

第 1 章 MySQL 数据库

1.1 章节目标

1.2 MySQL 介绍

1.2.1 任务 1-MySQL 简介

MySQL 是一种开源的关系型数据库管理系统 (RDBMS)，它使用 SQL (结构化查询语言) 语言进行数据的存储和访问。MySQL 的设计目标是速度、高可靠性和容易使用，特别是在 Web 应用方面，它是目前应用最为广泛的关系型数据库管理系统之一。

MySQL 最初由瑞典 MySQL AB 公司开发，现已被 Oracle 公司收购。MySQL 分为两个版本，一个是社区版 (Community Edition)，一个是企业版 (Enterprise Edition)。社区版是免费使用的开源软件，Enterprise 版包含更多高级特性，需要付费许可证来使用。

MySQL 被广泛应用于各种领域，包括大型 Web 系统、企业级应用、金融等领域。它支持多种操作系统，包括 Windows、Linux、Mac OS 等，同时也支持多种编程语言和开发框架，如 Java、PHP、Node.js、Django 等。

1.2.2 任务 2-MySQL 发展历史

MySQL 的历史发展可以清晰地分为几个重要的阶段，以下是根据参考文章中的相关信息归纳的 MySQL 历史发展：

1. 创建与早期发展

1994 年：MySQL 由瑞典的 Michael Widenius 和 David Axmark 创建，最初是为了满足个人和小型应用程序的需求。

1995 年：MySQL 首次发布，并成为开源项目。这使得更多的开发者能够参与 MySQL 的改进和扩展，为其快速发展奠定了基础。

2. 版本迭代与功能增强

1996 年-1997 年：MySQL 2.0 和 3.0 版本发布，引入了子查询、事务支持和索引等重要功能。

2000 年：MySQL 4.0 发布，带来了存储过程、触发器、视图和外键支持，进一步提高了 MySQL 在复杂应用中的可用性。

2003 年：MySQL 5.0 发布，继续增强了存储过程、触发器和视图的支持，并加入了更强大的查询优化功能，使 MySQL 在企业级应用中更具竞争力。

3. 收购与分支创建

2008 年：Sun Microsystems 收购了 MySQL AB，这一收购引发了关于 MySQL 未来的担忧。

2010 年：Oracle 收购了 Sun Microsystems，进一步引发了关于 MySQL 开源性质和未来发展的担忧。同年，由于担忧 Oracle 对 MySQL 的控制，MySQL 的创始人之一 Michael Widenius 创建了 MariaDB，这是一个基于 MySQL 代码的开源分支。

5.持续更新与优化

2010 年至今：尽管 MySQL 被 Oracle 收购，但 MySQL 社区版本仍然存在，继续得到开源社区的支持和开发。MySQL 5.5 及之后的版本继续引入性能和功能方面的改进，如 InnoDB 存储引擎的改进、复制和集群支持等。

2018 年：MySQL 8.0 发布，引入了许多新功能，包括窗口函数、通用表表达式（CTE）等，进一步增强了 MySQL 的功能和性能。

6.版本升级与兼容性

从 MySQL 5.7 到 8.0，再到更新的版本，每一次升级都带来了许多改进和新功能，从而提高了系统的稳定性和安全性。但需要注意的是，MySQL 的升级通常不支持直接跨越大版本，例如从 5.6 直接升级到 8.0 是不可行的，需要逐步升级。

总结来说，MySQL 自 1995 年首次发布以来，经历了多个版本的迭代和功能增强，逐渐从一个简单的 SQL 数据库发展成为一个功能强大、广泛应用于各种场景的关系型数据库管理系统。尽管经历了多次收购和开源社区的变迁，但 MySQL 依然保持着其开源、易用、高性能和可扩展性等核心特点，继续为全球的开发者和企业提供服务。

1.2.3 任务 3-MySQL 的特点与优势

MySQL 的特点与优势主要体现在以下几个方面：

1.开源与免费

MySQL 是一个开源的数据库管理系统，这意味着任何人都可以使用和修改其源代码，无需支付任何费用。这种开源的特性使得 MySQL 在全球范围内得到了广泛的应用，尤其是在中小企业和个人开发者中。同时，开源也促进了 MySQL 的快速发展，因为任何人都可以为其贡献代码，共同改进和优化系统。

2.高性能

MySQL 具有出色的性能表现，尤其是在处理大量数据和复杂查询时。它采用了优化的 SQL 查询算法和索引技术，能够快速响应各种查询请求。此外，MySQL 还支持多种存储引擎，如 InnoDB、MyISAM 等，每种存储引擎都有其独特的优势和适用场景，用户可以根据实际需求选择合适的存储引擎来提高性能。

3.易用性

MySQL 的语法简单易懂，学习成本较低。它提供了丰富的 API 接口和工具，使得开发者可以轻松地与数据库进行交互。同时，MySQL 还提供了完善的文档和社区支持，用户可以在遇到问题时快速找到解决方案。

4.稳定性

MySQL 经过多年的发展和完善，已经具备了很高的稳定性。它可以在各种操作系统上运行，并且支持大量的并发连接。MySQL 还提供了多种备份和恢复机制，确保数据的安全

性和可靠性。

5.可扩展性

MySQL 支持分布式架构和分片技术，可以轻松应对大规模数据和高并发场景。通过添加更多的服务器和节点，可以水平扩展 MySQL 的处理能力，满足不断增长的业务需求。此外，MySQL 还支持插件式存储引擎架构，可以方便地扩展新功能，满足特定场景的需求。

6.安全性

MySQL 提供了多种安全特性，如用户权限管理、数据加密、访问控制等，确保数据的安全性。它还支持 SSL/TLS 加密协议，可以在传输过程中保护数据的机密性和完整性。此外，MySQL 还提供了审计和日志功能，可以记录用户的操作行为，方便进行安全审计和故障排查。

7.广泛的社区支持

MySQL 拥有庞大的社区支持，包括开发者、用户、贡献者等。社区提供了丰富的文档、教程、示例代码等资源，帮助用户更好地使用 MySQL。同时，社区还提供了论坛、邮件列表等交流渠道，方便用户之间互相学习和交流。这种广泛的社区支持使得 MySQL 能够持续不断地发展和完善。

综上所述，MySQL 以其开源、高性能、易用性、稳定性、可扩展性、安全性和广泛的社区支持等特点与优势，成为了全球范围内广泛应用的数据库管理系统之一。

1.2.4 任务 4-MySQL 的应用场景

MySQL 的应用场景非常广泛，涵盖了各行各业以及不同类型的应用。以下是一些典型的 MySQL 应用场景：

1. Web 应用程序

内容管理系统（CMS）：如 WordPress、Drupal 和 Joomla 等流行的 CMS 平台，它们使用 MySQL 作为后端数据库来存储网站内容、用户信息、评论等数据。

电子商务网站：众多电子商务网站，包括自建电商平台和基于开源解决方案（如 PrestaShop、Magento 等）的平台，都依赖 MySQL 来管理产品目录、订单、客户信息等。

社交网络与论坛：类似于 Facebook、Twitter 这样的社交网络，以及各种在线论坛和社区，使用 MySQL 来存储用户资料、帖子、评论、点赞等信息。

2. 企业级应用

客户关系管理（CRM）系统：用于跟踪和管理公司与客户之间的交互，包括客户资料、销售机会、服务请求等。

企业资源规划（ERP）系统：集成企业内部各个部门的信息，如财务、供应链、人力资源等，MySQL 用于存储和查询这些关键业务数据。

数据分析与报告：MySQL 可用于构建数据仓库，支持 OLAP（联机分析处理）查询和报告生成，帮助企业做出基于数据的决策。

3. 软件开发与测试

软件开发环境：在软件开发过程中，MySQL 常用作后端数据库，支持应用程序的原型设计、开发和测试。

持续集成/持续部署（CI/CD）：在自动化的测试环境中，MySQL 数据库用于存储测试数据和结果。

4. 科研与教育

科学研究：在科学研究中，MySQL 可用于存储实验数据、模拟结果和研究资料，方便科学家进行数据分析和可视化。

教育环境：在教育机构和培训课程中，MySQL 常被用作教授数据库管理和 SQL 编程的实践平台。

5. 游戏开发

在线游戏：多人在线游戏（MMOG）和社交游戏使用 MySQL 来存储玩家资料、游戏状态、排行榜等信息。

游戏后台管理：游戏开发者利用 MySQL 管理游戏内的物品、角色、任务等游戏元素。

6. 移动应用开发

后端服务：MySQL 作为移动应用的后端数据库，存储用户数据、应用设置和同步信息。

推送通知服务：存储和管理推送通知的订阅者列表、发送记录等。

7. 物联网（IoT）

数据存储与分析：在物联网应用中，MySQL 可用于存储从各种传感器和设备收集的数据，并进行后续的分析和处理。

总的来说，MySQL 的灵活性和可靠性使其适用于各种规模和类型的应用场景，从简单的个人网站到复杂的企业级解决方案。

1.3 MySQL 环境搭建

1.3.1 任务 5-Windows10 下载 MySQL

官网下载地址：

<https://downloads.mysql.com/archives/installer/>

下载界面：

← → ↻ 🔍 downloads.mysql.com/archives/installer/ 📄 ☆ 🗑️ 👤

⚠️ 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:


Operating System:

Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-web-community-8.0.36.0.msi)</small>	Jan 2, 2024	2.1M	Download
Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-community-8.0.36.0.msi)</small>	Jan 2, 2024	285.3M	Download

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

MySQL open source software is provided under the [GPL License](#).

上面那个是需要联网安装，我们下载下面这个，直接本地安装。

名称	修改日期	类型	大小
 mysql-installer-community-8.0.36.0.msi	2024/5/30 10:28	Windows Installer ...	292,140 KB

1.3.2 任务 6-安装 MySQL

1、Windows10 子系统 Ubuntu 22.04 安装 MySQL

在 Windows 子系统 Ubuntu 22.04 上安装 MySQL，你可以使用以下步骤：

- (1) 打开 Windows 命令提示符或 PowerShell。
- (2) 输入以下命令以开启 Ubuntu 子系统的 MySQL 功能：

```
wsl --install -d Ubuntu-22.04
```

- (3) 启动 Ubuntu 子系统，并且更新包列表：

```
sudo apt update
```

- (4) 安装 MySQL 服务器：

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

- (5) 安装完成后，运行安全安装脚本来设置密码和调整安全选项：

```
sudo mysql_secure_installation
```

- (6) 启动 MySQL 服务：

```
sudo service mysql start
```

(7) 登录到 MySQL 以确保一切正常：

```
mysql -u root -p
```

当系统提示输入密码时，输入你在 `mysql_secure_installation` 过程中设置的密码。

以上步骤会在 Ubuntu 22.04 子系统上安装 MySQL 服务器，并确保服务正在运行。

远程连接问题解决：

```
use mysql;
```

```
select host,user,plugin from user;
```

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select host,user,plugin from user;
+-----+-----+-----+
| host      | user           | plugin           |
+-----+-----+-----+
| localhost | debian-sys-maint | caching_sha2_password |
| localhost | mysql.infoschema | caching_sha2_password |
| localhost | mysql.session    | caching_sha2_password |
| localhost | mysql.sys        | caching_sha2_password |
| localhost | root            | auth_socket       |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

