



中山大學  
SUN YAT-SEN UNIVERSITY

# 数据库实验报告

实 验 吴扬一、伍建 学 20337250 、 日 2023.1.5

人： 霖 号： 20337251 期：

院（系）： 计算机学院 专业（班级）： 网络空间安全

联 系 方

式：

实验题目： 实验8：数据库设计与应用开发大作业

小组分工：

吴扬一：设计系统、实现系统

伍建霖：分析系统、撰写报告

# 第一章 需求建模

## 1.1 系统功能性需求

本系统目标是通过建立一个系统使得管理员和经理可以更有效地管理数据库用户、内容与权限，项目经理、采购员和销售能够更方便地更新数据库内容，更方便地更新采购与销售地信息，从而提高交流与工作效率。

本系统主要面向五种用户角色：管理员、经理、项目经理、采购员和销售。针对这五种用户角色，系统的主要功能性需求有：

管理员：负责管理数据库的用户，即负责对数据库各级用户的增加和删除，同时也管理各个用户的权限。

经理：负责管理各个数据库用户、内容与权限

项目经理：负责管理 item 表，能够查询 item、order 和 supply 表的内容

采购员：负责管理 supply 和 supplier 表，能够查询 supplier、supply 和 item 表的内容

销售：负责管理 customer 和 order 表，能够查询 customer、order 和 item 表的内容

## 1.2 系统核心用例

本系统涉及到的用例比较多，本节主要介绍以下三个核心用例，它们分别是：更改用户权限、新增用户、新增数据库条目。

### 1.2.1 更改用户权限

用例：更改用户权限

范围：数据库系统

级别：用户目标

主要参与者：管理员、经理

涉众及其关注点：

1. 管理者：希望能够准确快速地授予变更职位的员工对应的权限。
2. 基层员工：希望能快速地拥有对应职位的数据库权限。

**前置条件：**员工的帐号必须已经添加到系统中。

**成功保证：**提示权限更改成功。

**主成功场景：**

1. 经理查询所有数据库内容。
2. 项目经理查询 item、order 和 supply 数据库内容。
3. 采购员查询 supplier、supply 和 item 数据库内容。
4. 销售查询 customer、order 和 item 数据库内容。

**特殊需求：**系统的设计保障系统的健壮性。

**发生频率：**不经常发生。

## 1.2.2 新增用户

**用例：**新增用户

**范围：**数据库系统

**级别：**用户目标

**主要参与者：**管理员、经理

**涉众及其关注点：**

3. 管理者：希望能够准确快速地为新入职的员工创建数据库用户。
4. 基层员工：希望能快速地拥有对应职位的数据库用户。

**前置条件：**数据库正常运行。

**成功保证：**提示用户创建成功。

**主成功场景：**

5. 经理查询所有数据库内容。
6. 项目经理查询 item、order 和 supply 数据库内容。
7. 采购员查询 supplier、supply 和 item 数据库内容。
8. 销售查询 customer、order 和 item 数据库内容。

**特殊需求：**系统的设计保障系统的健壮性。

**发生频率：**不经常发生。

### 1.2.3 增加数据库条目

**用例：**增加数据库条目

**范围：**数据库系统

**级别：**用户目标

**主要参与者：**经理、项目经理、采购员、销售

**涉众及其关注点：**

5. 管理者：希望能够员工能够准确快速地修改数据库，正常办公。

6. 基层员工：希望能快速地修改对应职位的数据库。

**前置条件：**员工的帐号必须已经添加到系统中。

**成功保证：**提示数据库修改成功。

**主成功场景：**

9. 经理可以新增所有数据库内容。

10. 项目经理可以新增 item 数据库内容。

11. 采购员可以新增 supplier、supply 数据库内容。

12. 销售可以新增 customer、order 数据库内容。

**特殊需求：**系统的设计保障系统的健壮性。

**发生频率：**经常发生。

## 第二章 架构设计

### 2.1 系统架构及原理

本系统采用 MVC 的设计模式，为了更加清晰地显示其架构图，将 model 层再细分为两层：Domain 层和 Foundation 层。Domain 层主要表示系统的数据模型，Foundation 层主要表示系统所使用的数据库类型支持和框架支持。

视图层：即 UI 层，在最顶层。其主要是显示与用户之间交互的界面，使用了 Bootstrap 框架为用户提供了一个可以跨平台的自适应的简约友好的界面，便于用户操作。

控制层，即 Controller 层，在第二层。其主要是连接 UI 层和 Domain 层，起到桥梁的作用。当用户通过 UI 层向系统发出请求时，控制层接收到请求并且调用 Domain 层中相应的模型进行处理，处理结束后再将结果保存在到模型中并返回给 UI 层。

领域层，即 Domain 层，在第三层。其主要是存放系统的各个模型数据。

基础层，即 Foundation 层，在最底层。其主要是为系统提供一些基础的服务。此处使用封装了 SQLAlchemy 框架的扩展来实现数据库 SQLite 的映射。

### 2.2 数据库设计

本系统数据库采用 Python 中的 PyMysql 模块来调用 MYSQL 数据库。

为了方便对系统五种角色的操作，本系统将用户设计成五种用户权限：管理员>经理>项目经理=采购员=销售。

#### 2.2.1 ER 图

本系统的 ER 图如图 2-1 所示：

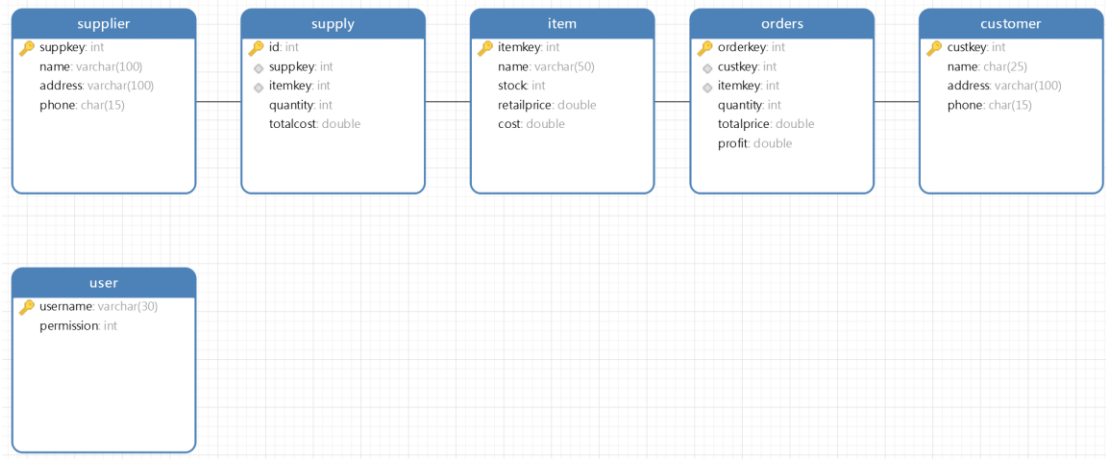


图 2-1：系统 ER 图

## 2.2.2 数据库表设计

本系统共有 6 张数据库表，分别是：customer, orders, item, supply, supplier 和 user 表。

表 2-1 对数据库表进行了整体的描述。

表 2-1：数据库表汇总

表名	说明
Customer	存放顾客信息的顾客表
Orders	存放订单信息的订单表
Item	存放库存信息的库存表
Supply	存放采购信息的采购表
Supplier	存放供应商信息的供应商表
User	存放用户信息的用户表

表 2-2 描述了 customer 表的详细信息。

表 2-2：顾客表

字段	数据类型	主键、外键	允许空	说明
Custkey	integer	主键	否	顾客编号
name	Char		是	顾客名
Address	Varchar		是	顾客地址
Phone	Char		是	顾客电话

表 2-3 描述了 orders 表的详细信息。

表 2-3：订单表

字段	数据类型	主键、外键	允许空	说明
Orderkey	Int	主键	否	订单编号
Custkey	Int		是	顾客编号
Itemkey	Int		是	库存编号
Quantity	Int		是	下单数量
Totalprice	Double		是	总金额
Profit	Double		是	盈利

表 2-4 描述了 item 表的详细信息。

表 2-4：库存表

字段	数据类型	主键、外键	允许空	说明
Item	Int	主键	否	库存编号
name	Varchar		是	库存姓名
Stock	Int		是	库存数量
Retailprice	Double		是	零售价
Cost	Double		是	成本

表 2-5 描述了 supply 表的详细信息。

表 2-5：采购表

字段	数据类型	主键、外键	允许空	说明
Id	Int	主键	否	采购号
Suppkey	Int	外键	是	供应商编号
Itemkey	Int	外键	是	库存编号
Quantity	Int		是	数量
Totalcost	Double		是	总价

表 2-6 描述了 supplier 表的详细信息。

表 2-6：供应商表

字段	数据类型	主键、外键	允许空	说明
----	------	-------	-----	----

Suppkey	Int	主键	否	供应商编号
Name	Varchar		是	供应商名
Address	Varchar		是	供应商地址
Phone	Char		是	供应商电话

表 2-7 描述了 user 表的详细信息。

表 2-7：用户表

字段	数据类型	主键、外键	允许空	说明
Username	Varchar	主键	否	用户名
Permission	Int		是	权限等级



## 第三章 模块设计

本系统按照用户群体划分为五个模块：管理员模块、经理模块、项目经理模块、采购员模块和销售模块。每个模块下面又分为多个子模块，比如管理员模块可以划分为：查询数据库模块、增加经理模块、删除经理模块、增加项目经理模块、删除项目经理模块、增加采购员模块、删除采购员模块、增加销售模块、删除销售模块。本章主要选取经理的增加销售模块、销售的增加顾客模块和管理员的授权模块进行详细的介绍。

### 3.1 增加销售模块

增加销售模块是增加用户模块的子模块，是经理用户和管理员用户的功能。目前仅支持一次操作创建一个用户，并支持自定义命名；后续考虑加入新增用户数量选项，以支持一次增加多个用户，并以序号自动命名，比如销售 1、销售 2 等等。其流程图如图 3-1 所示：

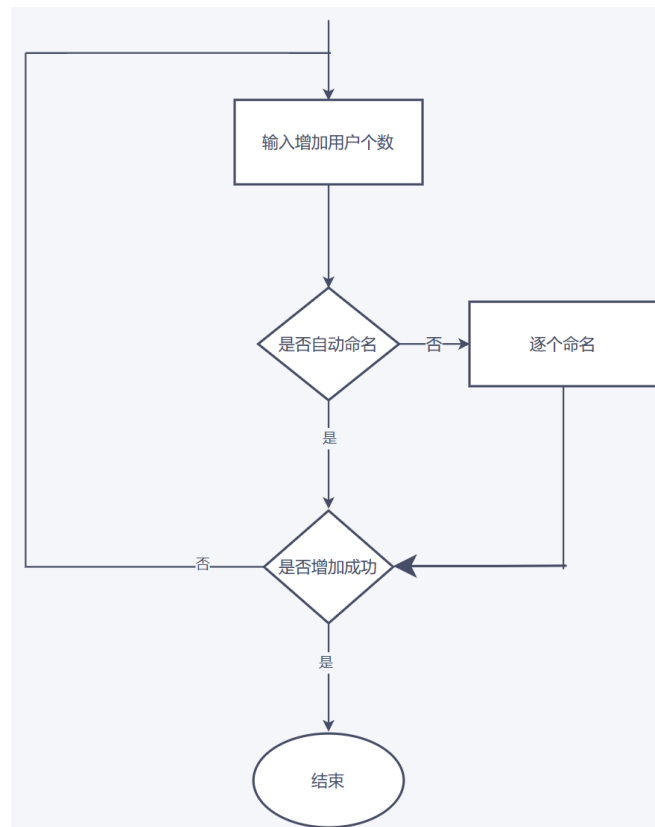


图 3-1：增加用户流程图

图 3-2 主要描述了增加一个用户的核心代码。当经理选择提交新增销售的用户名时，系统同时要求输入与用户名配对的密码，组合成新建用户的 sql 语句，然后执行该语句并提交。

```
def create_user(username, password):  
    query = "CREATE USER '{}'@localhost IDENTIFIED BY '{}'".format(  
        username, password)  
    execute_query(query)  
    conn.commit()
```

图 3-2：增加用户代码

### 3.2 增加顾客模块

增加顾客模块是增加数据库模块的子模块，是销售用户和经理用户的功能。销售需要先选择增加顾客，然后依次输入顾客名，地址和电话，接着判断是否增加成功，若不成功则需要重新操作。其流程图如图 3-3 所示：

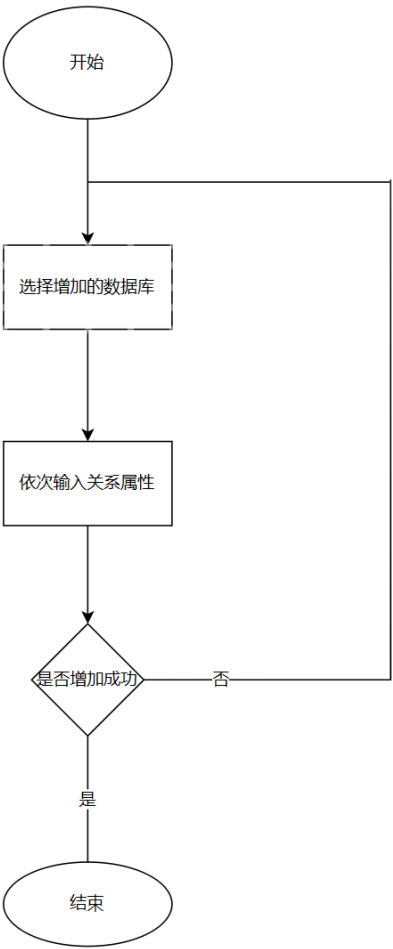


图 3-3：增加数据库流程图

增加顾客首先需要输入顾客名，地址和电话，接着将三个属性拼接成一个 sql 插入语句，然后执行该语句，并提交，若操作不成功将会输出错误信息，并要求重新操作。增加数据库的核心代码如下图 3-4 所示：

```
def add_customer(name, address, phone):  
    query = "INSERT INTO customer(name, address, phone) VALUES('{}', '{}', '{}')".format(  
        name, address, phone)  
    ret = execute_query(query)  
    conn.commit()  
    if ret != -1:  
        print('Customer added successfully.')
```

图 3-4：增加数据库的代码

### 3.3 授权模块

授权模块是管理员用户和经理用户的功能，可以给项目经理，采购员以及销售对应的数据库权限。管理员先输入被授权的用户名，再选择将要授予的权限，接着判断是否增加成功，若不成功则需要重新操作。其流程图如图 3-5 所示：

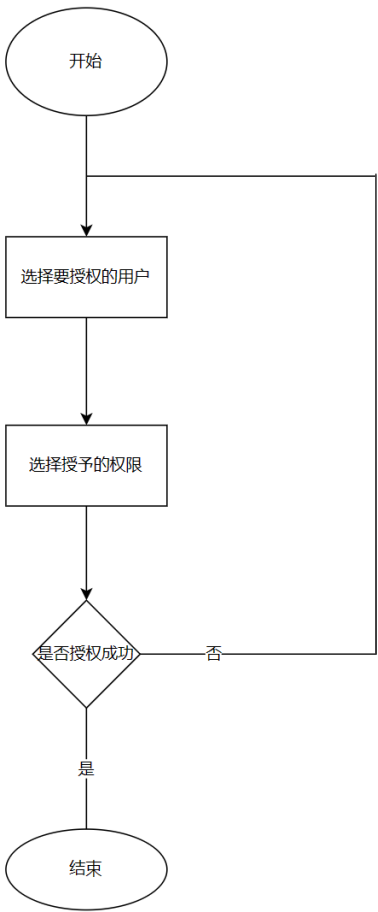


图 3-5：授予权限流程图

管理员先输入被授权的用户名，再选择将要授予的权限，组合成 sql 语句后执行并提交，接着判断是否增加成功，若操作不成功将会输出错误信息，并要求重新操作。其流程图如图 3-6 所示：

```
def grant_privilege(username, perm):  
    ret = None  
    if perm == 10:  
        query = "GRANT SalesToCRole TO '{}@localhost'.format(username)"  
        ret = execute_query(query)  
        conn.commit()  
    elif perm == 20:  
        query = "GRANT SalesToBRole TO '{}@localhost'.format(username)"  
        ret = execute_query(query)  
        conn.commit()  
    elif perm == 30:  
        query = "GRANT ItemManagerRole TO '{}@localhost'.format(username)"  
        ret = execute_query(query)  
        conn.commit()  
    elif perm == 100:  
        query = "GRANT ManagerRole TO '{}@localhost'.format(username)"  
        ret = execute_query(query)  
        conn.commit()  
    elif perm == 255:  
        query = "GRANT ALL PRIVILEGES ON *.* TO '{}@localhost'.format(  
            username)  
        ret = execute_query(query)
```

图 3-6: 授予权限的核心代码

## 第四章 部署与应用

### 4.1 部署图

本系统采用客户端直连数据库模式，当客户端和数据库完全部署应用后，客户端即可通过接口对数据库进行相应的访问。系统的部署图如图 4-1 所示：

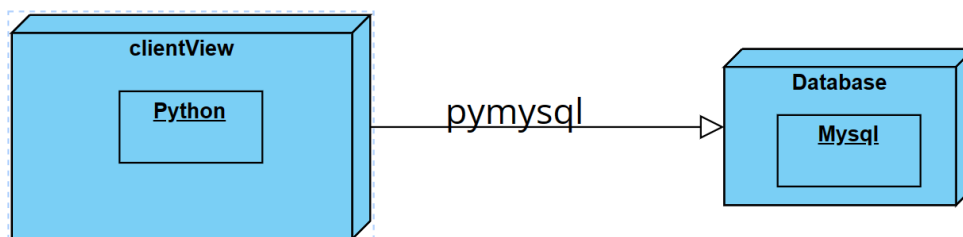


图 4-1：系统部署图

### 4.2 部署过程

#### 4.2.1 Python 环境配置

由于 Mac 和 Linux 系统已经实现对 Python 的安装和配置，而 Windows 系统没有。因此本节介绍 Windows 系统下 Python 的安装和配置。

首先到 Python 官网 <https://www.python.org/downloads/> 下载 2.7 的版本，然后进行安装。

安装成功后，需要配置 Python 的环境变量。首先进入控制面板，点击系统，然后在页面左侧点击“高级系统设置”，如图 4-2 所示：



图 4-2：系统页面

接着系统会出现一个弹窗“系统属性”，点击“高级”选项页面下的“环境变量”按钮，如图 4-3 所示：

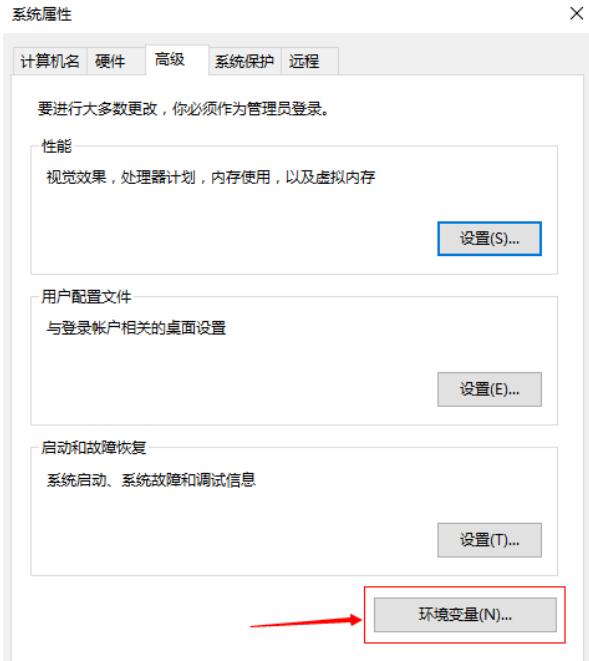


图 4-3：系统属性页面

然后在弹出的“环境变量”页面下方“系统变量”里找到 Path 选项并双击，如图 4-4 所示：

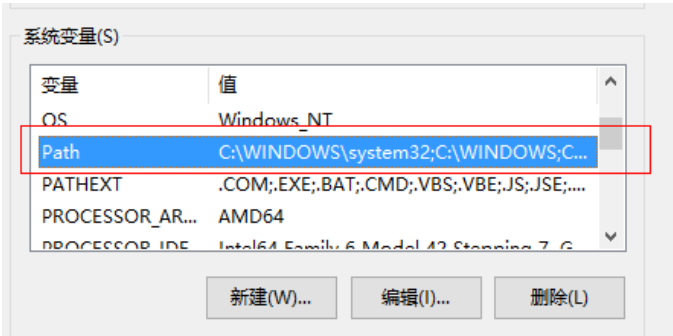


图 4-4：系统变量页面

接着在弹出的页面中添加 Python 的安装路径从而完成 Python 环境的配置。如图 4-5 所示：

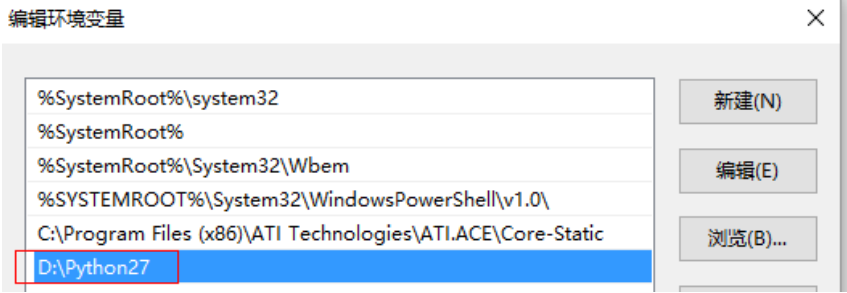


图 4-5：添加 Python 环境变量

## 4.2.2 Mysql 安装配置

### 一、下载

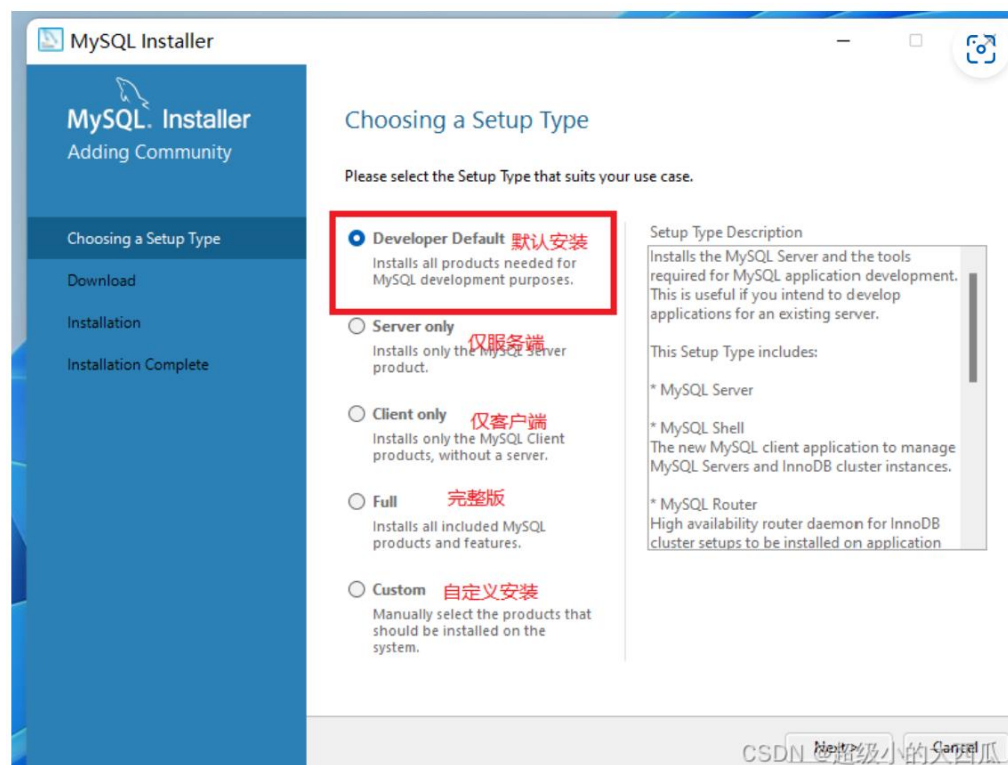
进入 mysql 官方网站并下载社区版。



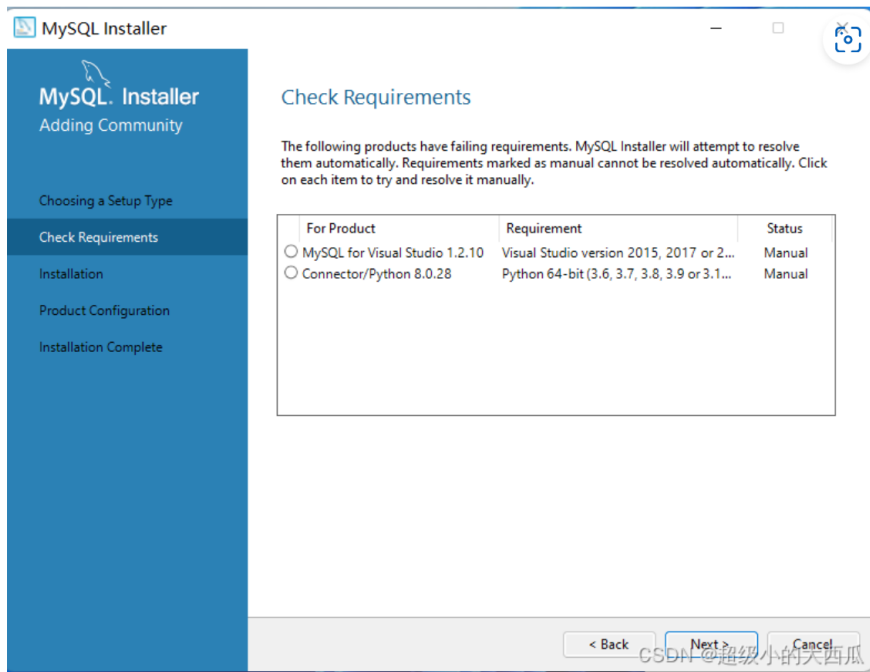
### 二、安装 mysql

1. 双击安装软件开始安装。

2. 下图是选择安装类型，选择默认安装就可以，然后点击“next”。

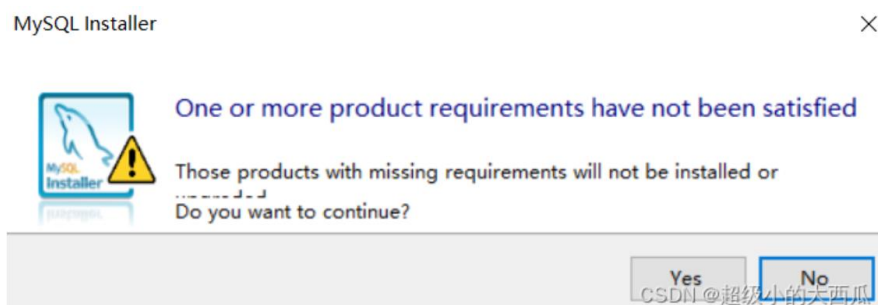


3. 检测需要的安装，这里我们直接点击 next

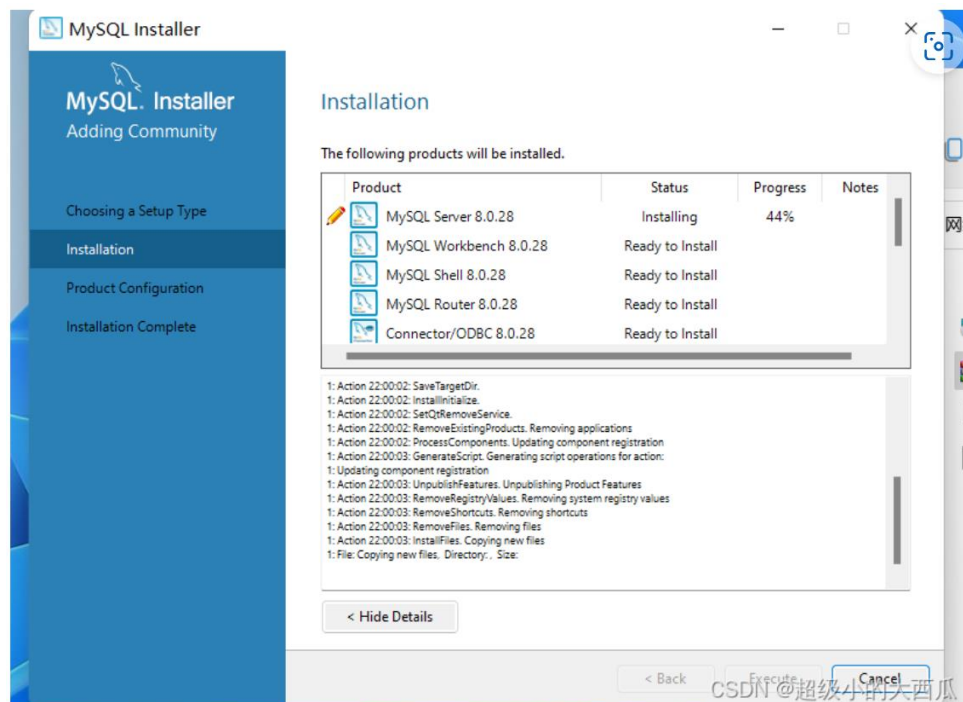


4. 点击 yes

MySQL Installer

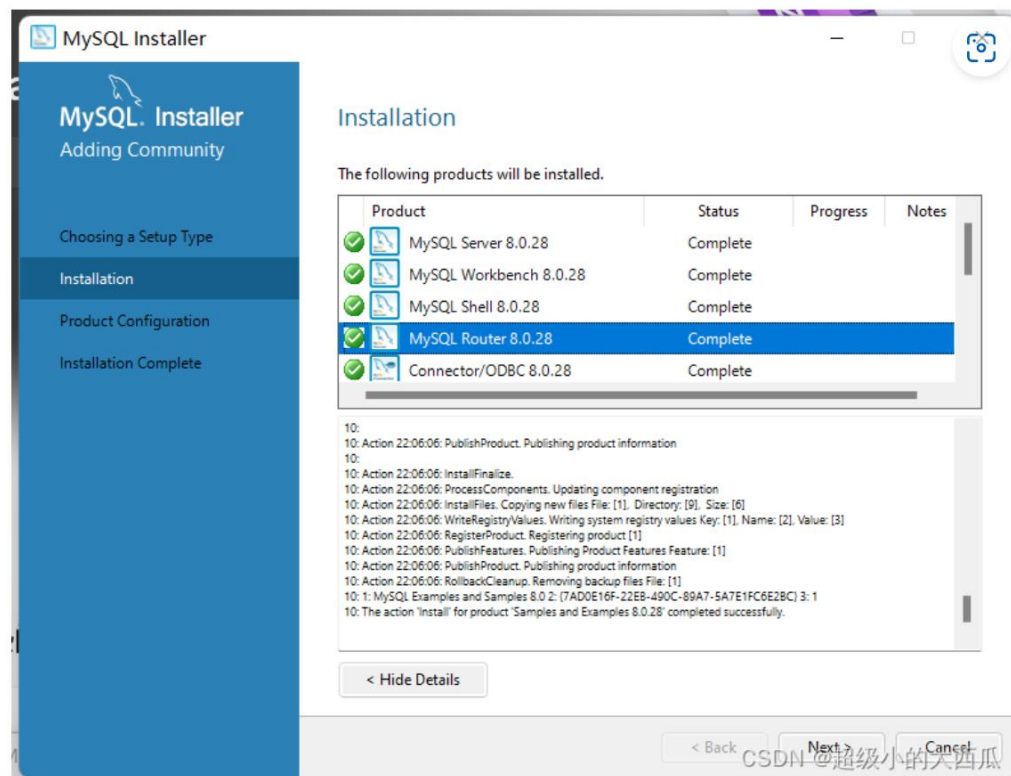


5. 点击 Execute 开始安装

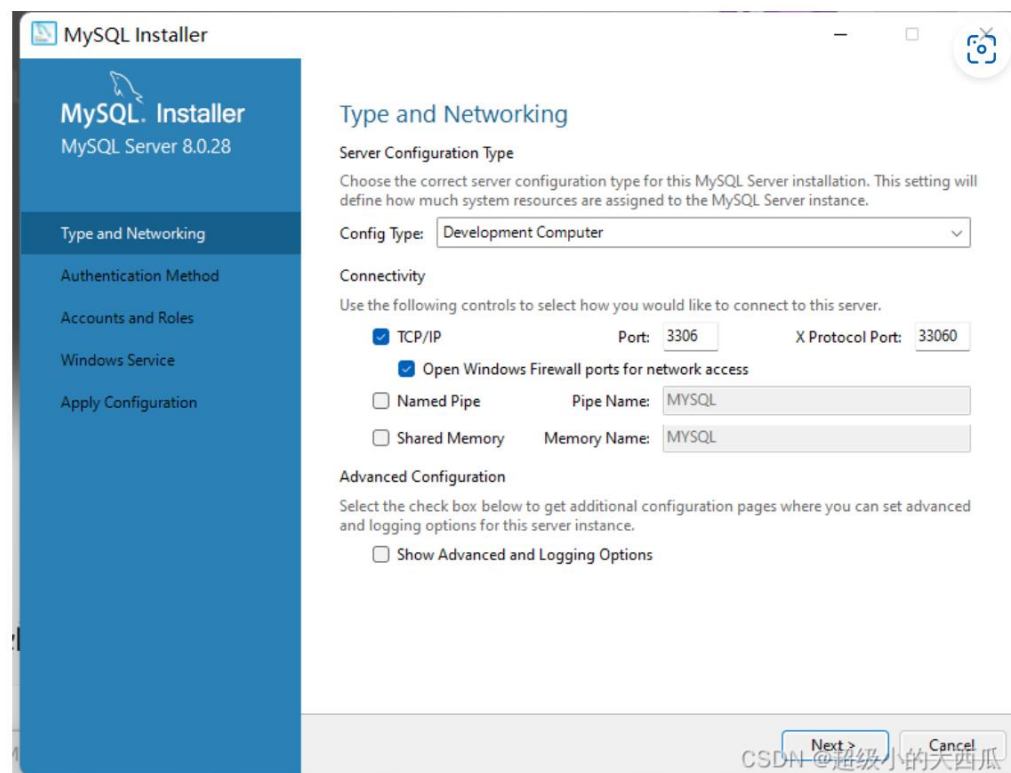




6.安装完成后点击 next

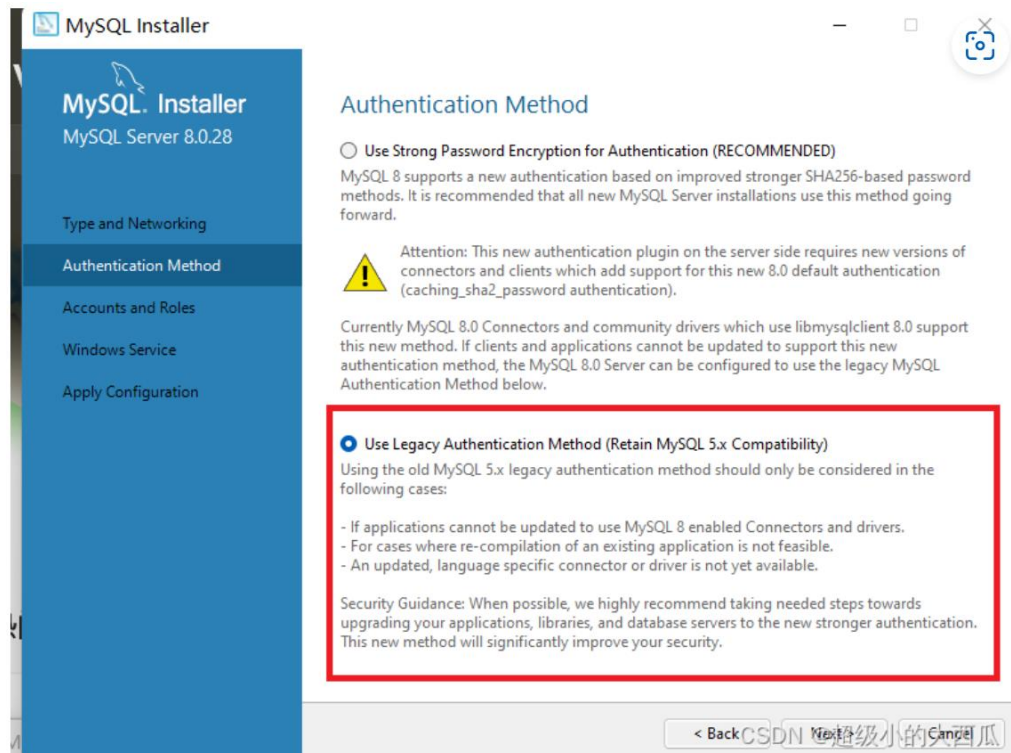


7.多次点击 next （直到出现以下页面），默认选第一个，点击“next”继续

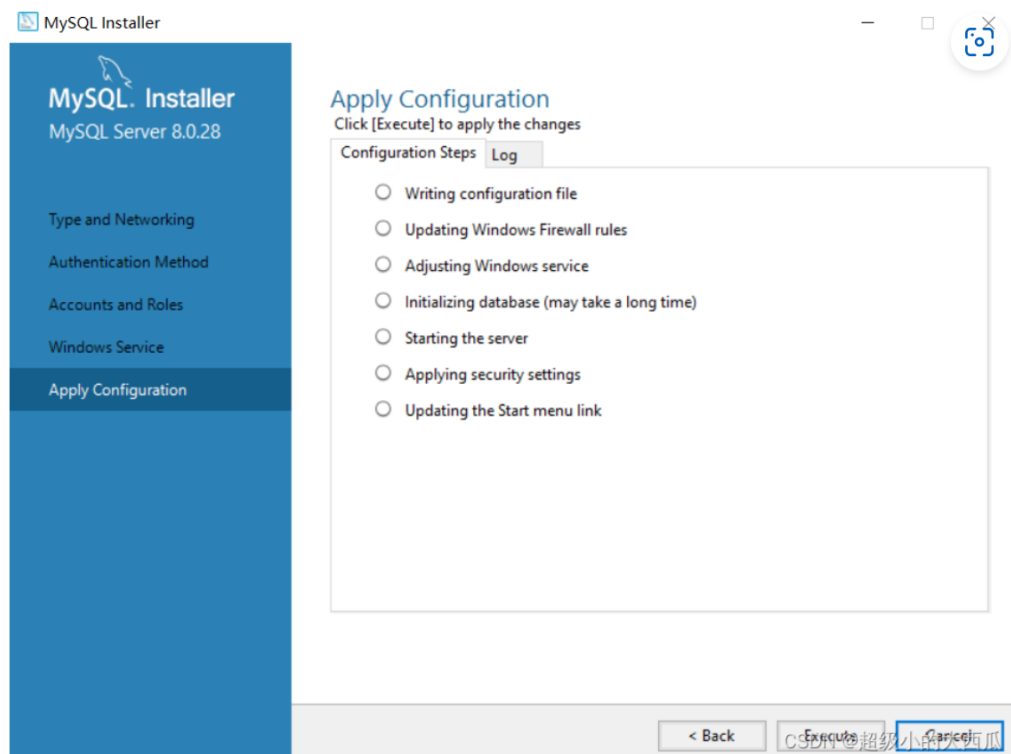


8.密码验证方式（Authentication Method）这一步很重要

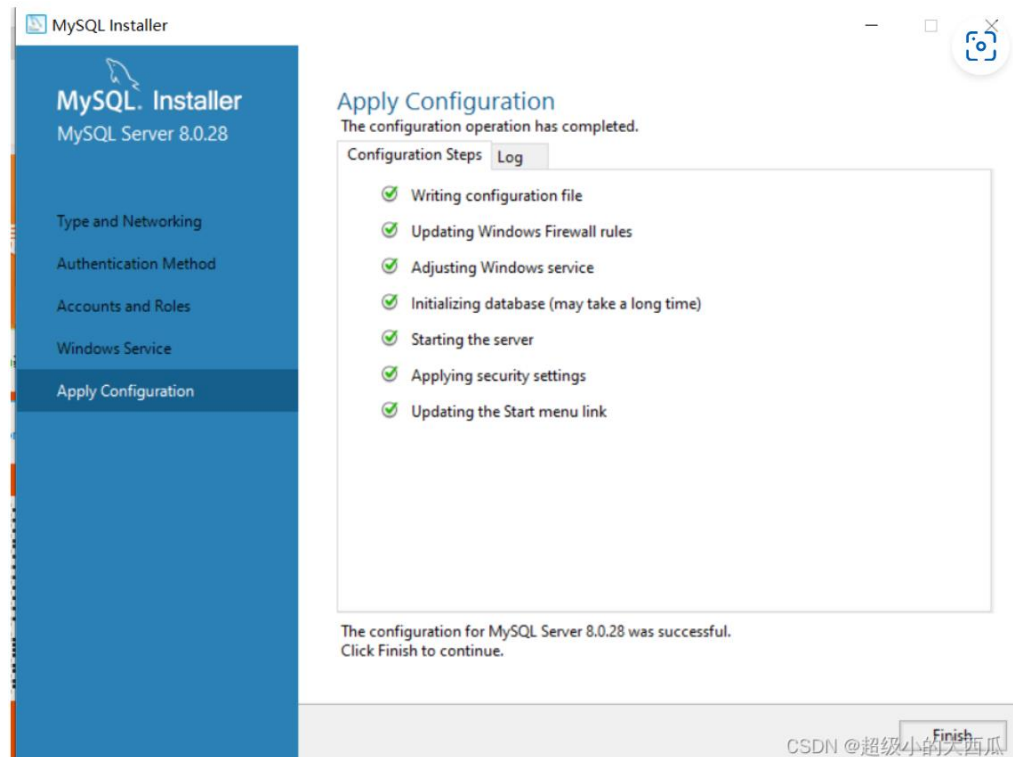
第一个是强密码校验，mysql 推荐使用最新的数据库和相关客户端，MySQL8 换了加密插件，所以如果选第一种方式，很可能你的 navicat 等客户端连不上 mysql8。所以这里一定要选第二个（下图红框的地方）



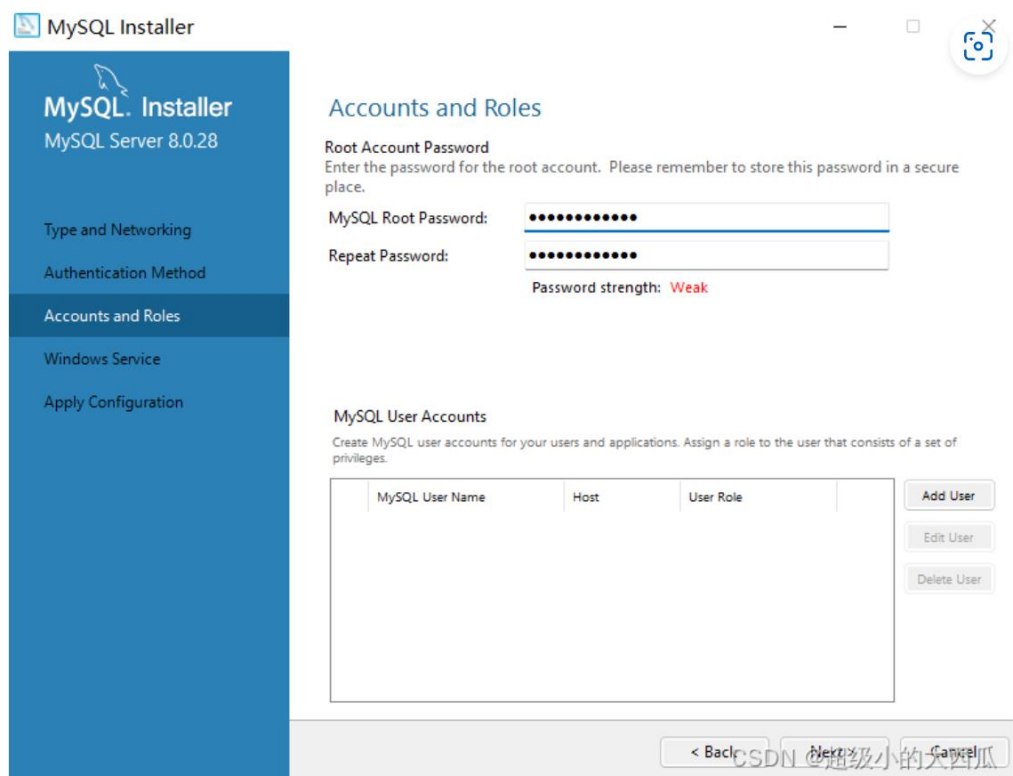
9.点击 Execute 开始安装



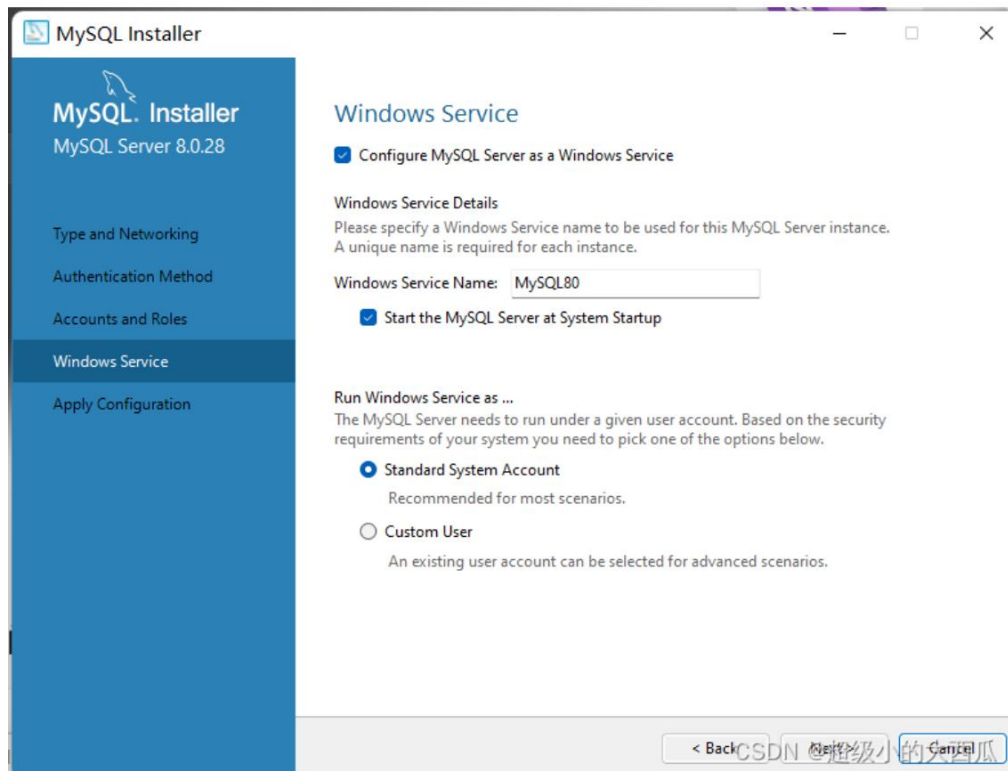
10.安装完成后点击 Finish



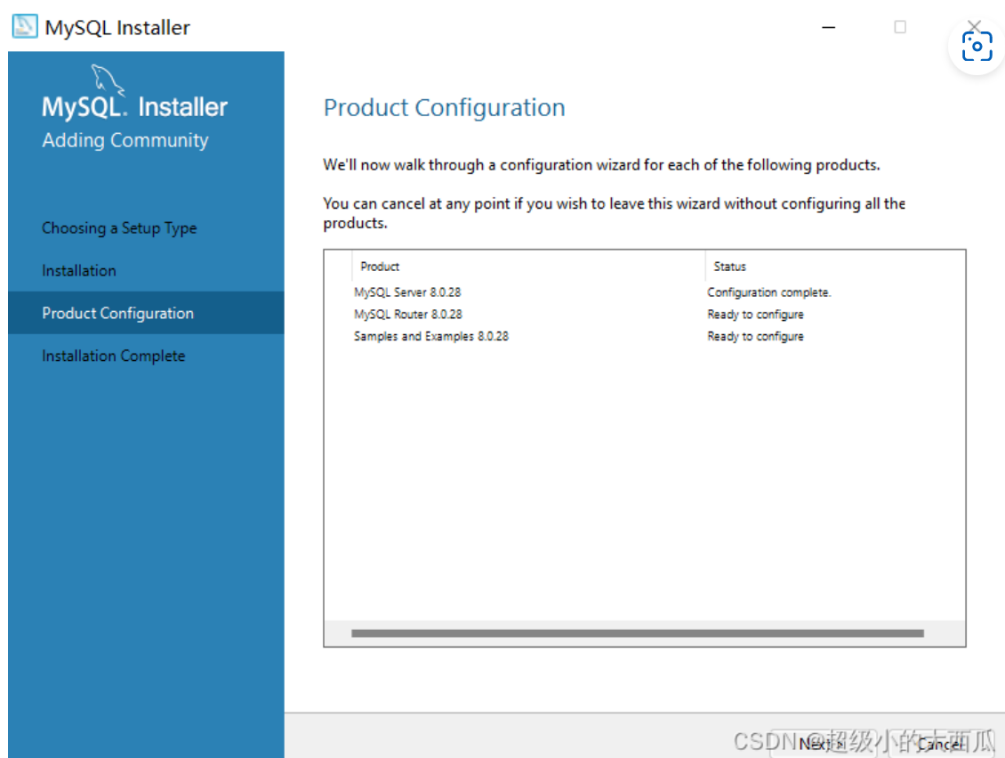
11.设置密码，需要牢记，后面要用这个密码链接数据库。输入完，点击 next 继续。



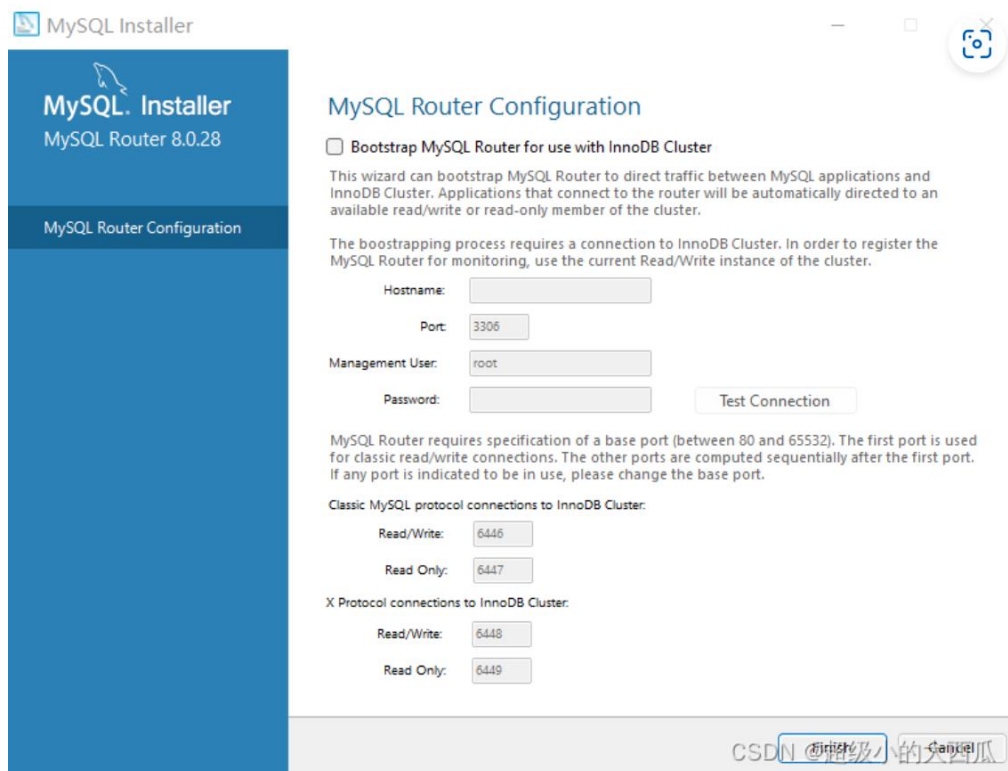
12.默认点击 next



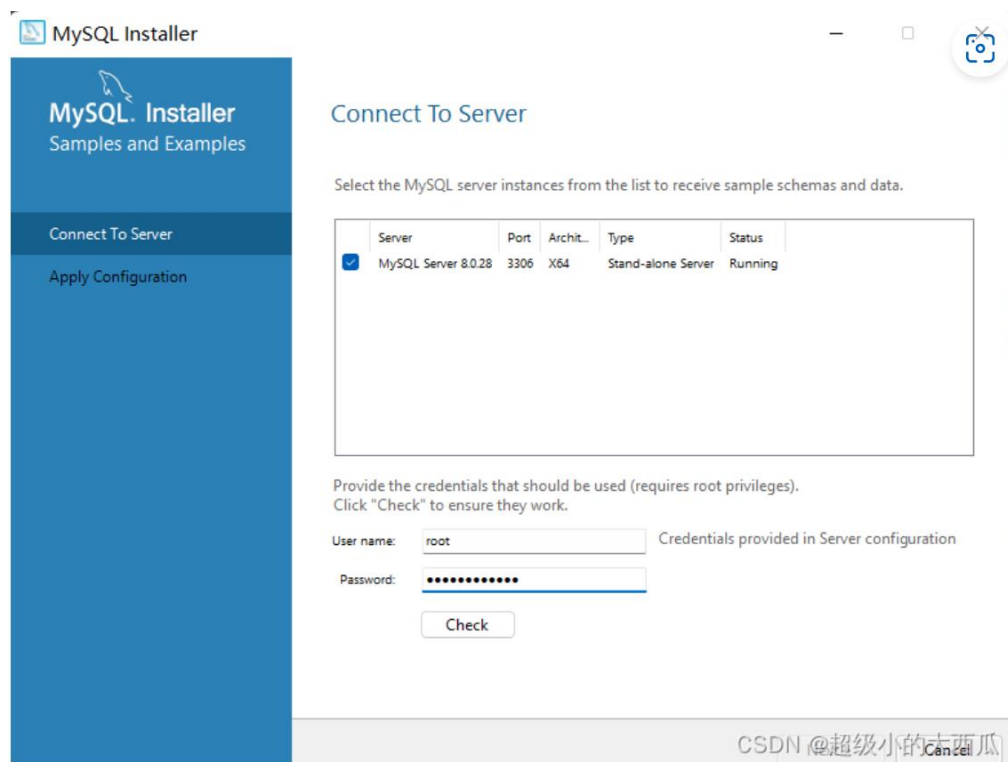
13. 点击 next



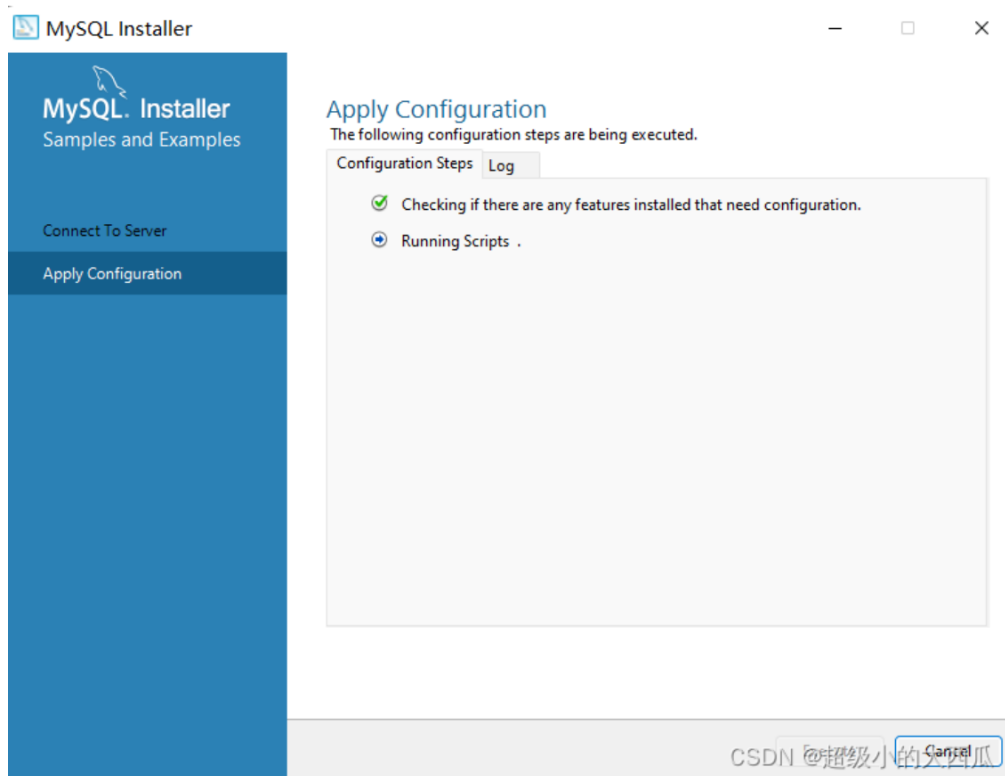
14. 默认点击 next



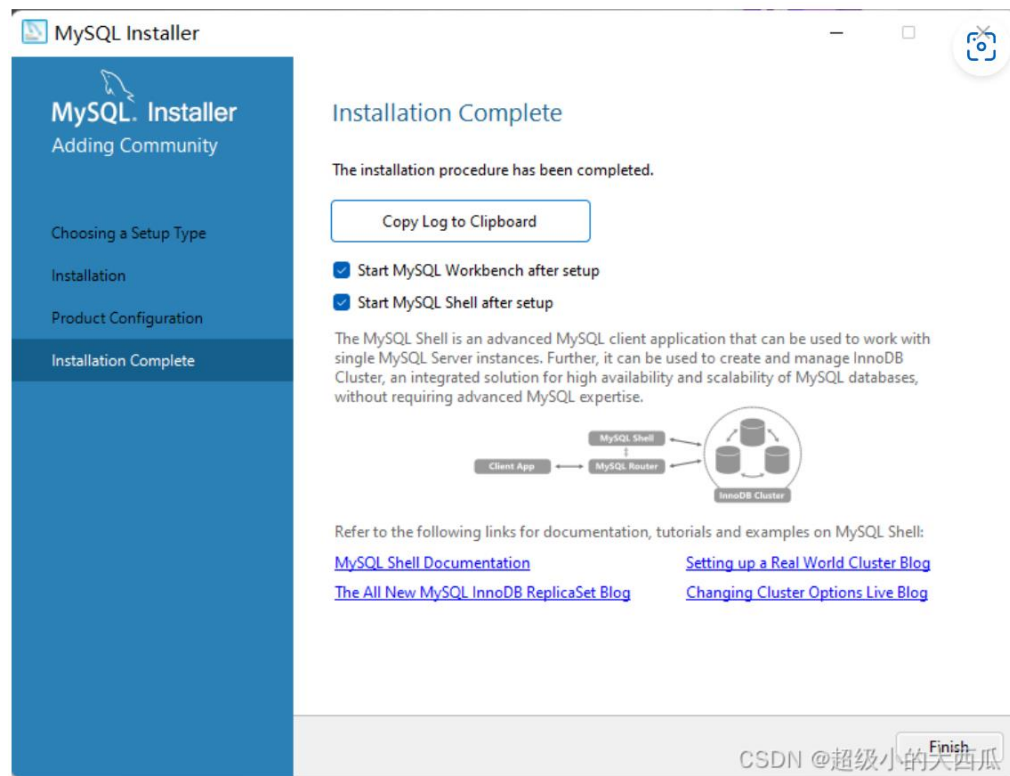
15.输入用户名和密码，点击 check，检查是否正确。成功了 Next 才可点击



16.点击 Execute 执行安装（完成后点击 Finish）



17.完成点击 Finish（第一个是 Mysql 的客户端，如果不使用只勾选第 2 个“start mysql shell after setup”，点击“finish”）（到这里数据库的安装就完成了）



### 4.2.3 PyMysql 及相关依赖的安装配置

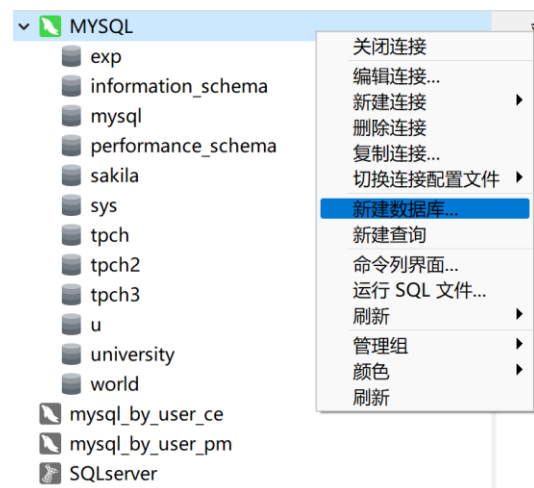
使用 Windows 系统打开终端命令行，进入 Python 安装目录下的 Script 文件夹下，然后依次执行以下命令：

```
pip install pymysql
```

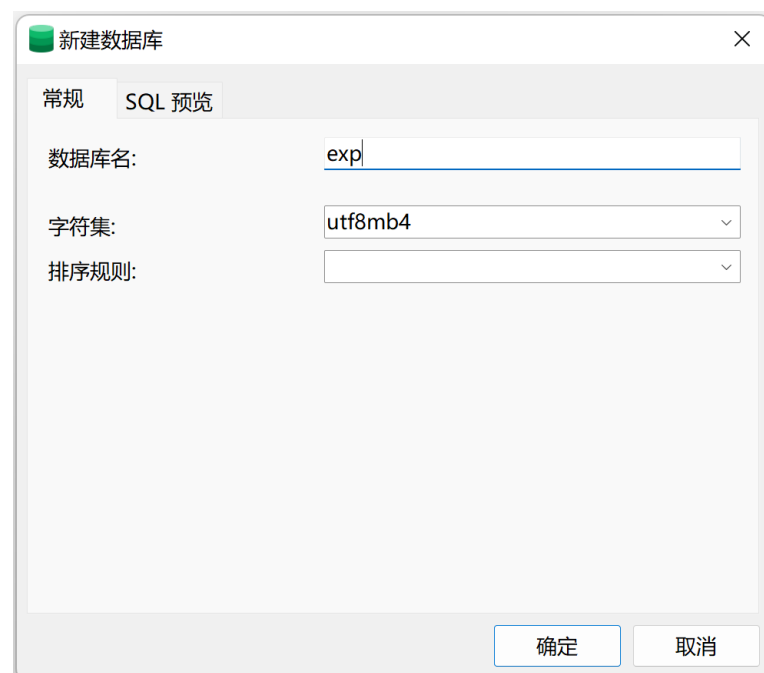
```
pip install getpass
```

### 4.2.4 建立数据库

打开 Navicat 等数据库管理软件，连接 Mysql 数据库，新建数据库。

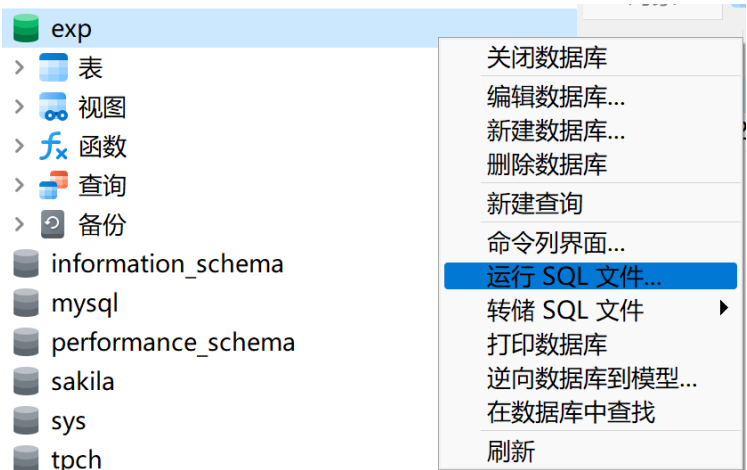


数据库名为 exp，字符集为 utf8mb4

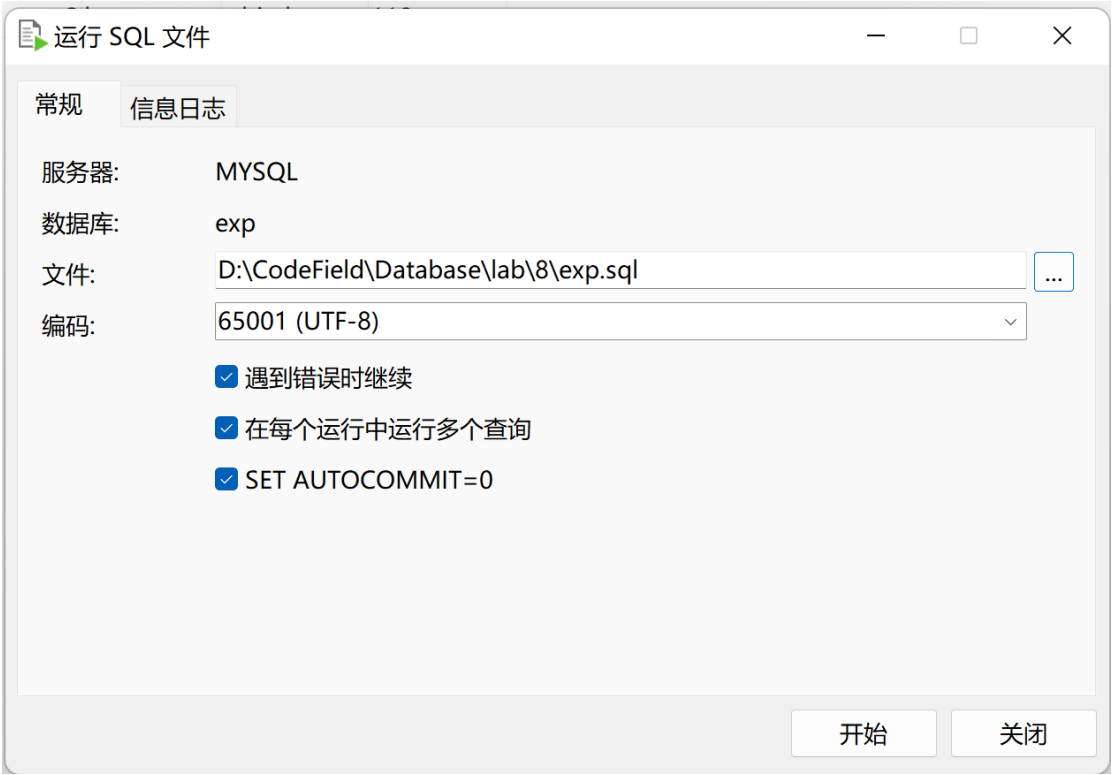




双击打开该数据库，接着右键选择运行 SQL 文件



接着选择 exp.sql 并点击开始，无报错后关闭。同样执行 role.sql 即可。



#### 4.2.5 运行

首先运行项目文件夹下的 `connect.py` 文件来运行系统，注意 `main` 函数中的 `user`，这里输入用户名，用户名和密码正确即可使用系统。



### 4.3 部分功能截图

管理员登陆页面如图所示：

```
Connection established
Welcome to the database
SELECT permission FROM user WHERE username = 'root'
Query successful.
You are an admin
Select the operation to perform:
1. Create user
2. Delete user
3. Grant permission
4. Revoke permission
5. Execute query
6. Activate role
q. quit
Enter choice:
```

经理登陆页面如图所示：

```
Connection established
Welcome to the database
SELECT permission FROM user WHERE username = 'manager'
Query successful.
You are a Manager
Select the operation to perform:
1. Add entry
2. Search entry
3. Update entry
4. Delete entry
5. Create user
6. Delete user
7. Grant permission
8. Revoke permission
q. quit
Enter choice: 1
```

项目经理登陆页面如图所示：

```
Connection established
Welcome to the database
SELECT permission FROM user WHERE username = 'itemmanager'
Query successful.
You are an Item Manager
Select the operation to perform:
1. Add item
2. Search item
3. Search order
4. Search supply
q. quit
Enter choice: q
```

采购员登陆页面如图所示：

```
Connection established
Welcome to the database
SELECT permission FROM user WHERE username = 'corporatesales'
Query successful.
You are a Corporate Sales
Select the operation to perform:
1. Add supplier
2. Search supplier
3. Add supply
4. Search supply
5. Search products
q. quit
Enter choice: q
```

销售登陆页面如图所示：

```
Connection established
Welcome to the database
SELECT permission FROM user WHERE username = 'salesperson'
Query successful.
You are a Salesperson
Select the operation to perform:
1. Add customer
2. Search customer
3. Add order
4. Search order
5. Search products
q. quit
Enter choice: 5
```

管理员授权页面如图所示（已授权）：

```
Enter choice: 3
Enter username: salesperson
Select permission:
1. Salesperson
2. Corporate Sales
3. Item Manager
4. Manager
5. Admin
Enter choice: 1
GRANT SalesToCRole TO 'salesperson'@localhost
Query successful.
INSERT INTO user(username,permission) VALUES('salesperson',10)
(1062, "Duplicate entry 'salesperson' for key 'user.PRIMARY'")
Query failed. Check for duplicate entries.
```

经理查询界面如图所示：

```

You are a Manager
Select the operation to perform:
1. Add entry
2. Search entry
3. Update entry
4. Delete entry
5. Create user
6. Delete user
7. Grant permission
8. Revoke permission
q. quit
Enter choice: 2
Search for:
1. Customer
2. Supplier
2. Customer name
3. Customer address
4. Customer phone
Enter choice: 1
Enter customer key: henry
SELECT * FROM customer WHERE custkey = 'henry'
Query successful.
Not found

```

销售增加顾客页面如图所示:

```

Connection established
Welcome to the database
SELECT permission FROM user WHERE username = 'salesperson'
1. Add customer
2. Search customer
3. Add order
4. Search order
5. Search products
q. quit
Enter choice: 1
Enter name: tony
Enter address: sysu
Enter phone: 119
INSERT INTO customer(name, address, phone) VALUES('tony', 'sysu', '119')
Query successful.
Customer added successfully.

```

Navicat 验证增加顾客操作成功:

对象 customer @exp (MYSQL) - 表			
custkey	name	address	phone
2	henry	whitehouse	110
3	tony	sysu	119