

实验1 实验指导 DBMS的安装和使用

一、实验目的

1. 通过安装数据库管理系统，初步了解DBMS的运行环境。
2. 了解并初步掌握DBMS交互界面、图形界面(命令行界面)和系统管理工具的使用。
3. 搭建本地实验平台。

二、实验环境

Windows 10 Pro for Workstations

数据库管理系统： MySQL

三、实验流程

1.根据某个DBMS的安装说明等文档，安装DBMS。

本人已经安装过mysql-8.0.28版本，如下图所示

```
C:\Users\MSI-NB>mysqlsh --version
mysqlsh    Ver 8.0.28 for Win64 on x86_64 - for MySQL 8.0.28 (MySQL Community Server (GPL))
```

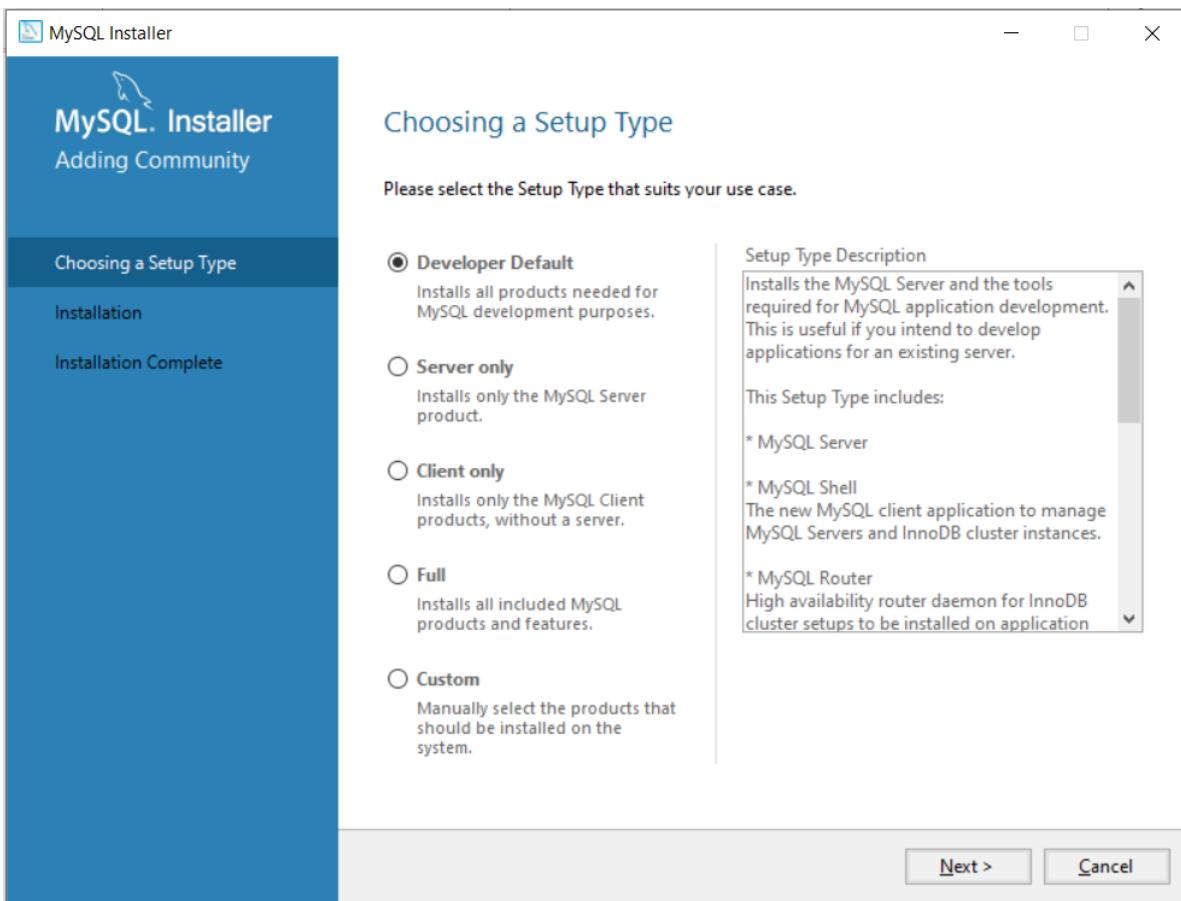
因此本文只简单介绍mysql非web版的安装过程

版本选择上我们选择社区版 (community version)，打开下方的下载页面

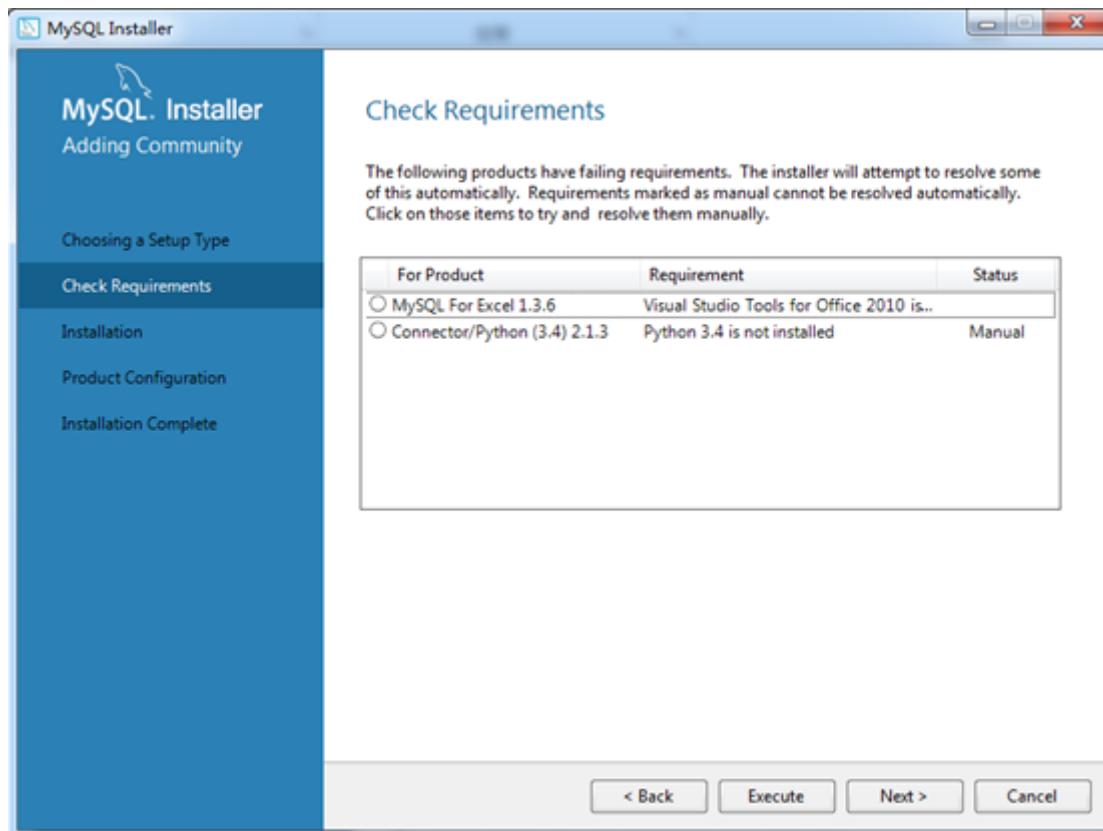
<https://downloads.mysql.com/archives/installer/>

根据需求选择我们需要的版本以及操作系统，这里我们选择8.0.19以及Microsoft Windows，选择下方的(mysql-installer-community-8.0.28.0.msi)下载即可。