得改，不然使用 `root` 账户无法进行本地和远程登录

```
update user set host = '%' where user = 'root';
```

```
flush privileges;
```

```
alter user 'root'@'%' identified with mysql_native_password by 'mysqlpassword';
```

```
flush privileges;
```

```
mysql> update user set host='%' where user = 'root';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> alter user 'root'@'%' identified with mysql_native_password by 'root';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

改完之后的配置

```
select host,user,plugin from user;
```

```
mysql> select host,user,plugin from user;
+-----+-----+-----+
| host      | user          | plugin                |
+-----+-----+-----+
| %         | root          | mysql_native_password |
| localhost | debian-sys-maint | caching_sha2_password |
| localhost | mysql.infoschema | caching_sha2_password |
| localhost | mysql.session  | caching_sha2_password |
| localhost | mysql.sys      | caching_sha2_password |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> exit
Bye
```

重启 mysql 再用超级账户进入

```
sudo service mysql restart
```

```
sean@sean-linux-lap:/$ sudo service mysql restart
[sudo] sean 的密码:
sean@sean-linux-lap:/$ sudo mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

刚刚账户密码已经修改掉了，没有超级账户这一说了啦

老老实实用账户密码登录

```
mysql -u root -p
```

```
sean@sean-linux-lap:/$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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> █
```

想要在同一局域网下，通过 **navicat** 连接我 **ubuntu** 上的 **mysql**，需要开启 **3306** 端口远程访问

首先下载两个包

使用 **netstat** 命令需要先下载 **net-tools** 工具包

```
sudo apt-get install net-tools
```

```
sean@sean-linux-lap:/$ sudo apt-get install net-tools
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
下列【新】软件包将被安装：
  net-tools
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 94 个软件包未被升级。
需要下载 204 kB 的归档。
解压缩后会消耗 819 kB 的额外空间。
获取:1 http://cn.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
已下载 204 kB，耗时 0秒 (24.2 kB/s)
正在选中未选择的软件包 net-tools。
(正在读取数据库 ... 系统当前共安装有 206633 个文件和目录。)
准备解压 .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
正在解压 net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
正在设置 net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
正在处理用于 man-db (2.10.2-1) 的触发器 ...
```

使用 **vim** 命令需要先下载 **vim** 工具包

```
sudo apt-get install vim
```



```
sean@sean-linux-lap:/usr/share/mysql$ sudo apt-get install vim
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
将会同时安装下列软件：
    vim-runtime
建议安装：
    ctags vim-doc vim-scripts
下列【新】软件包将被安装：
    vim vim-runtime
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 94 个软件包未被升级。
需要下载 8,556 kB 的归档。
解压缩后会消耗 37.6 MB 的额外空间。
您希望继续执行吗？ [Y/n] Y
```

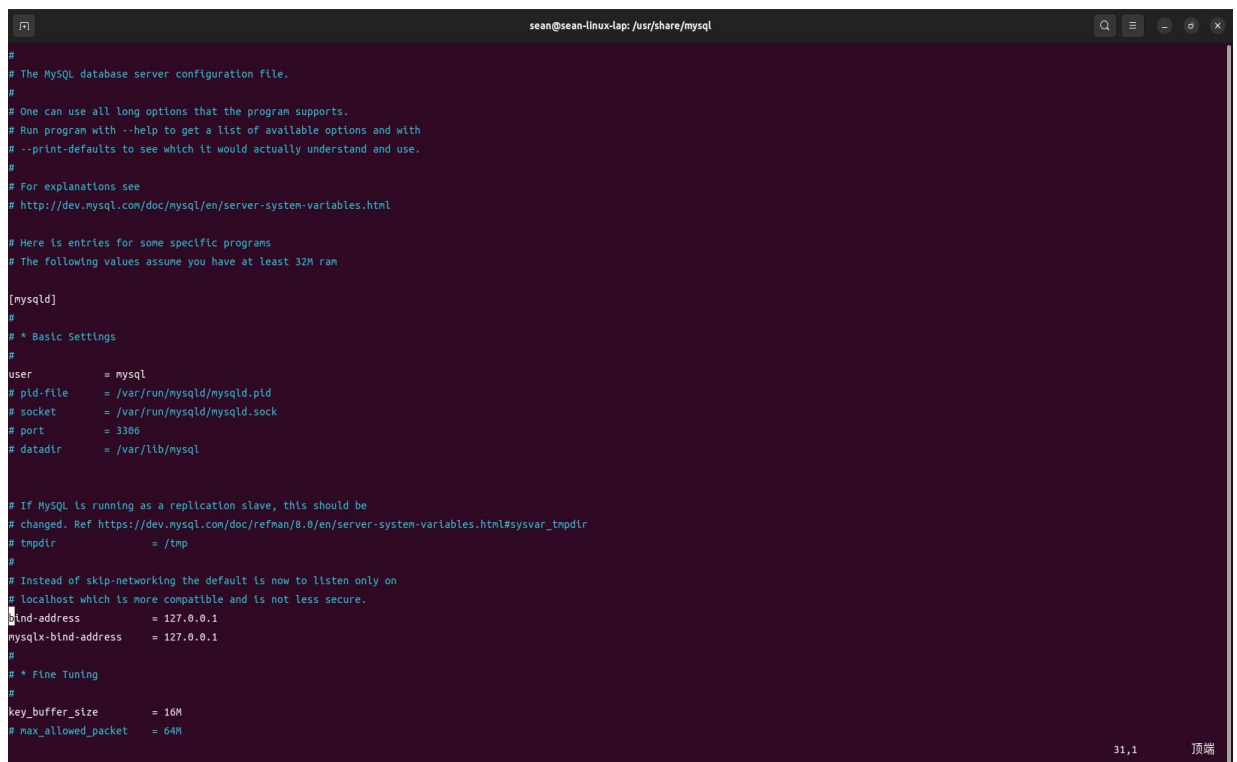
查看 3306 端口，可以看到现在 3306 端口绑定的是本地的 127.0.0.1 地址

```
netstat -an | grep 3306
```

修改 mysql 配置文件，由于这个文件是只读的，所以使用 `sudo` 命令

```
sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

将 `bind-address = 127.0.0.1` 注释掉



```
sean@sean-linux-lap: /usr/share/mysql
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysql/mysqld.pid
# socket             = /var/run/mysql/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir             = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
# bind-address        = 127.0.0.1
mysqlx-bind-address  = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 16M
# max_allowed_packet = 64M
```

重新启动并且查看端口信息

