

目录

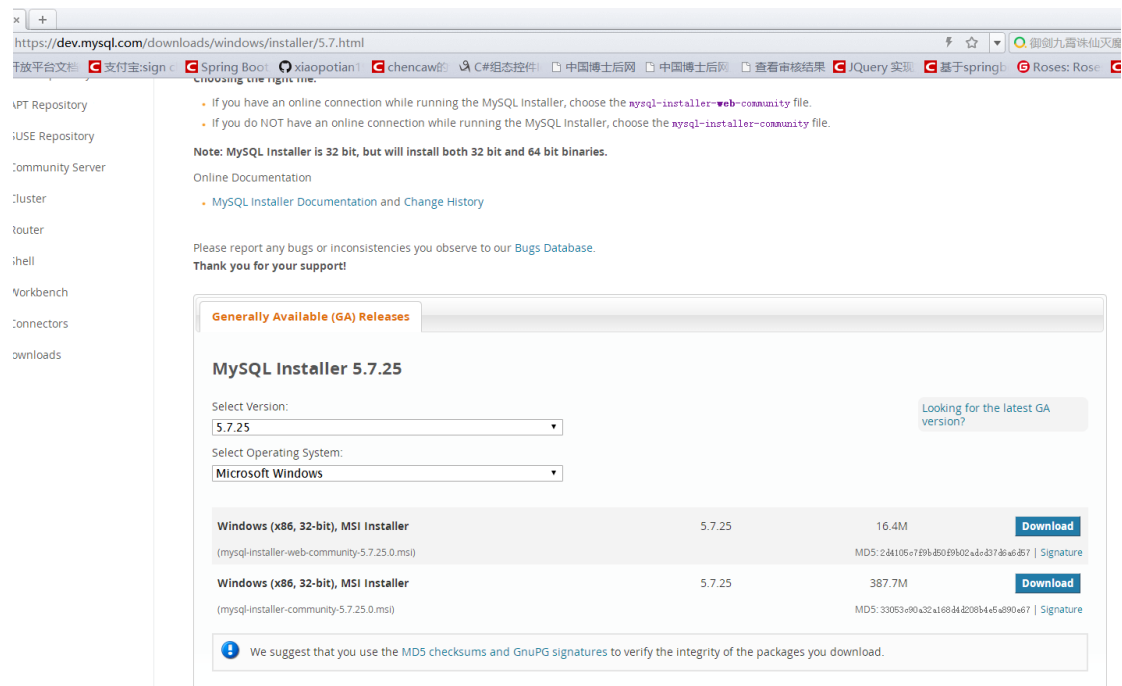
基础三、Mysql 的安装和使用	2
3.1 软件下载	2
3.2 Mysql 的安装	2
3.3 MySql 配置环境变量	15
3.3 MySql 的使用	19
3.3.1 登录	19
3.3.2 创建数据库	20
3.3.3 查看数据库	20
3.3.4 删除数据库	21
3.3.5 不报错删除数据库 if exists	21
3.3.6 使用该数据库 use	22
3.3.7 创建表	22
3.3.8 参看所有的表名	23
3.3.9 参看表的结构	23
3.3.10 在表中插入数据	23
3.3.11 查看刚才插入的数据	24
3.3.12 修改表名	24
3.3.13 修改表字段的数据类型	24
3.3.14 修改表字段名称	25
3.3.15 增加字段	25
3.3.16 删除字段	26
3.3.17 删除表	27
基础四、数据的增删改查四大操作	28
4.1 完整的创建一个例子	28
4.2 插入数据	29
4.3 查找数据	29
4.4 删除数据	33
4.5 修改数据	34

基础三、Mysql 的安装和使用

3.1 软件下载

1、Mysql 有社区免费版本安装，下载链接如下

<https://dev.mysql.com/downloads/windows/installer/5.7.html>



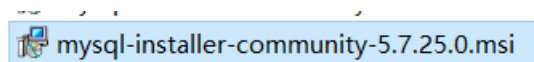
2、选择

Windows (x86, 32-bit), MSI Installer (mysql-installer-community-5.7.25.0.msi)	5.7.25	387.7M	Download
--	--------	--------	--------------------------