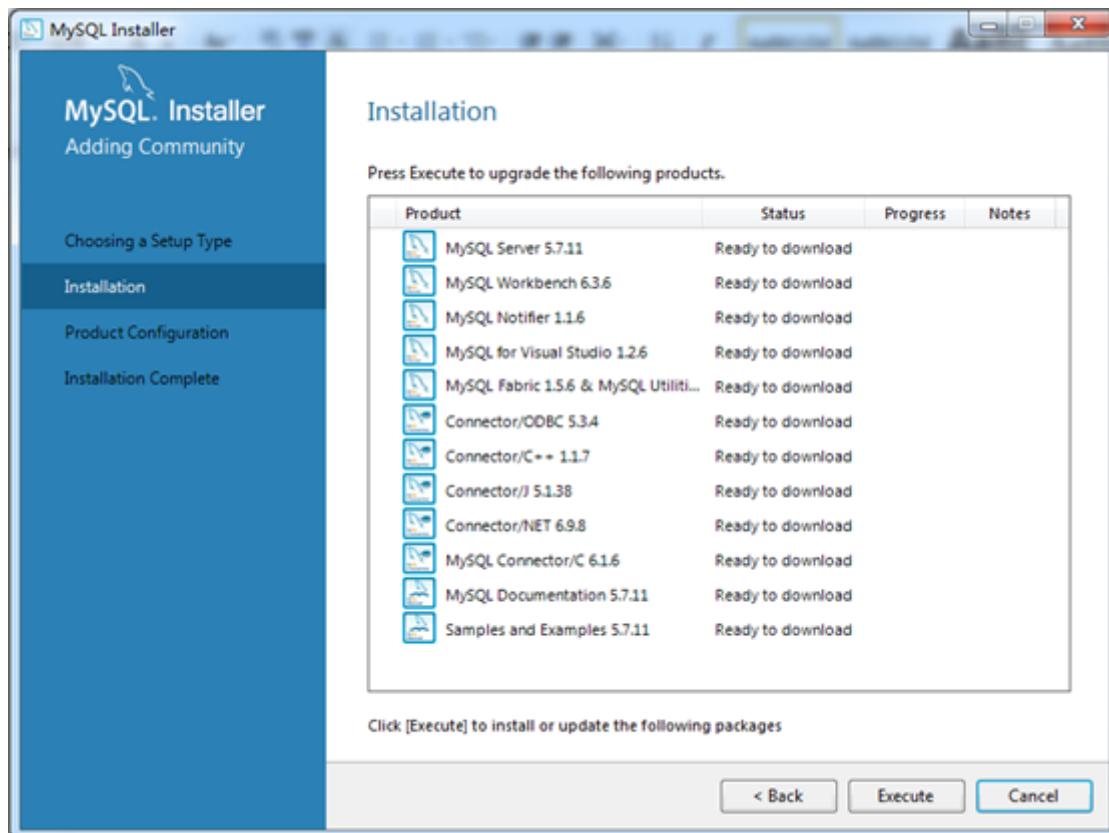
之后我们双击打开msi安装包



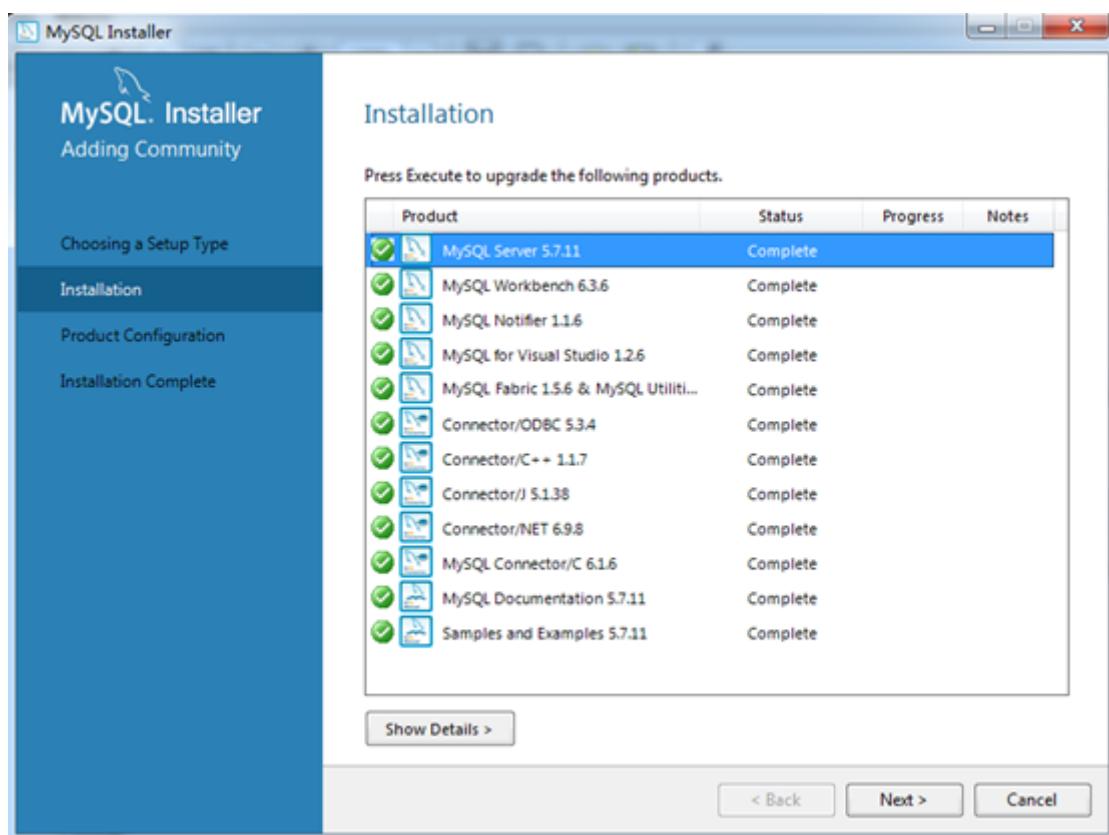
本人因已经安装过mysql，因此不再继续安装下去，这里点击下一步。



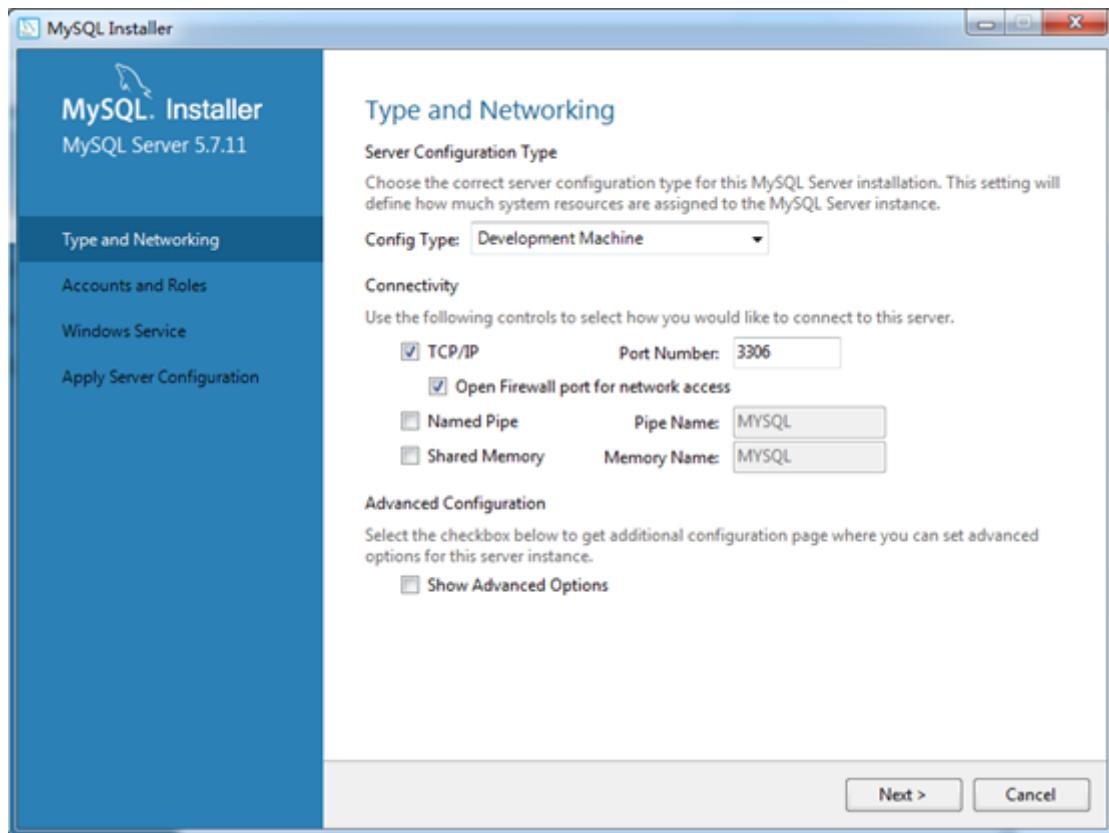
点击Execute



下载安装完毕后，点击next

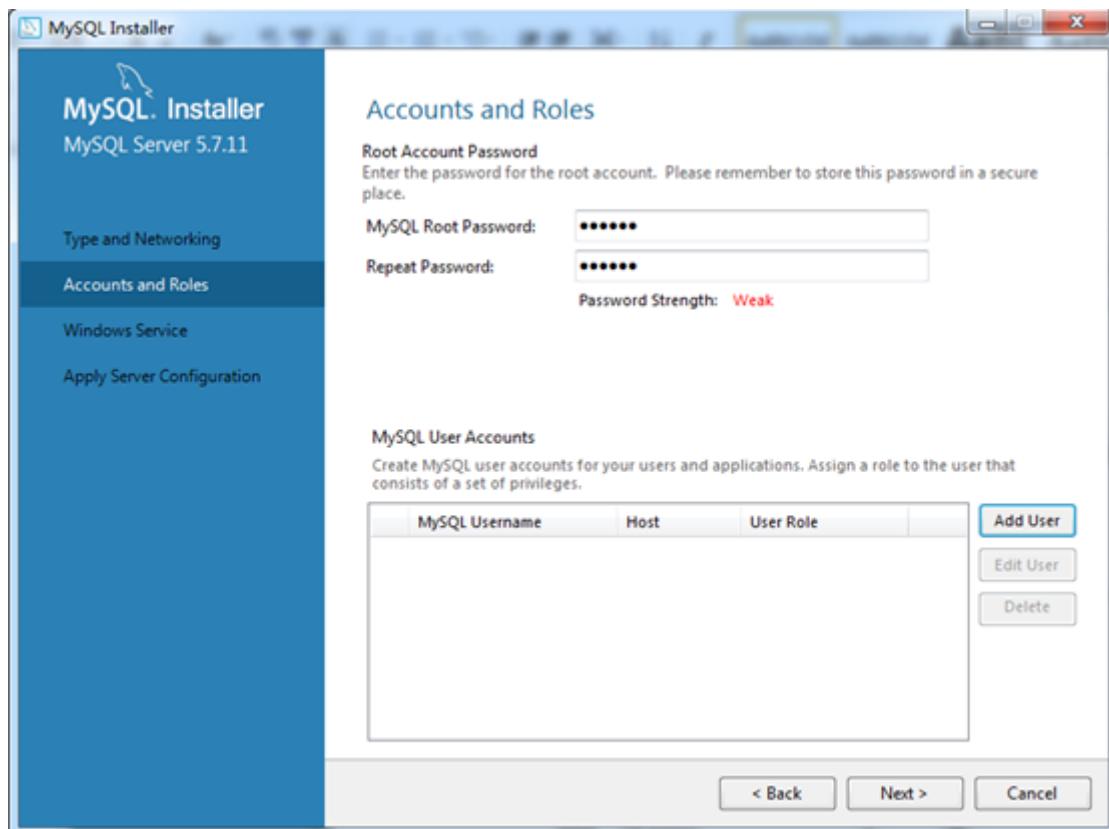


选择默认的网络连接

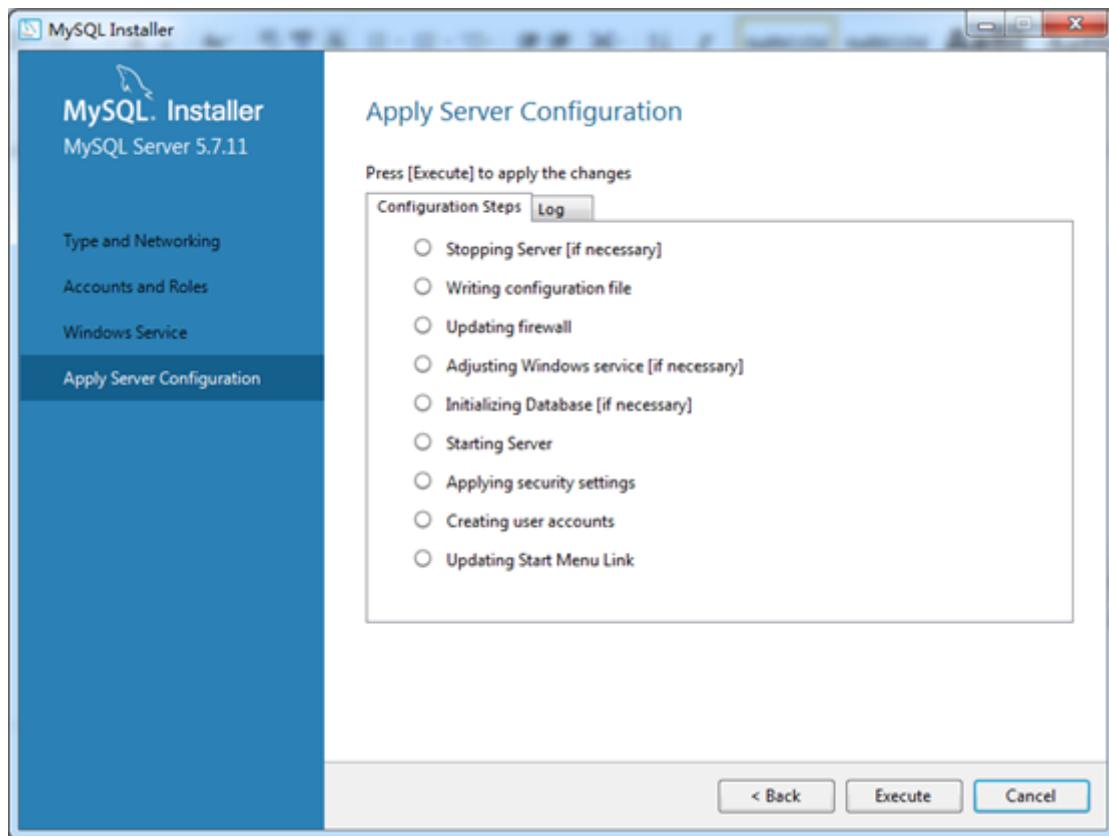


这里需要注意的是，mysql默认端口为3306，如无特殊情况请不要修改！

设置数据库管理员密码，然后点击Add User，添加当前用户作为数据库用户



点击Execute，执行数据库服务器配置



之后我们在命令行中输入

```
1 | mysql -uroot -p
2 | Password: root password
```

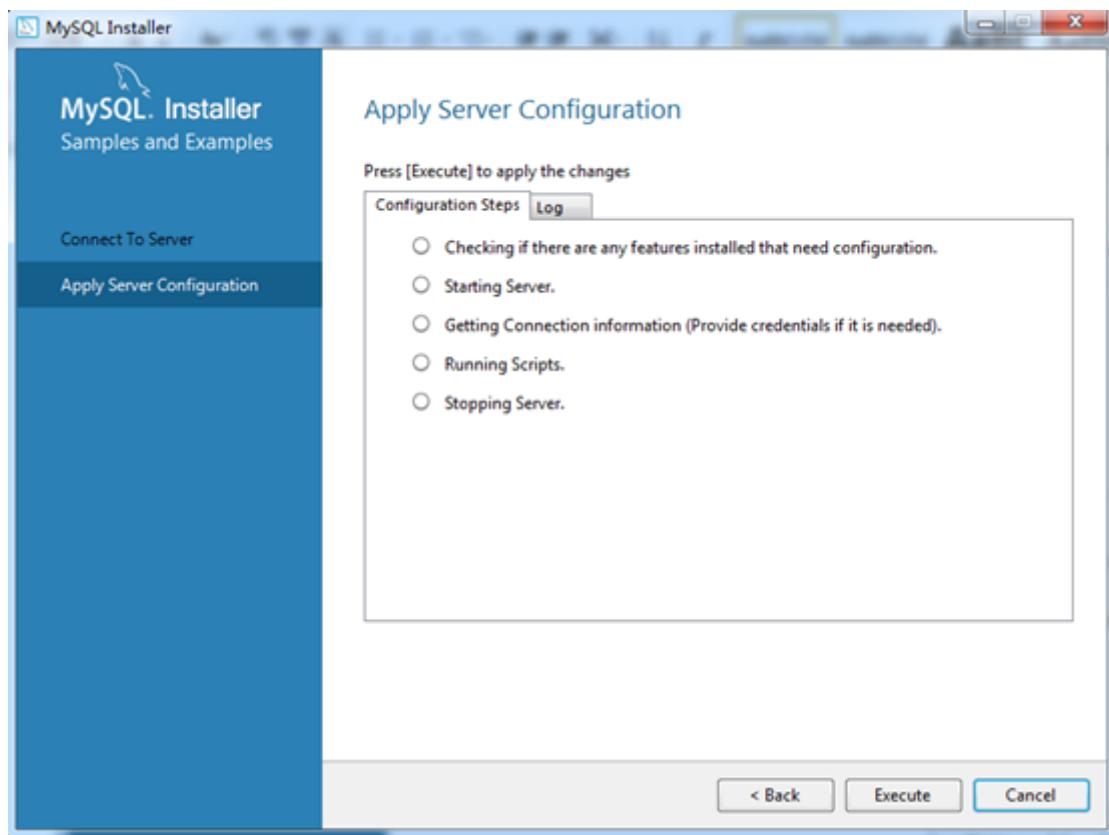
```
Files\Common was unexpected at this time.
