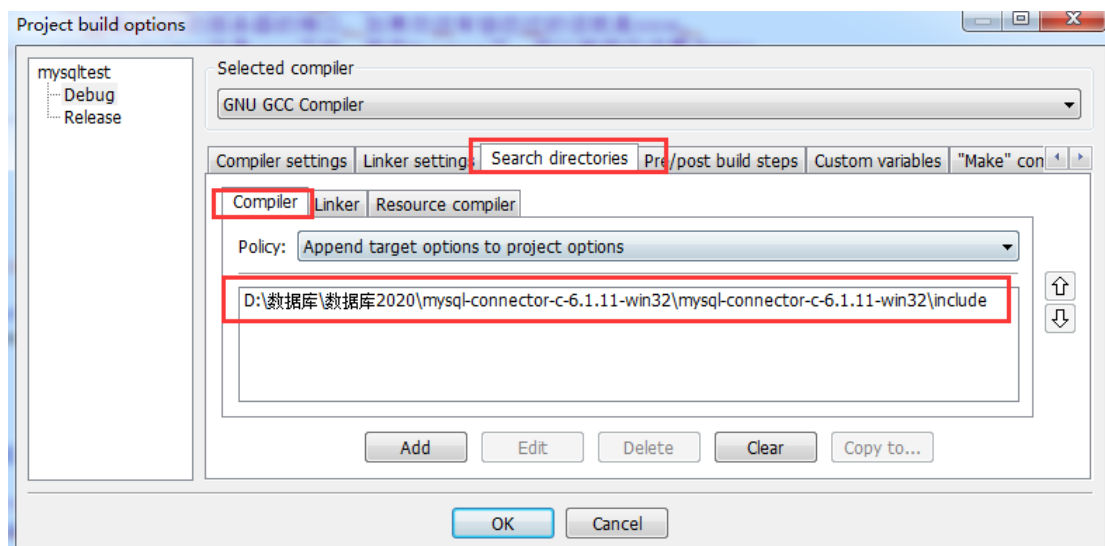
MySQL 安装路径下的 include 文件夹存放 C 语言编译相关的头文件（例如重要的 mysql.h），lib 文件夹存放相关的静态库（例如重要的 opt/libmysql.lib）。我们的 CodeBlocks 项目想要访问数据库，需要先进行配置。

本来配置的时候用 MySQL 安装路径下的 include、lib 就可以，但是安装的 MySQL 版本（64 位）跟 minGW（32 位）的不兼容，所以，另外下载了一个 mysql-connector-c-6.1.11-win32.zip，解压后放在 C:\。

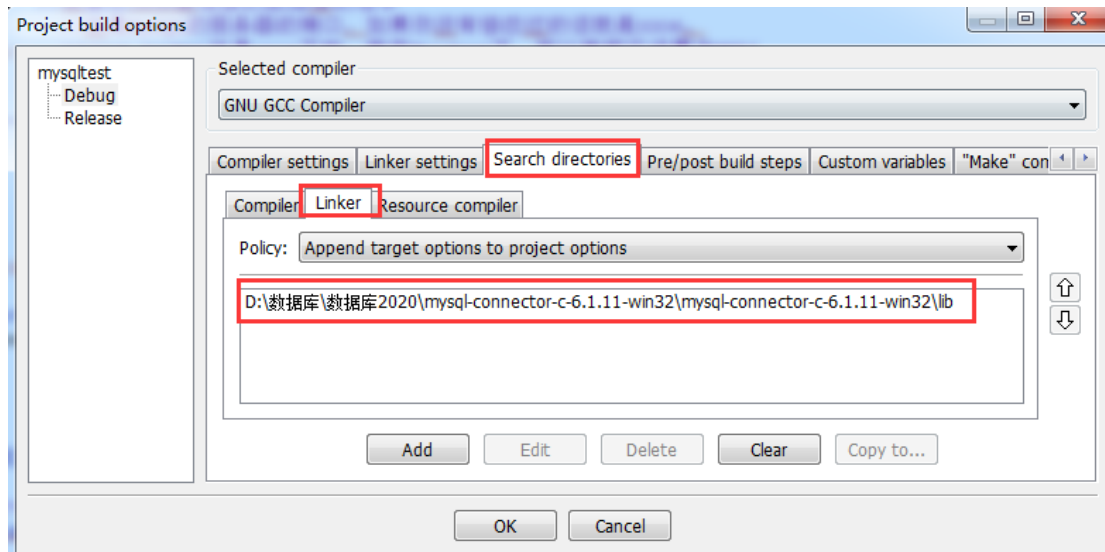
下载地址：<https://downloads.mysql.com/archives/c-c/>

### 配置步骤：

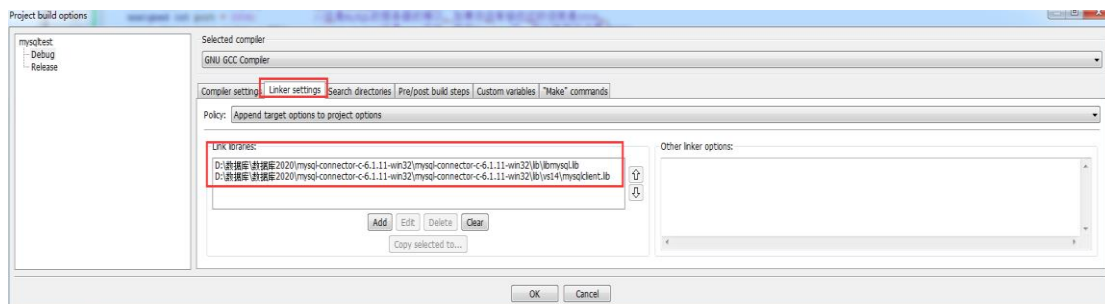
- 1、使用 Code::Blocks 建立空白工程。
- 2、在菜单 Project-->Build option-->点击 Search directories-->Compiler-->Add, 添加:



- 3、在菜单 Project-->Build option-->点击 Search directories-->Linker-->Add, 添加:



4、添加需要的库文件 lib/下的库文件 libmysql.lib 和 mysqlclient.lib。在菜单 Project-->Build option-->点击 Linker settings-->Add, 添加:



5、给个示例程序，示范数据库连接、查询、取结果集显示、释放结果集、关闭连接:

```

1 #include <stdio.h>
2 #include <winsock.h>
3 #include <mysql.h> //包含MySQL所需要的头文件
4 MYSQL mysql, *sock; //声明MySQL的句柄
5 void main(void)
6 {
7     const char * host = "127.0.0.1"; //因为是为本地测试，所以填写的是本地IP
8     const char * user = "root"; //这里改为你用户名，即连接MySQL的用户名
9     const char * passwd = "123456"; //这里改为你用户密码
10    const char * db = "company"; //这里改为你要连接的数据库的名字
11    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
12    const char * unix_socket = NULL; //unix_socket这是unix上的，我在Windows上，所以就把它设置为NULL
13    unsigned long client_flag = 0; //这个参数默认为0
14    const char * i_query = "select * from employee"; //查询语句
15    MYSQL_RES * result; //保存结果集
16    MYSQL_ROW row; //代表的是结果集中的一行
17    mysql_init(&mysql); //连接之前必须使用这个函数来初始化
18    if ( (sock = mysql_real_connect(&mysql, host, user, passwd, db, port, unix_socket, client_flag)) == NULL ) //连接MySQL
19    {
20        printf("连接失败，原因是: \n");
21        fprintf(stderr, "%s\n", mysql_error(&mysql));
22        exit(1);
23    }
24    else
25    {
26        fprintf(stderr, "连接MySQL成功! ! \n");
27    }
28    /*设置查询编码为gbk...以支持中文*/
29    mysql_query(&mysql, "set names gbk");
30
31    if ( mysql_query(&mysql, i_query) != 0 ) //如果连接成功，则开始查询
32    {
33        fprintf(stderr, "查询失败! \n");
34        exit(1);
35    }
36    else
37    {
38        if ( (result = mysql_store_result(&mysql)) == NULL ) //保存查询的结果
39        {
40            fprintf(stderr, "保存结果集失败! \n");
41            exit(1);
42        }
43        else
44        {
45            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
46            {
47                printf("name is %s\t", row[0]); //打印第1列的数据
48                printf("age is %s\t\n", row[1]); //打印第2列的数据
49            }
50        }
51    }
52    mysql_free_result(result); //释放结果集
53    mysql_close(sock); //关闭连接
54    system("pause");
55    return;
56 }

```

- a) 在程序中加入 `#include <mysql.h>`，让编译器能够找到所有 MySQL Client 的函数定义。随后，使用 `mysql_init` 函数准备连接，`mysql_init` 函数的声明为

**MYSQL \*mysql\_init(MYSQL \*mysql);**

其参数和返回值均为类型为 MYSQL 的结构体的指针。MYSQL 结构体是连接 MySQL 服务器的句柄，每一个句柄代表程序与 MySQL 服务器唯一的连接，所有 insert, select, delete 等操作都需要 MySQL 句柄作为参数以识别不同用户、数据库以及程序。若 `mysql_init` 操作成功，则返回刚刚初始化的句柄；否则，返回 NULL。

- b) 在句柄被初始化后，就可以使用 `mysql_real_connect` 函数或者 `mysql_connect` 函数连接 MySQL 服务器了，这两个函数的声明如下：

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

其中，第 1 个参数是 MySQL 句柄，第 2 至第 4 个参数分别为 host name、user name 及 password，第 5 个参数是待使用的数据库的名字，你也可把第 5 个参数设为 NULL，之后使用 `mysql_select_db` 函数来选择数据库，第 6 个参数是 MySQL 服务器的连接池，通常把它设为 `MYSQL_PORT`，第 7 个参数是 MySQL 服务器及 client 之间传输的渠道(socket 或 named pipe)，

但 MySQL 服务器会根据 host name 建立另一渠道，除非你有真的需要，否则你需在第 7 个参数填上 NULL。最后是 client\_flag，包括压缩协议、查询协议、加密协议等。

- c) 与 MySQL 服务器的连接成功后，使用 mysql\_close 函数断开连接，其函数声明为：

```
int mysql_close(MYSQL*);
```

- d) 在成功连接 MySQL 服务器后，便可以使用 mysql\_query 函数查询数据库。mysql\_query 函数的声明如下：

```
int mysql_query(MYSQL *mysql, const char *query);
```

第 1 个参数是 MySQL 连接的句柄，第 2 个参数是 SQL 语句。如果查询成功，mysql\_query 返回 0，否则返回非 0。

如果你的查询语句不需要结果返回，例如 DELETE、UPDATE、INSERT 等，mysql\_query 被执行后便完成整个操作了；如果你要执行 SELECT、SHOW、DESCRIBE 等，在存取结果前，必须使用 mysql\_store\_result 函数建立查询结果的句柄。mysql\_store\_result 的函数声明如下：

```
MYSQL_RES *mysql_store_result(MYSQL *mysql);
```

其输入参数为 MySQL 的连接句柄，其输出为 MYSQL\_RES 结构体指针类型。MYSQL\_RES 结构体表示的是 MySQL 数据库传递回来的查询结果，但我们并不会直接读取 MYSQL\_RES 的数据，而是使用 MYSQL\_ROW，也即是以行为单位读取数据。

注意 mysql\_result 是一个指针。因为 mysql\_store\_result 函数会自动分配内存存储查询结果，所以需要执行

```
mysql_free_result(MYSQL_RES*)来释放内存。
```

- e) 在提取结果前必须使用上面介绍过的 mysql\_store\_result 函数为查询结果分配内存，然后使用 mysql\_fetch\_row 函数逐行提取数据。mysql\_fetch\_row 函数的声明如下：

```
MYSQL_ROW mysql_fetch_row(MYSQL_RES *result);
```

其输入参数为查询结果句柄，输出的类型为 MYSQL\_ROW。MYSQL\_ROW 是一个数组结构，数组中每一个元素依次为该元组各个属性上的值。

结果的元组数可以用 mysql\_num\_rows 函数返回。mysql\_num\_rows 函数的声明为

```
int mysql_num_rows(MYSQL_RES*);
```

其输入参数为查询结果句柄，输出结果为元组数量。

请注意如果数据库很大，而又没有用 mysql\_free\_result 函数释放内存的话，很容易发生内存溢出的问题。

## 1.2 Java 语言操作 MySQL

### 1. 安装 JDK1.8

官网下载 `jdk-8u241-windows-x64.exe` 并安装。

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

### 2. 安装 Eclipse

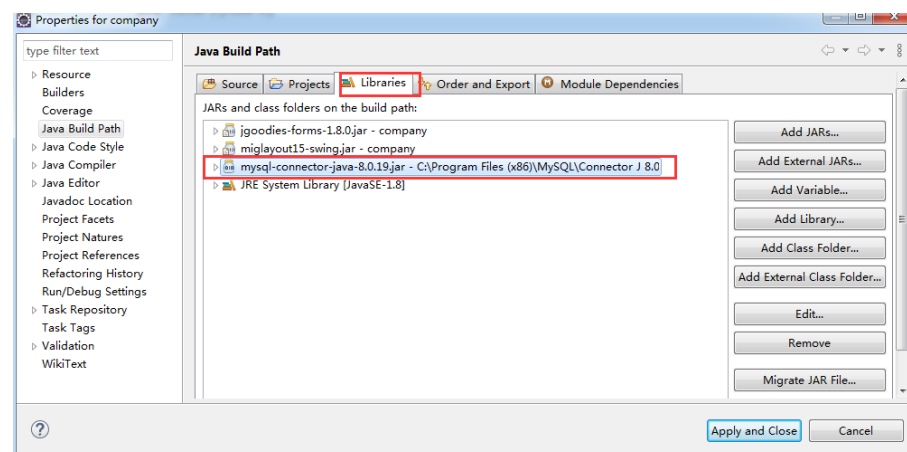
官网下载 `eclipse-inst-win64.exe` 并安装

<https://www.eclipse.org/downloads/>

### 3. 打开 eclipse，新建一个工程 Company，具体操作为：

点击 File ----> 点击 new ----> 点击 java project，在 project name 填工程名字，比如 Company，然后点击 finish。