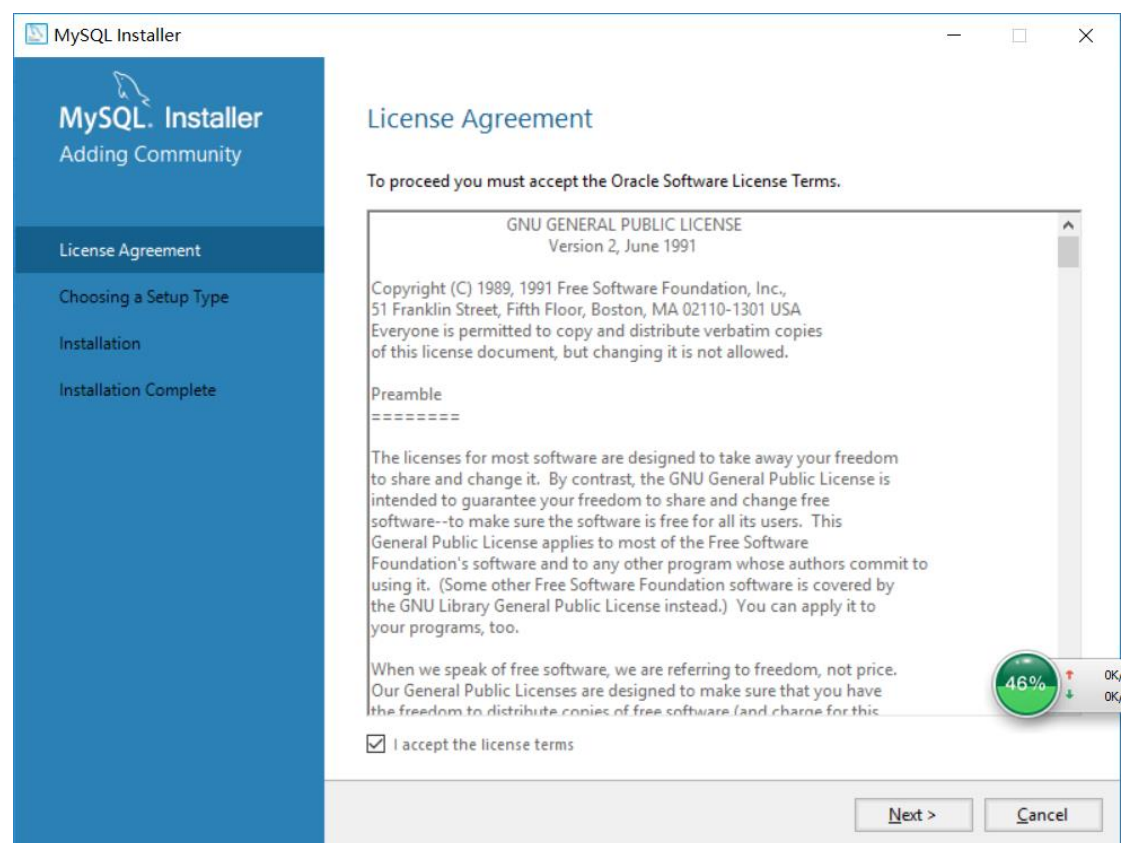
3、在跳出的界面上，选择 no thanks 直接下载

3.2 Mysql 的安装

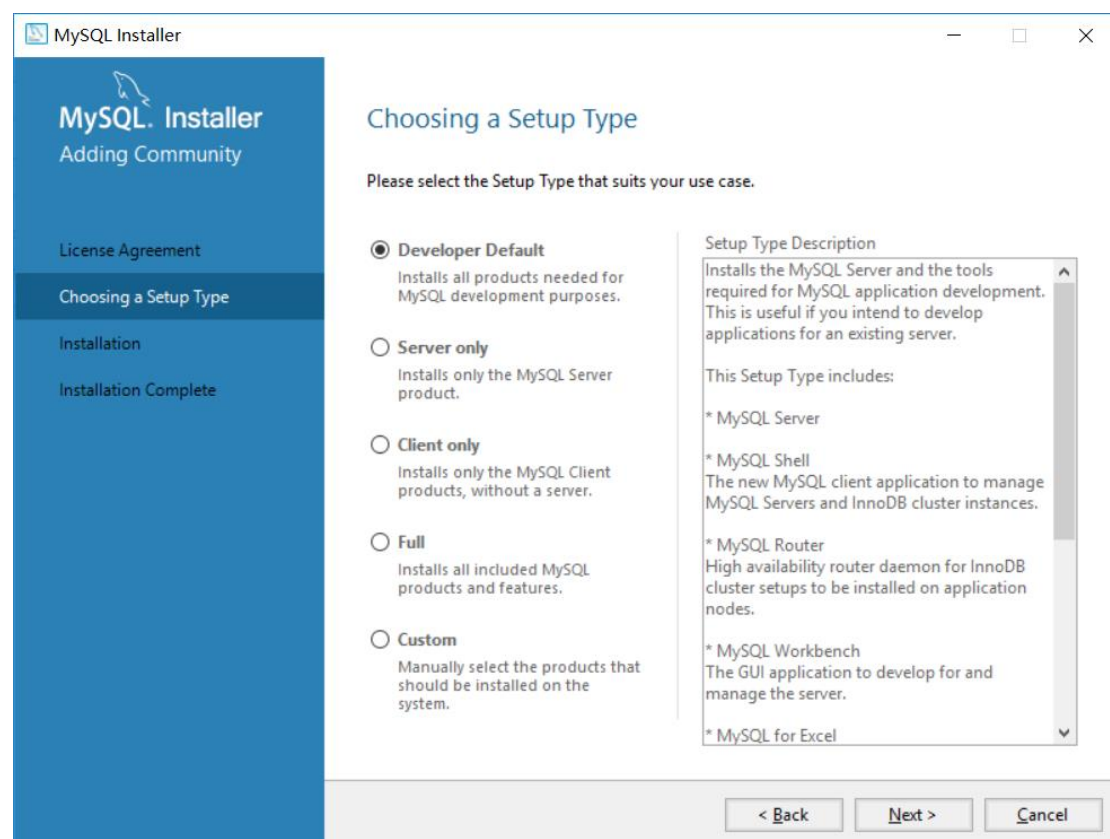
1、双击安装包进行安装



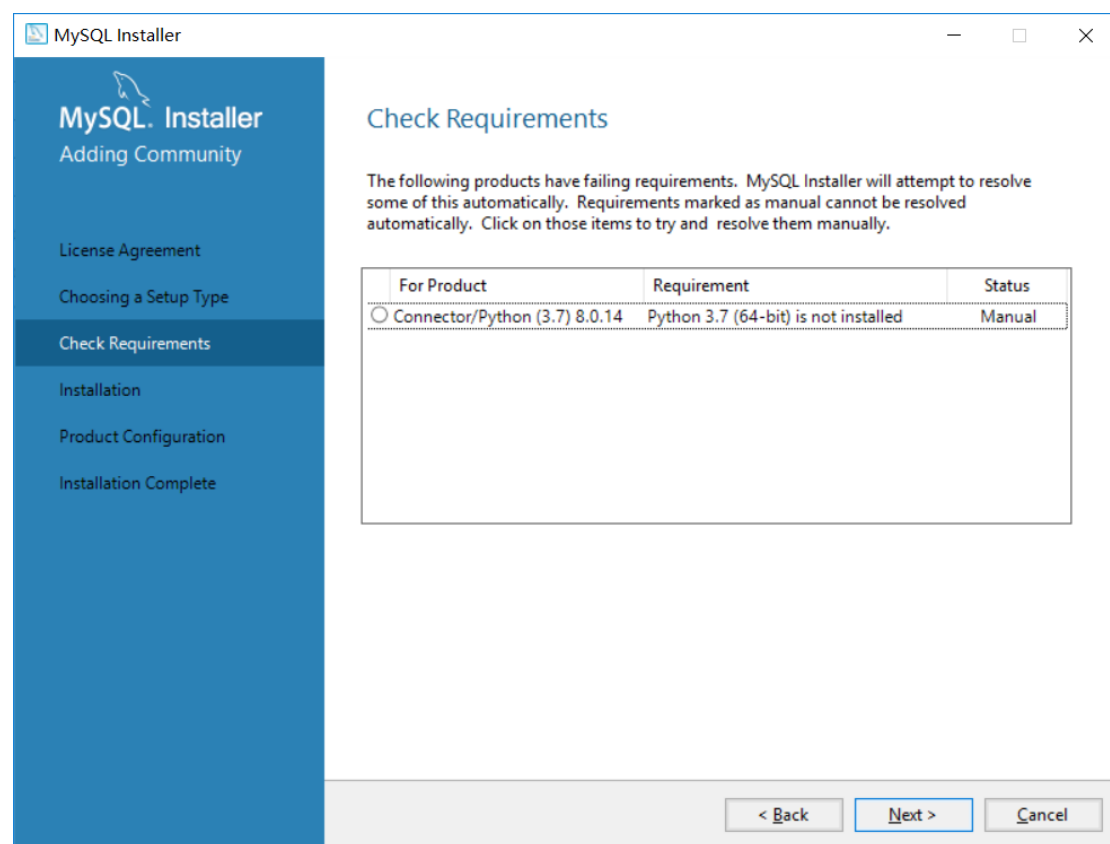
2、在跳出的界面中选择 I accept 点击 Next



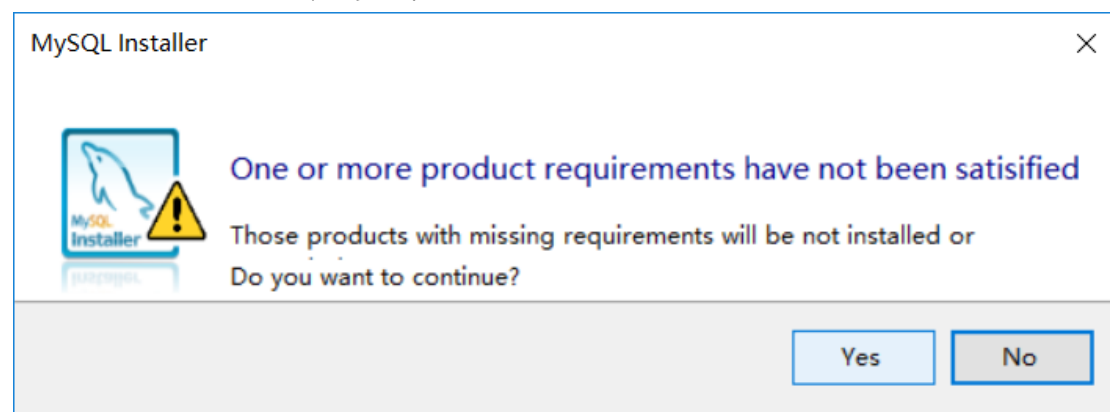
3、点击 Next 下一步



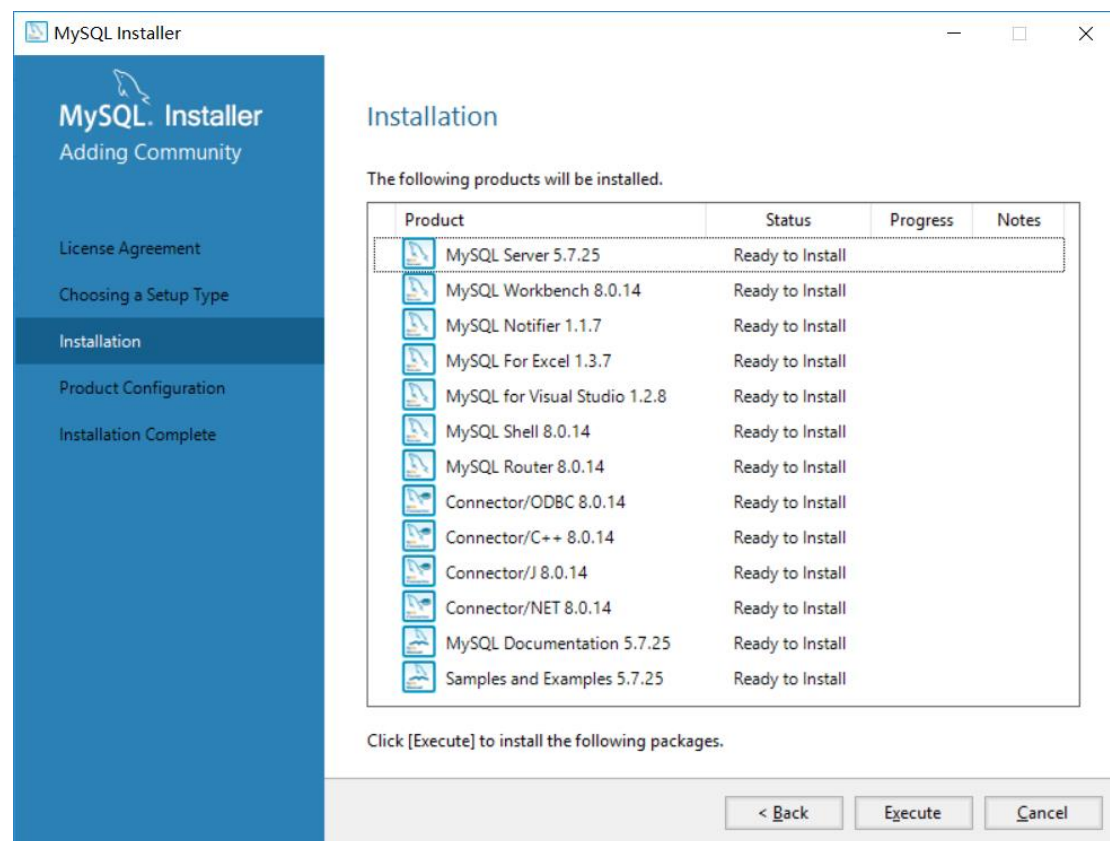
4、点击 Next 按钮



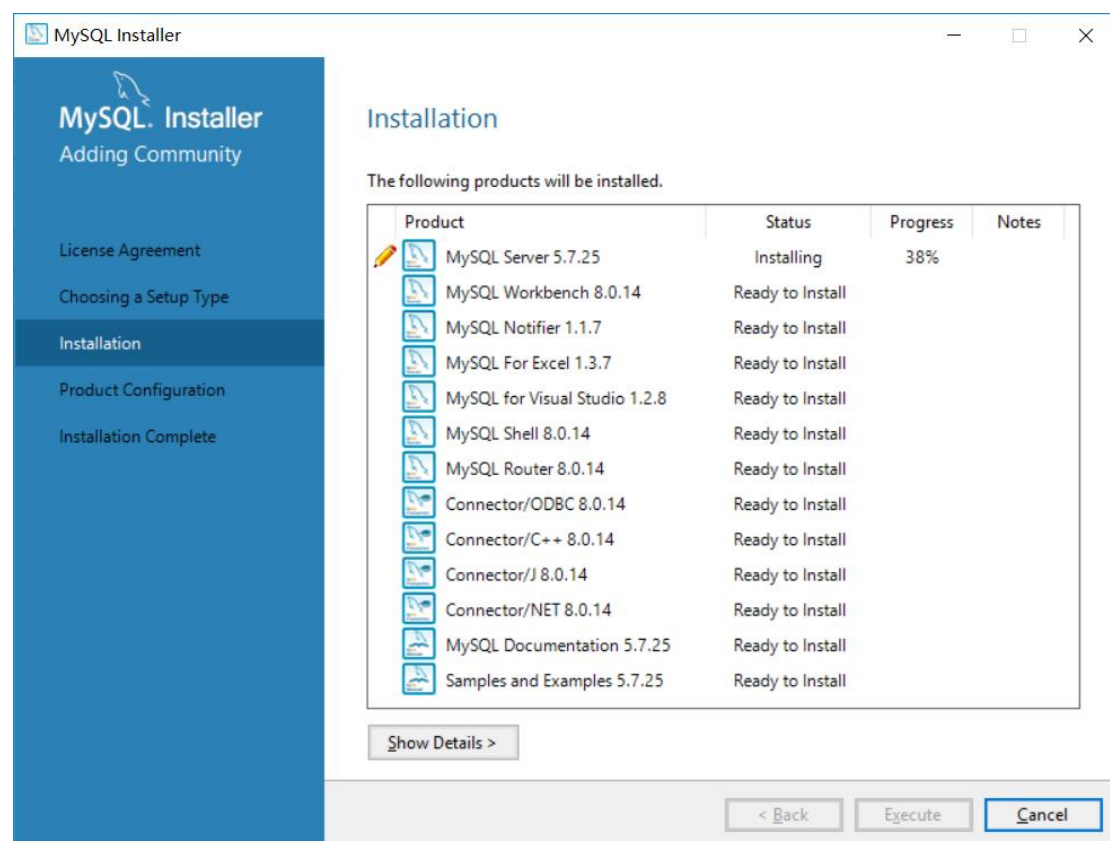