E:\cmder
λ mysqlsh -u root -p
MySQL Shell 8.0.28

Copyright (c) 2016, 2022, 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.

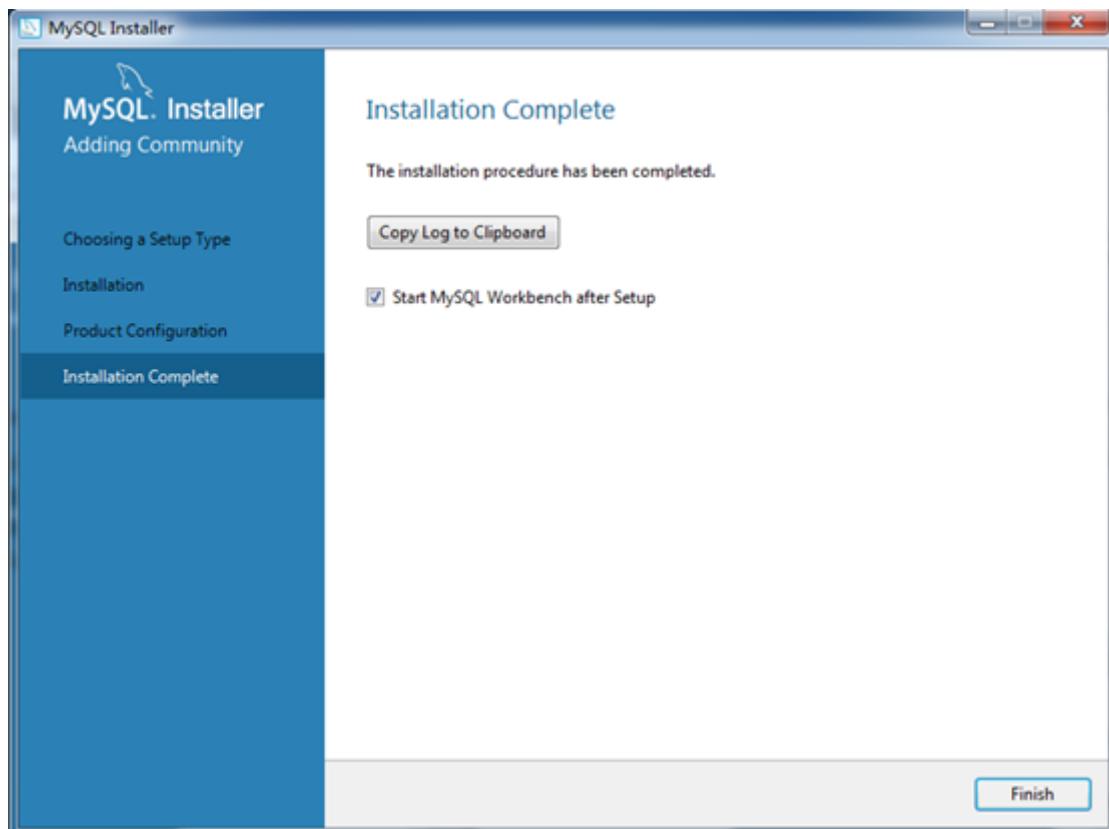
Type '\help' or '\?' for help; '\quit' to exit.
Creating a session to 'root@localhost'
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 32 (X protocol)
Server version: 8.0.28 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL localhost:33060+ ssl JS > \sql
Switching to SQL mode... Commands end with ;
MySQL localhost:33060+ ssl SQL > https://cmder.net/|
```