右键项目 Compan ----> build path----> configure build path，添加 `mysql-connector-java-8.0.19.jar`。



### 4. 使用 java 进行数据库操作

```

public static void main(String[] args) throws Exception
{
    String URL = "jdbc:mysql://127.0.0.1:3306/company?serverTimezone=Asia/Shanghai";
    String USER = "root";
    String PASSWORD = "123456";
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
    String sql = "select pname from project";
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    if (rs != null) {
        while (rs.next()) {
            System.out.println(rs.getString("pname"));
        }
    }

    sql = "update employee set salary = 2000 where essn = 02 ";
    stmt = conn.createStatement();
    int num = stmt.executeUpdate(sql);
    if (num > 0) {
        System.out.println("Update successfully!");
    }
    conn.close();
}

```

## 1.3 Python 语言操作 MySQL

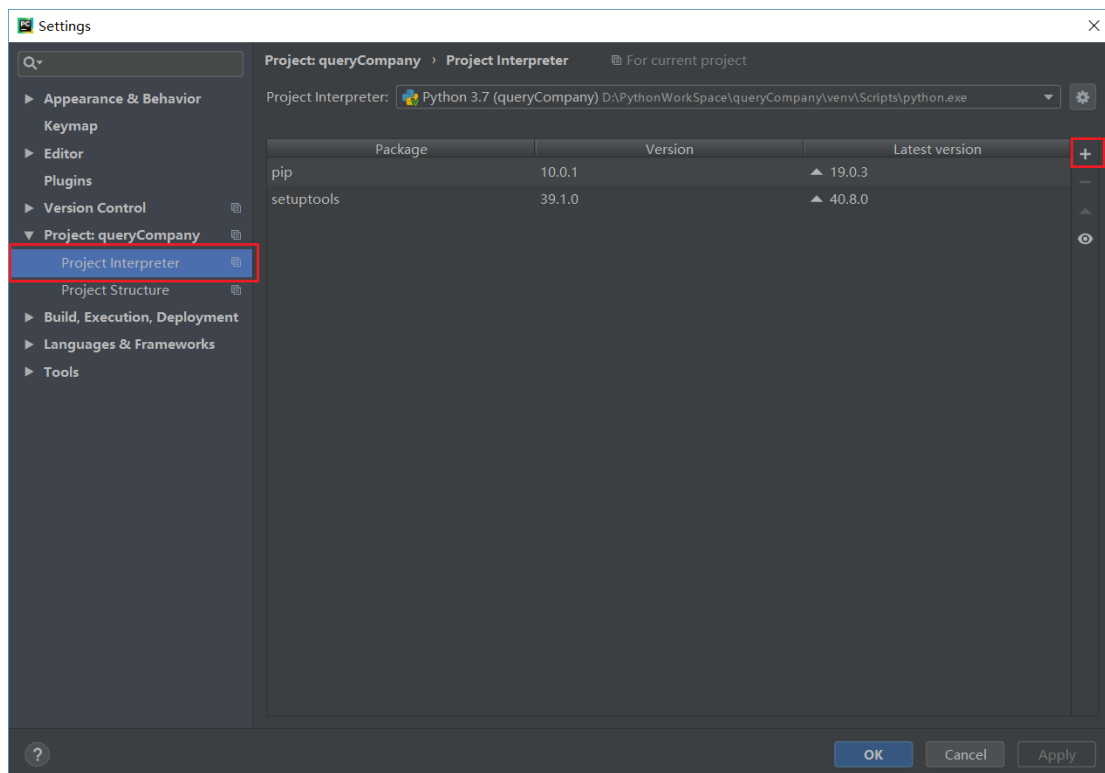
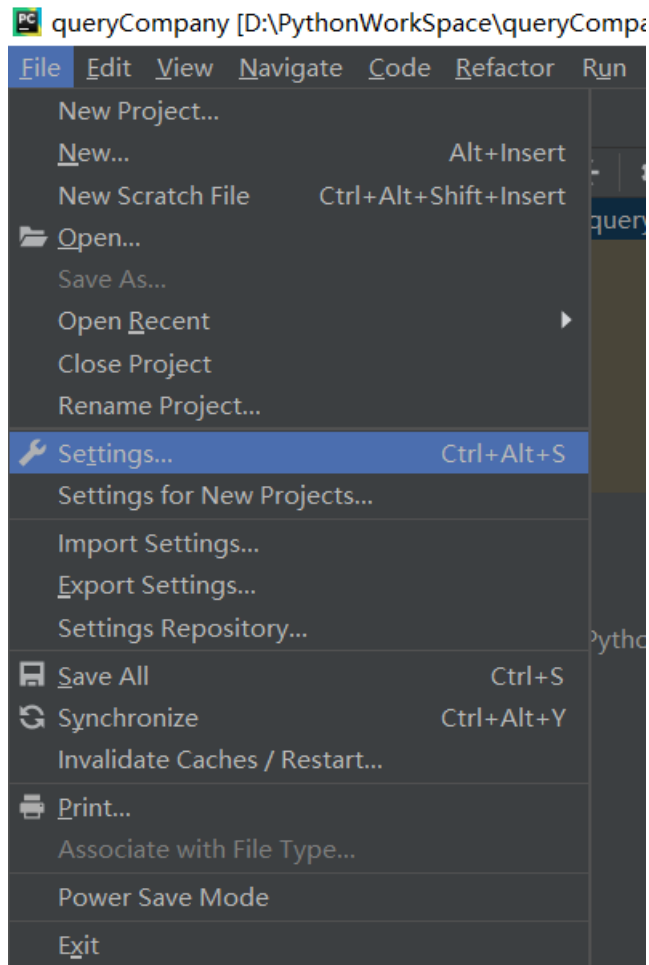
- 1.配置 python 环境（略）
- 2.安装 pymysql 库 命令: pip3 install pymysql