5、如果跳出了如下界面，则点击 Next



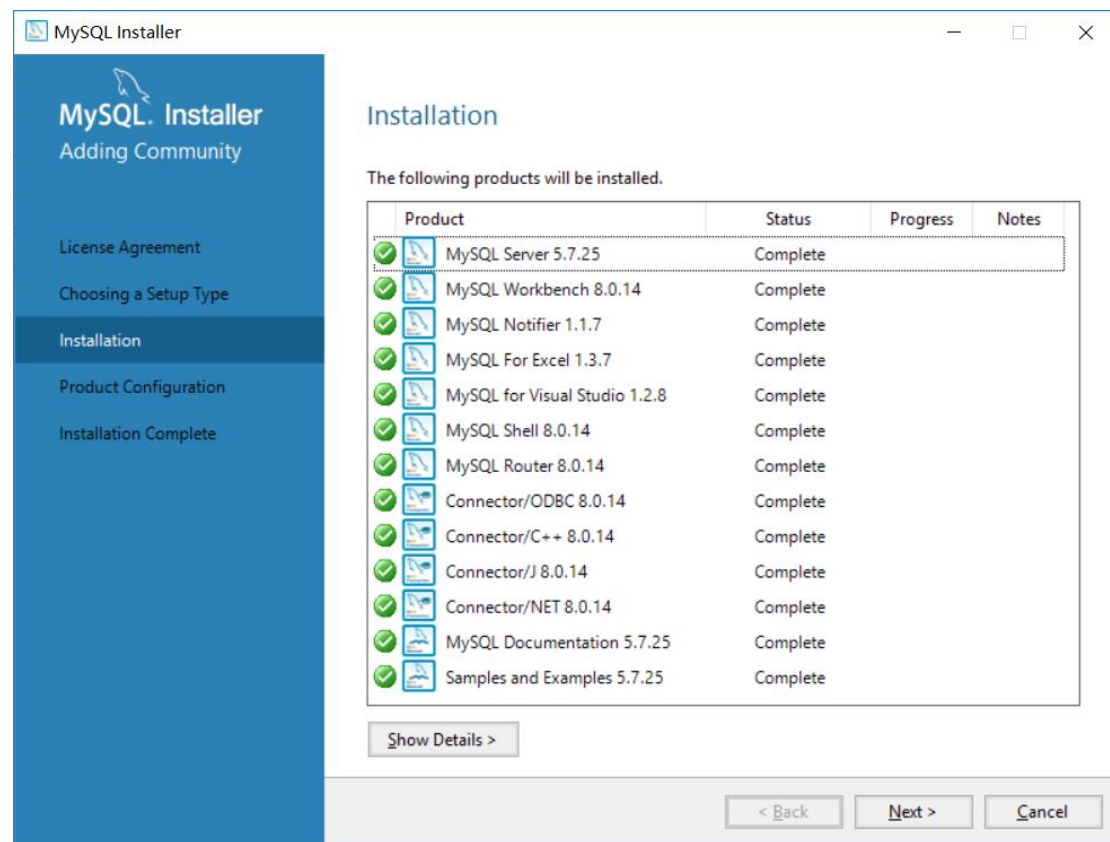
6、而后点击 Execute



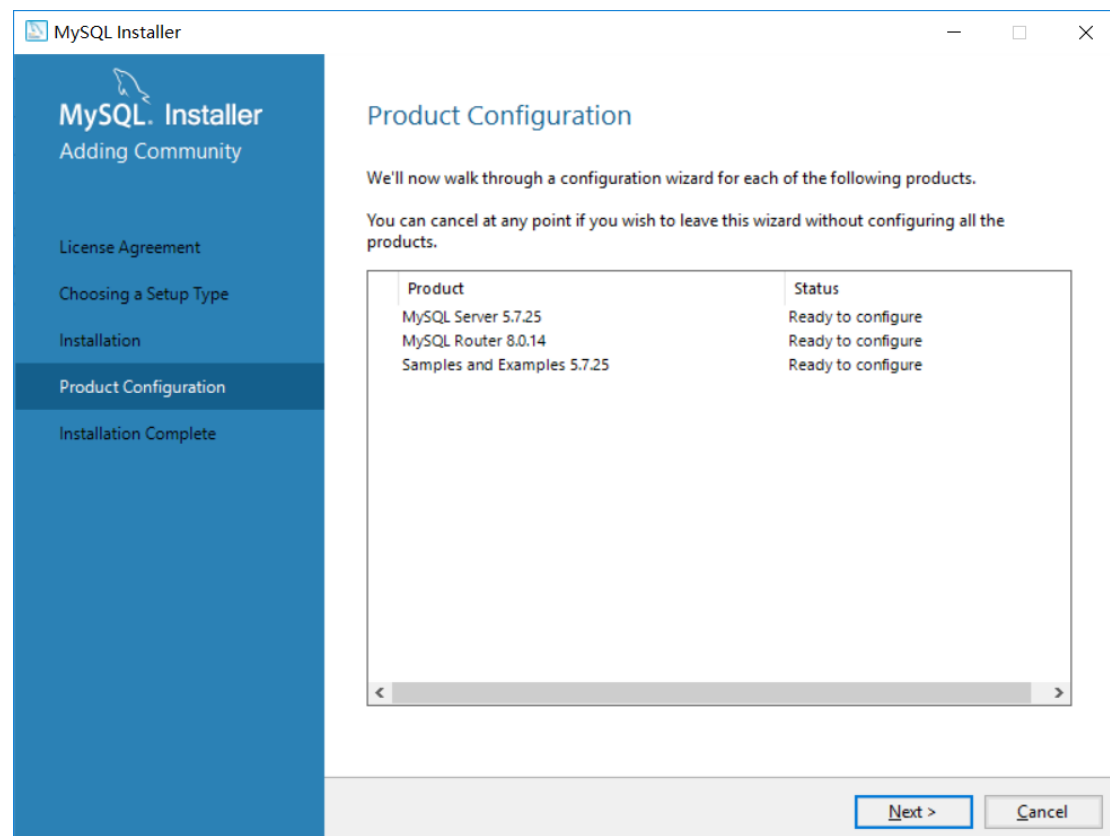
此时会显示安装进度



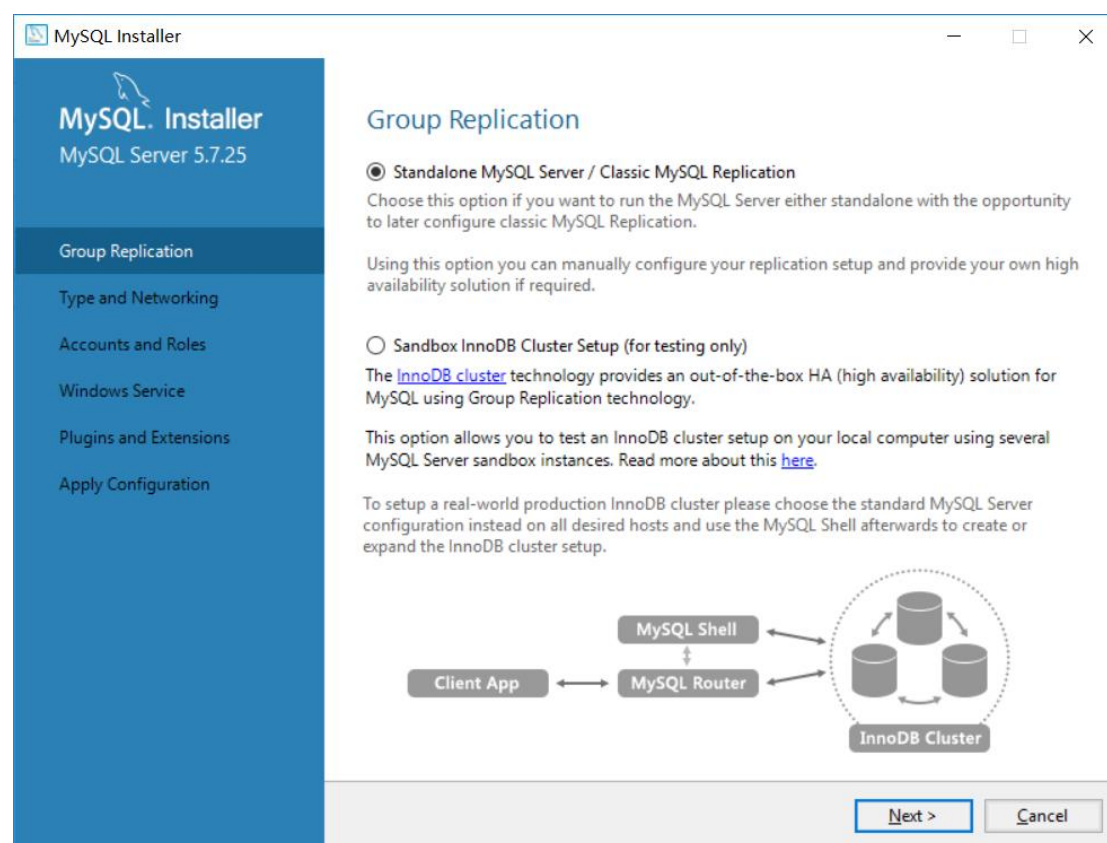
7、在上述界面中点击 Next



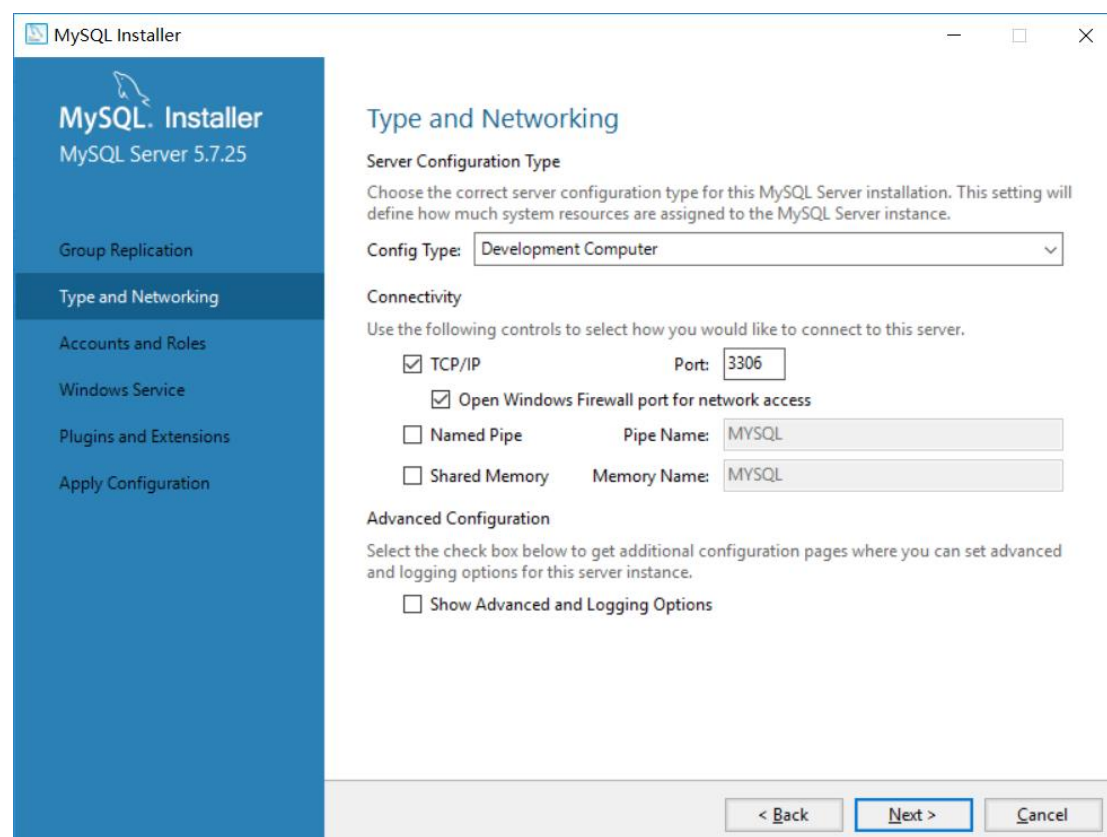
8、在上述界面中点击 Next



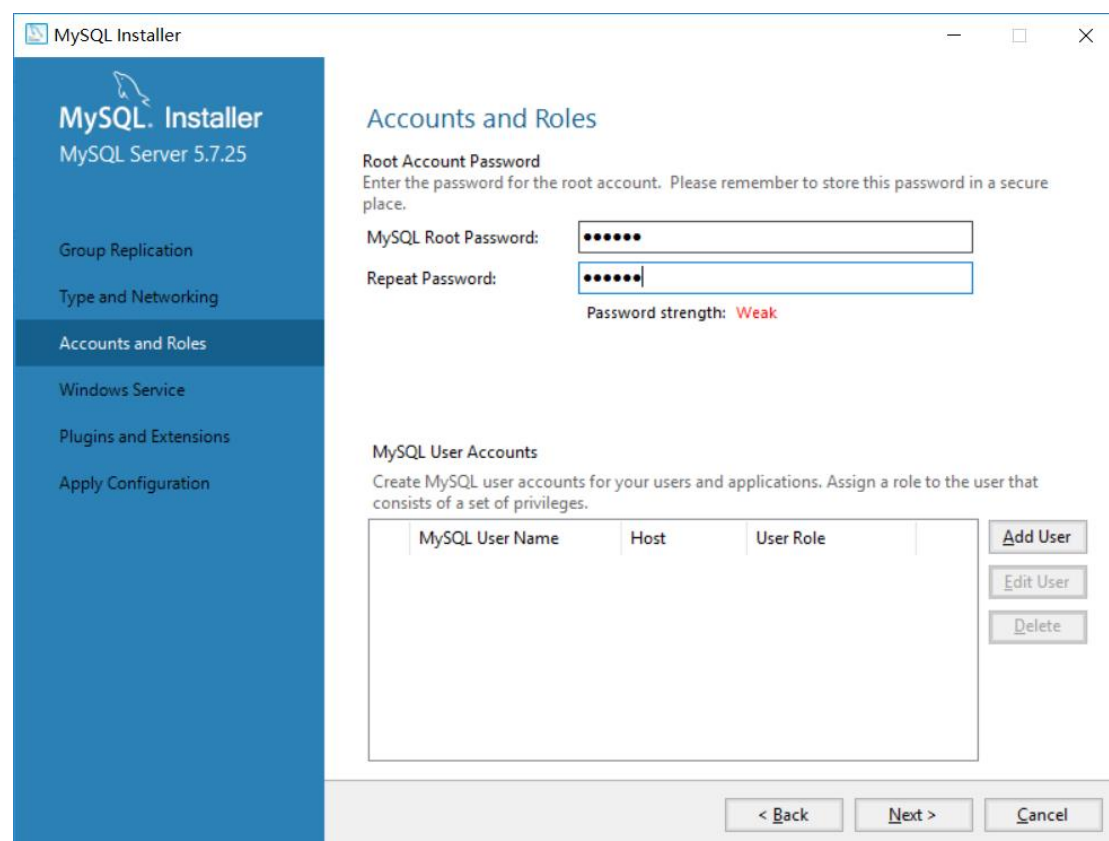
9、在上述界面中点击 Next



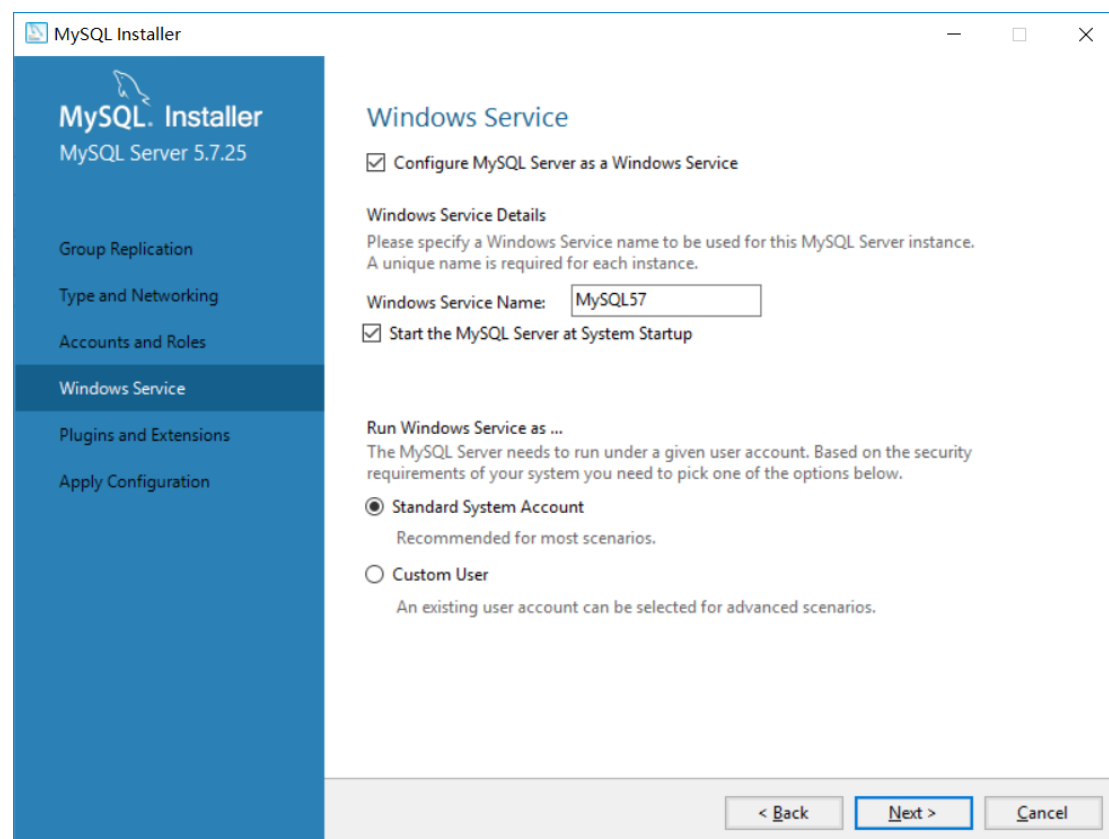