出现如图所示即为连接成功

点击Execute，执行样本数据库配置

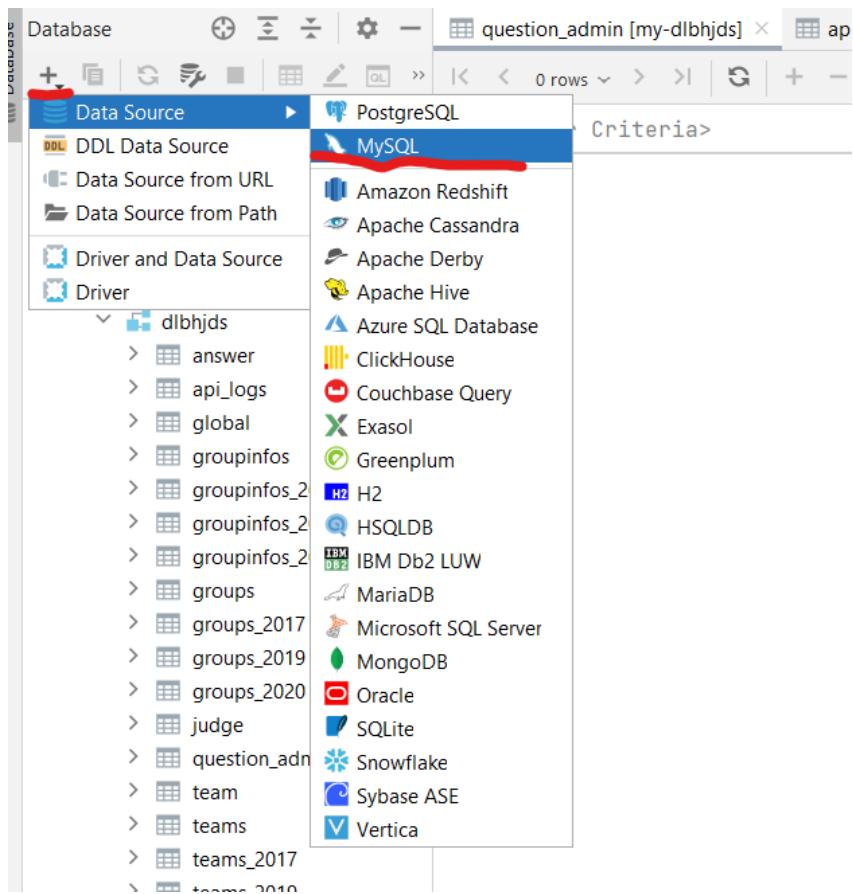


之后即可安装完毕

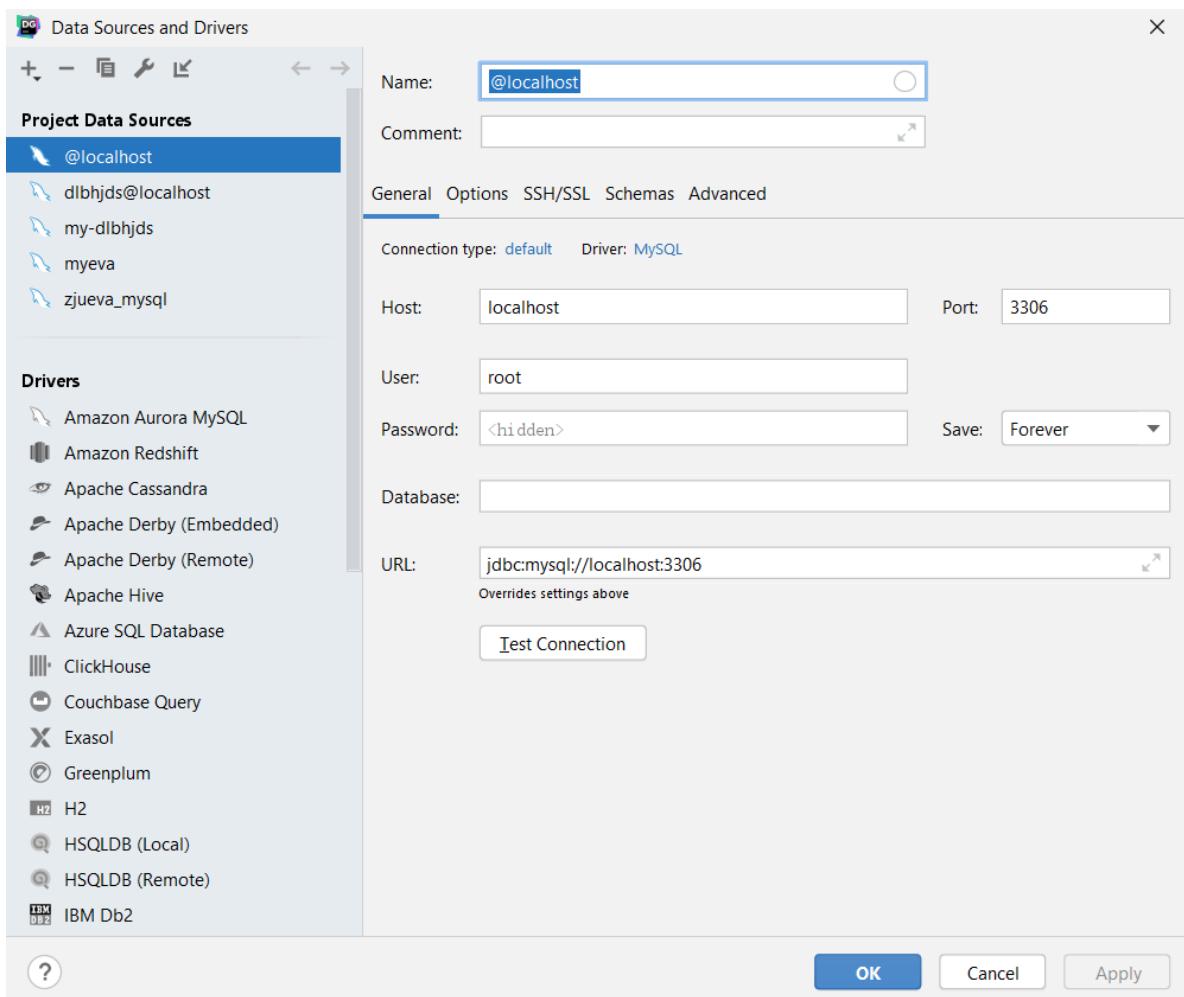


这里我们使用datagrip来进行数据库的可视化操作，本文采用的datagrip版本为2020.03，采用jetbrain教育用户激活

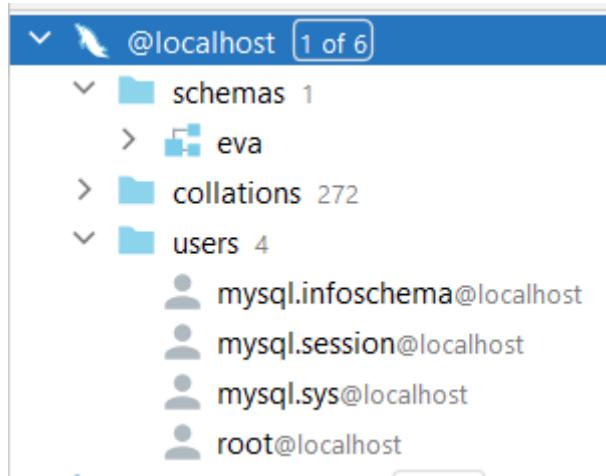
安装好datagrip后我们点击左侧的加号，选择Mysql



根据上面的配置选择合适的用户（本文采用root）以及相应的密码即可连接



可以看到左边的三个文件夹



证明我们连接成功！

2. 了解DBMS的用户管理。

mysql的用户信息全部保存在名为mysql的系统数据库中，主要用到user、db、tables_priv、columns_priv、procs_priv五张表，最重要的是user表。