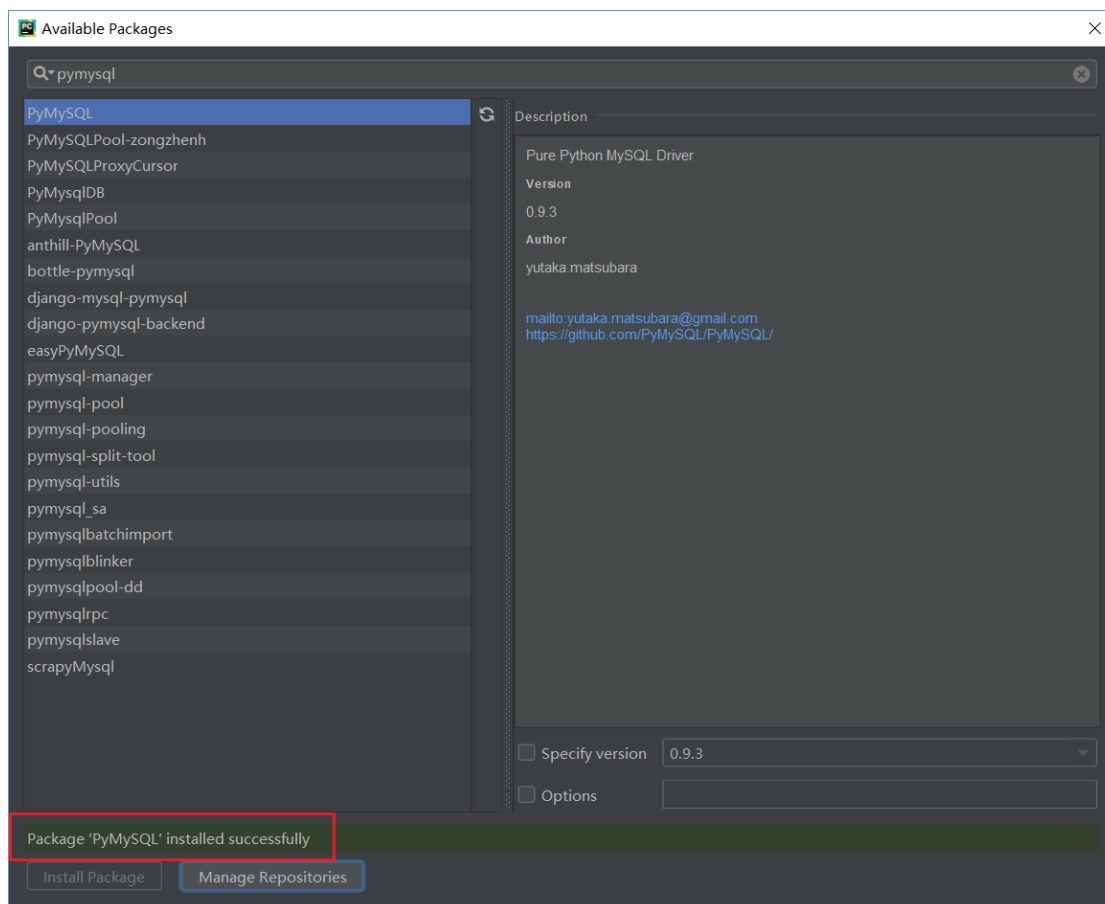
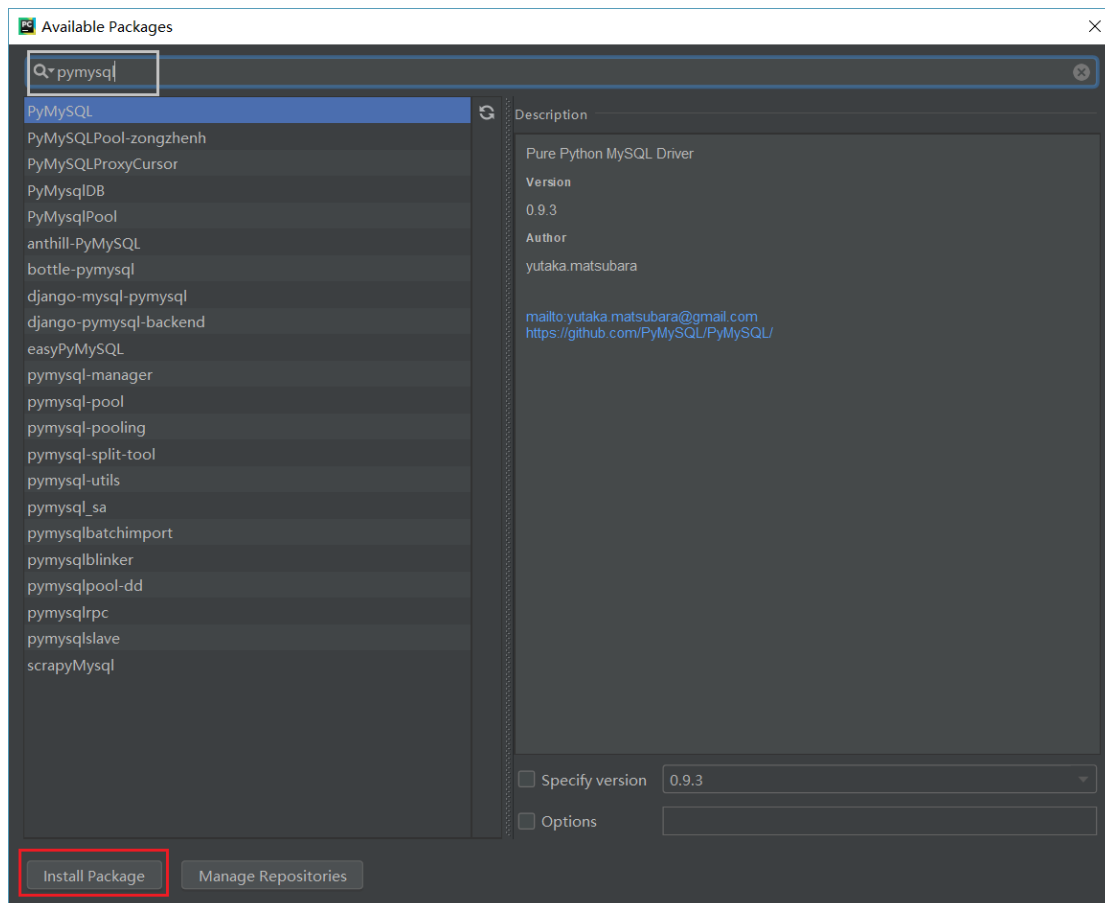
```

C:\Users\renyi>pip3 install pymysql
Collecting pymysql
  Downloading https://files.pythonhosted.org/packages/ed/39/15045ae46f2a123019aa968dfcba0396c161c
/PyMySQL-0.9.3-py2.py3-none-any.whl (47kB)
    100% |#####| 51kB 104kB/s
Installing collected packages: pymysql
Successfully installed pymysql-0.9.3
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