```
sudo service mysql restart
```

```
netstat -an | grep 3306
```

成功！



2、Windows10 安装 MySQL

双击安装包运行：



Developer Default 是开发者默认

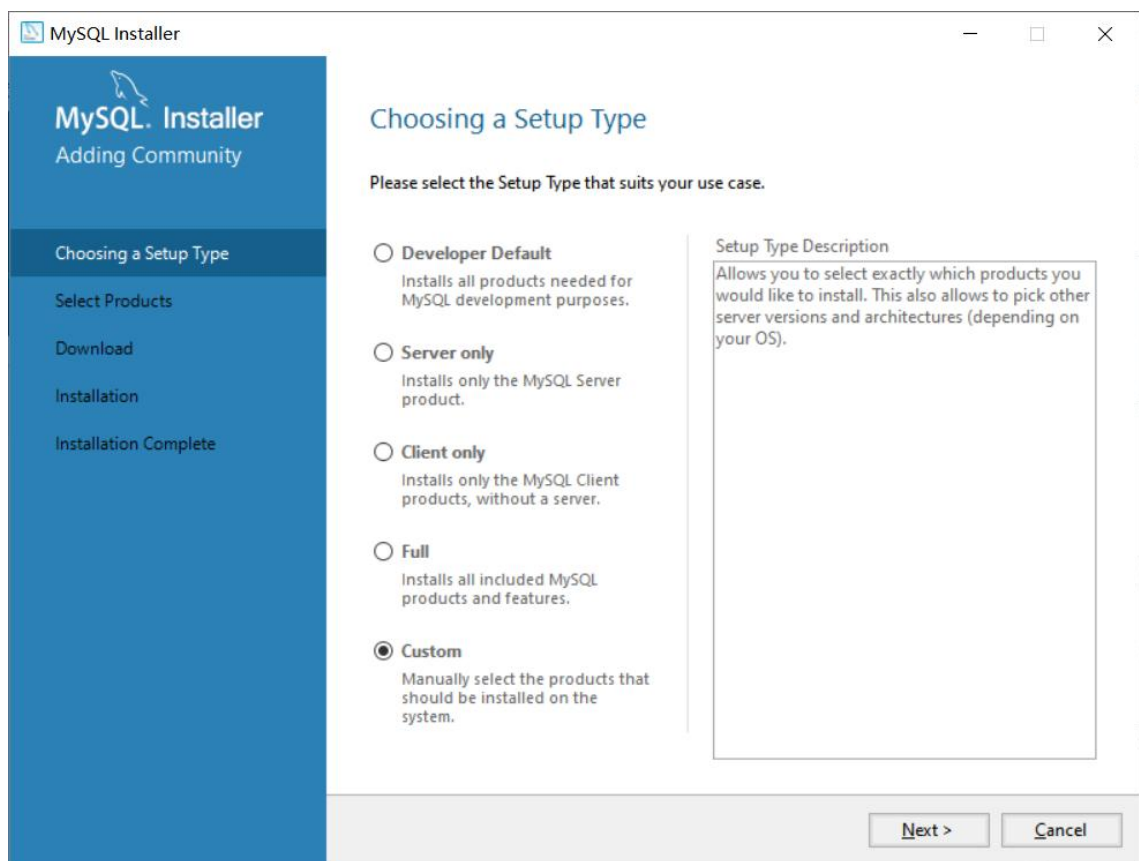
Server only 仅作为服务器安装

Client only 仅作为客户端安装

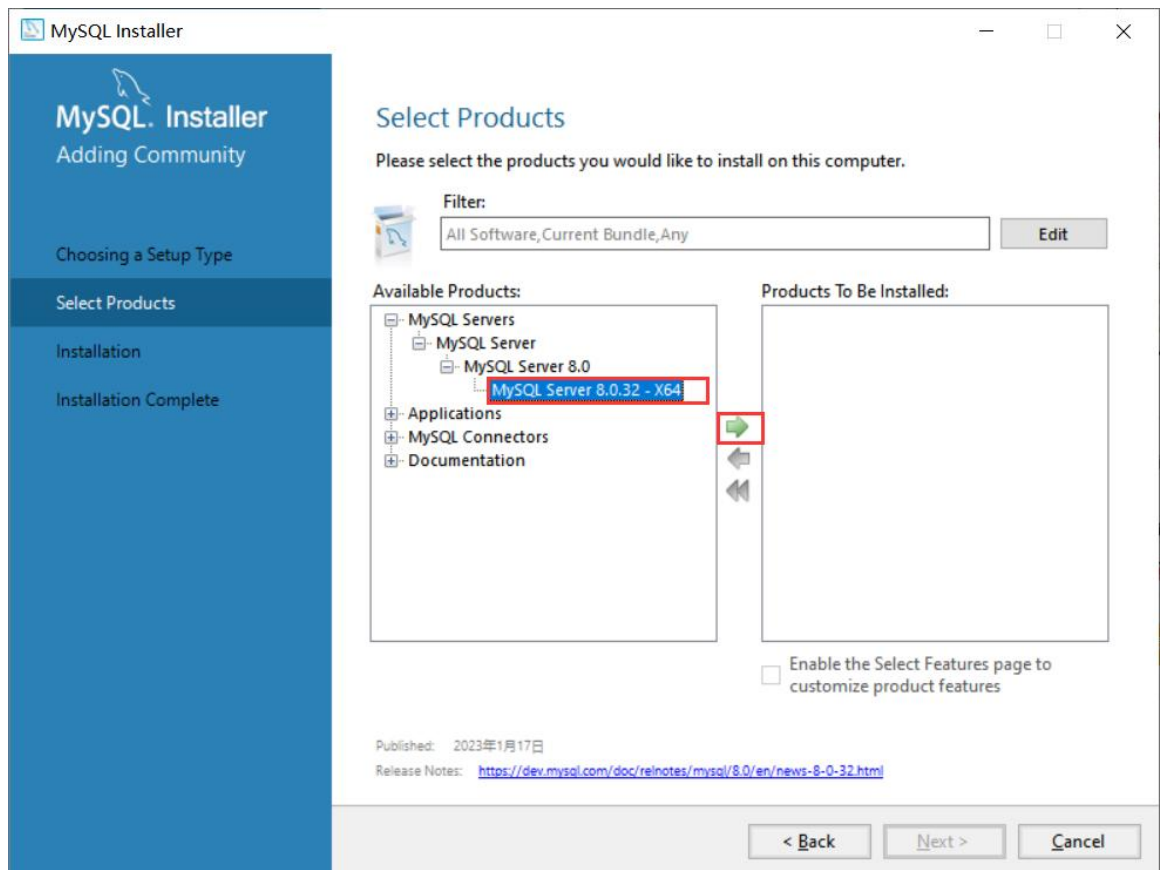
Full 是完整安装

Custom 是自定义安装

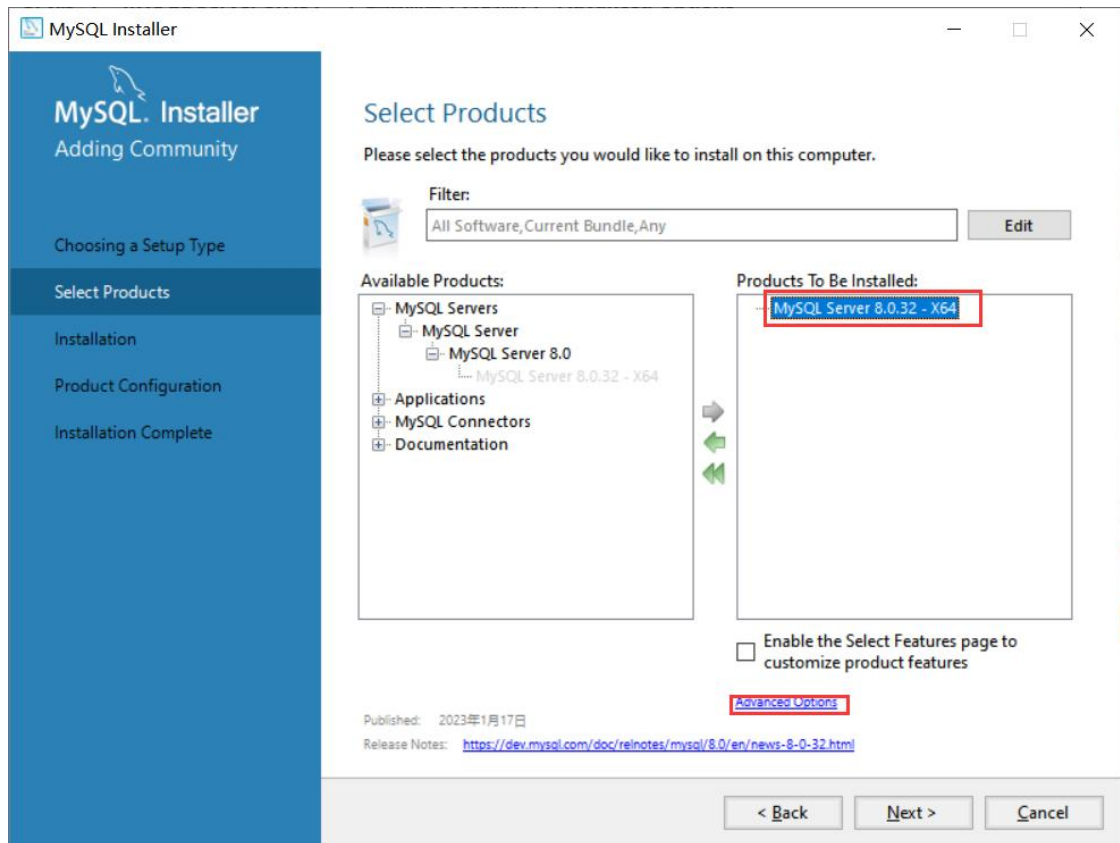
选自定义安装：



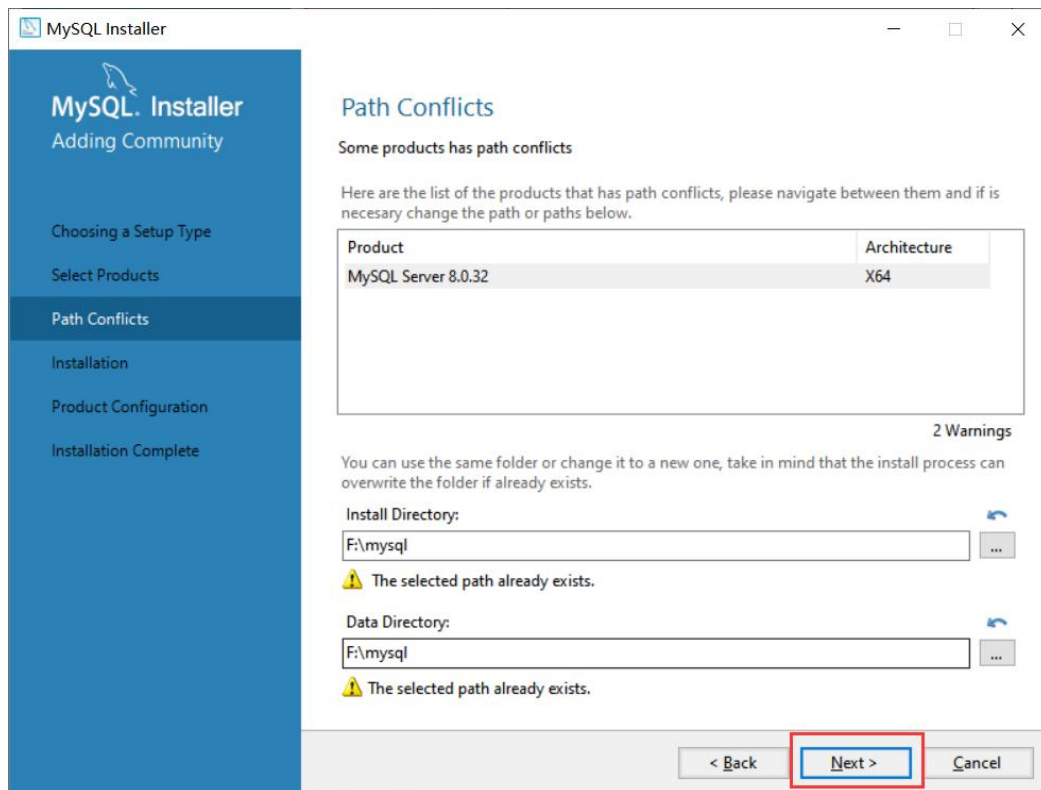
选 64 位安装程序:

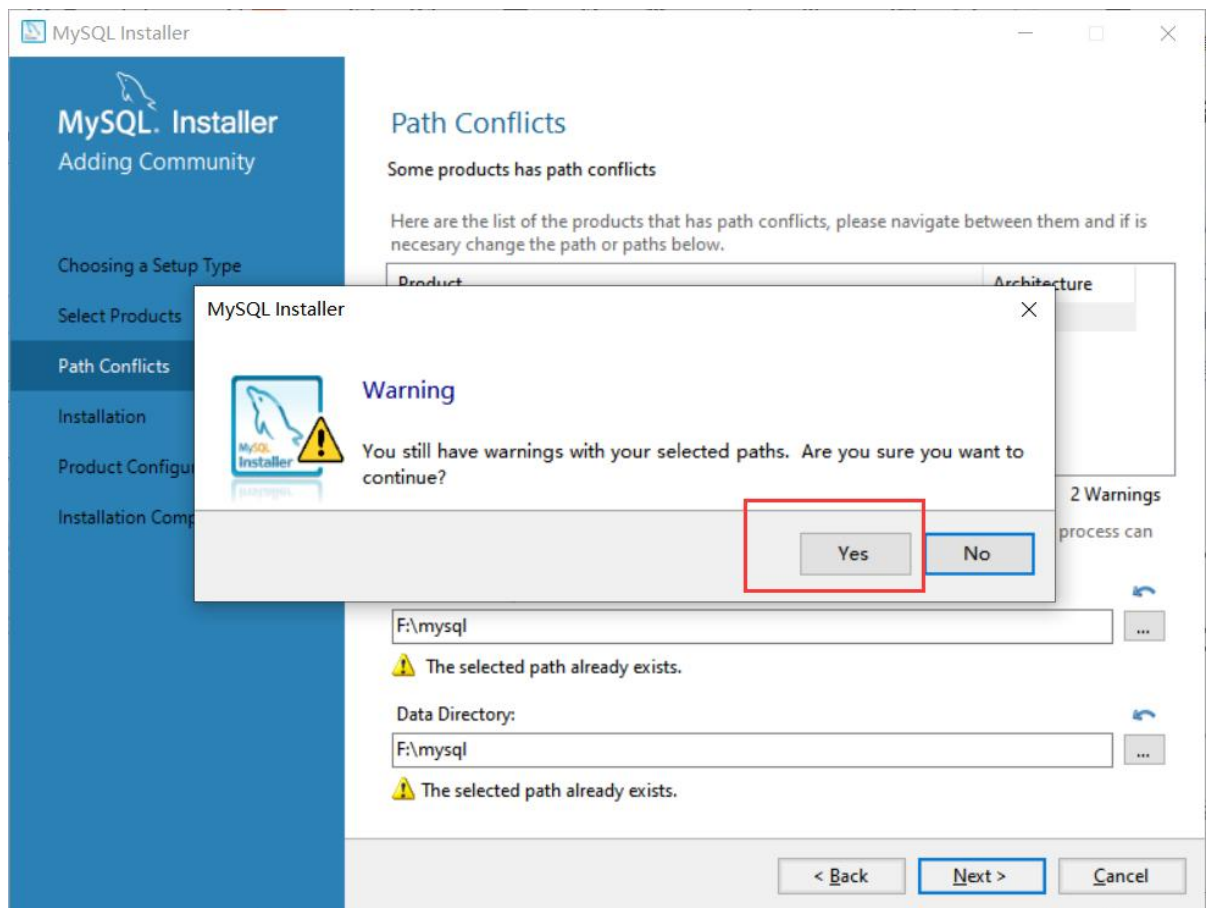


继续配置，这里是选择安装位置，选择合适的即可：

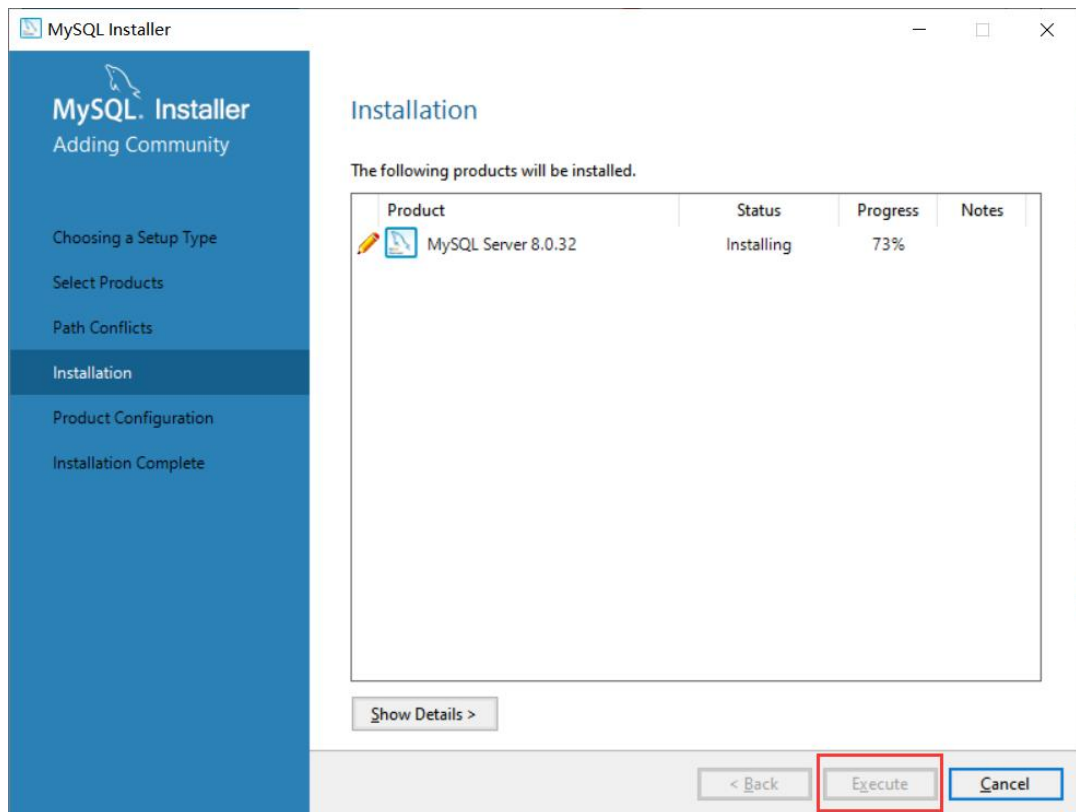


然后点 ok，然后 next:

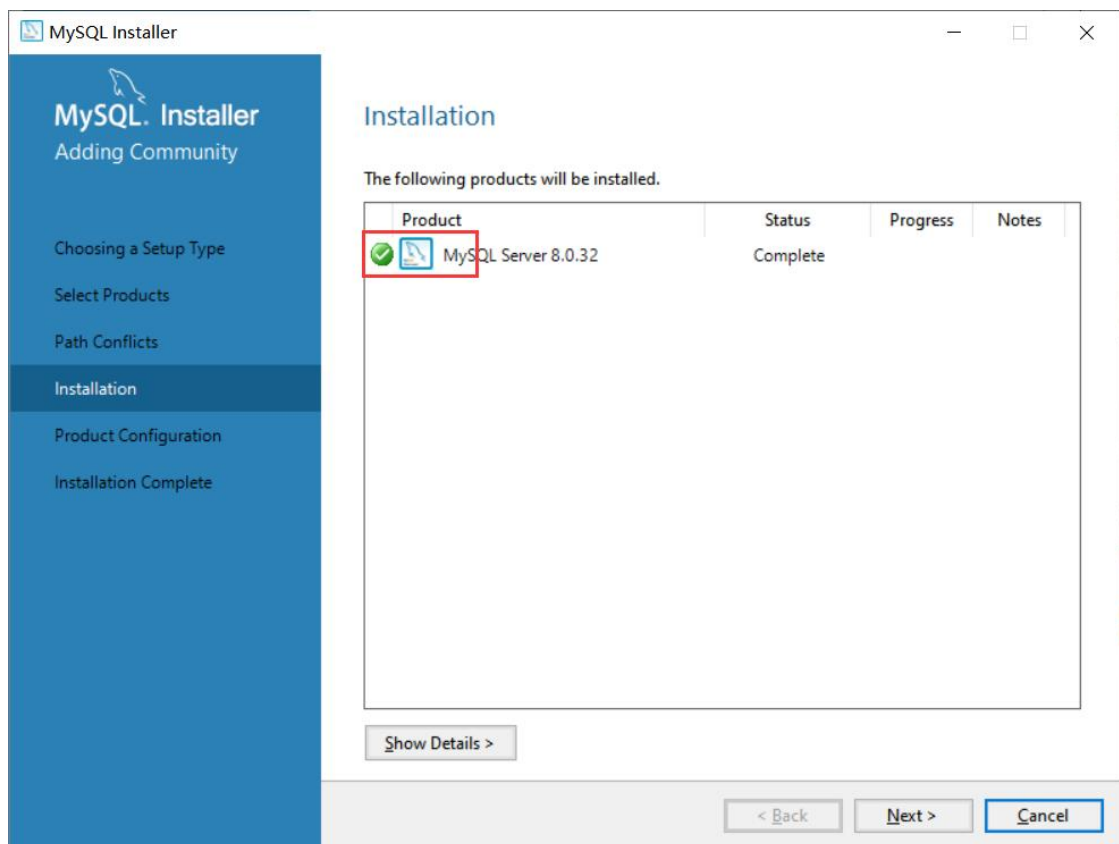




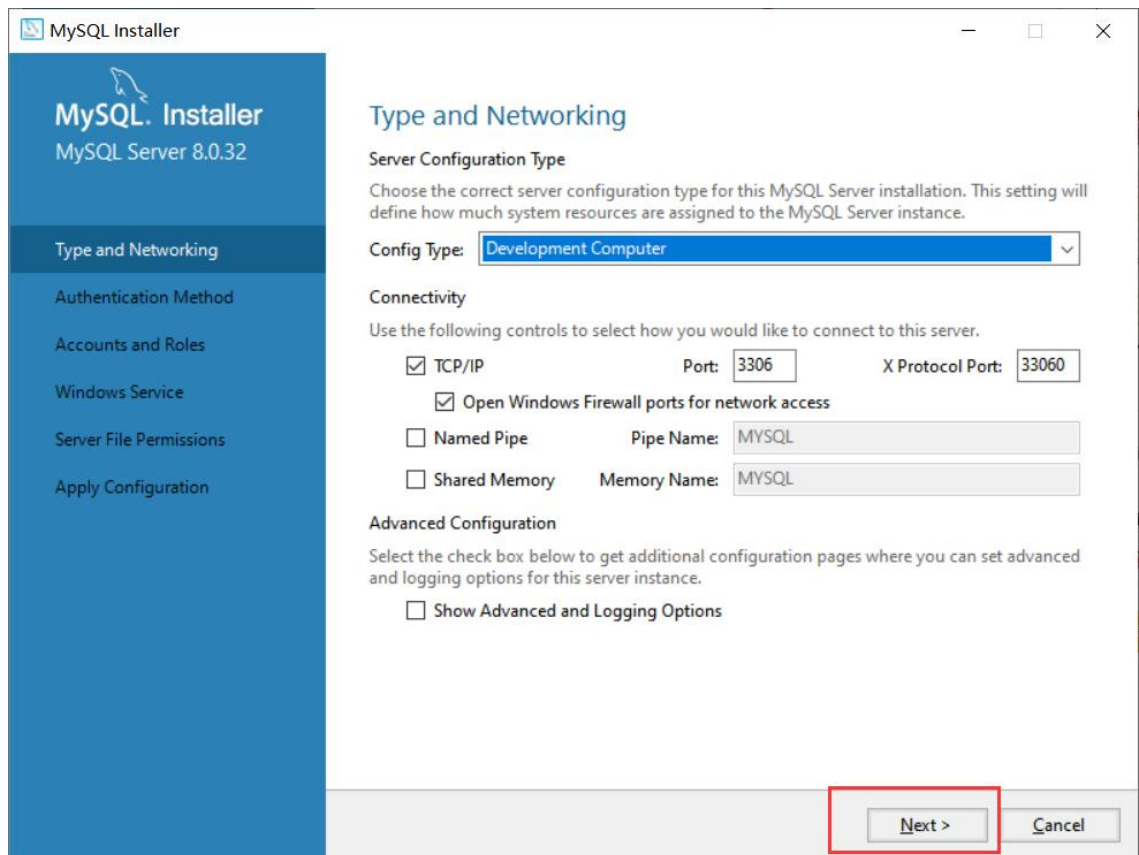
点击这个就会开始安装：



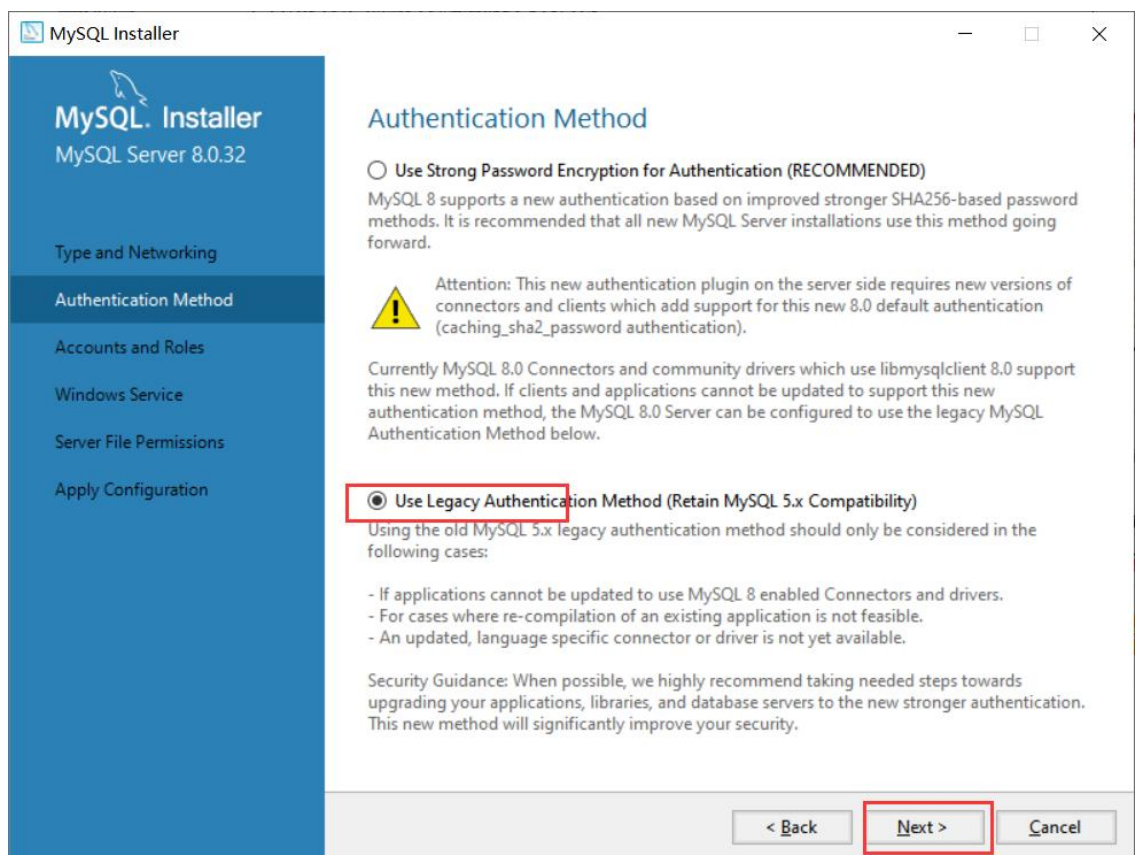
这里打勾说明装好了：



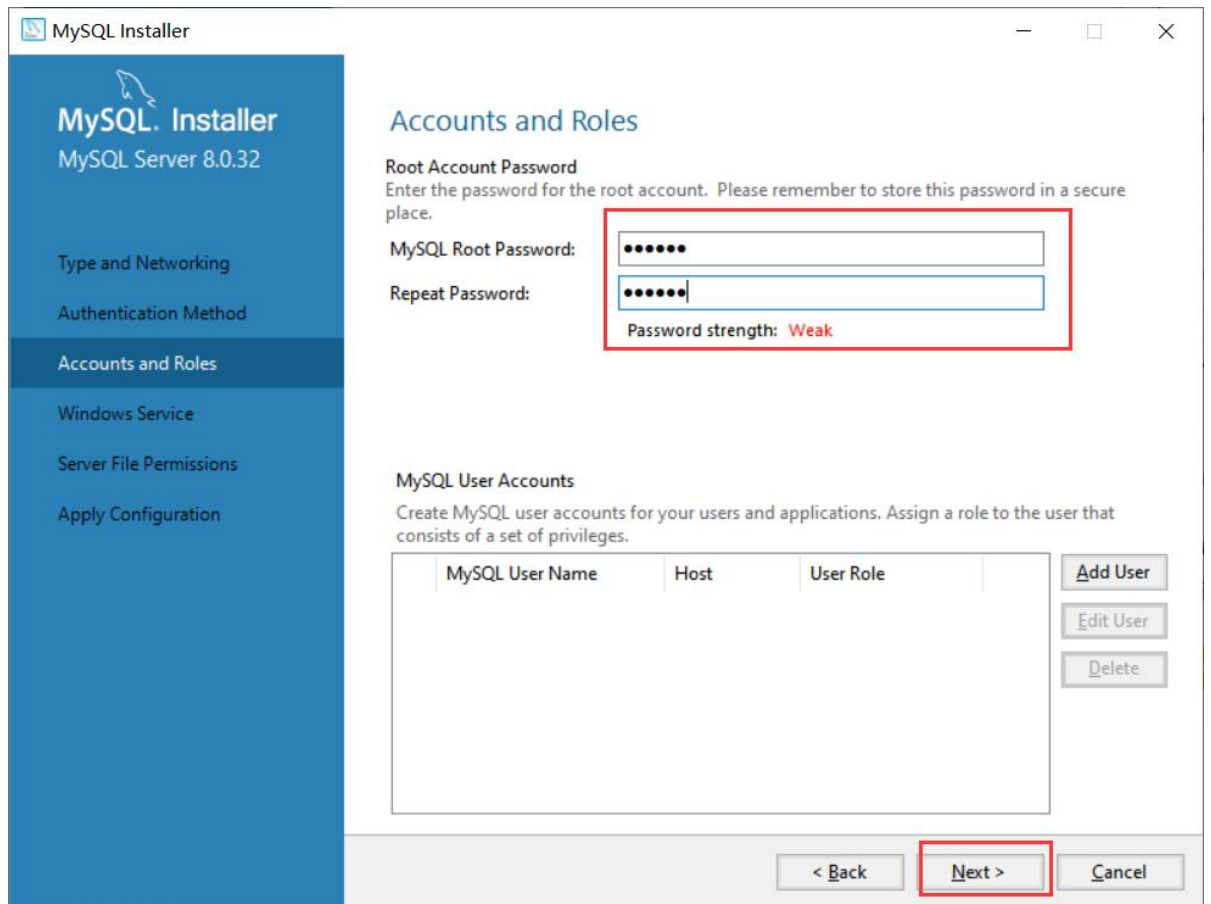
一路 next 到这个界面，我选择的是个人学习安装，其他无需修改：



这里是设置密码的方式，使用旧规则：



输入密码(root)和确认密码(root):