查阅资料可知：

- user表存储全局权限，适用于一个给定服务器中的所有数据库，在命令中展现形式为.；
- db表存储数据库权限，适用于一个给定数据库中的所有表，在命令中展现形式为[数据库名].*；
- tables_priv表存储表权限，适用于一个给定表中的所有列，在命令中展现形式为[数据库名].[表名]；
- columns_priv表存储列权限，适用于一个给定表中的单一列，在命令中展现形式为；

这里我们首先查看mysql.user表

```
1 | select * from mysql.user;
```

可以看到user表内容如下

| Host | User | Select_priv | Insert_priv | Update_priv | Delete_priv | Create_priv | Drop_priv | Reload_priv | Shutdown_priv | Process_priv | File_priv | Grant |
|-------------|------------------|-------------|-------------|-------------|-------------|-------------|-----------|-------------|---------------|--------------|-----------|-------|
| 1 localhost | mysql.infoschema | Y | N | N | N | N | N | N | N | N | N | N |
| 2 localhost | mysql.session | N | N | N | N | N | N | Y | N | N | N | N |
| 3 localhost | mysql.sys | N | N | N | N | N | N | N | N | N | N | N |
| 4 localhost | root | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |

这里值得注意是host（作用域），一般情况下都是localhost（本地），也可以是网段如192.168.1.%或全网%，网段和全网是用于远程连接mysql的。

另一种展示用户权限的指令为:show grants for '[用户名]'@'[使用范围]'

```
1 | show grants for '[user]'@'[host]';
2 | //show grants for 'root'@'localhost';
```

| Grants for root@localhost |
|--|
| 1 GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, ... |
| 2 GRANT APPLICATION_PASSWORD_ADMIN,AUDIT_ADMIN,BACKUP_ADMIN,BINLOG_ADMIN,BINLOG_E... |
| 3 GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION |

创建用户操作：

我们查看validate_password_policy (密码复杂度)、validate_password_length (密码长度)、validate_password_number_count (密码中数字字符长度)、validate_password_special_char_count (密码中特殊符号字符长度)、validate_password_mixed_case_count (密码中大小写字母长度) 这五个参数，使用命令为

```
1 | select @@[参数名];
```

创建用户命令为

```
1 | create user '[新用户名]'@'[作用域]' identified by '[密码]';  
2 | flush privileges; //刷新权限表,类似重新加载  
3 |  
4 | //now we use  
5 | create user 'zyc'@'localhost' identified by 'zhangyunce';
```

之后我们使用show grants命令查看该用户

```
1 | show grants for 'zyc'@'localhost';
```

可以看到目前是没有任何权限的

授权的命令为

```
1 | grant [权限] on [数据库名].[表名] to '[用户名]'@'[作用域]' identified by '[密码]';
```

其中权限为ALL PRIVILEGES或ALL是所有权限，还有单个权限select、update、insert、delete等，单个权限之间用逗号隔开，详细可以查看下mysql.user表的表结构。

如果不存在identified by '[密码]'时，密码维持原样。

这里我们给zyc用户赋予所有权限

```
1 | grant all privileges on *.* to 'zyc'@'localhost';
```

这时候再查看其权限

而撤销权限的方法和授权的方法类似

```
1 | revoke [权限] on [数据库名].[表名] from '[用户名]'@'[作用域]';  
2 | flush privileges; //属性权限表
```

我们使用下面的命令

```
1 | revoke all privileges on *.* from 'zyc'@'localhost';
```

这时候再查看其权限

```
Grants for cxz@localhost
1 GRANT USAGE ON *.* TO `cxz`@`localhost`
```

可以看到只剩下初始权限了，证明我们移除权限成功

最后是删除用户的方法

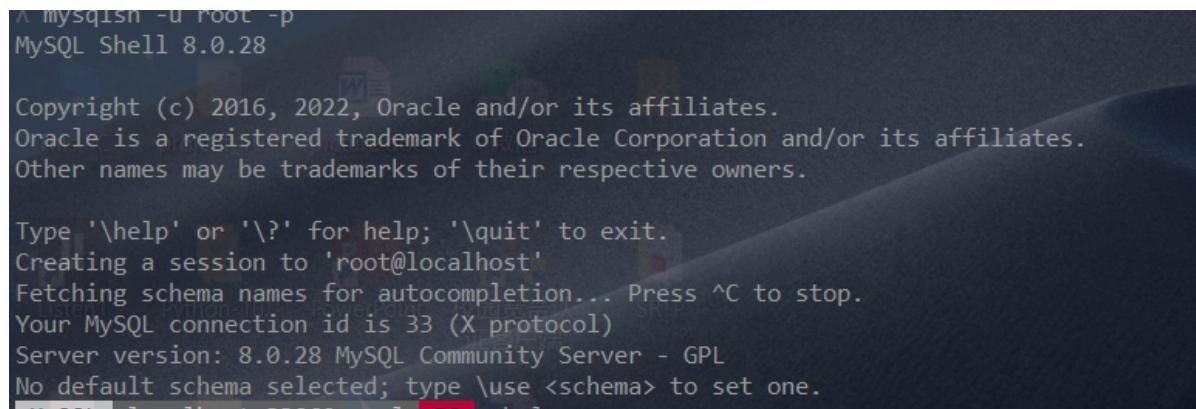
```
1 drop user '[用户名]'@'[作用域]';    //
2 flush privileges;    //刷新权限表!
```

以上就是mysql用户权限的基本操作，本段借鉴<https://www.cnblogs.com/diantong/p/11024294.html>并加以部分修改。

3.熟悉交互界面的基本交互命令

登录到mysql

```
1 mysql -uroot -p
2 passwd
```



mysqlsh -u root -p
MySQL Shell 8.0.28

Copyright (c) 2016, 2022, 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 '\?' for help; '\quit' to exit.
Creating a session to 'root@localhost'
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 33 (X protocol)
Server version: 8.0.28 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.

登录到mysql后，cli提示我们输入help查看交互命令

help命令还有很多神奇的有用处

```
1 help create
2 help show
3 help use
4 help select
5 ....
```

以上命令的截图由于篇幅原因不一一列举

```
MySQL [localhost:33060+ ssl JS] > \sql
Switching to SQL mode... Commands end with ;
MySQL [localhost:33060+ ssl SQL] > \help
The Shell Help is organized in categories and topics. To get help for a
specific category or topic use: \? <pattern>
2021 Zju Project3.zip lab2.docx
The <pattern> argument should be the name of a category or a topic.

The pattern is a filter to identify topics for which help is required, it can
use the following wildcards:
- ? matches any single character.
- * matches any character sequence.

The following are the main help categories:
- Shell Commands Provides details about the available built-in shell commands.
- SQL Syntax Entry point to retrieve syntax help on SQL statements.

The available topics include:
- The available shell commands.
- Any word that is part of an SQL statement.
- Command Line - invoking built-in shell functions without entering interactive
mode.

SHELL COMMANDS
The shell commands allow executing specific operations including updating the
shell configuration.

The following shell commands are available:
- \
- \connect (\c) Start multi-line input when in SQL mode.
- \disconnect Connects the shell to a MySQL server and assigns the
global session.
- \edit (\e) Disconnects the global session.
- \edit (\e) Launch a system editor to edit a command to be executed.
```

查看mysql版本

```
1 | mysql --version
```

查看当前所有的数据库

```
1 | show databases;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| eva      |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)
```

进入数据库

```
1 | use 数据库名;
```

```
mysql> use eva
Database changed
```

查看当前数据库所有的表

```
1 | show tables;
```

```
mysql> show tables;
+-----+
| Tables_in_eva |
+-----+
| __efmigrationshistory
| accessories
| bills
| billtranscationrecords
| dbconfigs
| newses
| notes
| tickets
| ticketworkers
| users
| wechatconfigs
| wechatusers
+-----+
```

查看某个表的字段结构

```
1 | desc 表名;
```

```
mysql> desc tickets
-> ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Id    | int  | NO   | PRI  | NULL   | auto_increment |
| Type  | int  | NO   |      | NULL   |
| Device | varchar(50) | NO   |      | NULL   |
| DeviceModel | varchar(50) | NO   |      | NULL   |
| Owner  | varchar(20) | NO   |      | NULL   |
| Phone  | varchar(20) | NO   |      | NULL   |
| Description | varchar(20) | NO   |      | NULL   |
| CreatedTime | datetime(6) | NO   |      | NULL   |
| Status  | int  | NO   |      | NULL   |
| IsConfirmed | tinyint(1) | NO   |      | NULL   |
| WeChatConfigId | int  | NO   |      | NULL   |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.01 sec)
```

查询表数据

```
1 | select * from 表;
```

```
mysql> select * from dbconfigs
      -> ;
+----+-----+-----+
| Id | Key            | Value          |
+----+-----+-----+
| 1  | MIGRATION_DATE | 02/07/2021 17:49:35 |
| 2  | APP_VERSION     | 2              |
| 3  | BILL_STATS_BASIC_COUNT | 0|0
| 4  | BILL_STATS_BASIC_AMOUNT | 0|0|0
+----+-----+-----+
4 rows in set (0.02 sec)
```

创建数据库

```
1 | create database name;
```

```
mysql> create database test;
Query OK, 1 row affected (0.02 sec)
```