10、在上述界面中点击 Next



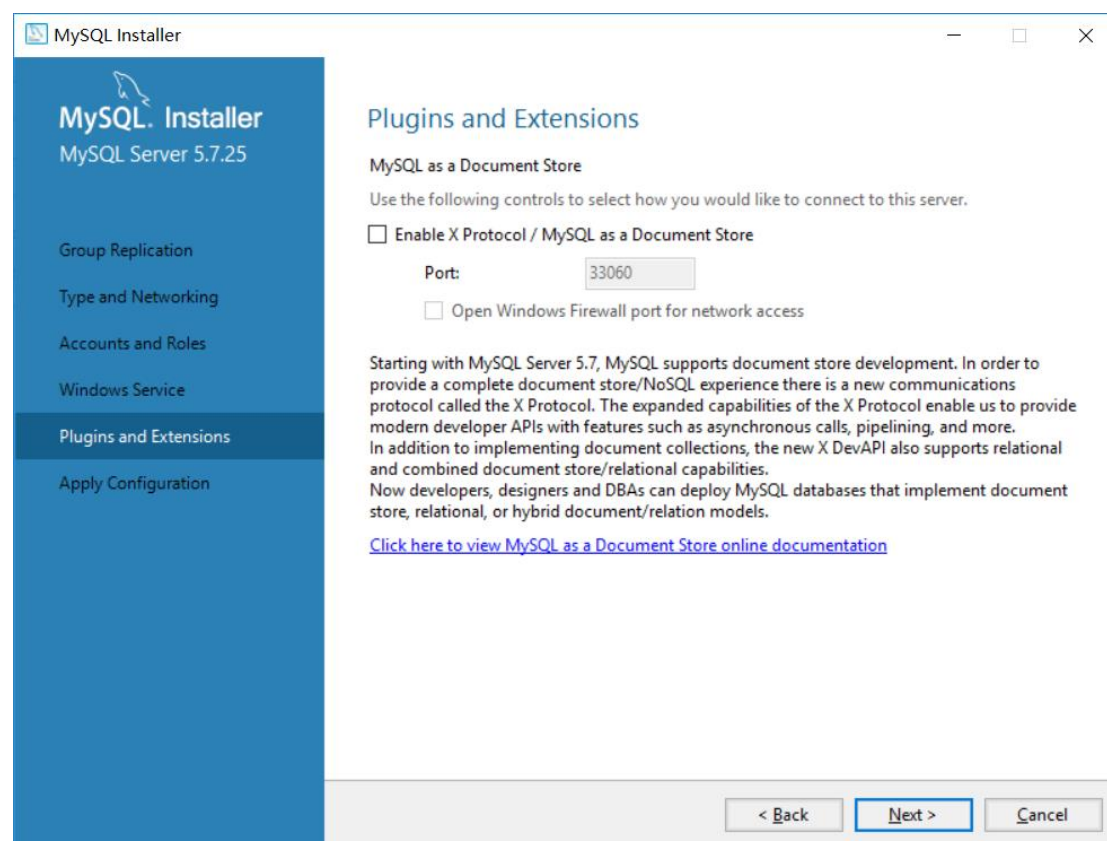
11、在上述界面中重复输入默认密码 111111，点击 Next



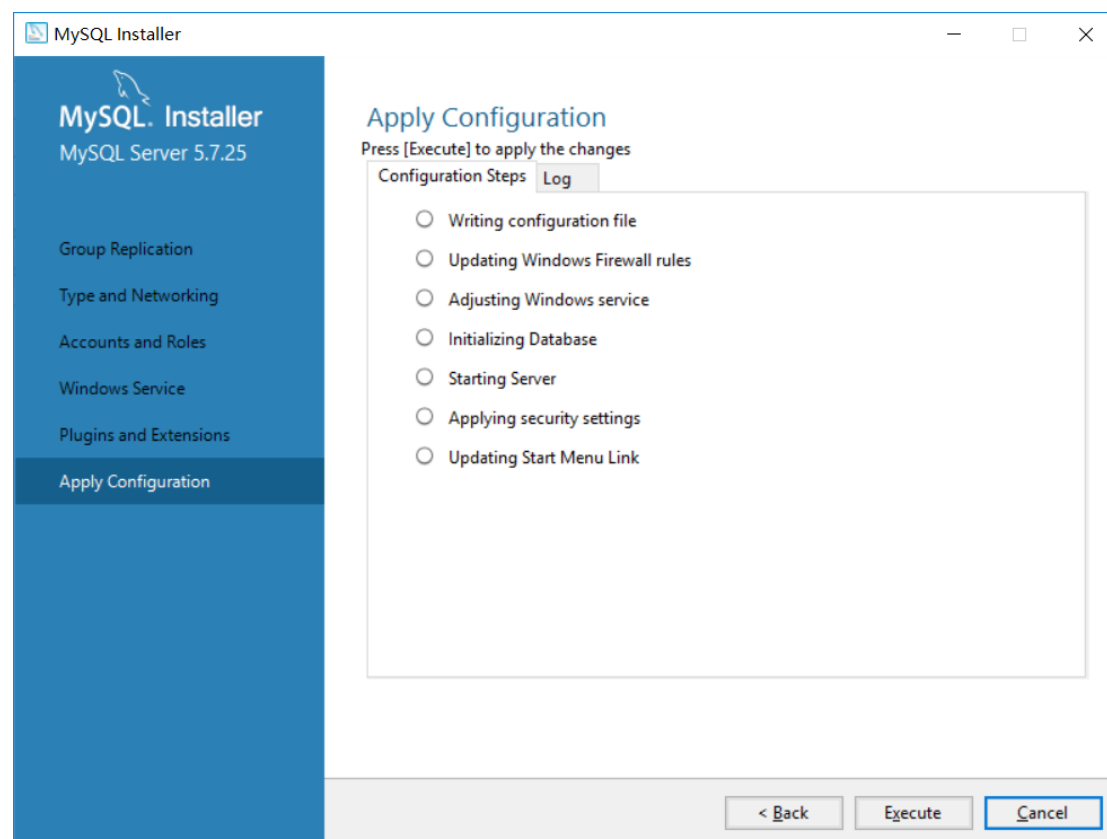
12、在上述界面中点击 Next



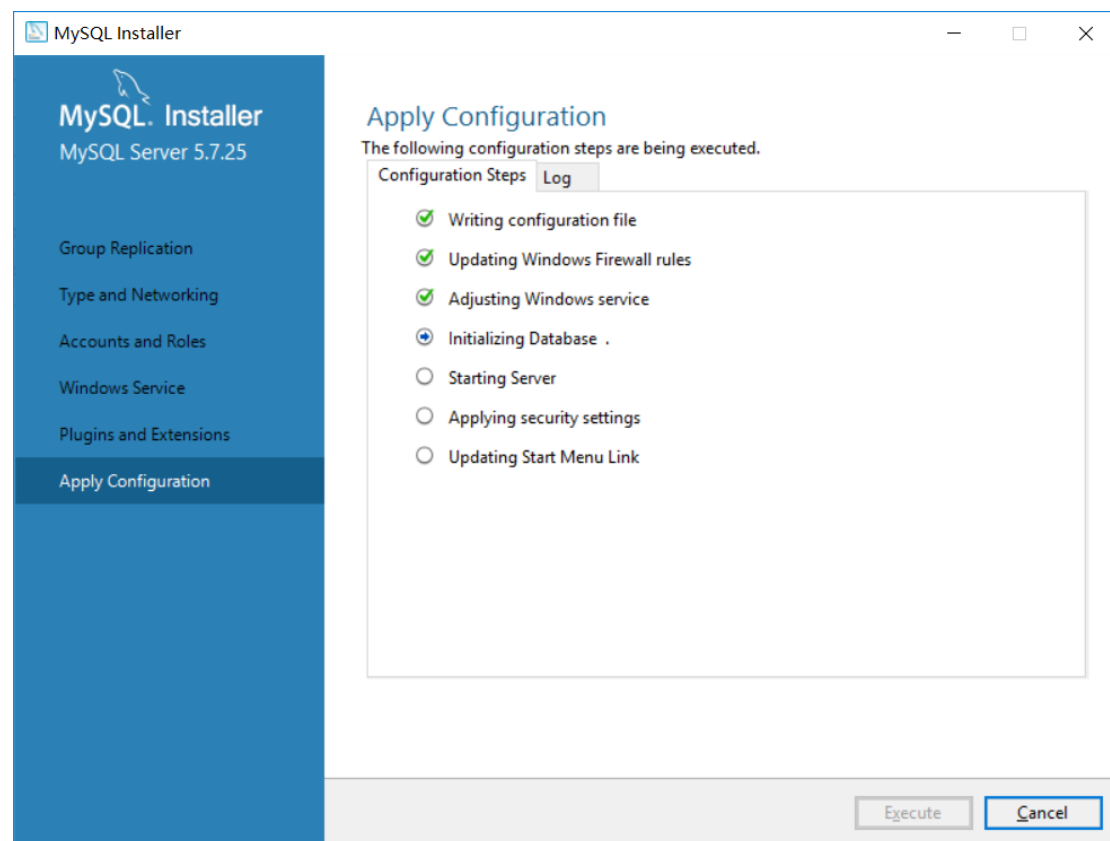
13、在上述界面中点击 Next



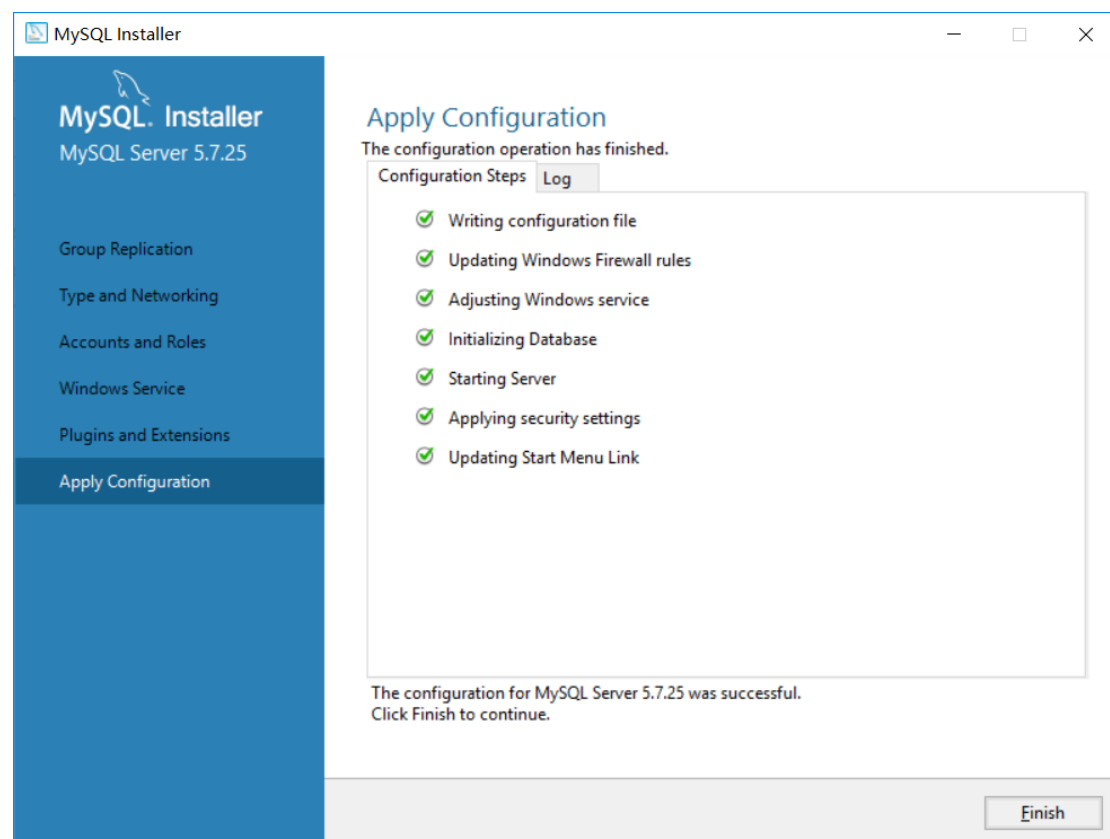
14、在上述界面中点击 Execute 完成安装



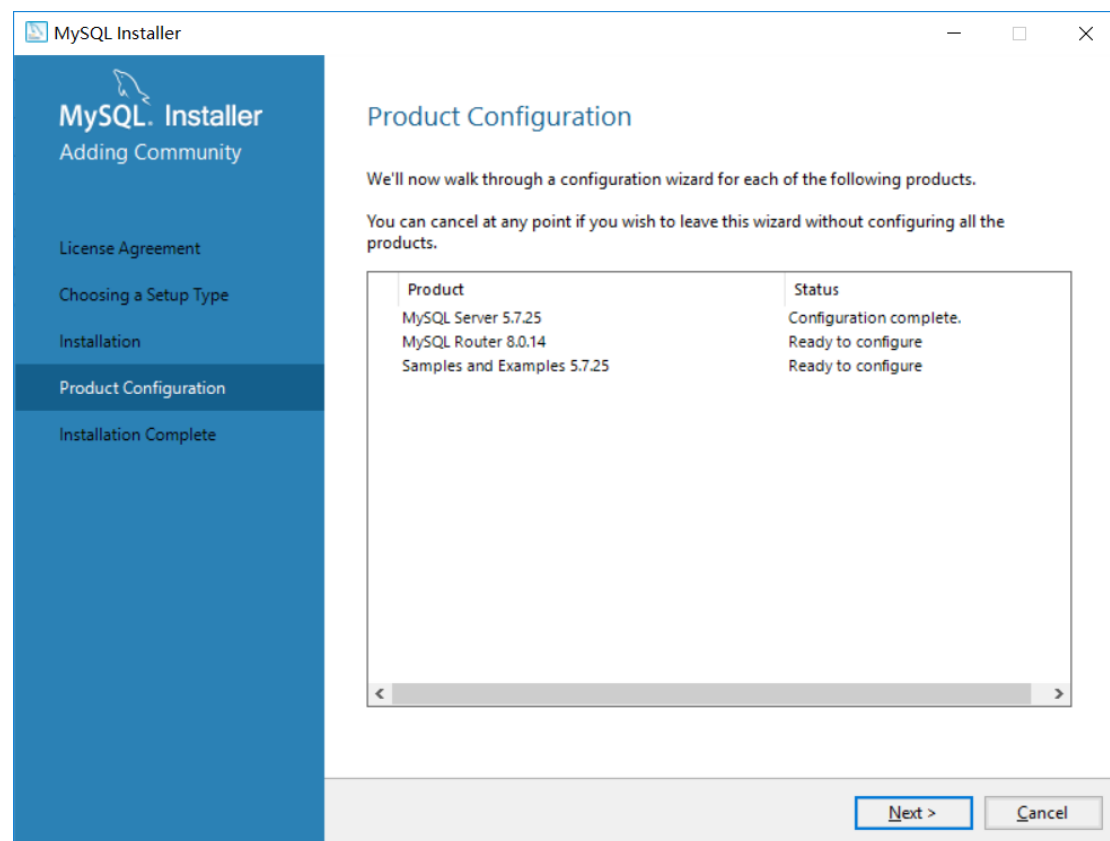
安装过程中会确认各项内容