The screenshot shows the 'Accounts and Roles' step of the MySQL Installer for MySQL Server 8.0.32. The left sidebar lists the installation steps: Type and Networking, Authentication Method, Accounts and Roles (selected), Windows Service, Server File Permissions, and Apply Configuration. The main area is titled 'Accounts and Roles' and contains two sections. The first section, 'Root Account Password', prompts the user to enter a password for the root account and repeat it. The password fields are masked with dots. A red box highlights these fields and the 'Password strength: Weak' indicator below them. The second section, 'MySQL User Accounts', instructs the user to create MySQL user accounts and assign roles. It features a table with columns for 'MySQL User Name', 'Host', and 'User Role'. To the right of the table are buttons for 'Add User', 'Edit User', and 'Delete'. At the bottom of the window, there are three buttons: '< Back', 'Next >' (highlighted with a red box), and 'Cancel'.

MySQL Installer

MySQL Server 8.0.32

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Server File Permissions

Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password strength: **Weak**

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL User Name	Host	User Role
-----------------	------	-----------

Add User

Edit User

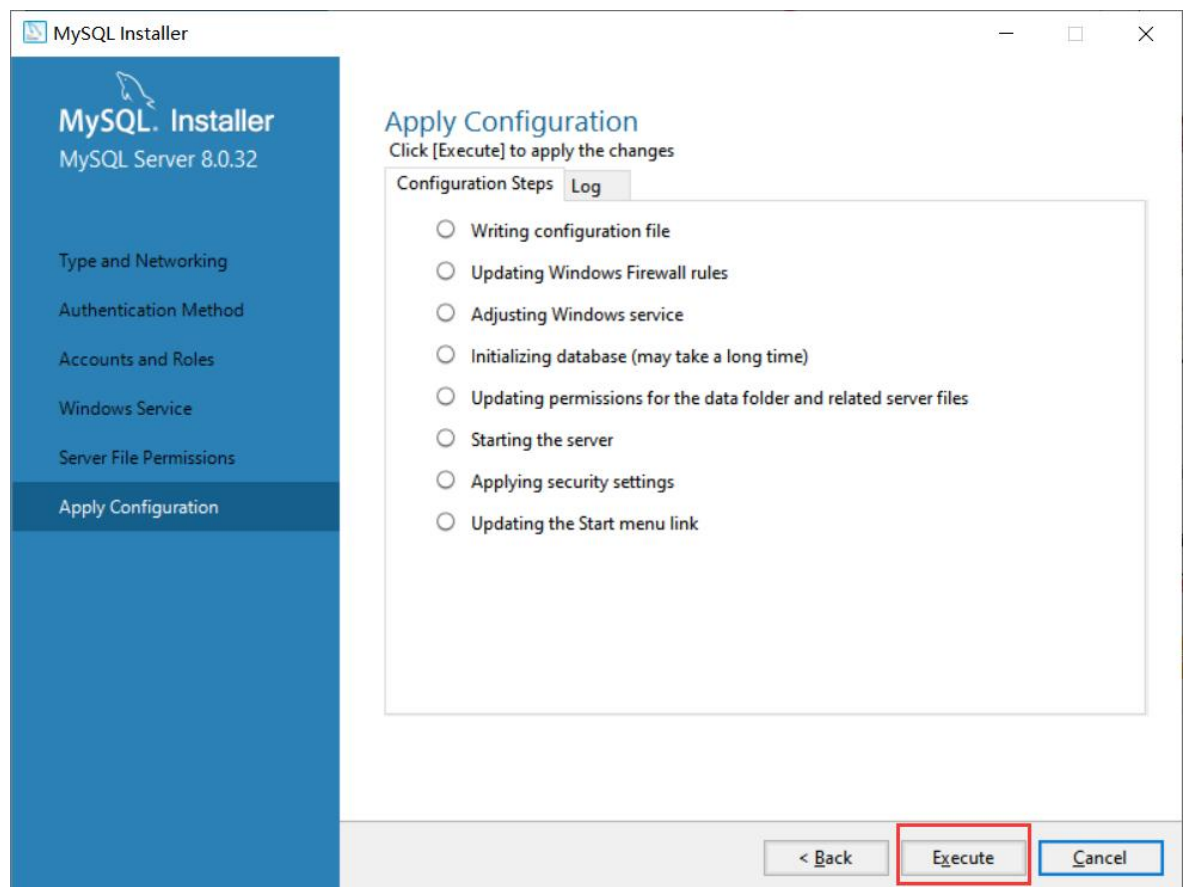
Delete

< Back

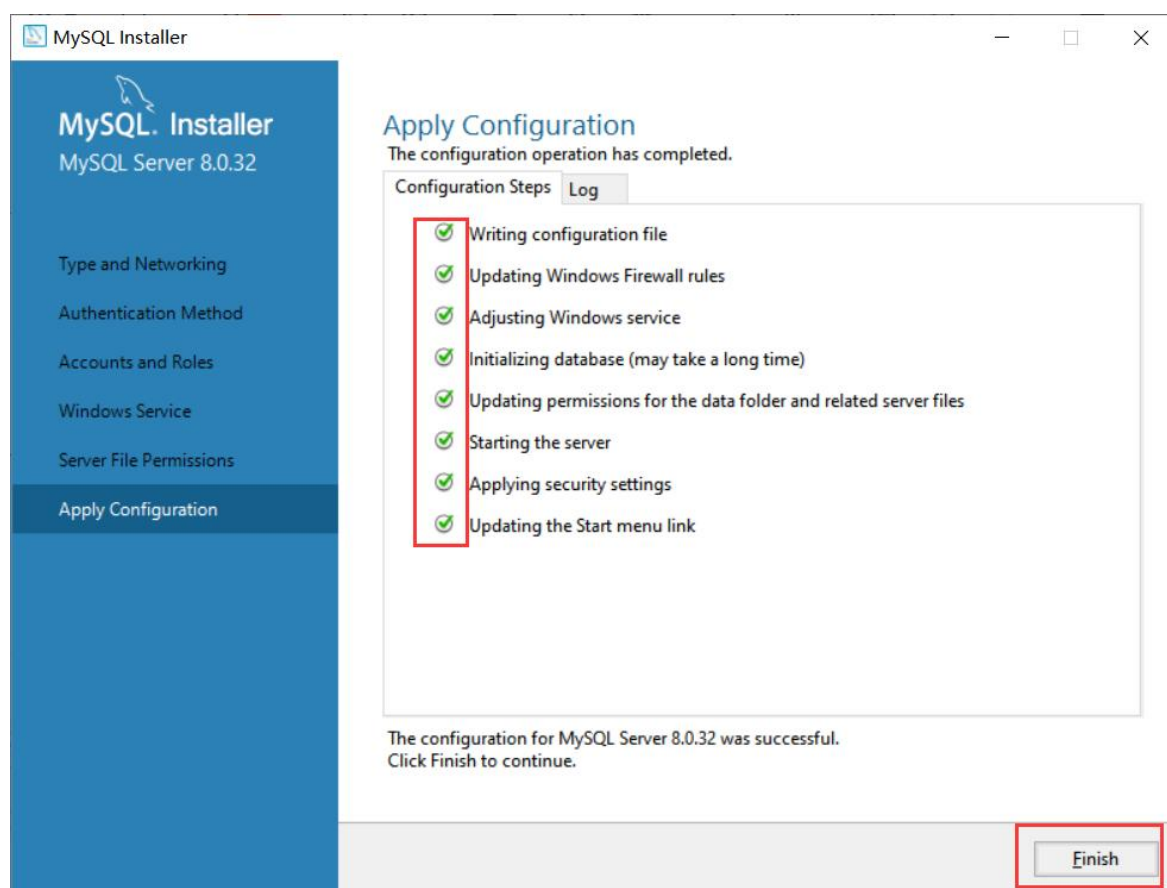
Next >

Cancel

然后又是一路 next，来到这个界面：

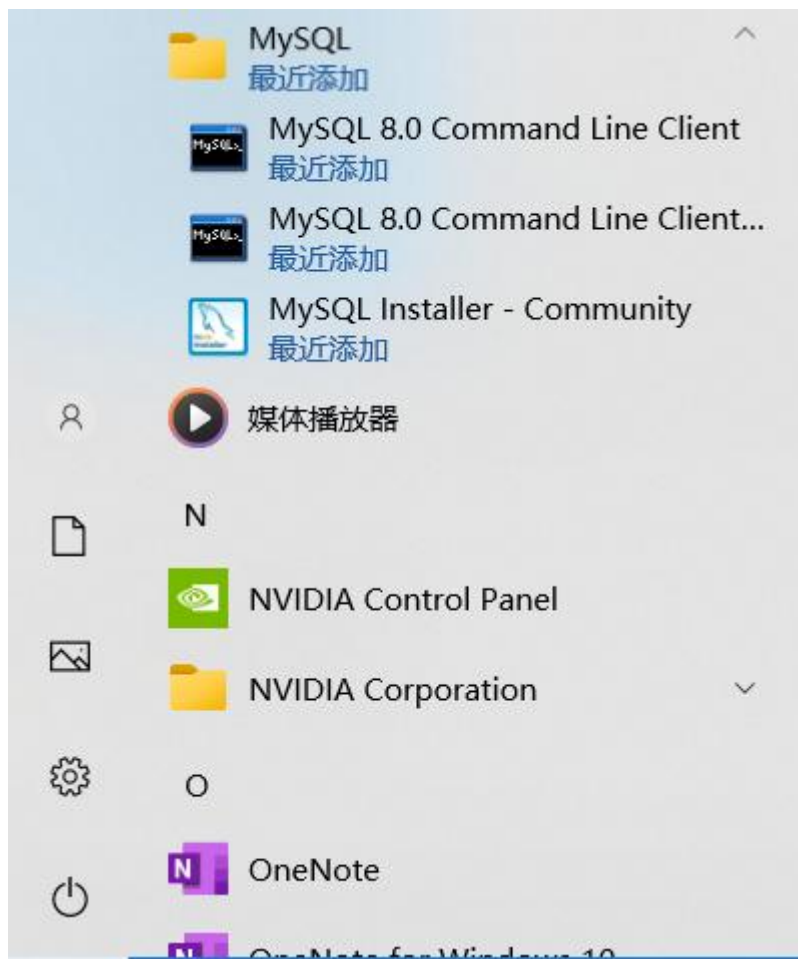


一直点到最后，然后点击 **finish** 就安装完成啦！



验证安装：

去开始菜单找这一项：



打开这个：



输入你设置的密码，就安装成功了

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

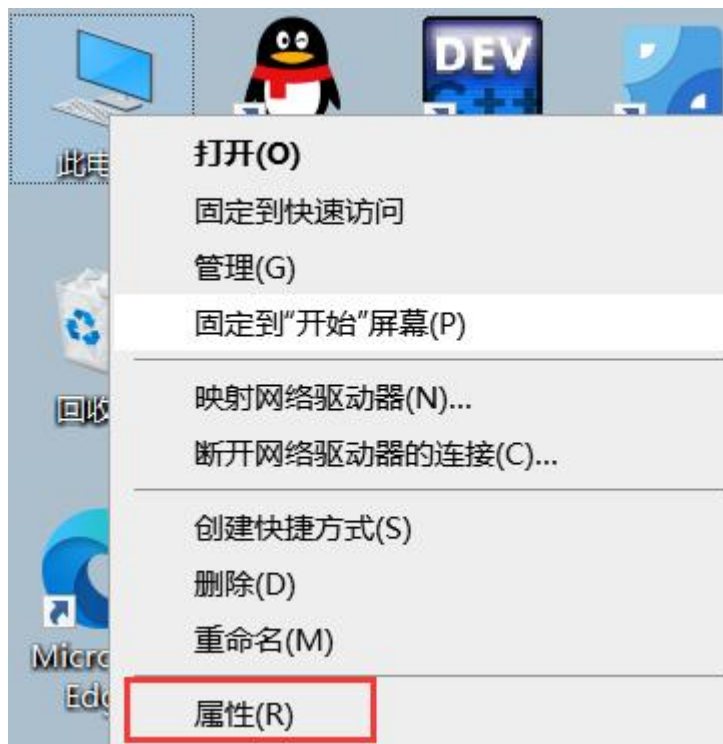
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> _
```