创建数据表

```
1 //创建操作非常复杂，这里不再一一列举
2 create table user (
3   -> id int(8) not null auto_increment,
4   -> name char(30),
5   -> primary key(id)
6   -> );
```

```
mysql> create table user (
      -> id int(8) not null auto_increment,
      -> name char(30),
      -> primary key(id)
      -> );
Query OK, 0 rows affected, 1 warning (0.07 sec)
```

添加数据

```
1 | insert into 表名 values(value,value);
```

```
mysql> insert into user values(1,"cxz");
Query OK, 1 row affected (0.02 sec)
```

```
mysql> insert into user values(2,"syf");
Query OK, 1 row affected (0.01 sec)
```

看起来也很美观

修改数据

```
1 | update 表 set name=value where xxxx;
```

删除列

```
1 | alter table 表名 drop 列名称;
```

```
mysql> alter table user drop name;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

删除数据表

```
1 | drop table 表名
```

```
mysql> drop table user
      -> ;
Query OK, 0 rows affected (0.03 sec)
```

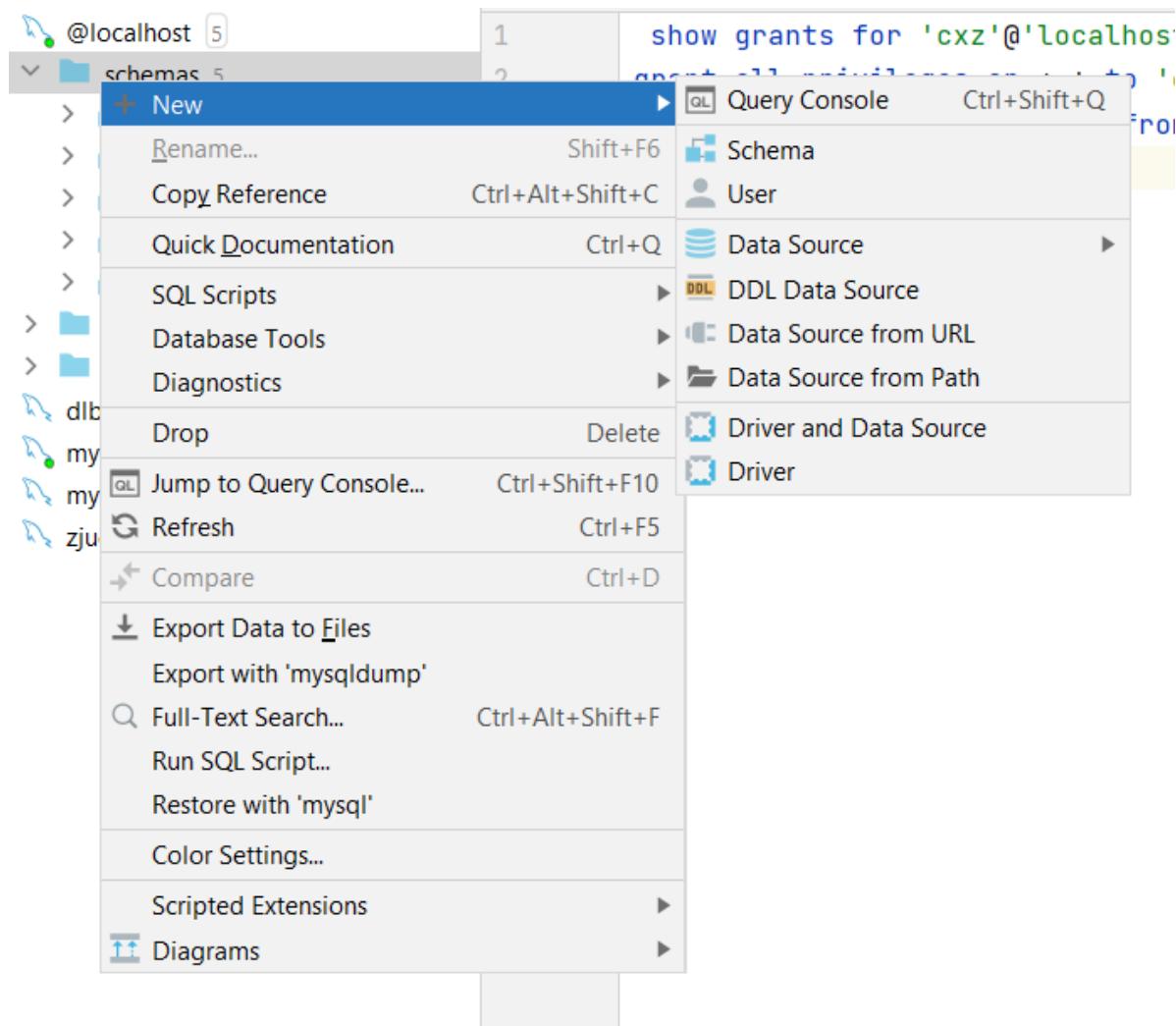
删除数据库