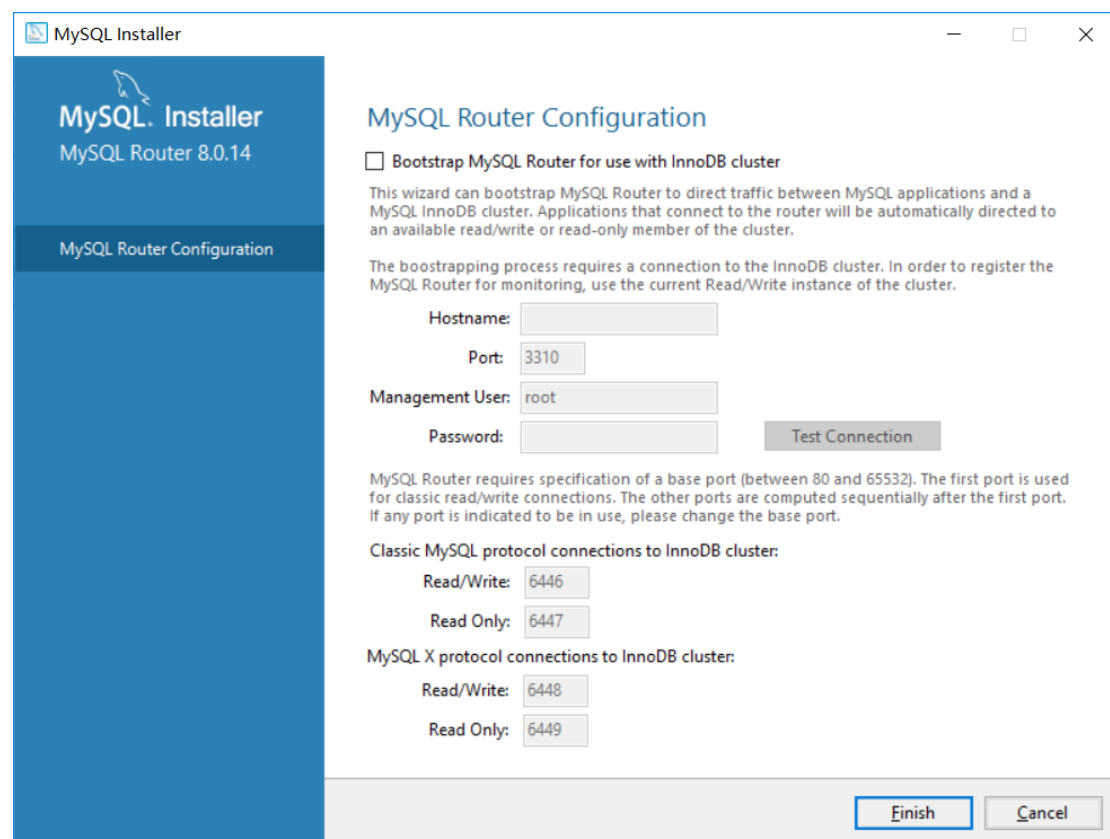
15、而后点击 Finish



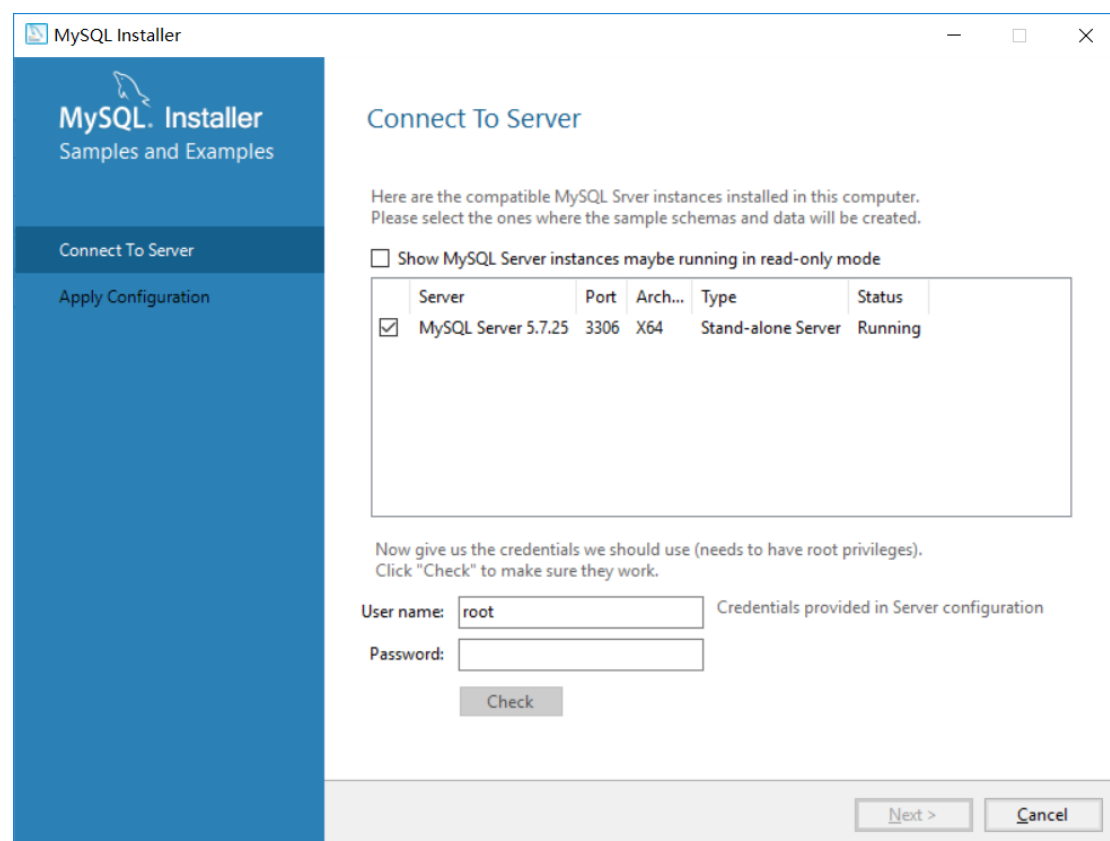
16、在如下界面上点击 next 按钮



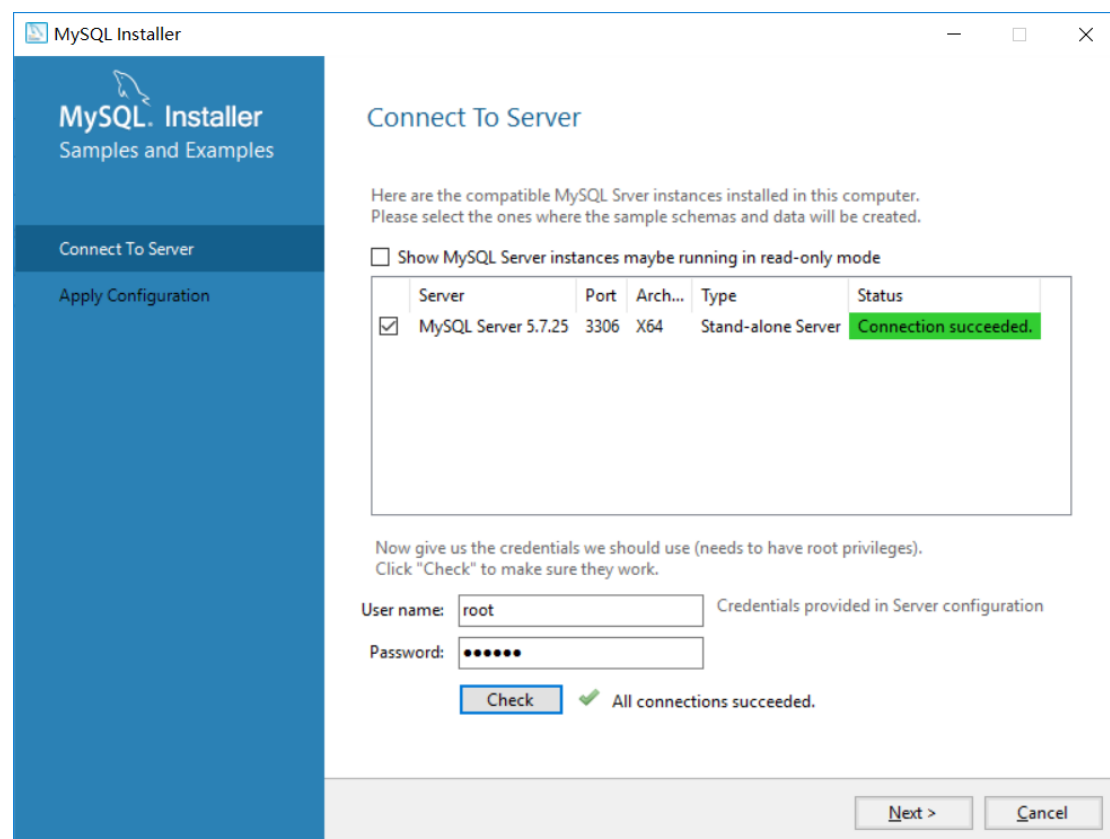
17、而后点击 finish



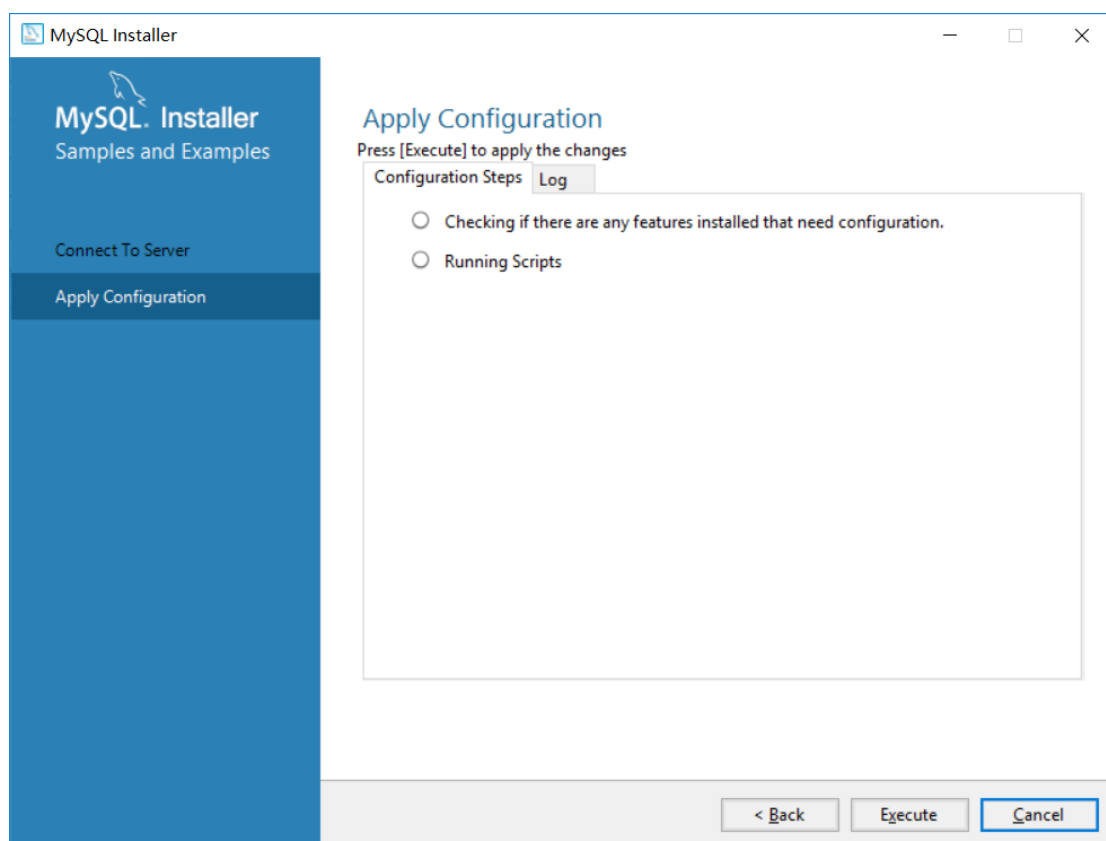
18、在接下来的界面中点击 next，出现如下界面



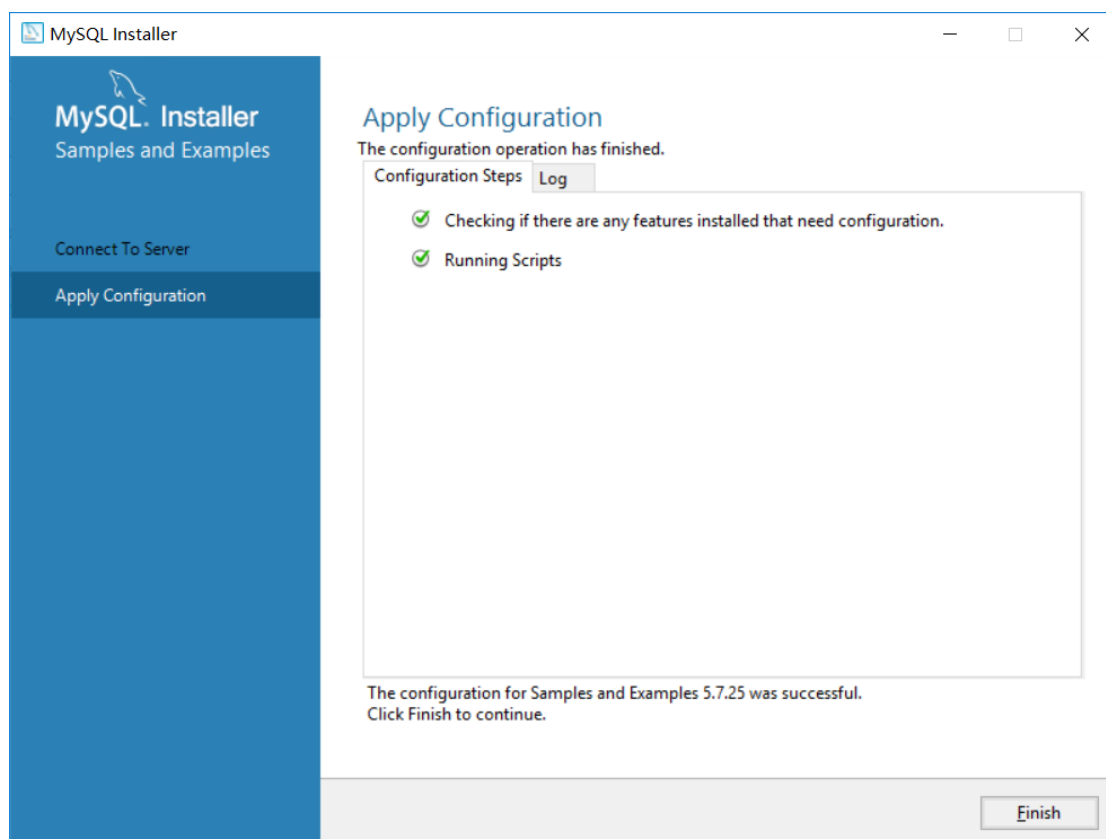
19、在 password 中输入 111111，并点击 Check 按钮，测试通过后点击 Next 按钮