环境变量配置：

此电脑，右键，点击属性



高级系统设置：

相关设置

[BitLocker 设置](#)

[设备管理器](#)

[远程桌面](#)

[系统保护](#)

[高级系统设置](#)

[重命名这台电脑](#)

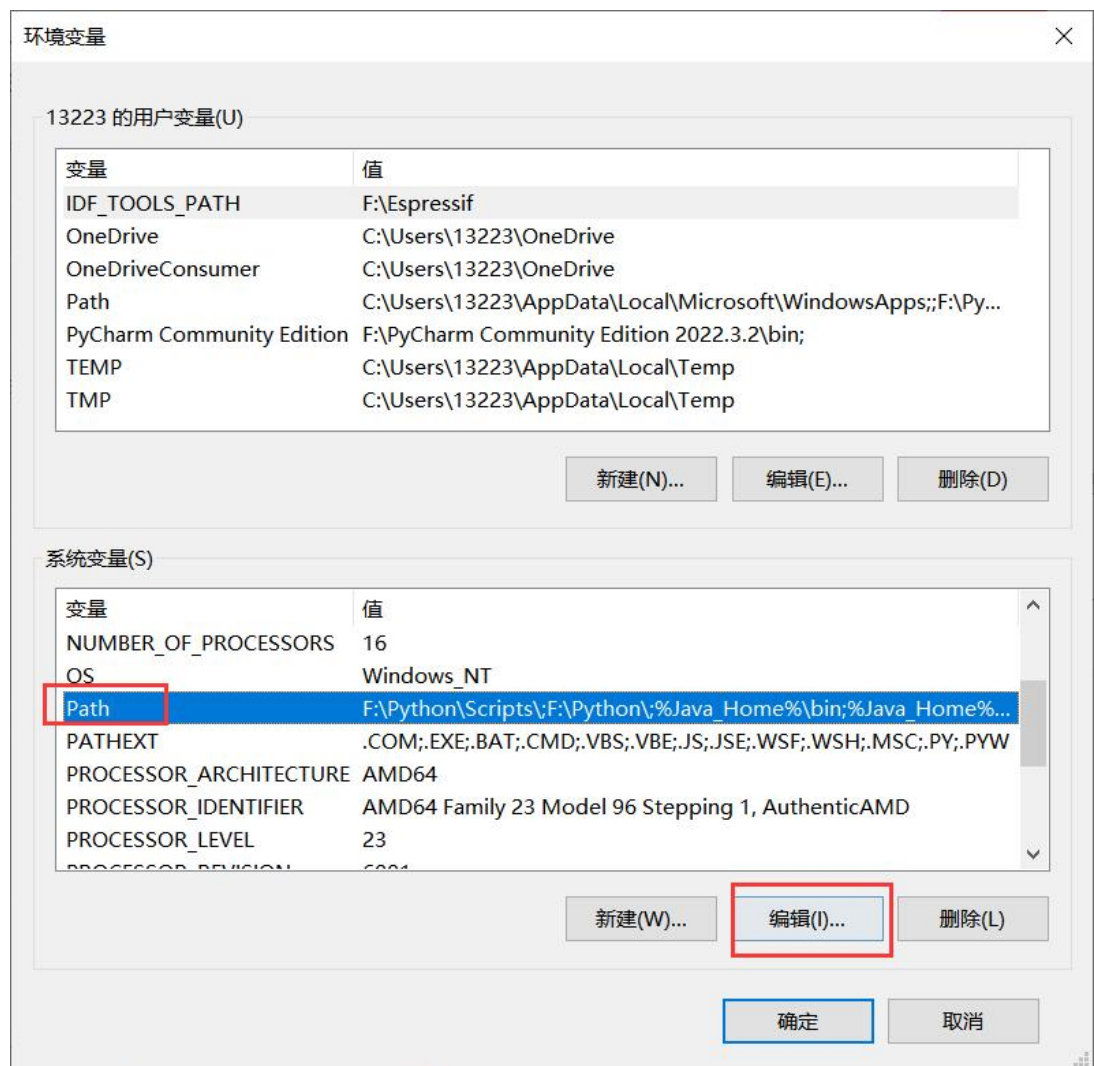
 [获取帮助](#)

 [提供反馈](#)

