

Day 1 环境安装配置

第0章：环境搭建

本章重点：

- 在电脑上安装MySQL数据库系统
- 安装客户端并连接到本机上的MySQL数据库
- 使用提供的脚本创建本教程所使用的示例数据库

1. MySQL 8.0 的安装

考虑到大家所使用的操作系统的不同，本教程分别提供了 windows 10, macOS和 centos平台上的MySQL 8.0 的安装流程，你可根据自己所使用电脑的操作系统选择以下三节中的一节进行学习和安装。

1.1 windows 下 MySQL 8.0 的下载安装

首先以最常见的 win10 为例，介绍 MySQL8.0 的下载安装。

1.1.1 下载

MySQL 针对个人用户和商业用户提供了不同的版本，MySQL 社区版(MySQL Community Edition) 是供个人用户免费下载的开源数据库，本教程将以MySQL 社区版为例进行安装连接和SQL查询的讲解。

MySQL 官网上的社区版软件的下载地址<https://dev.mysql.com/downloads/>，选择 MySQL Installer for Windows可以下载 windows 操作系统下的最新版 MySQL安装文件. 如果需要安装历史版本，可以选择最后的Download Archives后选择MySQL Installer，然后在新页面里选择所需历史版本的社区版。

如果想下载本教程所使用的 MySQL 8.0.21.0, 也可以在百度网盘下载，

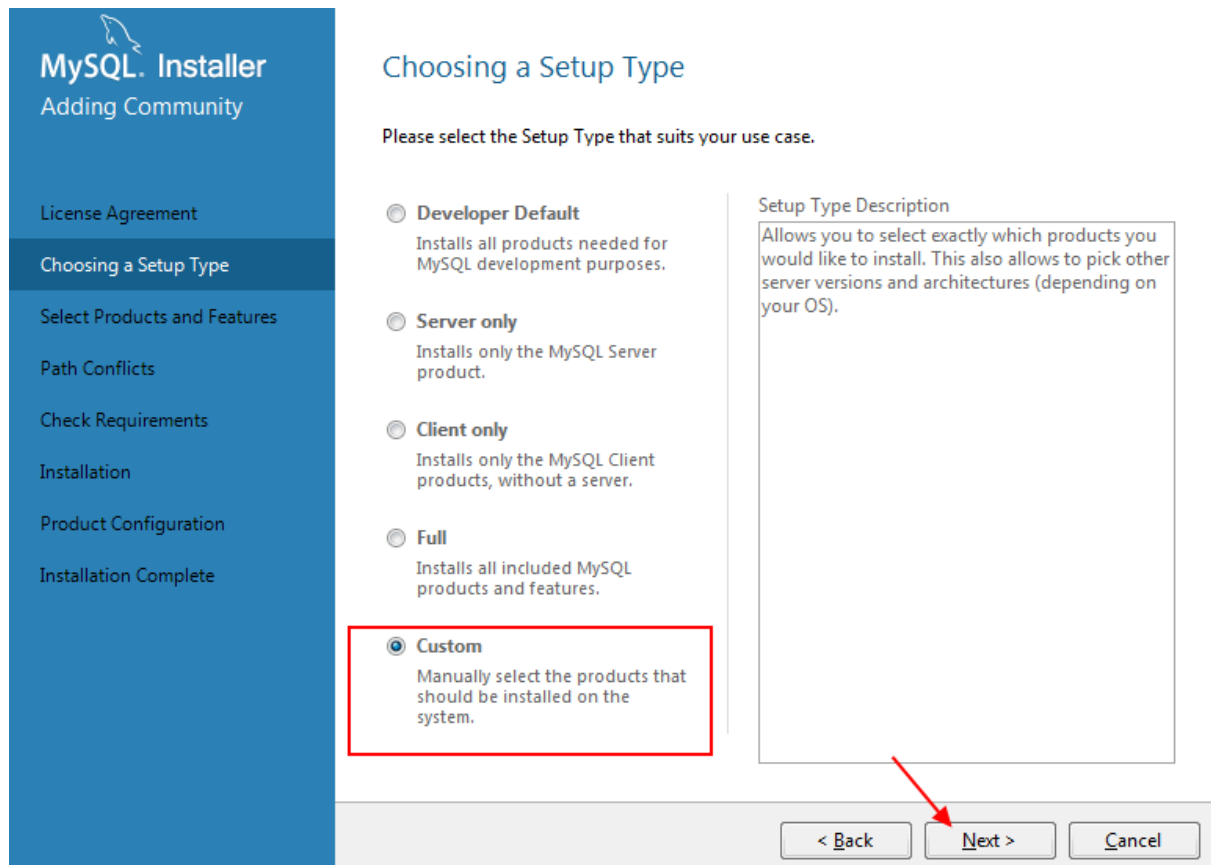
下载链接：<https://pan.baidu.com/s/1SOtMoVqgRXwa2qD0siHclg>提取码：80lf

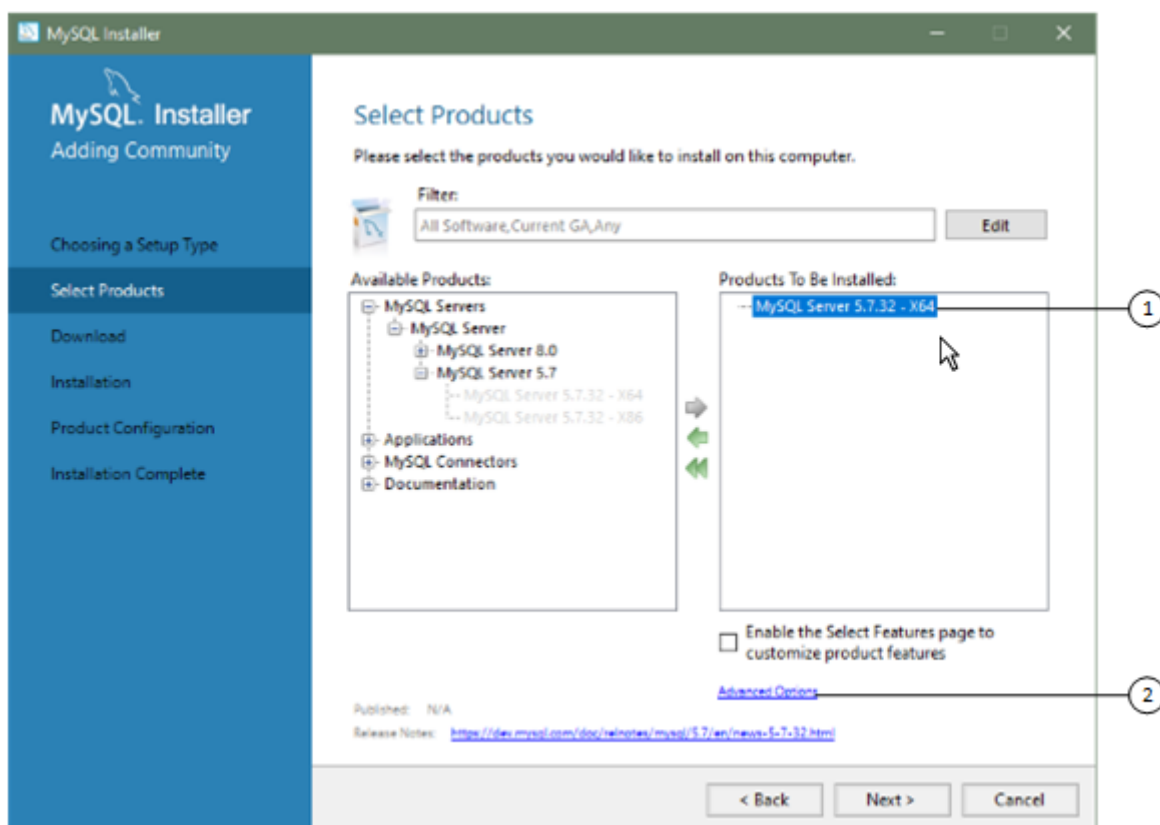
备用下载链接：<https://pan.baidu.com/s/1zK2vj50DvuAee-EqAcl-0A>提取码：80lf