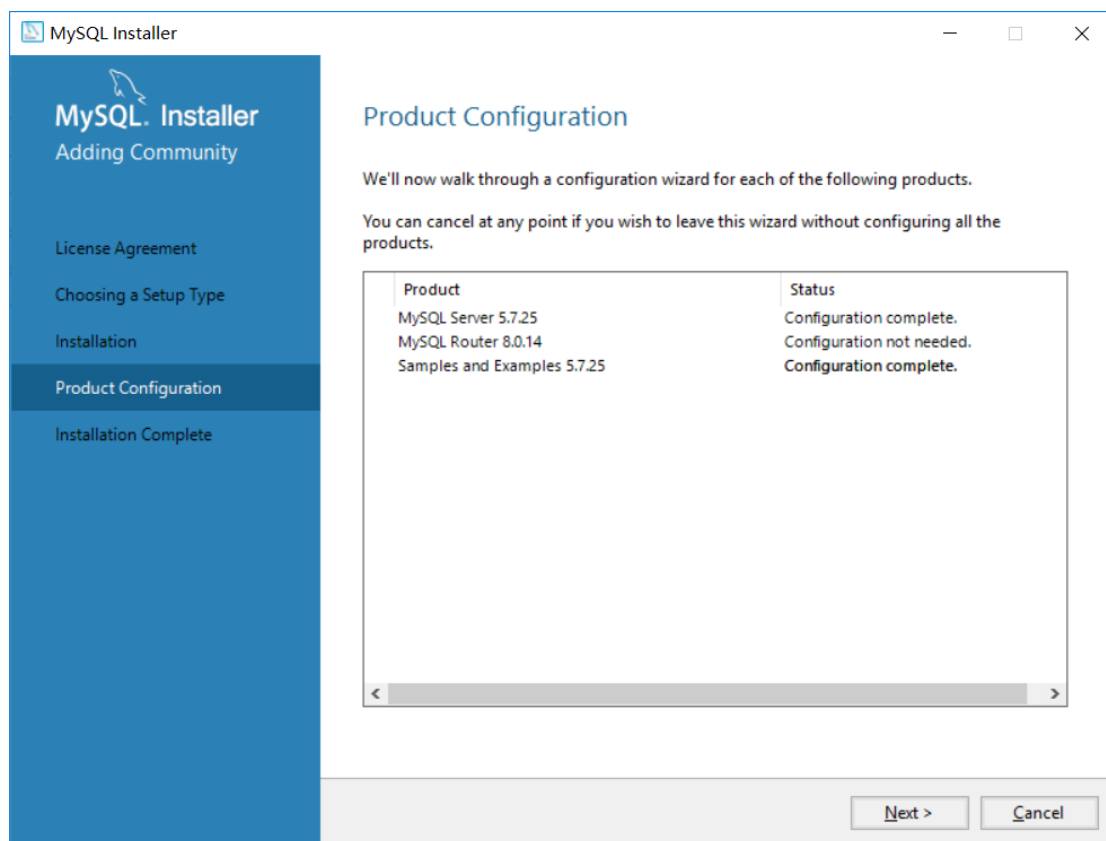
20、而后点击 Execute



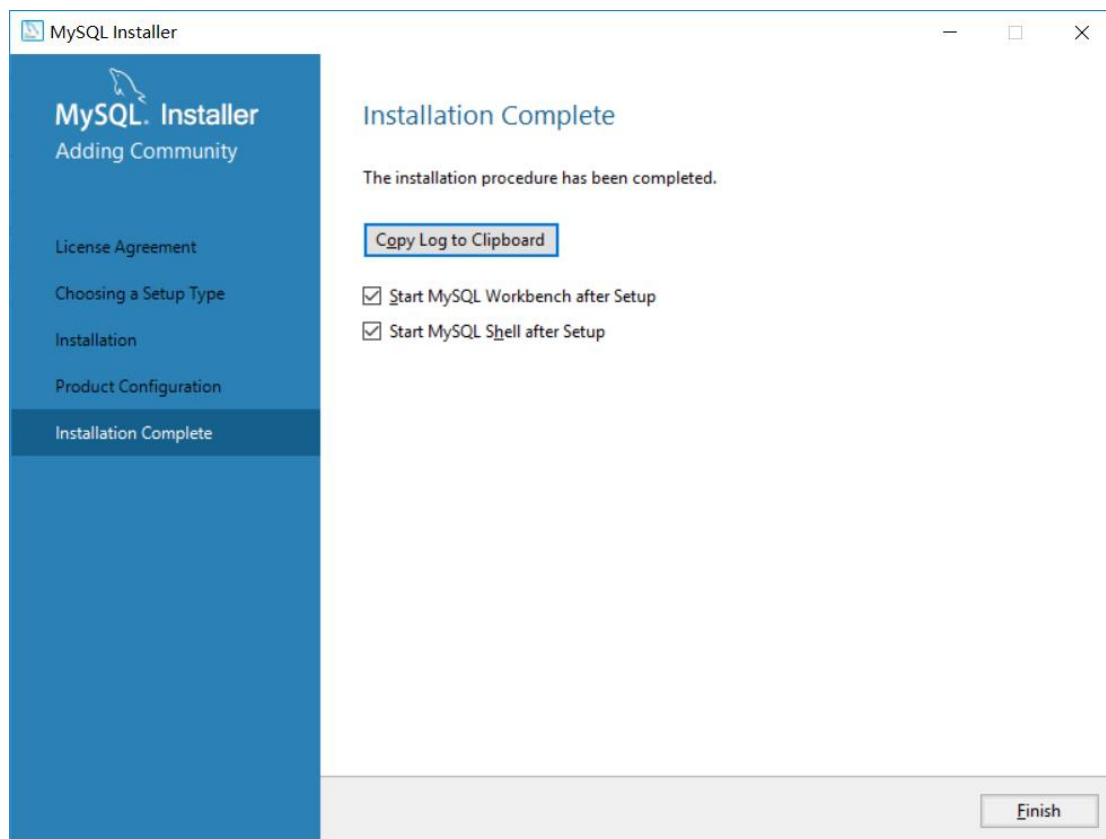
21、而后点击 finish



22、在出现的界面中点击 next



23、而后点击 finish



3.3 MySQL 配置环境变量

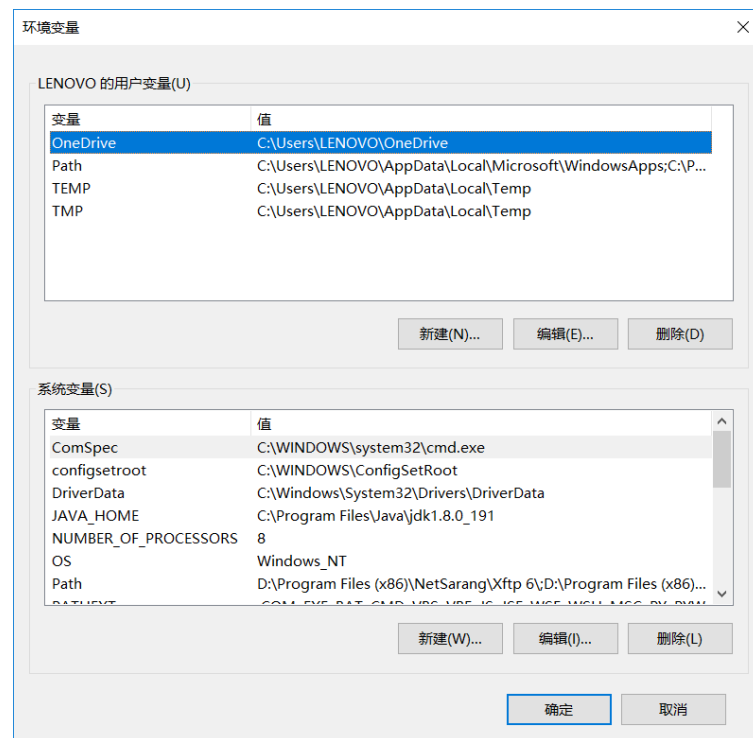
1、右键我的电脑



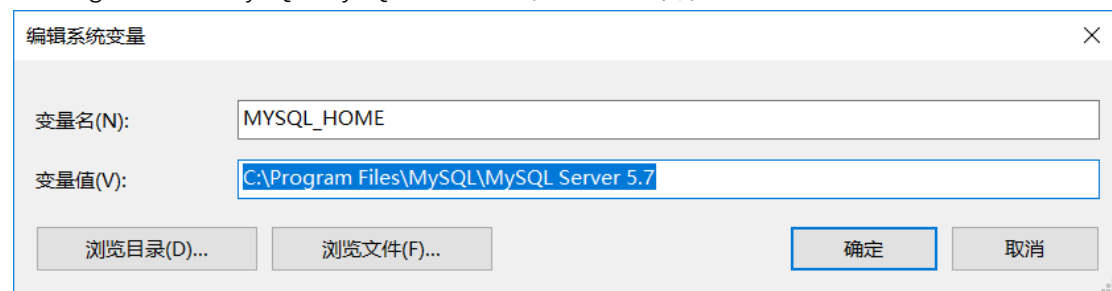
2、点击高级系统设置



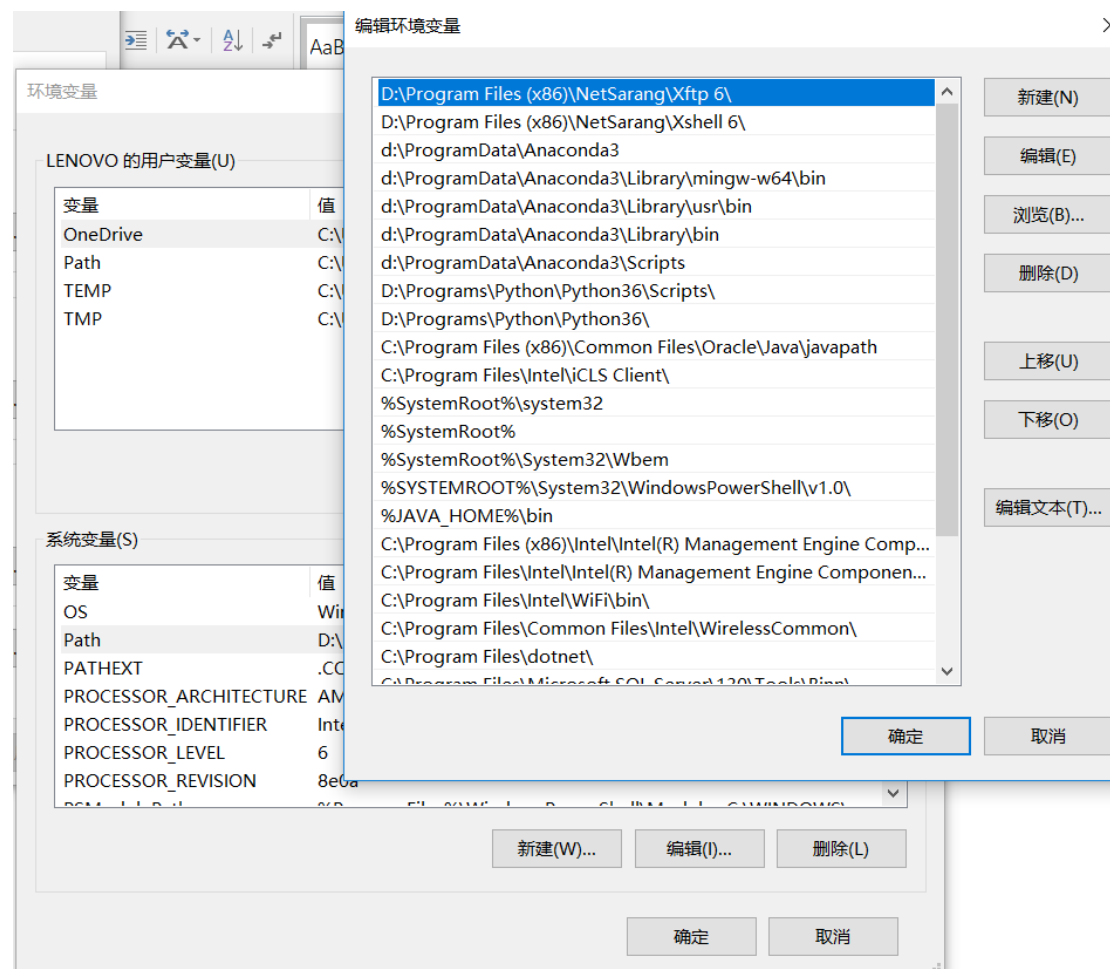
3、点击环境变量



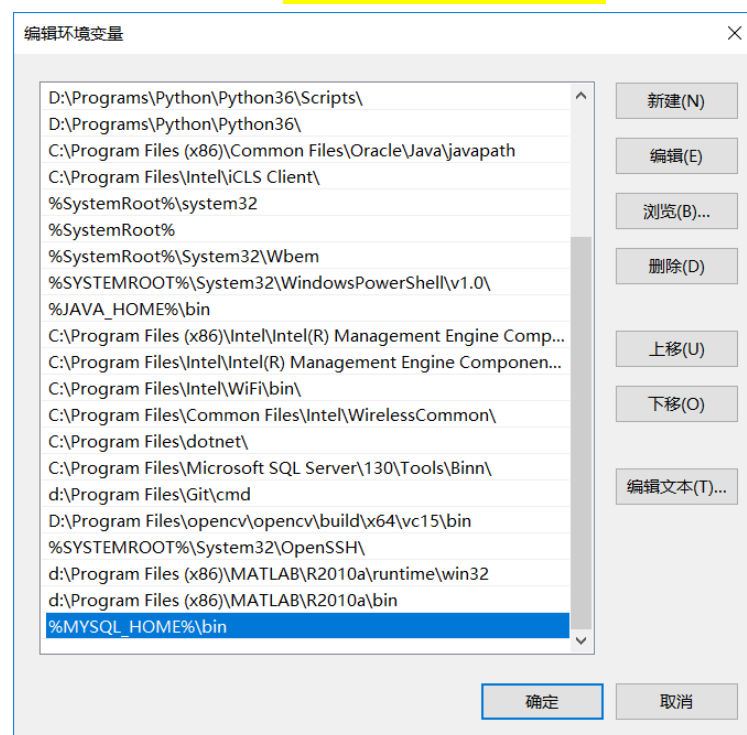
4、在下方的系统变量中点击新建按钮，输入系统变量名 MYSQL_HOME，输入系统路径 C:\Program Files\MySQL\MySQL Server 5.7，而后点击确定按钮



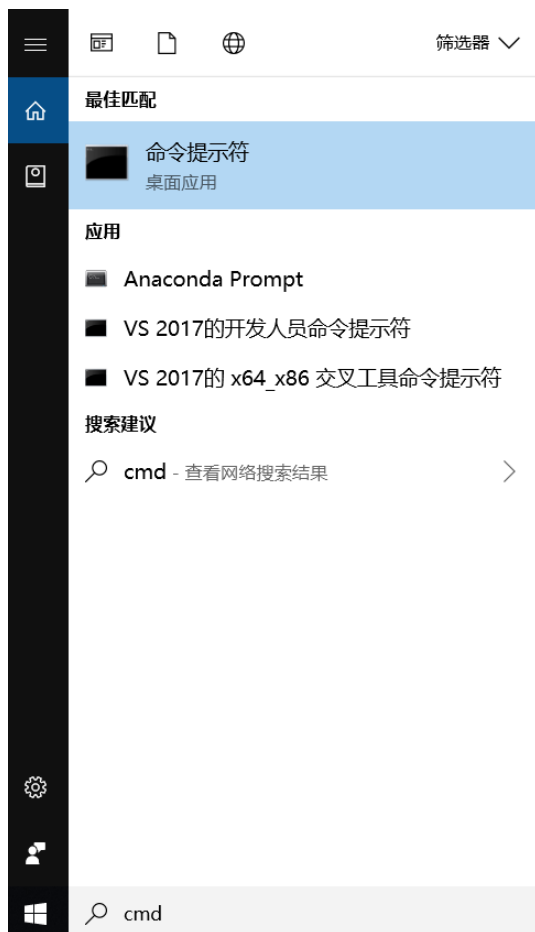
5、编辑 Path 系统变量：



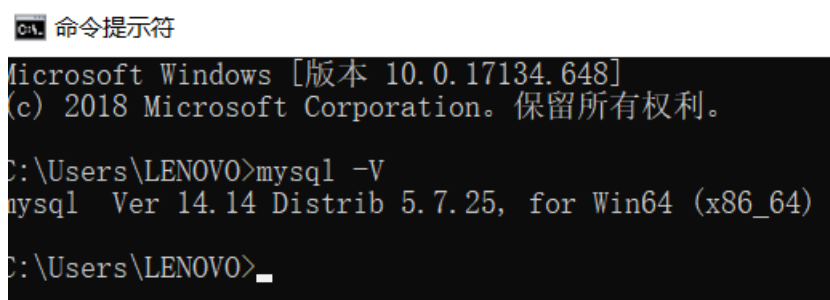
6、点击新建按钮，将 **%MYSQL_HOME%\bin** 添加到 path 变量，而后点击确定按钮



7、在起始菜单中输入 cmd，打开 CMD 命令



8、在跳出的 cmd 窗口中输入 mysql -V (V 必须大写)如下图所示，可以查看 mysql 当前版本



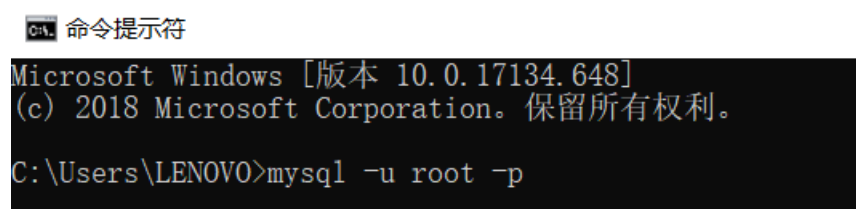
3.3 MySQL 的使用

3.3.1 登录

1、在开始菜单中输入 cmd，启动命令窗口镜像登录



2、在跳出的窗口中输入 `mysql -u root -p` 命令，



3、回车后输入你的 mysql 数据库密码，如教程中的默认密码 **111111**

命令提示符 - mysql -u root -p

```
Microsoft Windows [版本 10.0.17134.648]  
(c) 2018 Microsoft Corporation。保留所有权利。
```

```
C:\Users\LENOVO>mysql -u root -p  
Enter password: *****
```

4、出现如下界面，代表你已经进入 mysql

命令提示符 - mysql -u root -p

```
Microsoft Windows [版本 10.0.17134.648]  
(c) 2018 Microsoft Corporation。保留所有权利。  
  
C:\Users\LENOVO>mysql -u root -p  
Enter password: *****  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2  
Server version: 5.7.25-log MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.  
  
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>
```

3.3.2 创建数据库

1、首先创建一个数据库，如:chendb1，命令如下

格式: `create database <数据库名>;`

`create database chendb1;`

```
mysql> create database chendb1;
```

2、输入回车，可以看到数据库已建好

```
mysql> create database chendb1;  
Query OK, 1 row affected (0.02 sec)  
  
mysql>
```

3.3.3 查看数据库

1、输入 `show databases;`

```
mysql> show databases;_
```

2、输入回车，可以看到我们现在所有的数据库


```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| chendb1 |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.03 sec)

mysql>
```

3.3.4 删除数据库

1、命令测试如下

格式: `drop database <数据库名>;`

这儿我们测试

`drop database chendb1;`

```
mysql> drop database chendb1;_
```

2、输入回车，可以看到已经删除了该数据库

```
mysql> drop database chendb1;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

3.3.5 不报错删除数据库 if exists

1、在数据库已经删除的基础上，再次删除数据库试试，出现了 error

`drop database chendb1;`

```
mysql> drop database chendb1;
ERROR 1008 (HY000): Can't drop database 'chendb1'; database doesn't exist
mysql>
```

2、删除东西的判断可以加上 if exist，删除数据库，删除表都一样

格式: `drop database if exists <数据库名>;`

`drop database if exists chendb1;`

```
mysql> drop database if exists chendb1;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> _
```

此时，不会报错，但是提示 0 个东西成功，哈哈!!!

3.3.6 使用该数据库 use

1、重新创建被我们删除的数据库 chendb1

create database chendb1;

```
mysql> create database chendb1;  
Query OK, 1 row affected (0.00 sec)  
  
mysql>
```

2、使用该数据库，也就是接下来的操作都在该数据库中

格式: use <数据库名>;

use chendb1;

```
mysql> use chendb1;  
Database changed  
mysql>
```

3.3.7 创建表

1、创建表

格式: CREATE TABLE table_name (column_name column_type);

2、例子如下

```
CREATE TABLE `devinfo` (  
  `id` int(10) NOT NULL AUTO_INCREMENT COMMENT '序号',  