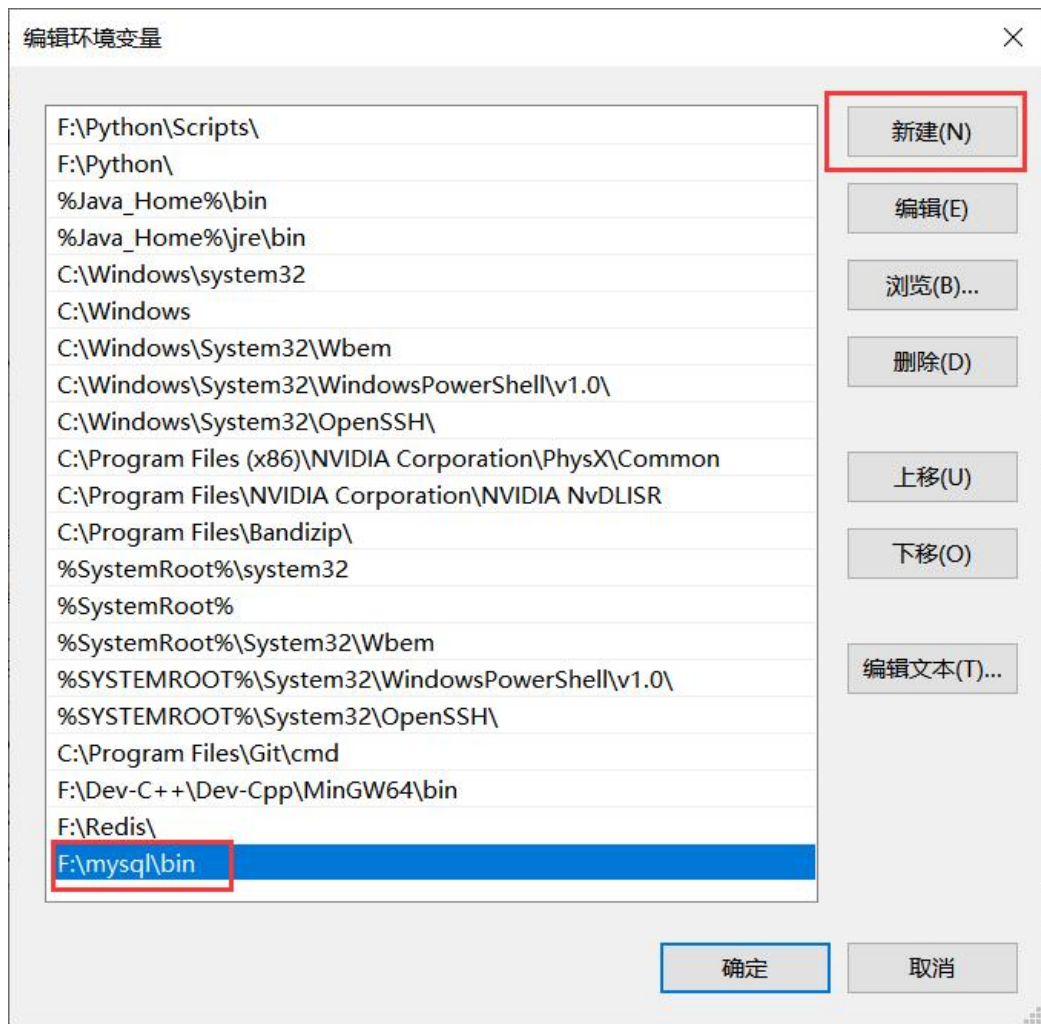
环境变量：



点中 PATH 然后编辑:



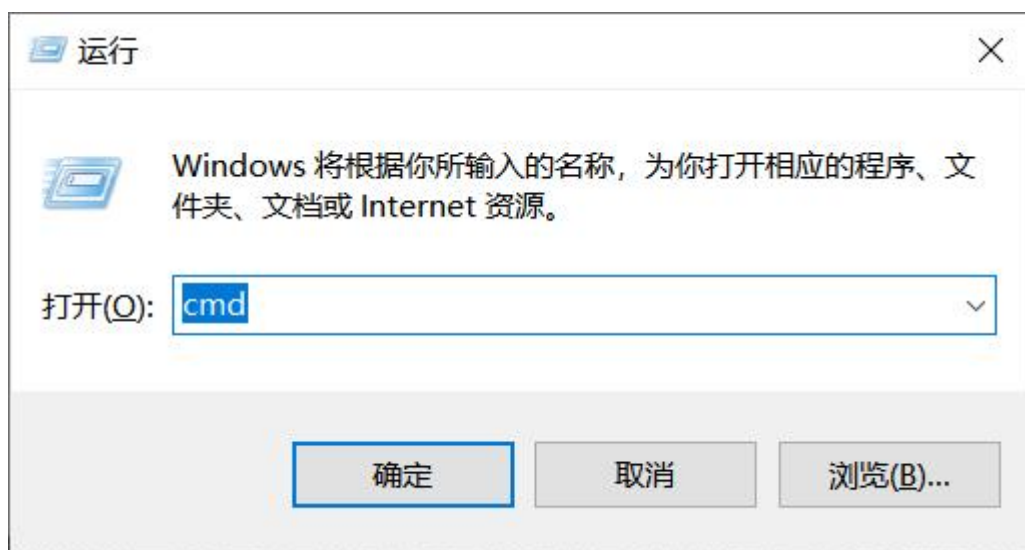
新建这个，填你的安装路径：



然后一直确定，最后直到所有窗口都没。

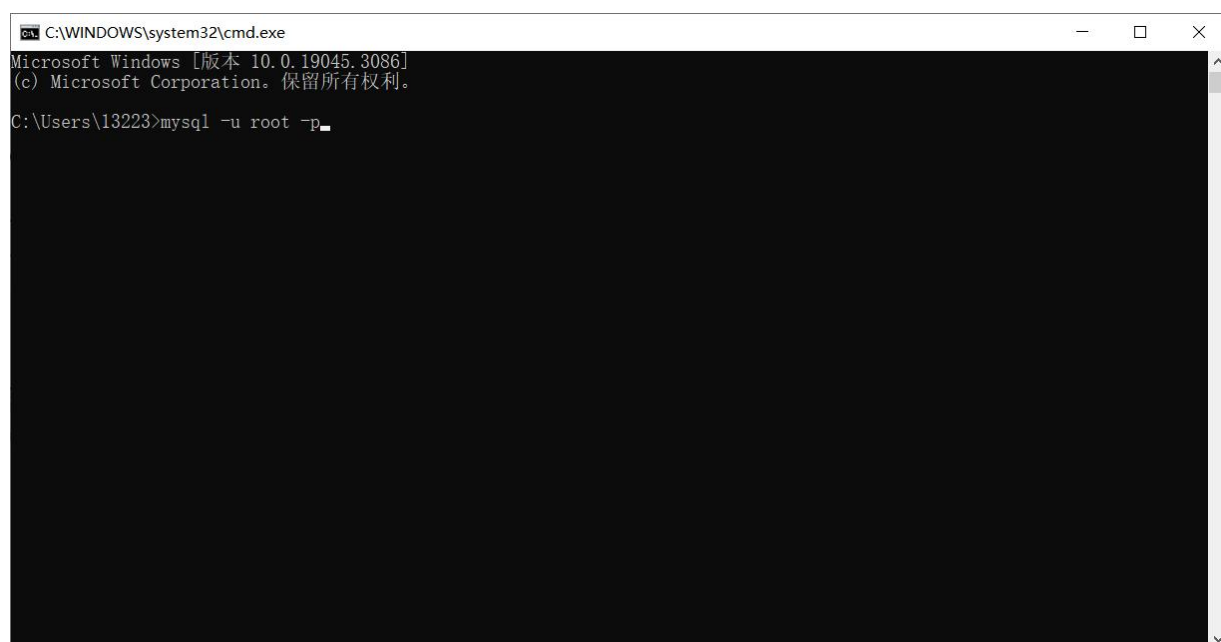
现在验证环境变量配置情况：

win + R:



输入这个：

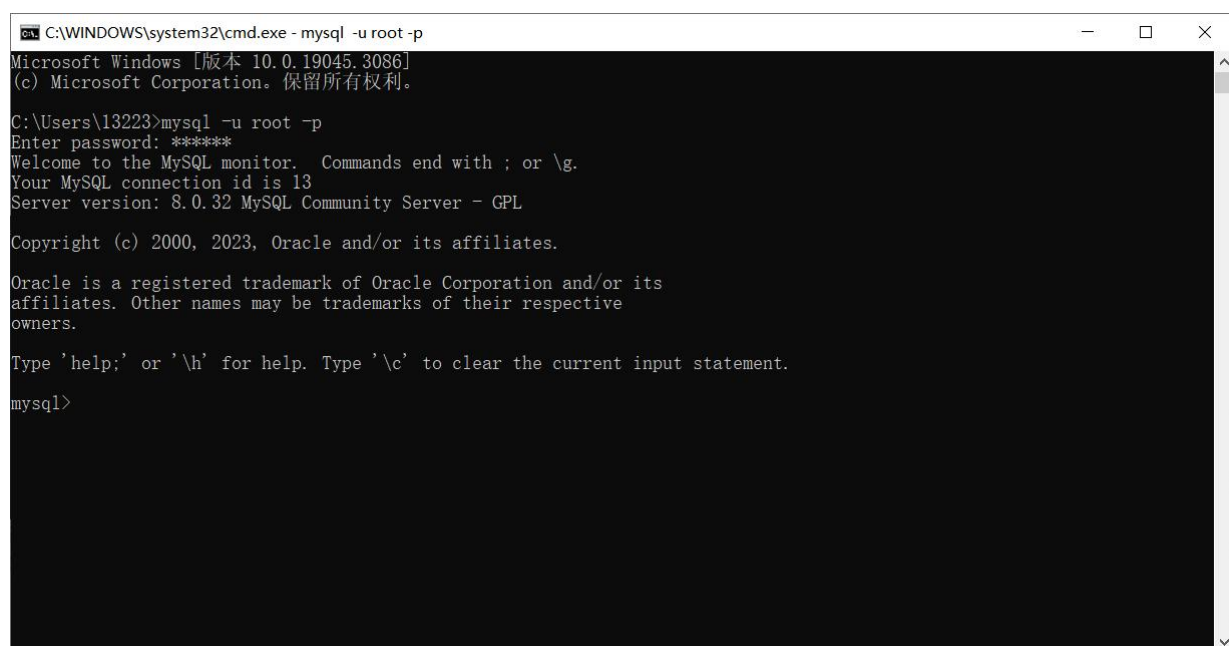
```
mysql -u root -p
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19045.3086]
(c) Microsoft Corporation。保留所有权利。

C:\Users\13223>mysql -u root -p
```

输入密码后如果是这个样子那么配置完成：



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p
Microsoft Windows [版本 10.0.19045.3086]
(c) Microsoft Corporation。保留所有权利。

C:\Users\13223>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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>
```

1.3.3 任务 7-安装 pymysql

通过下面这个命令进行安装：

```
pip install pymysql
```

pymysql 库: Python3 链接 mysql

备注:

ps:MySQLdb 只适用于 python2.x

python3 不支持 MySQLdb,取而代之的是 pymysql

运行会报: ImportError: No module named 'MySQLdb'

1.4 Python 下使用 MySQL

1.4.1 任务 8-python 连接 mysql

```
import pymysql as pmq

# 创建数据连接
con = pmq.connect(host='172.23.218.93', user='plate', password='@135qetQET',
database='plate')

# 操作游标
cur = con.cursor()

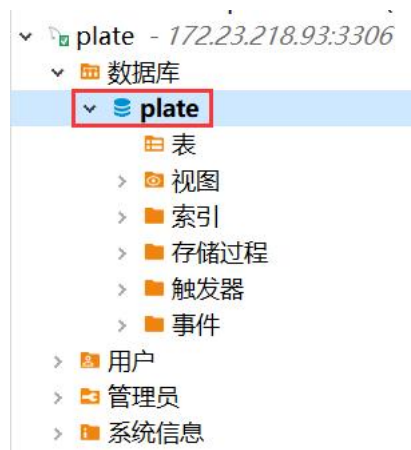
print(con)

print(cur)

cur.close()

con.close()
```

172.23.218.93 是数据库 ip, palte 是数据库用户名, @135qetQET 是数据库密码, plate 是数据库名。



图上的数据库 **plate** 已经建立好（建好之后，才能用上面代码去连接），建好之后，当前是没有表的，现在开始用 **Python** 进行建表，插入、查询，修改，删除等操作。

1.4.2 任务 9-建表 **t_parking**

在存储之前，先通过 **python** 创建表 **t_parking**，字段有 7 个，分别为(**id**, **park_id**, **license_plate**, **input_date**, **fee**, **output_date**, **status**)。

建表 **t_parking** 的 SQL 如下：