```
1 | drop database 库名;
```

```
mysql> drop database test;
Query OK, 0 rows affected (0.01 sec)
```

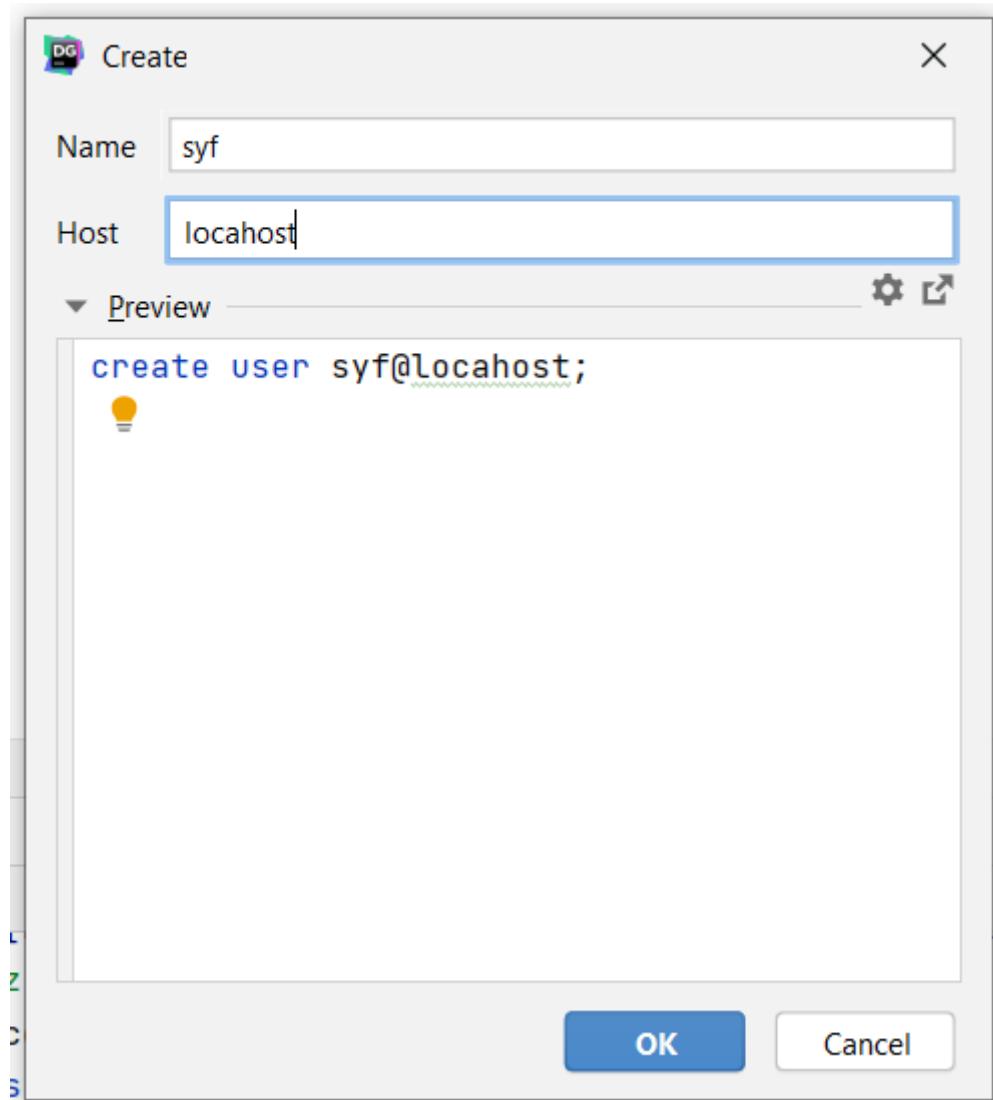
而更多复杂的操作，比如**复杂的查询操作，或者大幅度修改数据库结构**等操作由于篇幅原因，不再一一展示。

4.熟悉图形界面的功能和操作。(可选，例如使用命令行进行实验)

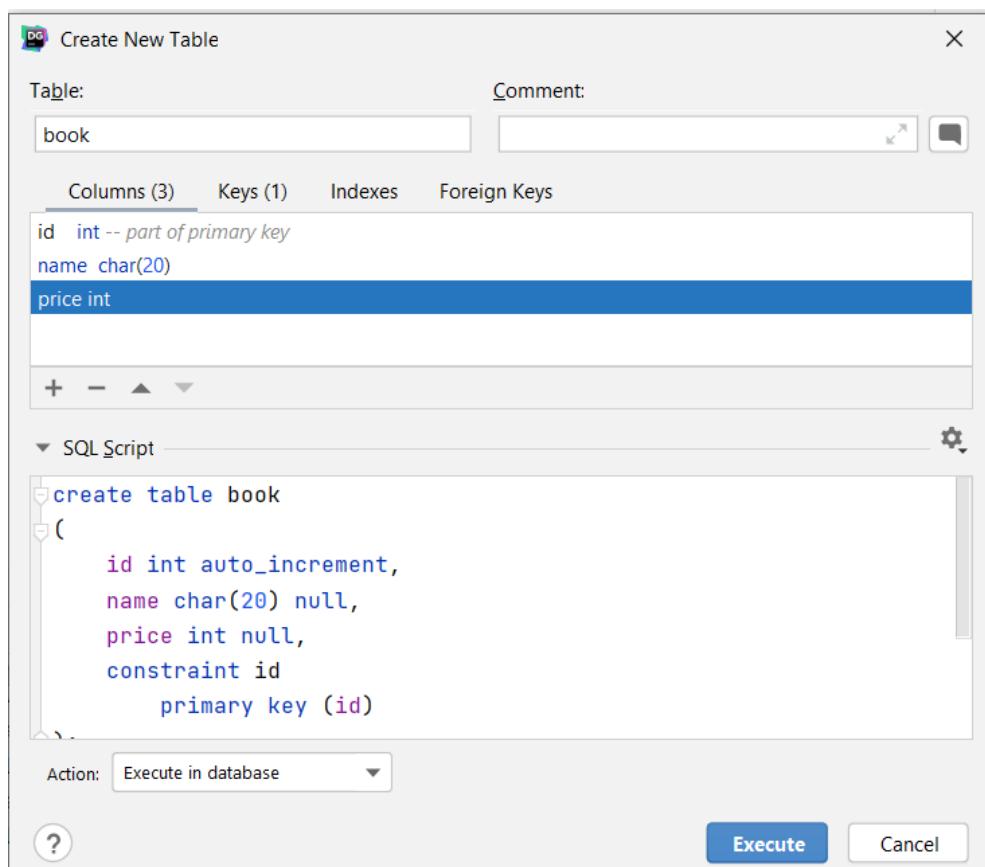


可以看到图形界面提供的操作还是非常丰富的，可以添加user、添加DataSource或者Schema，这里我们针对New中的操作进行说明

创建新用户



创建数据表



这里可以自己选择列的类型，datagrip会自动生成相应的SQL语句并执行

可以看到执行后我们创建了相应数据表：

The screenshot shows the DataGrip interface. On the left, the 'schemas' tree view is open, showing 'schemas 5' and 'eva' selected. Under 'eva', there are several tables: '_efmigrationshistory', 'accessories', 'bills', 'billtranscationrecords', 'book', and 'dbconfigs'. The 'book' table is currently selected. A preview window on the right shows the columns: id, name, and price, each with a small key icon indicating they are part of a primary key.

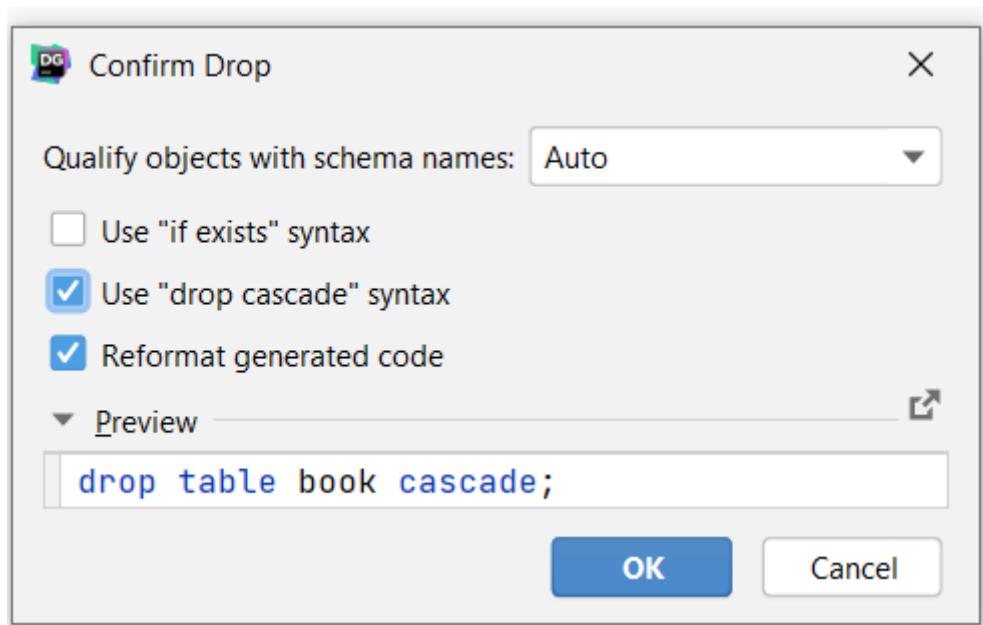
The screenshot shows the 'Modify Table' dialog in DataGrip. The 'Table' field is set to 'book'. The 'Comment' field is empty. Below it, there are tabs for 'Columns (3)', 'Keys (1)', 'Indexes', and 'Foreign Keys'. The 'Columns (3)' tab is selected, showing three columns: 'id int -- part of primary key', 'name char(30)', and 'price int default 10'. The 'price' column is currently selected. At the bottom of this section, there are buttons for adding (+), removing (-), and reordering (▲, ▼) columns. Below this is a 'SQL Script' section containing two generated SQL statements:

```
alter table book modify name char(30) null;  
alter table book alter column price set default 10;
```

At the bottom of the dialog, there is an 'Action' dropdown set to 'Execute in database', and three buttons: '?', 'Execute', and 'Preview'.

可以看到datagrip同时也帮我们生成了SQL语句

删除数据表



修改数据/插入数据方面

gui界面大大降低了CRUD的难度

点击+号创建新的一行，直接输入数据进行赋值后，点击右侧箭头即可上传至数据库，非常的简单方便

修改则直接编辑相应的栏目，之后点击右侧箭头即可上传至数据库，非常方便

| Table View | | | | | |
|---------------------|----|------|-------|--|--|
| Q <Filter Criteria> | | | | | |
| | id | name | price | | |
| 1 | 1 | syf | 1 | | |

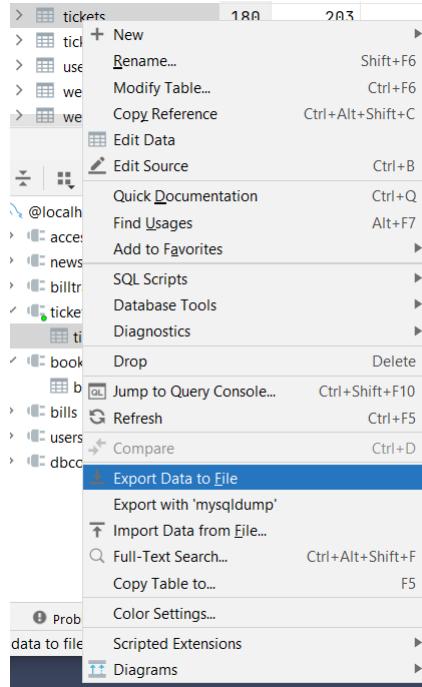
搜索

我们可以直接在搜索框中写简单的sql搜索语句，DataGrip会帮你自动生成相应的查询语句

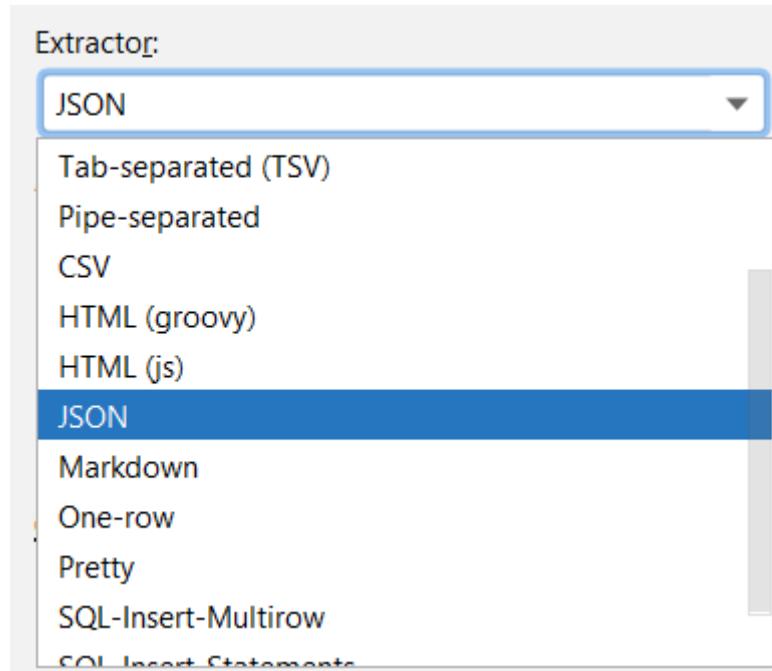
| Table View | | | | | | |
|-------------------|----|------|--------|-----------------|-------|-----|
| Q t.Device='Dell' | | | | | | |
| | Id | Type | Device | DeviceModel | Owner | |
| 1 | 7 | 1 | Dell | G3 | 周安 | 198 |
| 2 | 10 | 1 | Dell | IN1930MWc | 郑跃军 | 138 |
| 3 | 11 | 1 | Dell | XPS 15 | 孙文瑄 | 188 |
| 4 | 16 | 1 | Dell | Inspiron 13 | 邢雅琪 | 177 |
| 5 | 18 | 1 | Dell | IdeaPad 320s-15 | 林冰冰 | 188 |
| 6 | 32 | 1 | Dell | G3 3590 | 周安 | 198 |
| 7 | 33 | 1 | Dell | Inspiron 5557 | 赵淑楠 | 184 |
| 8 | 34 | 1 | Dell | G3 3580 | 金静文 | 199 |
| 9 | 39 | 1 | Dell | XPS 13-9360 | 郁珉昊 | 188 |
| 10 | 45 | 1 | Dell | Vostro 15-5459 | 沈雨欣 | 158 |

导入导出数据

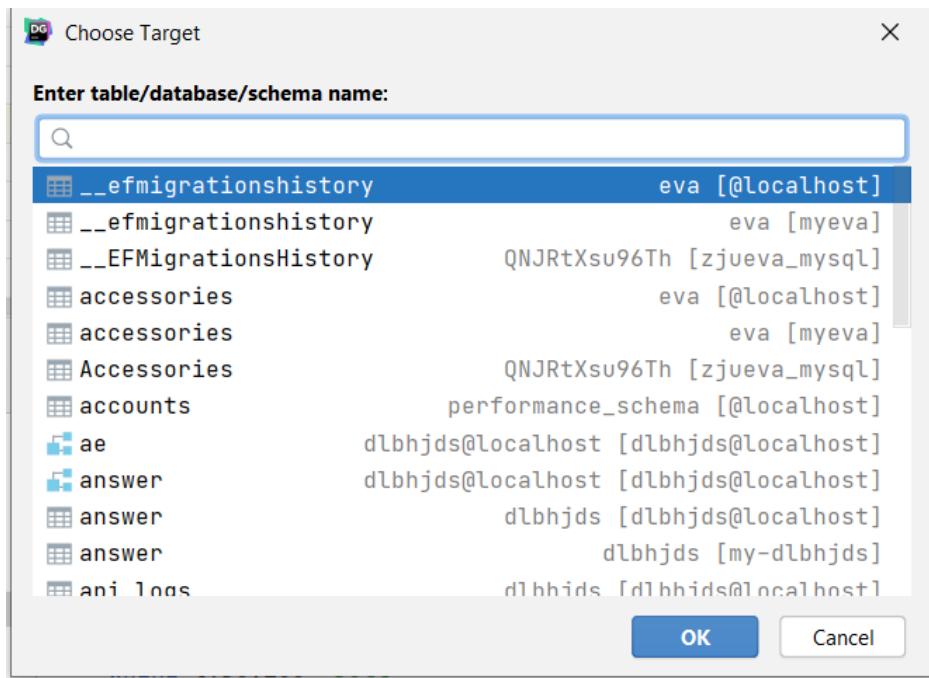
datagrip提供了丰富的导出数据方法，可以导出各种样式的数据格式，如下图



datagrip支持提供导出的格式非常的丰富，除了原生的sql insert、sql update之外们还有Json, csv, xml等等格式支持。



同时datagrip支持不同数据库间导入数据表，可以方便的在各个数据库中相互操作，如下图所示



四、遇到的问题及解决方法

- **datagrip连接数据库失败**

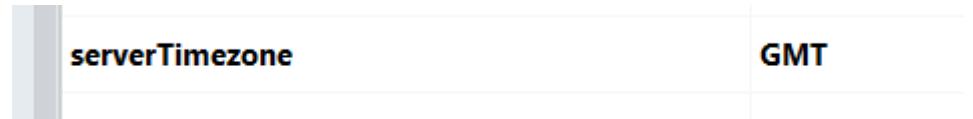
idea连接mysql报错Server returns invalid timezone. Go to 'Advanced' tab and set 'serverTimezone' property

原因：时区问题，MySQL驱动默认UTC时区。

解决方法：修改时区