```

如果用 PyCharm 的，执行以下步骤：打开 File→Settings







添加成功。

### 3.使用 python 进行数据库操作

```
# coding=gb2312
import pymysql

conn = pymysql.connect(host='localhost', user='root', password='MySQL1234', database='company')
# 使用cursor()方法创建一个游标对象 cursor
cursor = conn.cursor()
# 使用execute()方法执行SQL查询
cursor.execute('select essn from works_on where pno = 'P3')
# 使用 fetchall()获取所有数据
data = cursor.fetchall()
for i in data:
    print(i)

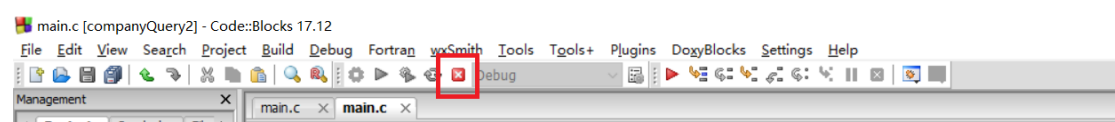
"""
如果需要建表、修改表等操作，执行cursor.execute(sql)
example:
sql = 'CREATE TABLE EMPLOYEE ( FIRST_NAME CHAR(20) NOT NULL, LAST_NAME CHAR(20), AGE INT, SEX CHAR(1))'
cursor.execute(sql)
"""

cursor.close()
conn.close()
```

## 2. 常见问题

### 2.1 程序运行后不要用 Abort 试图终止程序

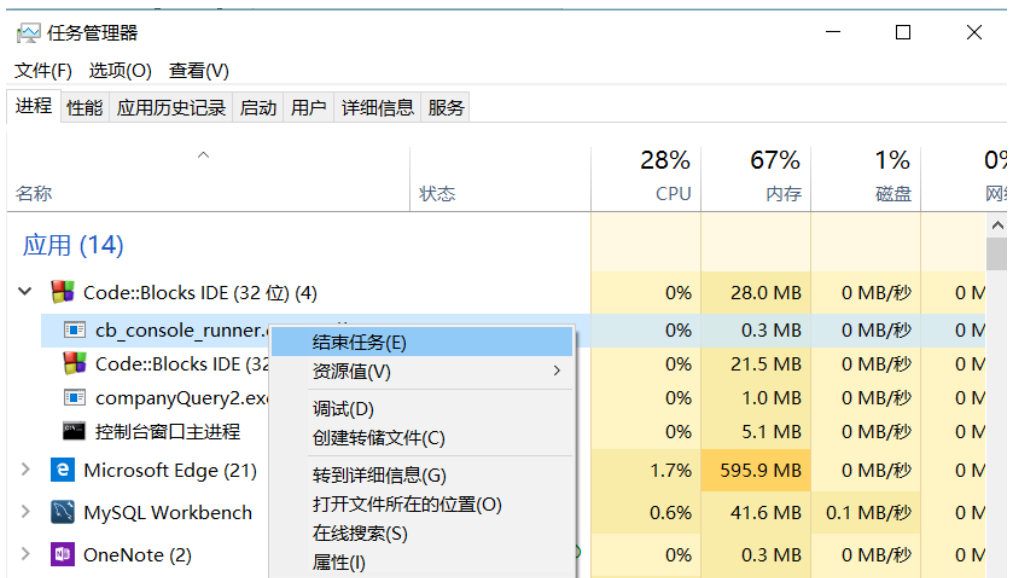
不要在程序运行中，点击下面这个按钮试图终止运行：



不过如果你已经这么干了，那就看看下文：

code::blocks 在执行编译后的控制台程序时是调用了一个控制台重指向器来执行程序的，这样你就不需要在程序里写程序结束后等待按键的代码以及提供程序的完整执行时间。但是如果你点击 code::blocks 的 Abort 按钮来结束程序的话，这个控制台重指向器不会被关闭，导致无法再执行程序。

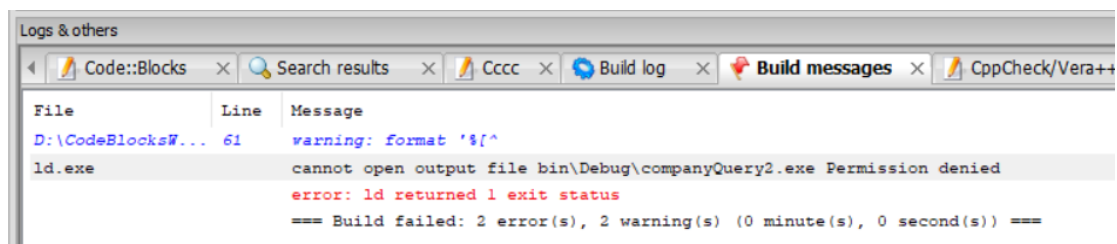
解决办法：到任务管理器的任务列表那里找到名为 cb\_console\_runner.exe 的程序，强制结束它即可。



正确方式：

- ①在运行结果黑框右上角的 X 点击一下，将其关掉；
- ②在黑框界面按任意键结束运行。

## 2.2 Cannot open output file permission denied



mingw32 编译器时经常遇到这个问题，原因有两个：

1. 有正在 run 的 exe 文件。只要把它终结就好；
2. 目录名字中有空格或汉字。只要改成全英文就行了；
3. 可以在 windows 控制台 cmd 中强制杀死该进程 `taskkill -F -IM XXX.exe`。