```
-- plate.t_parking definition

CREATE TABLE `t_parking` (
  `id` int NOT NULL AUTO_INCREMENT COMMENT '编号',
  `park_id` int NOT NULL COMMENT '停车场编号',
  `license_plate` varchar(10) NOT NULL COMMENT '车牌号',
  `input_date` datetime NOT NULL COMMENT '入场时间',
  `fee` float DEFAULT '0' COMMENT '停车费',
  `output_date` datetime DEFAULT NULL COMMENT '出场时间',
  `status` tinyint(1) DEFAULT '1' COMMENT '状态:1-入场, 0-出场',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

整体建表的 **Python** 代码如下：

```
import pymysql as pmq

#connect(ip,user,password,dbname)

con = pmq.connect(host='172.23.218.93',user='plate',password='@135qetQET',database='plate')

#操作游标

cur = con.cursor()

# 创建 t_parking 表

parking_sql= ""
```

```

CREATE TABLE t_parking(
    id INT AUTO_INCREMENT PRIMARY KEY NOT NULL COMMENT '编号',
    park_id INT NOT NULL COMMENT '停车场编号',
    license_plate VARCHAR(10) NOT NULL COMMENT '车牌号',
    input_date DATETIME NOT NULL COMMENT '入场时间',
    fee FLOAT DEFAULT 0 COMMENT '停车费',
    output_date DATETIME COMMENT '出场时间',
    status BOOLEAN DEFAULT TRUE COMMENT '状态:1-入场, 0-出场'
);

'''
# 执行 sql 语句
cur.execute(parking_sql)
# 提交到数据库执行
con.commit()

```

1.4.3 任务 10-t_parking 插入数据

利用 python 已经建好表，接着可以将数据插入到数据库表中，代码如下：

```

import pymysql as pmq

con = pmq.connect(host='172.23.218.93', user='plate', password='@135qetQET',
database='plate')

# 操作游标
cur = con.cursor()

sql_data = (0, '鲁 V3L69', '2024-05-31 09:08:09', 0, None, 1)

# t_parking 表插入数据
sql_insert = "INSERT INTO plate.t_parking(park_id, license_plate, input_date, fee,
output_date, status) VALUES(%s, %s, %s, %s, %s, %s);"

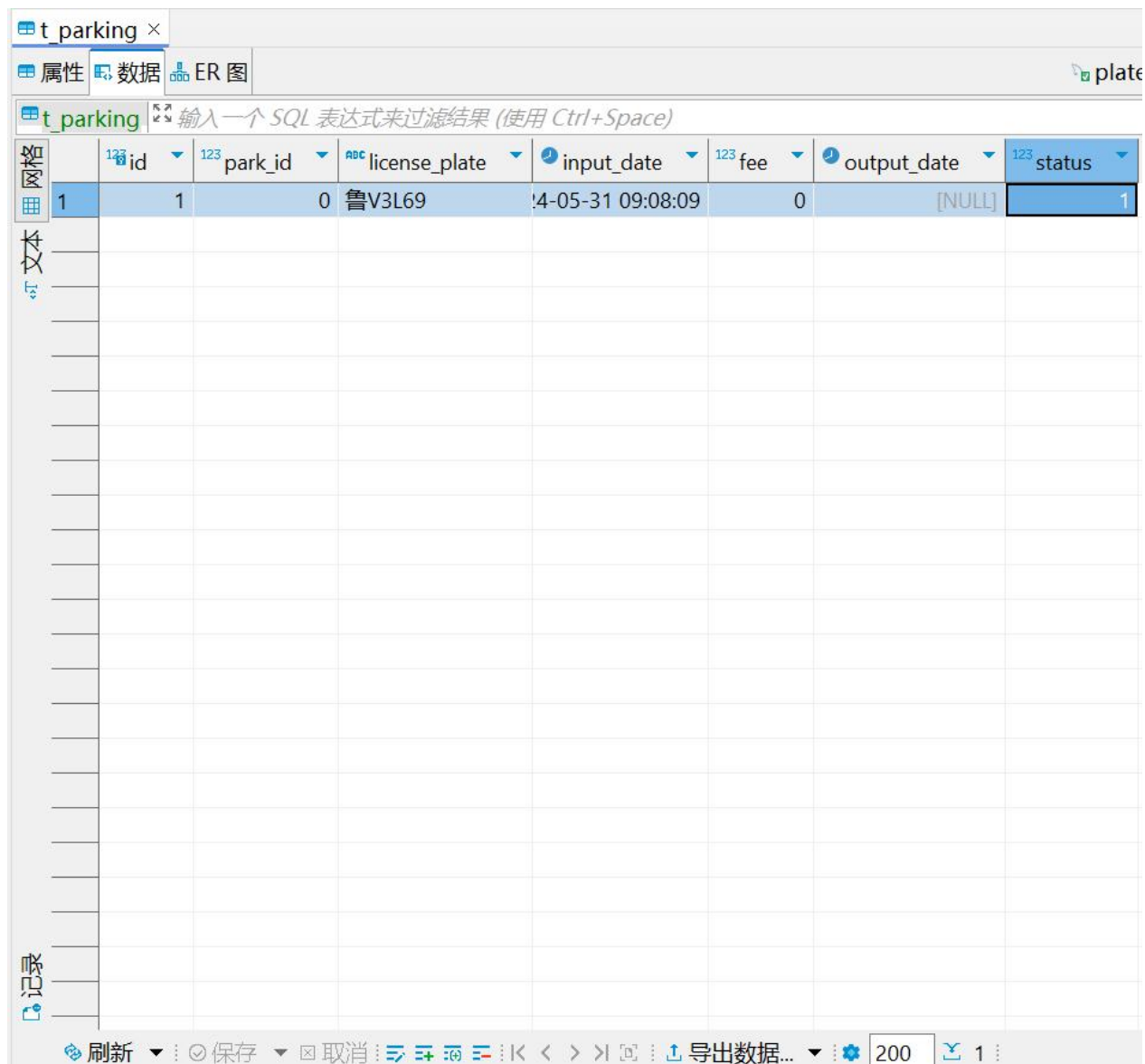
# 执行 sql 语句
cur.execute(sql_insert, sql_data)

```

```
# 提交到数据库执行
```

```
con.commit()
```

执行结果如下：



The screenshot shows a database management interface with a table named 't_parking'. The table has the following columns: id, park_id, license_plate, input_date, fee, output_date, and status. The first row contains the values: 1, 0, 鲁V3L69, 4-05-31 09:08:09, 0, [NULL], and 1. The interface includes tabs for '属性' (Properties), '数据' (Data), and 'ER图' (ER Diagram). The '数据' tab is active, showing the table data. The status bar at the bottom indicates 200 records and 1 page.

	id	park_id	license_plate	input_date	fee	output_date	status
1	1	0	鲁V3L69	4-05-31 09:08:09	0	[NULL]	1

1.4.4 任务 11-查询表 t_parking 数据

查询表中所有数据，代码如下：

```
import pymysql as pmq

con = pmq.connect(host='172.23.218.93', user='plate', password='@135qetQET',
database='plate')
```

```
# 操作游标
cur = con.cursor()

sql_data = (0)

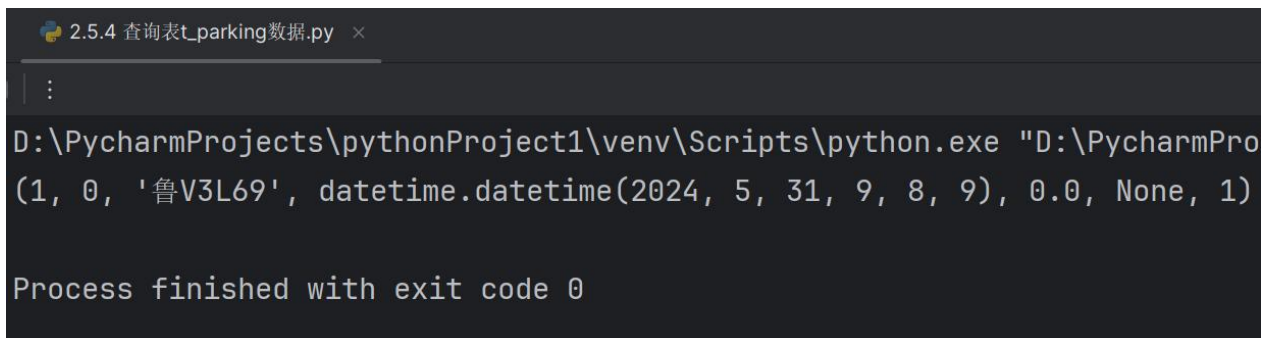
# t_parking 表插入数据
sql_select = """SELECT id, park_id, license_plate, input_date, fee, output_date, status
                FROM plate.t_parking
                WHERE park_id=%s;"""

# 执行 sql 语句
cur.execute(sql_select, sql_data)

# 获取结果并逐行打印数据
results = cur.fetchall()
for row in results:
    print(row)

# 提交到数据库执行
con.commit()
```

执行结果如下：



```
2.5.4 查询表t_parking数据.py x
:
D:\PycharmProjects\pythonProject1\venv\Scripts\python.exe "D:\PycharmPro
(1, 0, '鲁V3L69', datetime.datetime(2024, 5, 31, 9, 8, 9), 0.0, None, 1)
Process finished with exit code 0
```

1.4.5 任务 12-修改表 t_parking 数据

更新表 t_parking 数据，主要是停车费、出场时间及状态，代码如下：

```
import pymysql as pmq
```

```
con = pmq.connect(host='172.23.218.93', user='plate', password='@135qetQET',
database='plate')

# 操作游标

cur = con.cursor()


sql_data = (3, '2024-05-31 09:32:09', 0, '鲁 V3L69')

# t_parking 表插入数据

sql_update = """UPDATE plate.t_parking
                SET fee=%s, output_date=%s, status=%s
                WHERE license_plate=%s;"""

# 执行 sql 语句

cur.execute(sql_update, sql_data)


# 提交到数据库执行

con.commit()
```

执行结果如下：

[illegible]

1.4.6 任务 13-删除表 t_parking 数据

根据车牌号，删除表中数据，代码如下：

```
import pymysql as pmq

con = pmq.connect(host='172.23.218.93', user='plate', password='@135qetQET',
database='plate')

# 操作游标

cur = con.cursor()

sql_data = (0)
```

```
# t_parking 表插入数据

sql_delete = "DELETE FROM plate.t_parking

                WHERE park_id=%s;"

# 执行 sql 语句

cur.execute(sql_delete, sql_data)

# 提交到数据库执行

con.commit()
```

执行结果，如下：

[illegible]

1.5 章节总结

1.6 课后作业

结合自己设计的表，练习在 Python 中 MySQL 的使用。