```
1 # 设置全局时区 mysql> set global time_zone = '+8:00';
2 Query OK, 0 rows affected (0.00 sec)
3 # 设置时区为东八区 mysql> set time_zone = '+8:00';
4 Query OK, 0 rows affected (0.00 sec)
5 # 刷新权限使设置立即生效 mysql> flush privileges;
6 Query OK, 0 rows affected (0.00 sec)
```

之后在advanced setting中加入



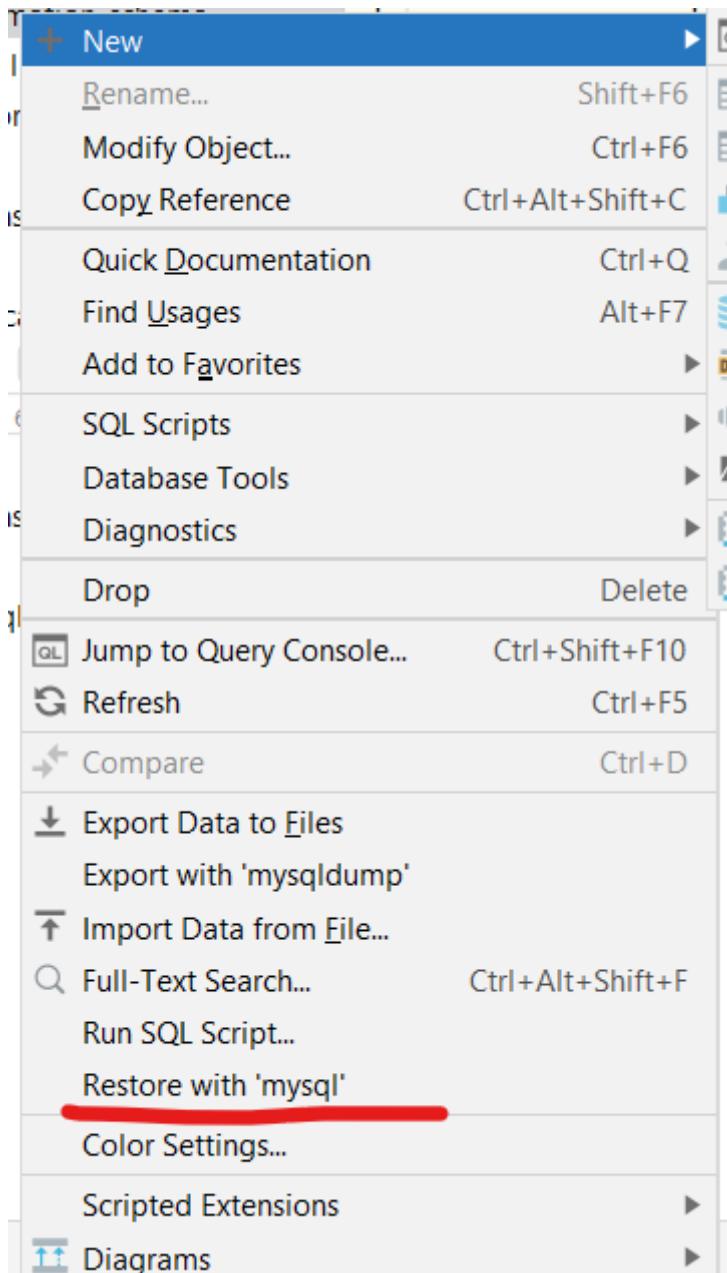
即可成功连接

- **数据库迁移过于麻烦**

mysql提供了一种自带的迁移工具：**mysqldump**,可以选择某个数据库并将整个数据库的结构和数据全部导出为.sql后缀的文件，可以让迁移工作变得非常简便

之后Run，即可在路径上得到.sql后缀的文件

之后我们选择导入



- **php数据库连接失败**

报错很奇怪，这里不再贴上。驱动为mysqli，php配置确认无误。

解决方法：将localhost改为127.0.0.1即可？我没懂为啥127.0.0.1和localhost不同，并没有开http代理

- **不同应用需要使用不同的数据库->Docker**

mysql安装比较复杂，针对不同的应用程序，我们需要安装若干个mysql，并将其开在不同端口，这本身就是一件让人头大的事情，但是我们可以使用docker来设置。

docker安装过程这里不再详述

我们只需要在目标文件夹下建立一个名为docker-compose.yml的文件，里面内容如下：

```
version: "3"
services:
  db:
    image: mysql:5
    restart: always
    ports:
      - 127.0.0.1:3306:3306
    volumes:
      - ./data/db_backup:/var/backups
      - ./data/mysql:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: "peleL3ypx5D1"
      MYSQL_DATABASE: "alibinjia"
      MYSQL_USER: "alibinjia"
      MYSQL_PASSWORD: "XCUycySH"
~
```

里面我们可以自己配置相应的端口映射，然后使用

```
1 | docker-compose up -d
```

即可便捷的在本地的3306端口开一个mysql DBMS

使用Docker我们可以便捷的同时开启若干个数据库并统一管理，是一个非常有必要使用的应用。

五、总结

mysql的安装比较简单，而且安装包已经将相关的依赖同时一并安装，只需要一直点点点就可以，因此安装过程并没有遇到很多的问题。

而在本次报告中，我们详细介绍了mysql命令行界面以及使用datagrip图形化界面操作数据库的方法，也了解了若干数据迁移的方法，让我加深了对数据库基本操作的理解以及命令行的操作方法，为之后更好的使用、设计数据库打下了坚实的基础。