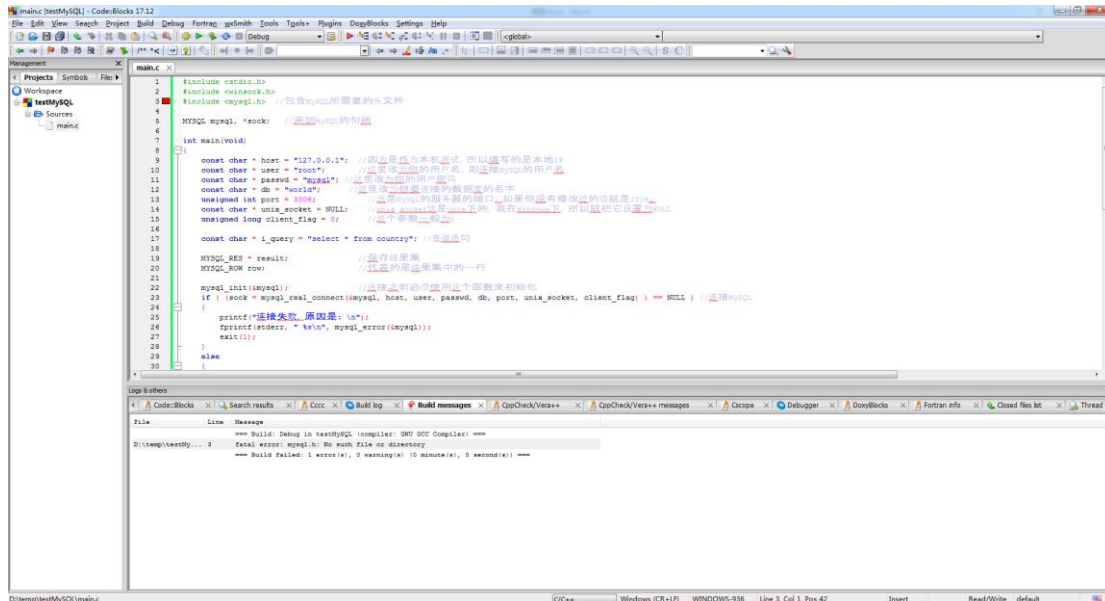
命令提示符

```

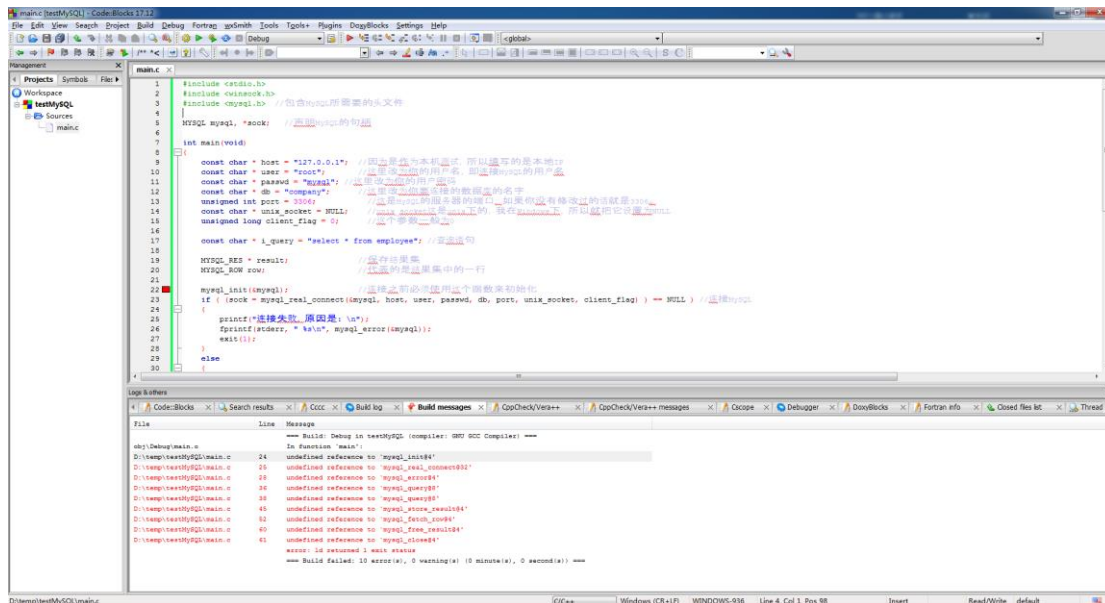
C:\Users\grace>taskkill -F -IM companyQuery2.exe
成功: 已终止进程 "companyQuery2.exe", 其 PID 为 10152。
C:\Users\grace>
  
```

## 2.3 Fatal error: mysql.h: No such file or directory

找不到 mysql.h 文件，请回到 1.1 节。



## 2.4 Undefined reference to 'mysql-init@4'



这个错误，是由于安装的 MySQL 版本（64 位）跟这里 Codeblocks（32 位）的不兼容。

解决办法：

（1）4.1 节配置用到的 include 还有 lib，不要使用 MySQL 安装目录下的，另外下载专门的 mysql-connector-c-6.1.11-win32.zip，用这个压缩包里的。

下载地址：<https://downloads.mysql.com/archives/c-c/>

## MySQL Product Archives

MySQL Connector/C (Archived Versions)

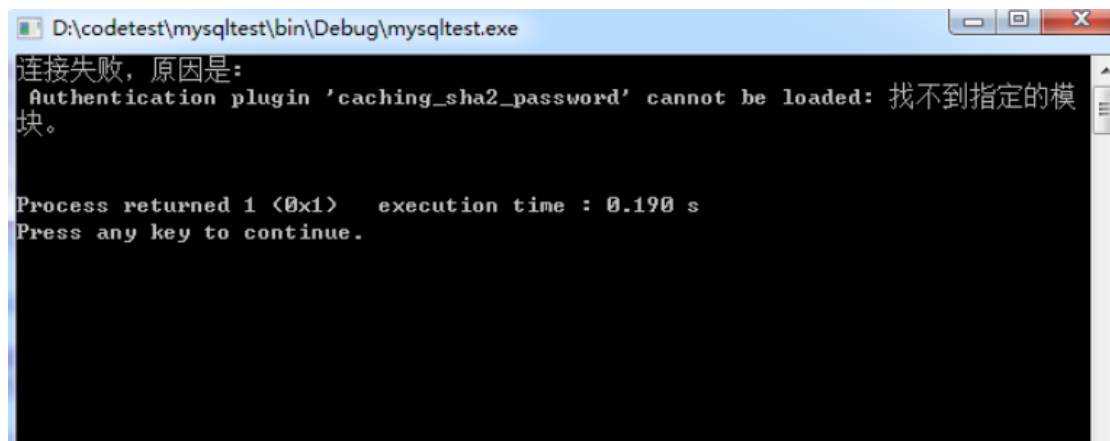
Please note that these are old versions. New releases will have recent bug fixes and features!  
To download the latest release of MySQL Connector/C, please visit [MySQL Downloads](#).