  `temp` varchar(6) NOT NULL COMMENT '环境温度',  
  `humi` varchar(6) NOT NULL COMMENT '环境湿度',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='设备传感信息记录表';
```

(1) 如果你不想字段为 NULL 可以设置字段的属性为 NOT NULL， 在操作数据库时如果输入该字段的数据为 NULL， 就会报错。

(2) AUTO_INCREMENT 定义列为自增的属性，一般用于主键，数值会自动加 1。

(3) PRIMARY KEY 关键字用于定义列为主键。 您可以使用多列来定义主键，列间以逗号分隔。

(4) ENGINE 设置存储引擎，CHARSET 设置编码。

3、在 mysql 的 cmd 界面中复制该信息，完成创建

```
mysql> CREATE TABLE `devinfo` (  
->  `id` int(10) NOT NULL AUTO_INCREMENT COMMENT '序号',  
->  `temp` varchar(6) NOT NULL COMMENT '环境温度',  
->  `humi` varchar(6) NOT NULL COMMENT '环境湿度',  
->  PRIMARY KEY (`id`)  
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='设备传感信息记录表';  
Query OK, 0 rows affected (0.06 sec)  
  
mysql>
```

3.3.8 参看所有的表名

1、命令格式

格式: `show tables;`

2、在 `mysql` 的 `cmd` 界面中输入 `show tables;` 而后回车，可以看到该数据库中的所有的表

```
mysql> show tables;
+-----+
| Tables_in_chendb1 |
+-----+
| devinfo            |
+-----+
1 row in set (0.00 sec)
```

3.3.9 参看表的结构

1、命令格式

格式: `desc <表名>;`

`desc devinfo;`

2、在 `mysql` 的 `cmd` 界面中输入 `desc devinfo;` 而后回车，可以看到表的结构

```
mysql> desc devinfo;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id     | int(10)       | NO   | PRI | NULL    | auto_increment |
| temp   | varchar(6)    | NO   |     | NULL    |                |
| humi   | varchar(6)    | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

3.3.10 在表中插入数据

1、插入数据的基础格式

`INSERT INTO table_name (field1, field2,...fieldN) VALUES (value1, value2,...valueN);`

2、例子

`insert into devinfo (temp, humi) values ('26.7', '99.2');`

3、在 `mysql` 的 `cmd` 界面中输入该信息而后回车，可以看到数据已经插入了

```
mysql> insert into devinfo (temp, humi) values ('26.7', '99.2');
Query OK, 1 row affected (0.01 sec)

mysql>
```

3.3.11 查看刚才插入的数据

- 1、查找数据的基础格式，（先简单的全部查找）

select * from table_name;

- 2、在 mysql 的 cmd 界面中输入 select * from devinfo;而后回车，可以看到数据已经插入了

```
mysql> select * from devinfo;
+----+-----+-----+
| id | temp | humi |
+----+-----+-----+
| 1  | 26.7 | 99.2 |
+----+-----+-----+
1 row in set (0.00 sec)
```

3.3.12 修改表名

- 1、基本的操作方法

格式：alter table 旧表名 rename 新表名;

alter table devinfo rename devinfofnew;

- 2、在 mysql 的 cmd 界面中输入 alter table devinfo rename devinfofnew; 而后回车，

```
mysql> alter table devinfo rename devinfofnew;
Query OK, 0 rows affected (0.30 sec)

mysql>
```

- 3、查看表名，在 mysql 的 cmd 界面中输入 show tables;而后回车，可以看到表的名字已修改

```
mysql> show tables;
+-----+
| Tables_in_chendb1 |
+-----+
| devinfofnew       |
+-----+
1 row in set (0.00 sec)
```

3.3.13 修改表字段的数据类型

- 1、基本的操作方法：

格式：alter table 表名 modify 属性名 数据类型;

接着上述的表操作：

alter table devinfofnew modify temp varchar(10);

- 2、在 mysql 的 cmd 界面中输入 alter table devinfofnew modify temp varchar(10); 而后回车，

```
mysql> alter table devinfonew modify temp varchar(10);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

3、在 mysql 的 cmd 界面中输入 `desc devinfonew;` 而后回车，可以看到修改后表的结构

```
mysql> desc devinfonew;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10)       | NO   | PRI | NULL    | auto_increment |
| temp  | varchar(10)   | YES  |     | NULL    |                |
| humi   | varchar(6)    | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

此时，可以看到，temp 的字段数已经修改为 10

3.3.14 修改表字段名称

1、基本的操作方法：

格式：alter table 表名 change 旧属性名 新属性名 新数据类型；

接着上述的表操作：

```
alter table devinfonew change humi humidity varchar(12);
```

2、在 mysql 的 cmd 界面中输入上述命令而后回车，

```
mysql> alter table devinfonew change humi humidity varchar(12);
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

3、在 mysql 的 cmd 界面中输入 `desc devinfonew;` 而后回车，可以看到修改后表的结构

```
mysql> desc devinfonew;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(10)       | NO   | PRI | NULL    | auto_increment |
| temp       | varchar(10)   | YES  |     | NULL    |                |
| humidity   | varchar(12)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

此时，可以看到，humi 的字段名称已修改为 humidity, 字段数已经修改为 12

3.3.15 增加字段

<https://www.cnblogs.com/sqb/p/5806797.html>

1、基本的操作方法：

格式：alter table 表名 add 属性名 1 数据类型 [完整性约束条件] [first | after 属性名 2];

alter table devinfonew add recordtime varchar(20) after id;

2、在 mysql 的 cmd 界面中输入上述命令而后回车，

```
mysql> alter table devinfonew add recordtime varchar(20) after id;
Query OK, 0 rows affected (0.49 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

3、在 mysql 的 cmd 界面中输入 desc devinfonew;而后回车，可以看到修改后表的结构

```
mysql> desc devinfonew;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(10)       | NO   | PRI | NULL    | auto_increment |
| recordtime | varchar(20)   | YES  |     | NULL    |                |
| temp       | varchar(10)   | YES  |     | NULL    |                |
| humidity   | varchar(12)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

此时，可以看到，recordtime 的字段名称添加在 id 字段后面,字段数为 20

3.3.16 删除字段

1、基本的操作方法：

格式：alter table 表名 drop 属性名;

alter table devinfonew drop recordtime;

2、在 mysql 的 cmd 界面中输入上述命令而后回车，

```
mysql> alter table devinfonew drop recordtime;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

3、在 mysql 的 cmd 界面中输入 desc devinfonew;而后回车，可以看到修改后表的结构

```
mysql> desc devinfonew;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(10)       | NO   | PRI | NULL    | auto_increment |
| temp       | varchar(10)   | YES  |     | NULL    |                |
| humidity   | varchar(12)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```


此时，可以看到，recordtime 的字段已被删除了

3.3.17 删除表

1、基本格式：

格式： `drop table if exists <表名>`

`drop table if exists devinfoew`

2、在 mysql 的 cmd 界面中输入上述 `drop table if exists devinfoew;`命令而后回车，

```
mysql> drop table if exists devinfoew;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql>
```

基础四、数据的增删改查四大操作

4.1 完整的创建一个例子

1、登录数据库

- (1) 在开始菜单中输入 cmd，启动命令窗口镜像登录
- (2) 在跳出的窗口中输入 `mysql -u root -p` 命令，如教程中的默认密码 `111111`

```
cmd 命令提示符 - mysql -u root -p
Microsoft Windows [版本 10.0.17134.648]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\LENOVO>mysql -u root -p
Enter password: *****
```

2、查看已有的数据库

```
show databases;
```

3、创建数据库

```
create database chendbtest;
```

4、查看数据库

```
show databases;
```

5、使用该数据库 chendbtest;

```
use chendbtest;
```

6、创建表格 user

```
CREATE TABLE `user` (
  `id` int(10) NOT NULL AUTO_INCREMENT COMMENT '序号',
  `username` varchar(20) NOT NULL COMMENT '用户名',
  `pwd` varchar(20) NOT NULL COMMENT '密码',
  `tel` varchar(15) NOT NULL COMMENT '电话',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='用户记录表';
```

5、查看表格

```
show tables;
```

4.2 插入数据

1、插入数据

插入数据的主要格式：

INSERT INTO table_name (field1, field2,...fieldN) VALUES (value1, value2,...valueN);

```
insert into user(username,pwd,tel) values ('chen1','1111111','13067841001');
insert into user(username,pwd,tel) values ('chen2','1111111','13067841002');
insert into user(username,pwd,tel) values ('chen3','1111111','13067841003');
```

2、在 mysql 的 cmd 界面中依次输入上述三条指令并回车，

```
mysql> insert into user(username,pwd,tel) values ('chen1','1111111','13067841001');
Query OK, 1 row affected (0.00 sec)

mysql> insert into user(username,pwd,tel) values ('chen2','1111111','13067841002');
Query OK, 1 row affected (0.00 sec)

mysql> insert into user(username,pwd,tel) values ('chen3','1111111','13067841003');
Query OK, 1 row affected (0.00 sec)

mysql>
```

4、在 mysql 的 cmd 界面 `select * from user;` 敲入回车后如下图所示

```
mysql> select * from user;
+----+-----+-----+-----+
| id | username | pwd | tel |
+----+-----+-----+-----+
| 1 | chen1 | 1111111 | 13067841001 |
| 2 | chen2 | 1111111 | 13067841002 |
| 3 | chen3 | 1111111 | 13067841003 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

4.3 查找数据

1、条件查询

(1) 格式： `select 字段 from 表名 where 条件;`

`select username, tel from user where username = 'chen2';`

where 后面的条件可以用>、<、>=、<=、!=等多种比较运算符，多个条件之间可以用 or、and 等逻辑运算符

(2) 在 mysql 的 cmd 界面中依次输入上述指令并回车，

```
mysql> select username, tel from user where username = 'chen2';
```

username	tel
chen2	13067841002

```
1 row in set (0.02 sec)

mysql>
```

2、排序和限制

(1) 格式: select * from 表名 [where 条件] [order by field1 [desc/asc], field2 [desc/asc]...];
select * from user order by tel desc; //查询用户表并按电话降序排列。

- 1) .desc 降序排列, asc 升序排列
- 2) .order by 后面可以跟多个不同的排序字段, 每个排序字段都可以有不同的排序顺序。
- 3) .如果排序字段的值一样, 则相同的字段按照第二个排序字段进行排序。
- 4) .如果只有一个排序字段, 则字段相同的记录将会无序排列。

(2) 在 **mysql** 的 **cmd** 界面中依次输入上述指令并回车,

```
mysql> select * from user order by tel desc;
```

id	username	pwd	tel
3	chen3	1111111	13067841003
2	chen2	1111111	13067841002
1	chen1	1111111	13067841001

```
3 rows in set (0.00 sec)

mysql>
```

3.查询不重复的记录

(1) 格式: select distinct 字段 from 表名;
select distinct pwd from user; //查询密码不相同的用户;
select distinct username, pwd from user; //查询用户名和密码同时不同的学生

- 1). distinct 必须放在最开头
- 2). distinct 只能使用需要去重的字段进行操作。 ----也就是说我 distinct 了 name, age 两个字段, 我后面想根据 id 进行排序, 是不可以的, 因为只能 name, age 两个字段进行操作。
- 3). distinct 去重多个字段时, 含义是: 几个字段 同时重复 时才会被 过滤。

(2) 在 **mysql** 的 **cmd** 界面中依次输入上述指令并回车,

```
mysql> select distinct pwd from user;
```

pwd
1111111

```
1 row in set (0.01 sec)

mysql> select distinct username, pwd from user;
```

username	pwd
chen1	1111111
chen2	1111111
chen3	1111111

```
3 rows in set (0.00 sec)

mysql>
```

4.聚合查询

select 字段 fun_name from 表名 [where 条件] [group by field1,field2...] [with rollup] [having 条件];

- 1). fun_name 表示要做的聚合操作，也就是说聚合函数，常用的有：sum(求和)、count(*) (记录数)、max(最大值)、min(最小值)。
- 2). group by 关键字 表示要进行分类聚合的字段。比如要按照部门分类统计员工数量，部门就应该写在 group by 后面。
- 3). with rollup 是可选语法，表明是否对分类聚合后的结果进行再汇总
- 4). having 关键字表示对分类后的结果再进行条件过滤。

公司员工表 A 如下 (编号, 姓, 名, 薪水)：

	id	xing	ming	salary
▶	1	Guo	Chuan	23
	2	Li	Si	75
	3	Li	Siyuan	51
	4	Guo	Si	26
	5	Guo	Siyuan	60
	6	Guo	Chuan	92
	7	Li	Siyuan	100
	8	Guo	Siyuan	2
▶	9	Guo	Songtian	71

(1)统计总人数

```
select count(1) from A;
```

	count(1)
▶	9

(2)统计各个姓的人数

```
select xing, count(1) from A group by xing;
```

	xing	count(1)
▶	Guo	6
	Li	3

(3)既要统计各个姓的人数，又统计总人数

```
select xing, count(1) from A group by xing with rollup;
```

	xing	count(1)
▶	Guo	6
	Li	3
	HULL	9

(4)统计人数大 4 的姓

```
select xing, count(1) from A group by xing having count(1)>4;
```

	xing	count(1)
▶	Guo	6

(5) 统计薪水总额，最低薪资，最高薪资

`select count(1),min(salary),max(salary) from A;`

	count(1)	min(salary)	max(salary)
▶	9	2	100

5.表连接

表连接分为内连接和外连接。

他们之间最主要的区别：内连接仅选出两张表中互相匹配的记录，外连接会选出其他不匹配的记录。

以下是**员工表 staff** 和**职位表 deptno**:

	id	name
▶	1	Steve
	2	Jobs
	3	Drake
	4	Tom
	5	Jay
	6	LI
	7	Chuan
	8	Anna
	9	Satan

	name	deptname
▶	Steve	tech
	Jobs	seal
	Drake	tech
	Tom	seal
	Jay	tech
	LI	hr
	Chuan	ceo
	Anna	seal
		driver

(1) 内连接

`select staff.name,deptname from staff,deptno where staff.name=deptno.name;`

	name	deptname
▶	Steve	tech
	Jobs	seal
	Drake	tech
	Tom	seal
	Jay	tech
	LI	hr
	Chuan	ceo
	Anna	seal

外连接 分为左连接和右连接

左连接：包含所有左边表中的记录，甚至是右边表中没有和他匹配的记录。

右连接：包含所有右边表中的记录，甚至是右边表中没有和他匹配的记录。

(2) 外连接(左连接):

`select staff.name,deptname from staff left join deptno on staff.name=deptno.name;`

	name	deptname
▶	Steve	tech
	Jobs	seal
	Drake	tech
	Tom	seal
	Jay	tech
	LI	hr
	Chuan	ceo
	Anna	seal
	Satan	NULL

(3) 外连接（右连接）：

`select deptname,deptno.name from staff right join deptno on deptno.name=staff.name;`

	deptname	name
▶	tech	Steve
	seal	Jobs
	tech	Drake
	seal	Tom
	tech	Jay
	hr	LI
	ceo	Chuan
	seal	Anna
	driver	

4.4 删除数据

1、查看数据库中有什么

在 mysql 的 cmd 界面 `select * from user;` 敲入回车后如下图所示

```
mysql> select * from user;
+----+-----+-----+-----+
| id | username | pwd | tel |
+----+-----+-----+-----+
| 1 | chen1 | 1111111 | 13067841001 |
| 2 | chen2 | 1111111 | 13067841002 |
| 3 | chen3 | 1111111 | 13067841003 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

2、基本格式

`DELETE FROM table_name [WHERE Clause]`

- (1) 如果没有指定 WHERE 子句，MySQL 表中的所有记录将被删除。
- (2) 你可以在 WHERE 子句中指定任何条件
- (3) 您可以在单个表中一次性删除记录。
- (4) 当你想删除数据表中指定的记录时 WHERE 子句是非常有用的

3、我们的删除例子

```
DELETE FROM user where tel = '13067841002'
```

4、在 mysql 的 cmd 界面上输入命令，敲入回车后如下图所示

```
mysql> delete from user where tel='13067841002'
-> ;
Query OK, 1 row affected (0.25 sec)

mysql>
```

5、在 mysql 的 cmd 界面输入 `select * from user;` 敲入回车后如下图所示

```
mysql> select * from user;
+----+-----+-----+-----+
| id | username | pwd | tel |
+----+-----+-----+-----+
| 1 | chen1 | 1111111 | 13067841001 |
| 3 | chen3 | 1111111 | 13067841003 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

可以看到，tel='13067841002'的用户记录已经被删除了

4.5 修改数据

1、查看数据库中有什么

在 mysql 的 cmd 界面输入 `select * from user;` 敲入回车后如下图所示

```
mysql> select * from user;
+----+-----+-----+-----+
| id | username | pwd | tel |
+----+-----+-----+-----+
| 1 | chen1 | 1111111 | 13067841001 |
| 3 | chen3 | 1111111 | 13067841003 |
+----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

2、基本格式

```
UPDATE table_name SET field1=new-value1, field2=new-value2 [WHERE Clause]
```

- (1) 你可以同时更新一个或多个字段。
- (2) 你可以在 WHERE 子句中指定任何条件。

(3) 你可以在一个单独表中同时更新数据。

(4) 当你需要更新数据表中指定行的数据时 WHERE 子句是非常有用的。

3、基本的例子

```
UPDATE user SET username='chennew', tel='18933221155' WHERE id=3;
```

4、在 mysql 的 cmd 界面上述命令，敲入回车后如下图所示

```
mysql> UPDATE user SET username='chennew', tel='18933221155' WHERE id=3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

5、在 mysql 的 cmd 界面 `select * from user;` 敲入回车后如下图所示

```
mysql> select * from user;
```

id	username	pwd	tel
1	chen1	1111111	13067841001
3	chennew	1111111	18933221155

```
2 rows in set (0.00 sec)

mysql>
```