1.1.2 安装

找到刚才下载好的msi文件, 双击开始安装. 初学者建议采用完全安装模式(Full)进行安装（缺点是不能修改安装目录）

如果想修改安装路径, 需要选择 **custom** 安装模式。包括安装路径和数据存放路径, 注意要安装组件至右侧, 再进行点选, 之后点击 **Advanced Options** 按钮进行修改。





安装过程中, 部分模块会依赖其他组件(每台电脑上列出的依赖项很可能会有不同)。

如果你的电脑之前没有安装过这些组件, 则需要额外进行安装, 此处点击 Execute 按钮即可:

在这些所依赖的组件的安装过程中, 只需要一路选择"同意"并逐个安装就可以了:

安装好一个组件后, 点击关闭按钮, 自动开始安装下一个组件(这一步根据操作系统版本可能会略有不同)

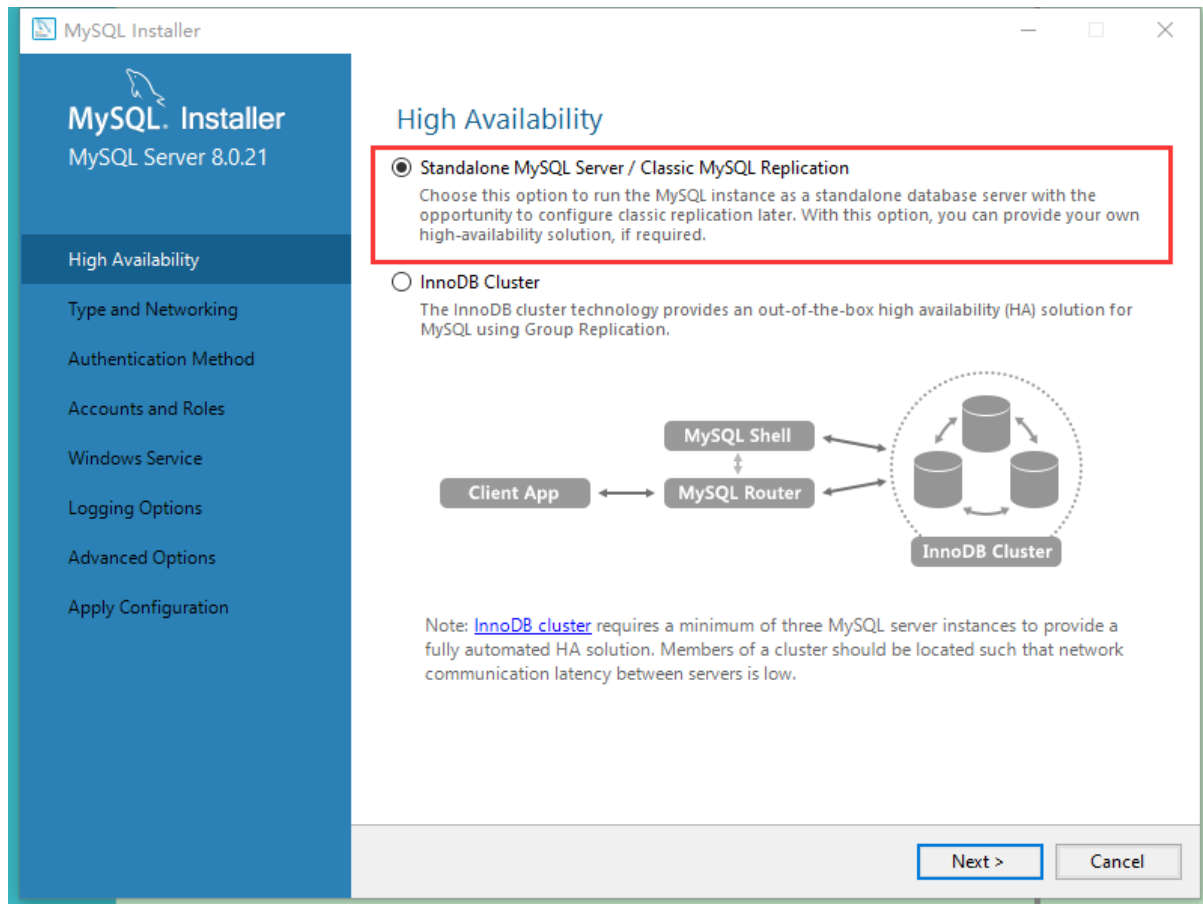
正常情况下, 会将所有组件安装成功. 但可能会有个别组件未安装成功. 个别组件没有呈现绿色是因为你的电脑中缺少某个程序, 例如, 如果你的电脑没有安装 Python 环境, 则该项目就不会呈现绿色. 待下边剩下 3 个按钮且上方大部分组件为绿色时, 即可点击 Next:

如果有个别组件未安装成功, 此时可以先选择 Yes, 忽略个别组件的安装。

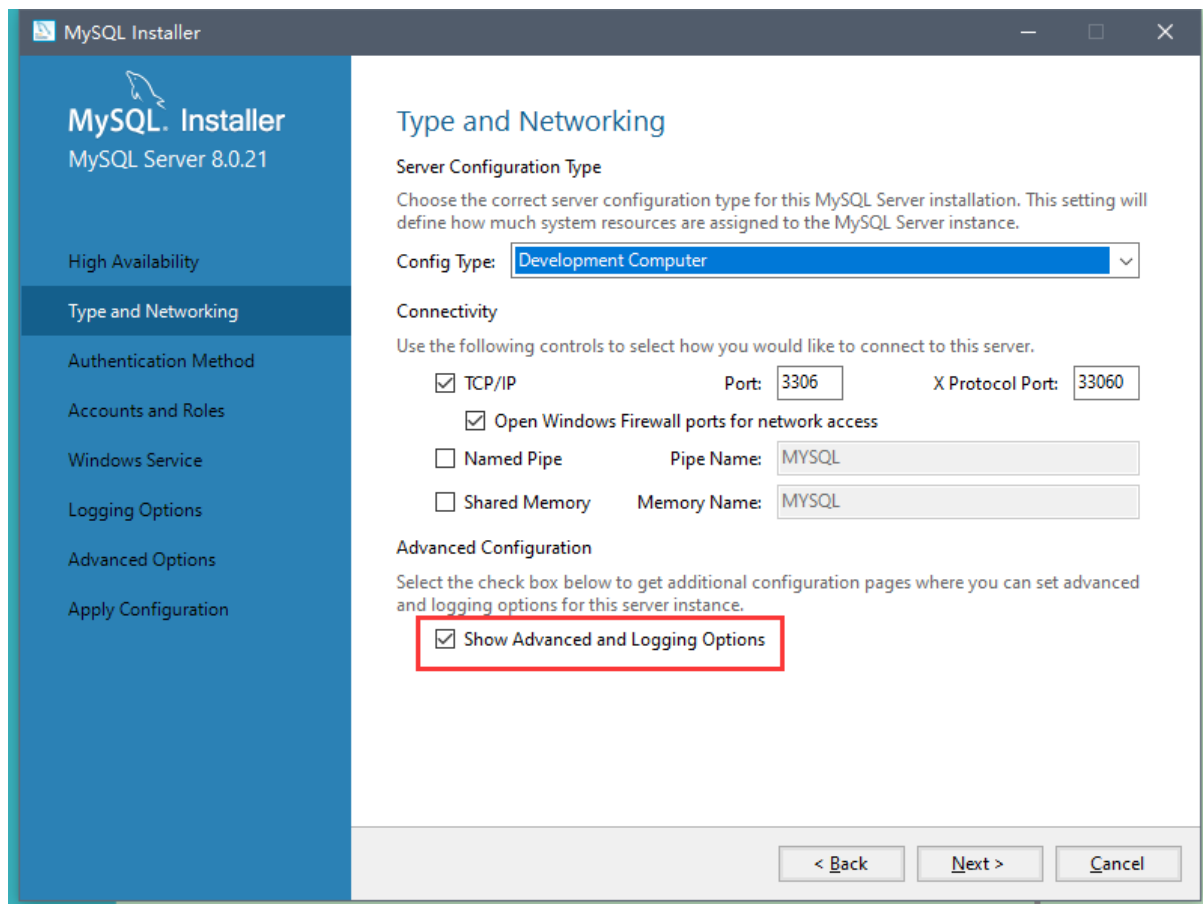
点击 Excute, 开始安装服务器软件MySQL Server, 连接和查询软件MySQL Workbench及其他相关软件等内容。

稍等片刻, 安装完成后, 点击 Next

下图这一步是选是否以集群方式安装 MySQL, 我们选择默认的第一个, 然后点击 Next:

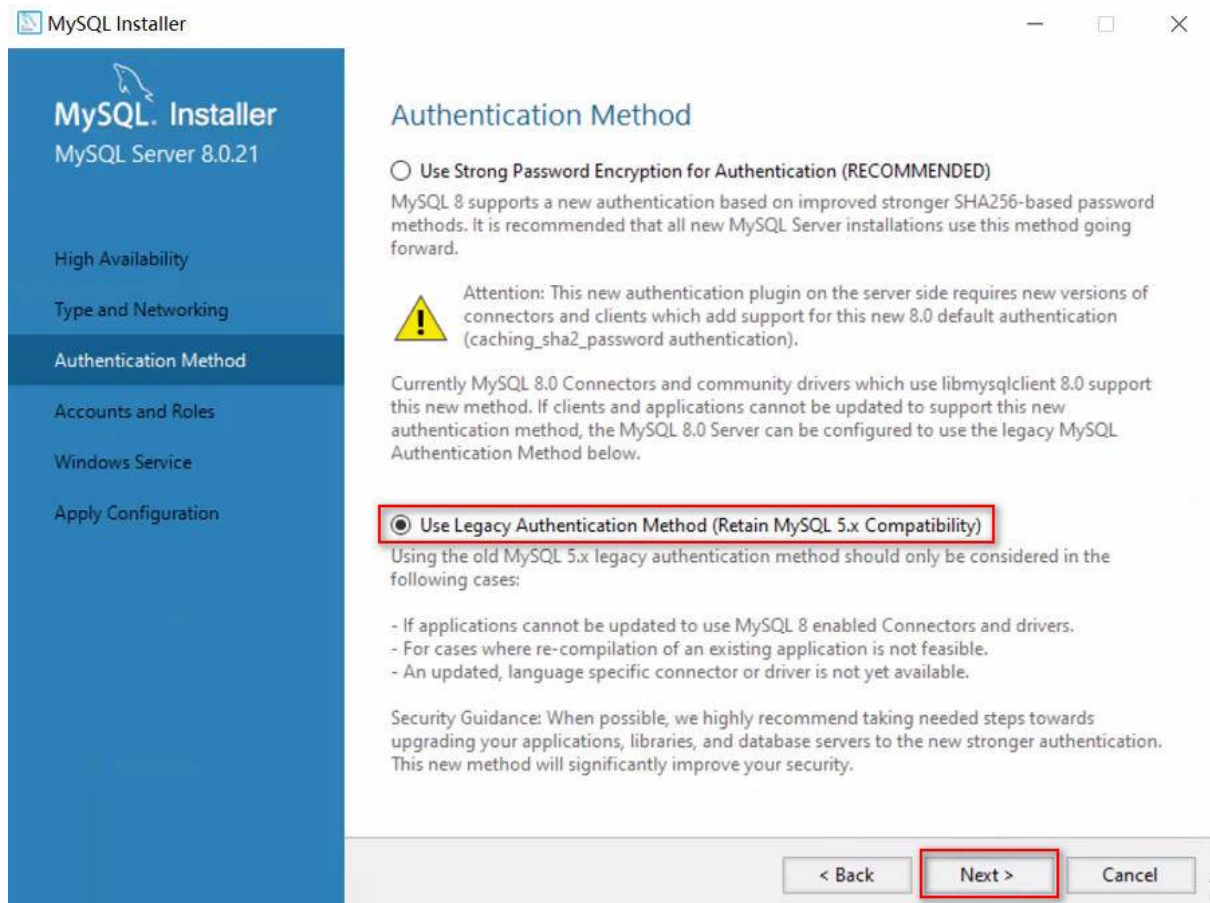


此处上边的各种相关配置保持默认即可, 勾选最下边的"Show Advanced and Logging Options"框, 然后点击 Next:

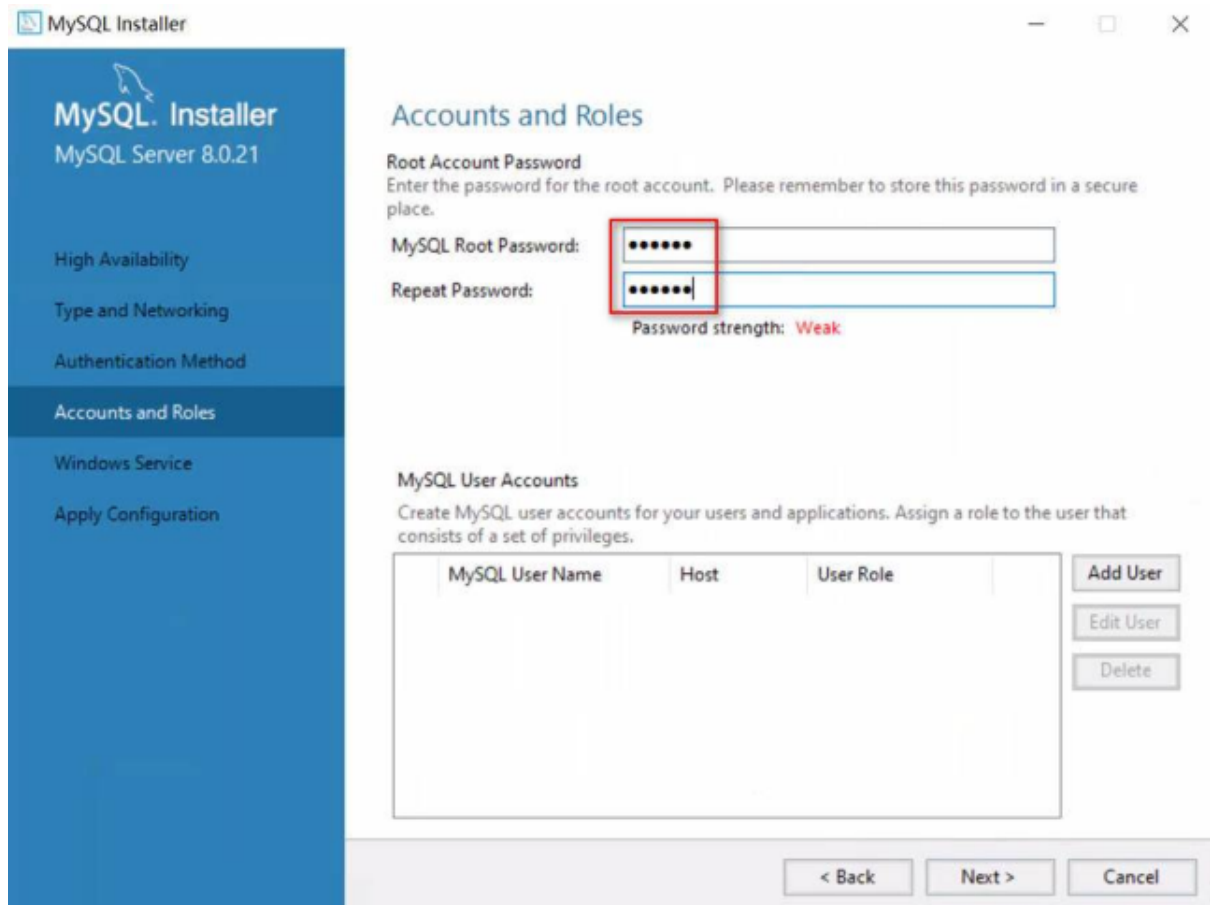


下图是密码强度的设置, 第一种模式为强密码校验模式, MySQL 8.0 推荐使用最新的数据库和客户端, 更换了加密插件, 这可能导致第三方客户端工具无法连接数据库。

第二种加密方式沿袭了 MySQL 5.x 的加密方式, 对第三方工具连接不敏感, 我们仅为了学习 SQL 查询, 并不需要很高的安全性, 因此此处请务必选择第二种方式(非常重要):



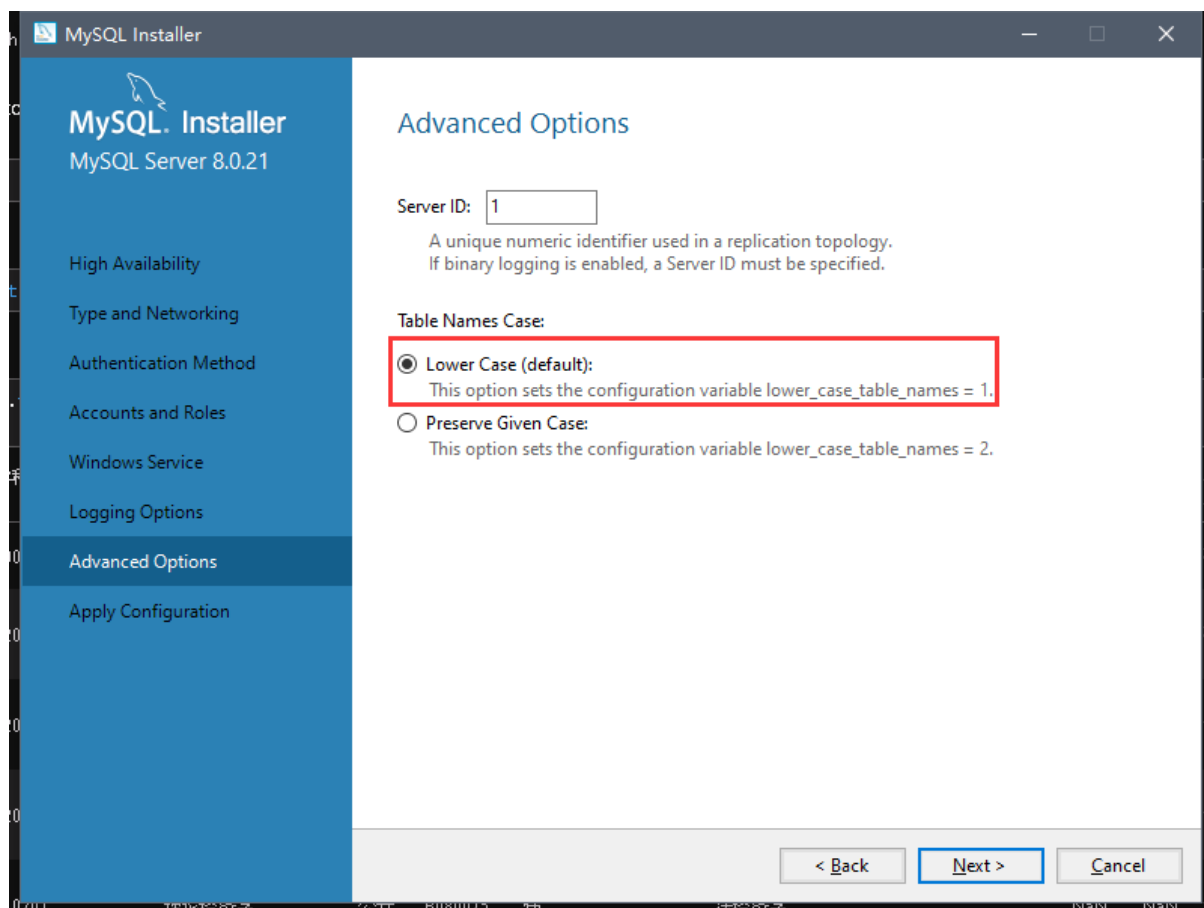
在这一步设置 MySQL 的 root 账户密码, 由于上一步选择了第二个选项, 因此这里可以设置为较简单容易记忆的而密码, 例如"123456". 建议设置比较简单的密码, 并将密码记录下来以防遗忘, 忘记密码是一件麻烦事。



此处保持默认即可, 如果 windows service name 右侧有黄色警告图标显示, 表示名称重复, 手动更换一个名称即可, 然后点击 Next:

Logging Options 这里使用默认设置即可, 我们的学习中暂时用不到这些设置, 直接点击 Next:

下图是设置是否大小写敏感的. 这一步非常重要, 由于windows系统是大小写不敏感的, 请大家务必使用第一个选项Lower Case。



点击 Execute

完成安装后, 在下图中点击 Finish 回到安装的主进程:

现在, 你的电脑上就已经安装了MySQL的服务器软件、用于连接服务器进行查询的MySQL Workbench, 以及其他程序语言连接MySQL的驱动, 此外还安装了几个示例数据库, 但本教程将采用《SQL基础教程》一书中的示例数据库, 该数据库的创建和数据导入将在后续进行介绍。

1.2 macOS 下下载安装 (略)

1.3 Linux 下 MySQL 8.0 的下载安装

本节我们以 centos 版本的 Linux 操作系统为例, 介绍下载安装 MySQL8.0.21 的过程。
示例操作系统的linux 版本: centos-release-7-7.1908.0.el7.centos.x86_64

在 CentOS 系统中默认是安装了 MariaDB 的, 但是我们需要的是 MySQL, 我们可以直接下载安装 MySQL, 安装 MySQL 可以覆盖 MariaDB。

关于 MariaDB:

MariaDB 数据库管理系统是 MySQL 的一个分支, 主要由开源社区在维护, 采用 GPL 授权许可. 开发这个分支的原因之一是: 甲骨文公司收购了 MySQL 后, 有将 MySQL 闭源的潜在风险, 因此社区采用分支的方式来避开这个风险. MariaDB 的目的是完全兼容 MySQL, 包括 API 和命令行, 使之能轻松成为 MySQL 的替代品. 但在两个分支经过了几年的各自迭代之后, 在一些方面二者出现了一些差异。

1.3.1 安装步骤:

首先, 从[MySQL官网](https://dev.mysql.com/get/mysql80-community-release-el7-2.noarch.rpm)下载 MySQL 的 Yum Repository. 根据 CentOS 和 MySQL 的版本, 选择下载相对应的文件。本文选择红色方框的文件。

Yum 是一个基于 rpm 的软件包管理器, 它可以从指定的服务器自动下载 RPM 包并且安装, 可以自动处理依赖性关系, 并且一次安装所有依赖的软体包, 无须繁琐地一次次下载、安装。

下载命令:

```
wget <https://dev.mysql.com/get/mysql80-community-release-el7-2.noarch.rpm>
```

用 yum 命令安装下载好的 rpm 包.

```
yum -y install mysql80-community-release-el7-2.noarch.rpm
```

安装 MySQL 服务器.

```
yum -y install mysql-community-server
```

这一步由于要下载安装文件, 可能会花一些时间. 安装完成后就会覆盖掉之前的 mariadb。

当出现如下图所示的内容, 则代表 MySQL 就安装完成了。



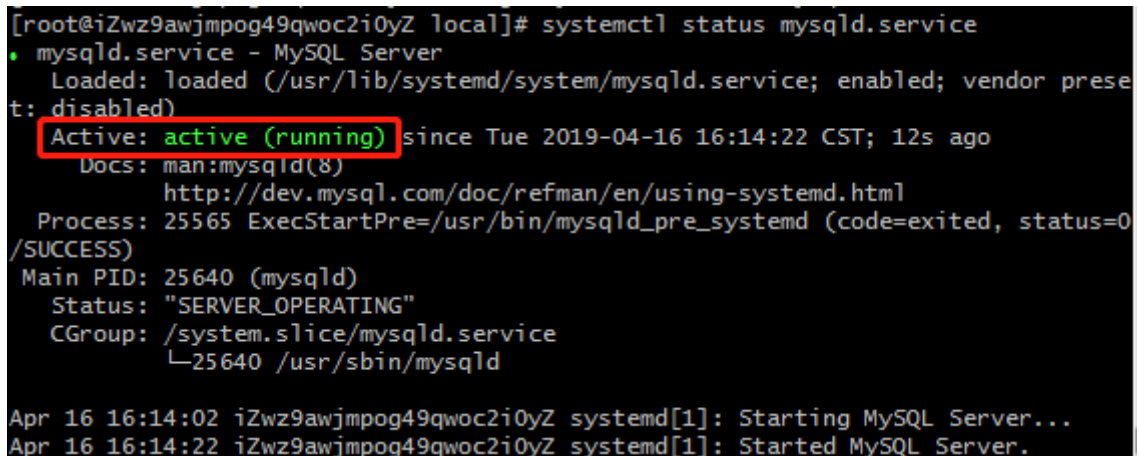
1.3.2 MySQL 数据库设置

启动 MySQL

```
systemctl start mysqld.service
```

查看 MySQL 运行状态, Active 后面的状态代表启动服务后为 active (running), 停止后为 inactive (dead), 运行状态如图：

```
systemctl status mysqld.service
```



```
[root@iZwz9awjmpog49qwoc2i0yZ local]# systemctl status mysqld.service
• mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor prese
t: disabled)
   Active: active (running) since Tue 2019-04-16 16:14:22 CST; 12s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Process: 25565 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0
/SUCCESS)
   Main PID: 25640 (mysqld)
      Status: "SERVER_OPERATING"
     CGroup: /system.slice/mysqld.service
            └─25640 /usr/sbin/mysqld

Apr 16 16:14:02 iZwz9awjmpog49qwoc2i0yZ systemd[1]: Starting MySQL Server...
Apr 16 16:14:22 iZwz9awjmpog49qwoc2i0yZ systemd[1]: Started MySQL Server.
```

重新启动 Mysql 和停止 Mysql 的命令如下：

```
service mysqld restart #重新启动 Mysql
systemctl stop mysqld.service #停止 Mysql
```

此时 MySQL 已经开始正常运行, 不过要想进入 MySQL 还得先找出此时 root 用户的密码, 通过如下命令可以在日志文件中找出密码：

为了加强安全性, MySQL8.0 为 root 用户随机生成了一个密码, 在 error log 中, 关于 error log 的位置, 如果安装的是 RPM 包, 则默认是/var/log/mysqld.log. 只有启动过一次 mysql 才可以查看临时密码

使用命令:

```
grep 'temporary password' /var/log/mysqld.log
```

查看初始的随机密码:

```
[root@iZwz9awjmpog49qwoc2i0yZ ~]# grep "password" /var/log/mysqld.log
2019-04-16T08:14:18.662081Z 5 [Note] [MY-010454] [Server] A temporary password i
s generated for root@localhost: GKKfdw+Dw4kq
```

登录 root 用户

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

然后输入上述查到的初始密码. 登录后还不能做任何查询或建库操作, 因为 MySQL 默认必须修改初始的随机密码之后才能操作数据库, 否则会报错:

修改密码为"123456", 注意结尾要有分号, 表示语句的结束。

```
ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';
```

这里有个问题, 新密码设置的时候如果设置的过于简单会报错:

原因是因为 MySQL 有密码设置的规范, 具体是与 `validate_password_policy` 的值有关:

MySQL 完整的初始密码规则可以通过如下命令查看:

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| validate_password.check_user_name | ON |
| validate_password.dictionary_file | |
| validate_password.length | 8 |
| validate_password.mixed_case_count | 1 |
| validate_password.number_count | 1 |
| validate_password.policy | MEDIUM |
| validate_password.special_char_count | 1 |
+-----+-----+
```

密码的长度是由 `validate_password_length` 决定的, 而 `validate_password_length` 的计算公式是:

```
validate_password_length = validate_password_number_count + validate_password_special_
char_count + (2 * validate_password_mixed_case_count)
```

如果想要设置简单的密码必须要修改约束, 修改两个全局参数:

- **validate_password_policy**代表密码策略, 默认是 1: 符合长度, 且必须含有数字, 小写或大写字母, 特殊字符. 设置为 0 判断密码的标准就基于密码的长度了. 一定要先修改两个参数再修改密码

```
mysql> set global validate_password.policy=0;
```

- **validate_password_length**代表密码长度, 最小值为 4

```
mysql> set global validate_password.length=4;
```

修改完,如图

```
mysql> SHOW VARIABLES LIKE 'validate_password%';
```

Variable_name	Value
validate_password.check_user_name	ON
validate_password.dictionary_file	
validate_password.length	4
validate_password.mixed_case_count	1
validate_password.number_count	1
validate_password.policy	LOW
validate_password.special_char_count	1

此时密码就可以设置的很简单, 例如 1234 之类的. 到此数据库的密码设置就完成了。

但此时还有一个问题, 就是因为安装了 Yum Repository, 以后每次 yum 操作都会自动更新, 如不需要更新, 可以把这个 yum 卸载掉:

```
[root@localhost ~]# yum -y remove mysql80-community-release-el7-2.noarch
```

在 CentOS 中 MySQL 的主要配置所在的目录:

- 1 /etc/my.cnf 这是 mysql 的主配置文件
- 2 /var/lib/mysql mysql 数据库的数据库文件存放位置
- 3 /var/log/mysql 数据库的日志输出存放位置

一些可能会用到的设置:

设置表名为大小写不敏感:

1.用root帐号登录后, 使用命令

```
systemctl stop mysqld.service
```

停止MySQL数据库服务, 修改vi /etc/my.cnf, 在[mysqld]下面添加

```
lower_case_table_names=1
```

这里的参数 0 表示区分大小写，1 表示不区分大小写。

2.做好数据备份，然后使用下述命令删除数据库数据(删除后，数据库将恢复到刚安装的状态)

```
rm -rf /var/lib/mysql
```

3.重启数据库 ,此时数据库恢复到初始状态。

```
service mysql start
```

4.重复安装时的操作, 查找临时密码

```
grep 'temporary password' /var/log/mysqld.log
```

5.修改密码. 密码8位以上, 大小写符号数据. 如想要使用简单密码, 需按照上述安装流程中的步骤进行操作。

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '*****';update user set host = "%" where user='root';
```

6.刷新MySQL的系统权限相关表

```
FLUSH PRIVILEGES;
```

此时, MySQL的表名的大小写不再敏感。

1.3.3 设置远程连接:

如果你想要在另外一台电脑上连接 centos 上安装的 MySQL, 那么还需要一些其他的设置。

首先需要将 MySQL 设置为可以远程连接，设置 mysql 库的 user 表中帐号的 host 为%，%表示可以从任意 ip 连接 mysql, 为了安全性, 也可以设置为你自己所用于连接 centos 上 MySQL 的电脑所使用的 IP。

其次, MYSQL 8.0 内新增加 mysql_native_password 函数, 通过更改这个函数密码来进行远程连接。

例如, 更改 root 用户的 native_password 密码

```
ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY'MyPass@123';
```

接下来为 centos 的防火墙开启 MySQL 所使用的 3306 端口,并重新加载防火墙:

```
firewall-cmd --zone=public --add-port=3306/tcp --permanent  
firewall-cmd --reload
```

完成上述设置后, 重新启动 MySQL 服务:

```
service mysqld restart #重新启动 Mysql
```

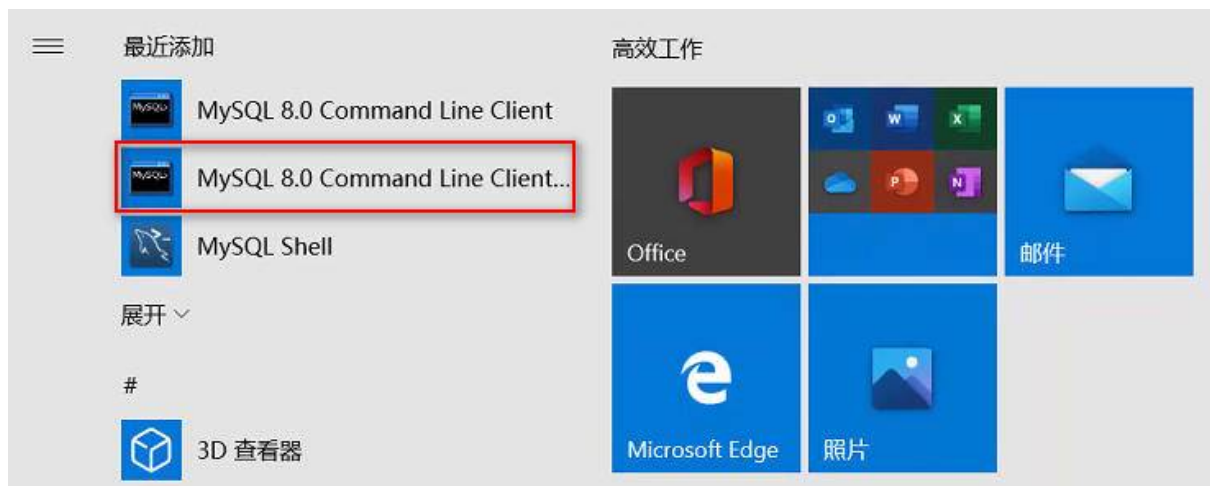
最后, 在另外一台电脑上, 使用下一节介绍的各类客户端工具进行连接测试。

2. 连接 MySQL 并执行 SQL 查询

在安装成功MySQL后, 我们可以通过开始菜单->控制面板->管理工具->服务中查找并开启或关闭MySQL服务. 开启服务后, MySQL Server将以后台服务的形式在你的电脑上运行. 如果想要进行查询, 还需要使用命令行工具或者其他更加友好的应用程序连接到MySQL服务。

2.0 使用命令行方式连接MySQL服务

连接数据库的最基本的方式, 是使用命令行. 以win10为例, 在完成 MySQL 服务的安装后, 在开始菜单找到刚才安装好的 MySQL 8.0 命令行工具,单击即可打开:



输入你在安装过程中为root用户设置的密码即可连接到数据库, 看到如下界面, 表示数据库安装成功,并且你已经使用命令行连接到了数据库(图中红框表示数据库版本), 在最后的 "mysql>" 后即可输入SQL命令执行各种数据库操作。

但是使用命令行在做数据查询时, 在查询结果的显示和导出方面有诸多不便, 特别是当我们对 SQL 查询不熟练的时候, 这种方式不利于我们进行查询语句的修改和调试. 因此本教程将选择查询界面更加友好的客户端工具(使用MySQL Workbench)来连接数据库, 这种通过命令行连接的方式暂时不再使用。

常见的可用于连接MySQL数据库并进行查询等操作的管理工具包括开源软件 MySQL Workbench, HeidiSQL和DBeaver, 以及商业软件Navicat(有免费版和14天试用版),SQLyog(有免费的社区版)和DataGrip等等。

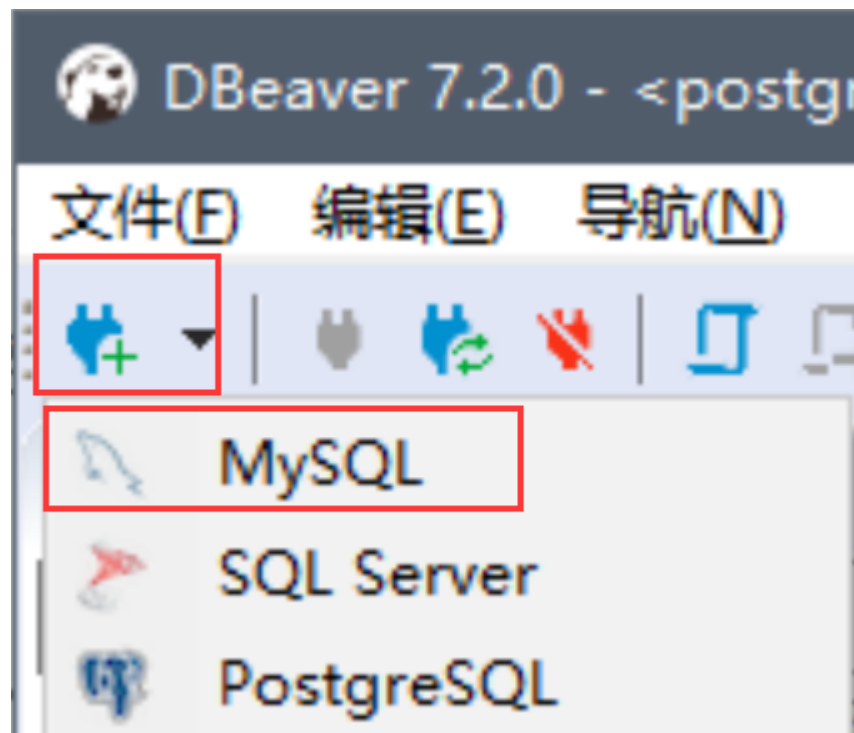
2.1 [选学]使用 MySQL Workbench 连接 MySQL

2.2 [选学]使用 HeidiSQL 连接 MySQL

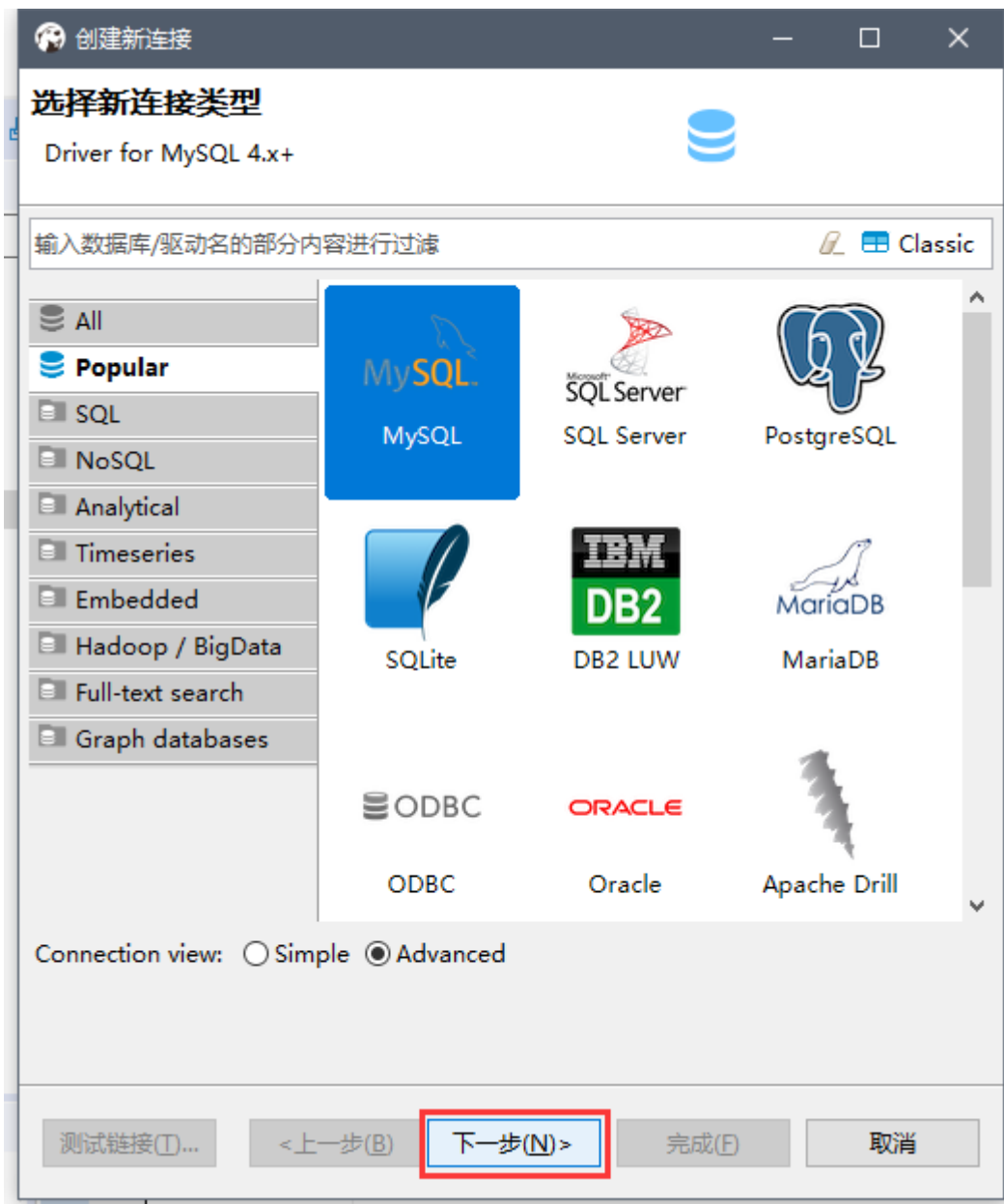
2.3 [选学]使用 DBeaver 连接 MySQL

DBeaver 是一款基于 Java 开发的 **免费和开源(GPL)** 的通用数据库管理工具和 SQL 客户端, 提供windows, macOS和Linux全平台支持, 能够连接包括 MySQL, PostgreSQL, Oracle, DB2, MSSQL, Sybase, Mimer, HSQLDB, Derby 等主流数据库软件在内的绝大多数兼容 JDBC 驱动的数据库. DBeaver 提供一个图形界面用来查看数据库结构、执行 SQL 查询和脚本, 浏览和导出数据, 处理 BLOB/CLOB 数据, 修改数据库结构等等。

由于是开源软件, 大家可直接从官网(<https://dbeaver.io/>)下载, 安装完成后, 打开软件, 点击"文件"菜单下的"新建连接"图标, 并选择 MySQL:



然后点击下一步:



在下方弹出的设置窗口中, "服务器地址"使用默认的localhost, "数据库"暂时留空, 然后在下方填入自己设置的密码, 最后点击测试连接。

创建新连接

连接设置

MySQL 连接设置

常规 驱动属性 SSH Proxy SSL

Server

服务器地址: localhost 端口: 3306

数据库:

认证 (Database Native)

用户名: root

密码: [masked] ☒ Save password locally

Advanced

服务器时区: 自动检测

本地客户端: MySQL Router 8.0

① 可以在连接参数中使用变量。 连接详情(名称、类型 ...)

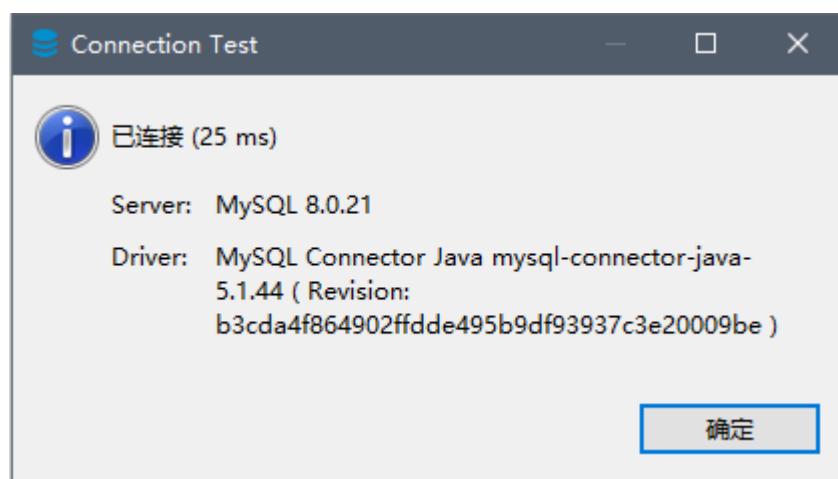
驱动名称: MySQL 编辑驱动设置

测试链接(T)... <上一步(B) 下一步(N)> 完成(F) 取消

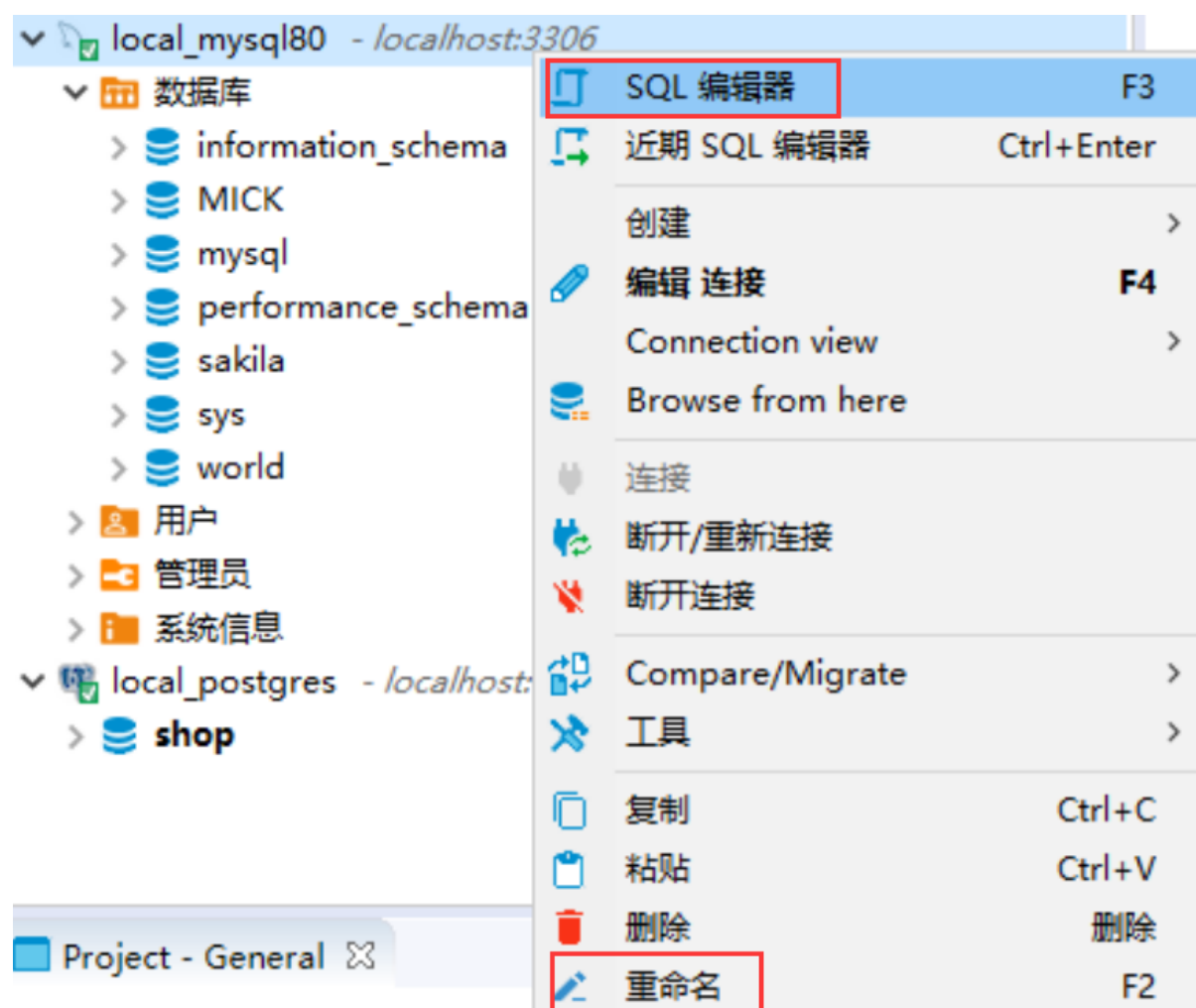
本次学习主要使用一个新建的 shop 数据库, 但因为相关数据我们还没导入(建库建表及数据导入的脚本在本章第三节), 因此数据库这里先留空. 等下一步导入数据后, 再进一步修改连接参数。

首次连接时会提示需要下载驱动程序,

完成驱动程序的下载后, 再次点击"测试连接", 如果弹出如下对话框, 则表示连接成功:

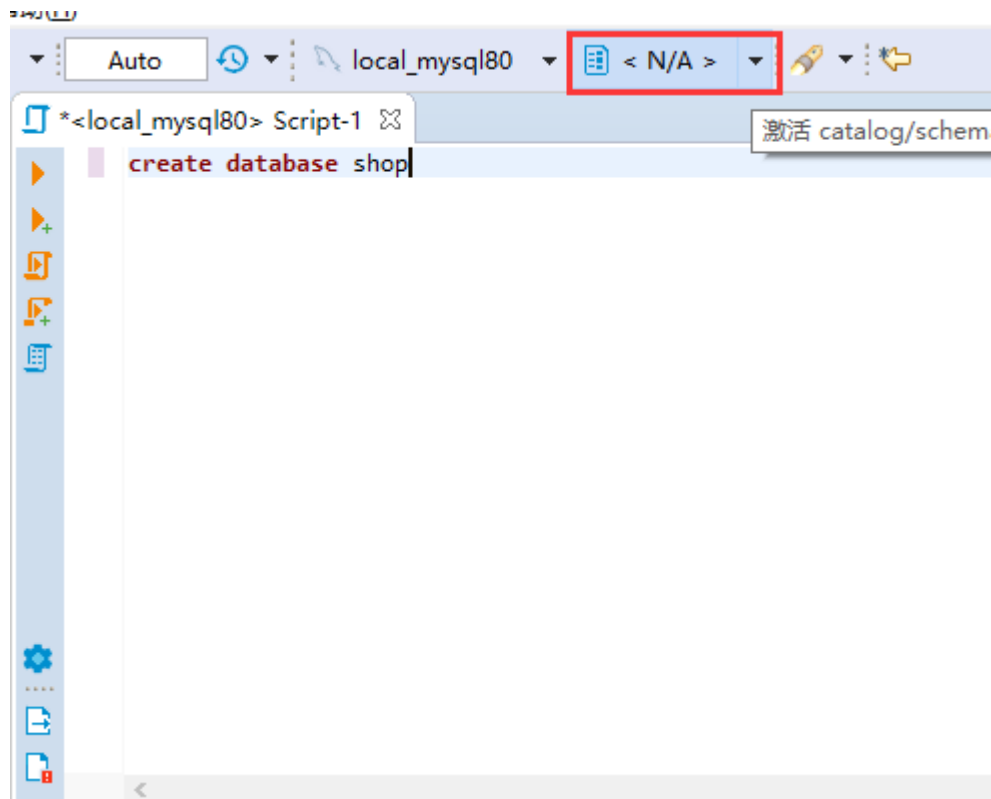


点击确定并保存连接后, 以后就可以直接双击这个连接进行数据库连接了. 如果需要保存多个数据库连接, 可以使用快捷键 F2 或选中当前连接并点击鼠标右键后选择"重命名"为当前连接起一个便于识别的名称。

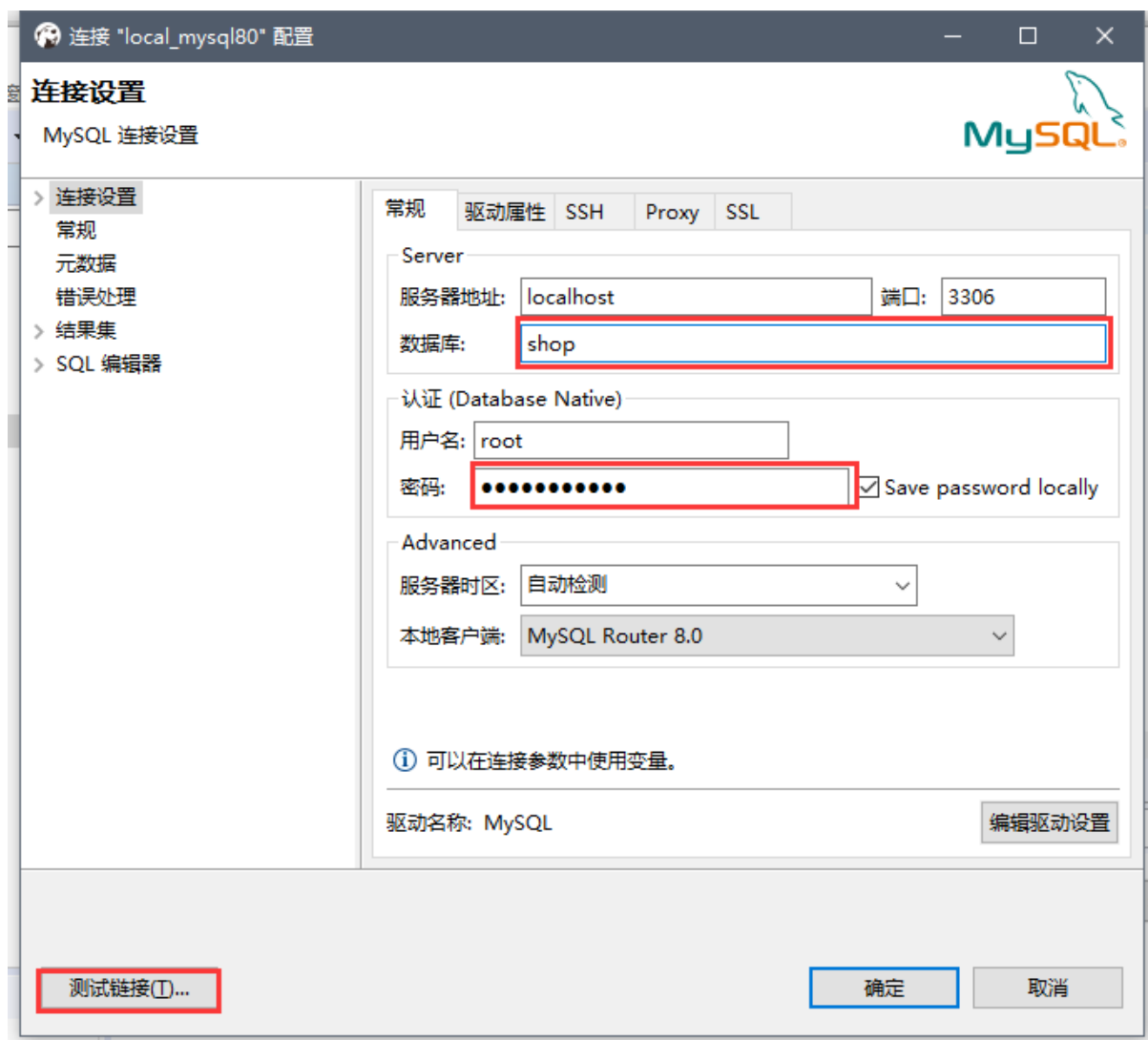


接下来就可以开始写你的第一个 SQL 语句了: 如上图这样,选中刚刚保存的连接, 鼠标右键选第一个按钮"SQL 编辑器", 或者直接使用快捷键 F3, 就会打开一个 SQL 编辑器, 然后就可以在这里编写 SQL 语句了。

例如我们可以使用如下语句完成本教程所使用的示例数据库的创建:



注意, 在使用创建数据库的语句时, 是无需在红框中选中数据库的, 但其他所有的创建表,导入数据, 和最常用的查询语句, 都需要选中相应的数据库. 在执行上述语句后, 我们可以回到连接设置里, 把默认数据库连接改为刚才创建的 shop。



2.4 [选学]使用 Navicat 连接 MySQL

2.5 [选学]使用 SQLyog 连接 MySQL

2.6 [选学]DataGrip的安装和连接MySQL

3.1 创建学习用的数据库

根据《SQL基础教程》提供的MySQL版本的数据库,数据表的创建以及数据导入的代码,经过一些修改,创建了一份 sql 脚本,运行该脚本可以一步到位地创建本文档所需的数据库shop及其中所有的表,并插入本教程所需要的所有数据。

下述SQL脚本可用于创建本教程所使用的示例数据库 `shop` 以及数据库中表的创建和数据的插入。

详见 `./materials/shop.sql`

存储引擎使用 InnoDB, 字符集改为 utf8mb4 以更好地支持中文。所有表名所使用的英文字母都改为小写(后续章节中,SQL 查询中的表名也需要相应修改为小写)所有列名所使用的英文字母确认为小写(后续章节中,SQL 查询中的列名也需要相应修改为小写)存在问题的数据, 例如 inventoryproduct 表的 inventory_id 列应为 P 开头的, 已修正为正确的数据。需测试 SQL 脚本在命令行及各个客户端中是否能被正确执行.

- MySQL Workbench 已测试通过(在win10 使用 MySQL Workbench 8.0.21)
- DBeaver 已测试通过(使用"执行 SQL 脚本(CTR+x)") (在 win10 使用 DBeaver7.2.0)
- HeidiSQL 已测试通过(在win10 使用HeidiSQL 11.0.0)
- navicat 已测试通过(在win10&win7 使用 navicat premium 15.0.17)
- sqlyog 已测试通过(在win10 使用 SQLyog 12.09)
- 命令行 win10 下测试未通过. 插入中文数据时提示" Data too long for column 'product_name' at row 1", 建议使用客户端进行数据库的创建和数据导入。