Product Version:   
Operating System:   
OS Version:

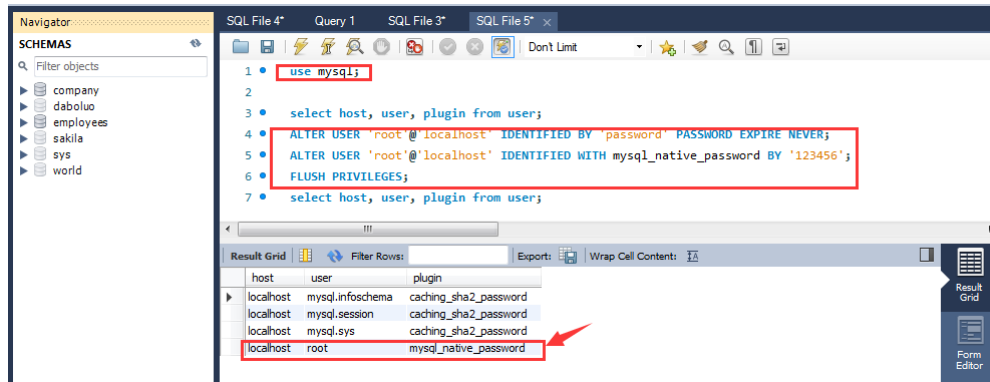
Windows (x86, 32-bit), MSI Installer (mysql-connector-c-6.1.11-win32.msi)	Jul 13, 2017	7.0M	<a href="#">Download</a>
Windows (x86, 64-bit), MSI Installer (mysql-connector-c-6.1.11-win64.msi)	Jul 13, 2017	7.6M	<a href="#">Download</a>
Windows (x86, 32-bit), ZIP Archive (mysql-connector-c-6.1.11-win32.zip)	Jul 17, 2017	6.9M	<a href="#">Download</a>
Windows (x86, 64-bit), ZIP Archive (mysql-connector-c-6.1.11-win64.zip)	Jul 17, 2017	7.5M	<a href="#">Download</a>
Windows (x86, 32-bit), ZIP Archive Debug Binaries & Test Suite (mysql-connector-c-6.1.11-win32-debug.zip)	Jul 17, 2017	17.9M	<a href="#">Download</a>
Windows (x86, 64-bit), ZIP Archive Debug Binaries & Test Suite (mysql-connector-c-6.1.11-win64-debug.zip)	Jul 17, 2017	18.4M	<a href="#">Download</a>

(2) 也可以将 codeblocks 的编译器换成 mingW64，这样就可以用 mysql 安装目录下的 include、lib 了。

## 2.5 Authentication plugin 'caching\_sha2\_password' cannot be loaded: 找不到指定的模块



出现这个问题的原因是 MySQL8.0 之前的版本中加密规则是 mysql\_native\_password，而在 MySQL8.0 之后，加密规则是 caching\_sha2\_password，解决问题的方法是把 MySQL 用户登录密码加密规则还原成 mysql\_native\_password。



## 2.6 ModuleNotFoundError: No module named 'pymysql'

运行报错:

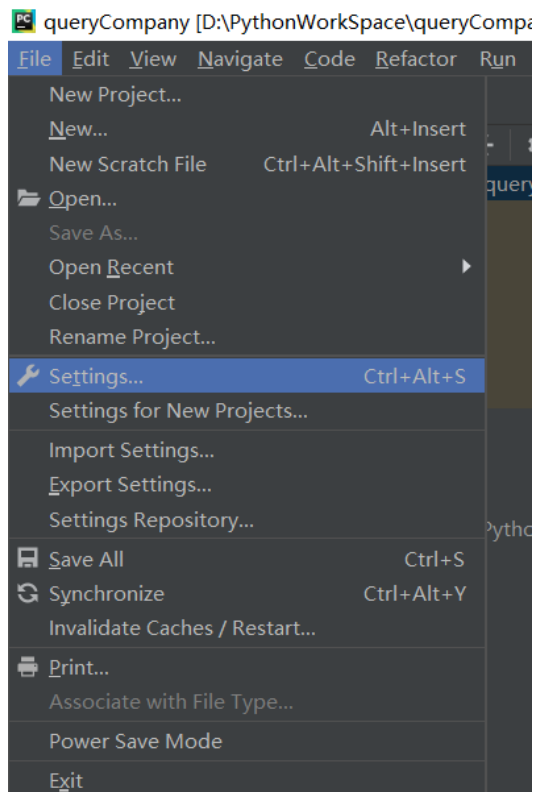
```

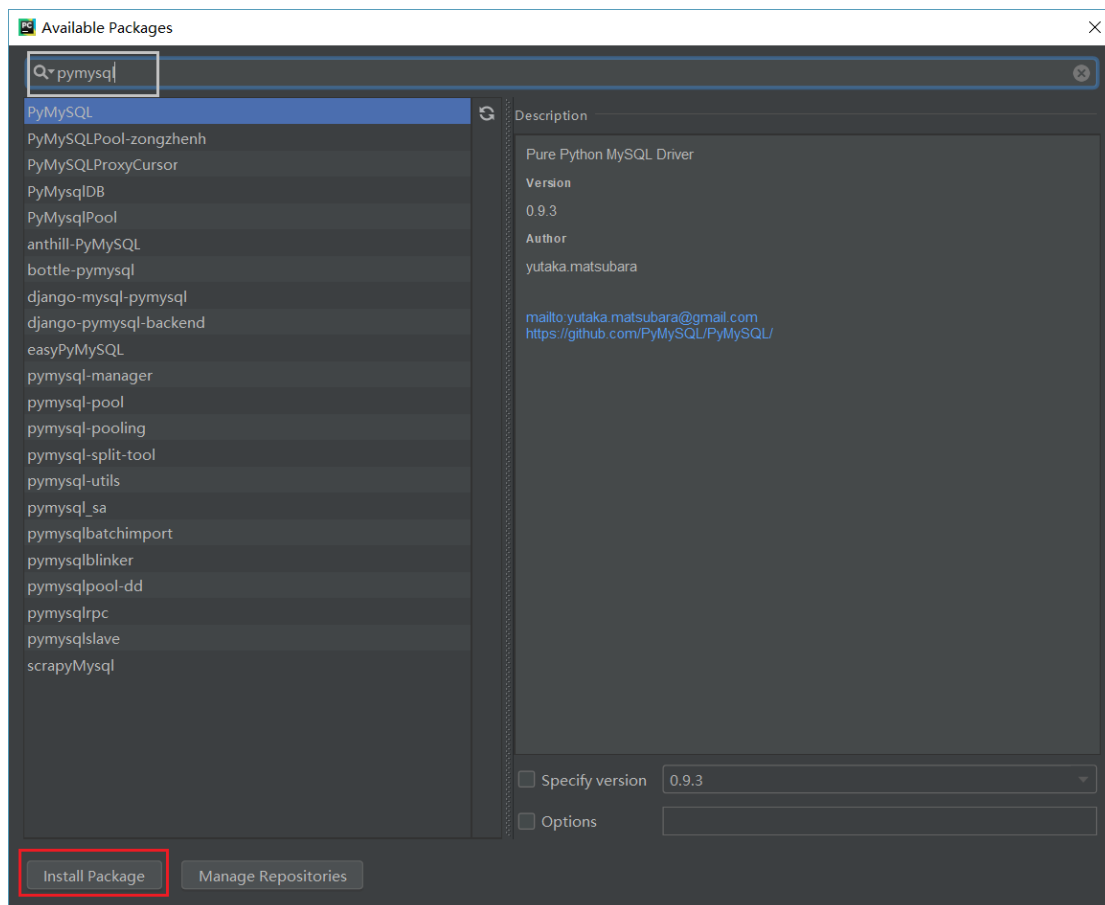
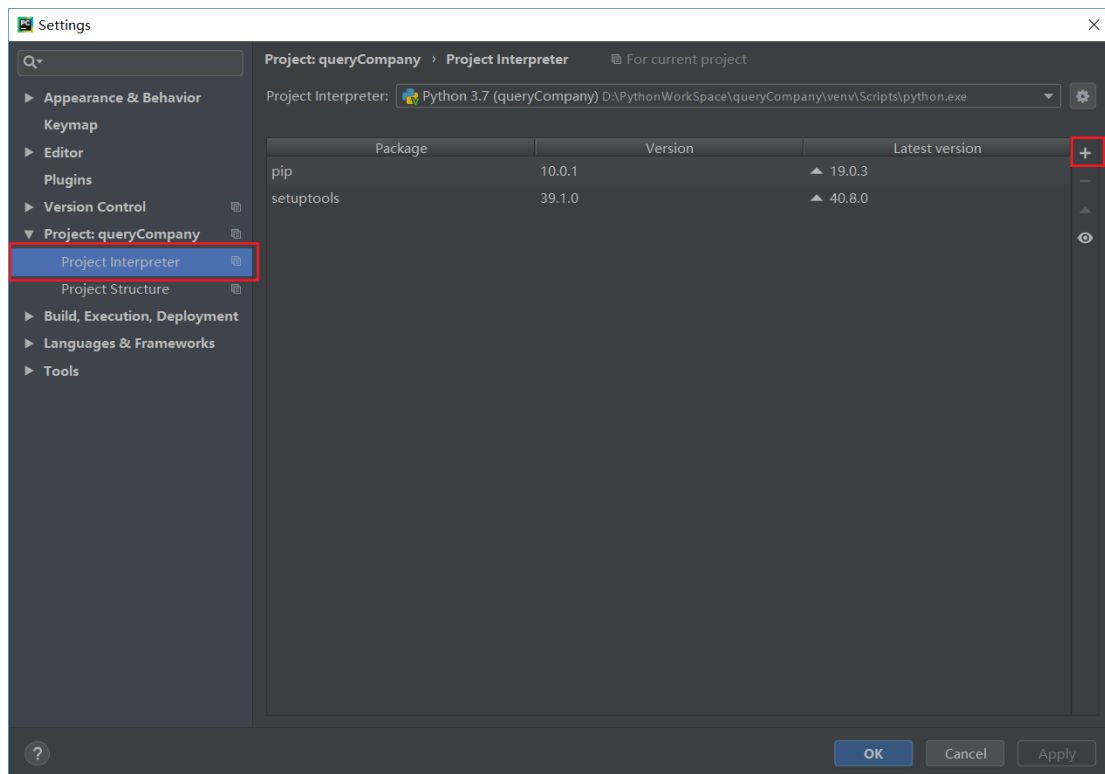
D:\PythonWorkSpace\queryCompany\venv\Scripts\python.exe D:/PythonWorkSpace/queryCompany/queryCompany2.py
Traceback (most recent call last):
  File "D:/PythonWorkSpace/queryCompany/queryCompany2.py", line 4, in <module>
    import pymysql
ModuleNotFoundError: No module named 'pymysql'

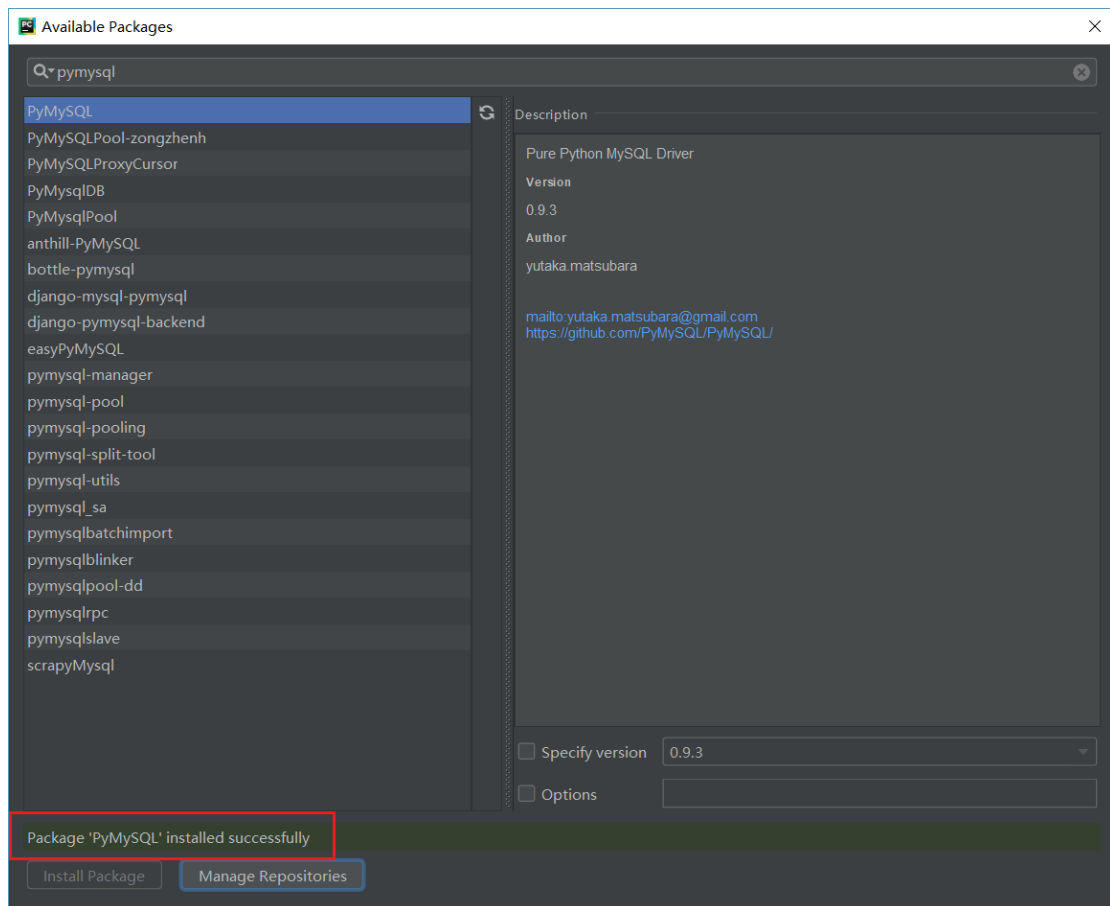
Process finished with exit code 1

```

解决办法:







添加成功:

