

#MySQL安装和核心语法

朱俊泽

2351114

##1MySQL安装

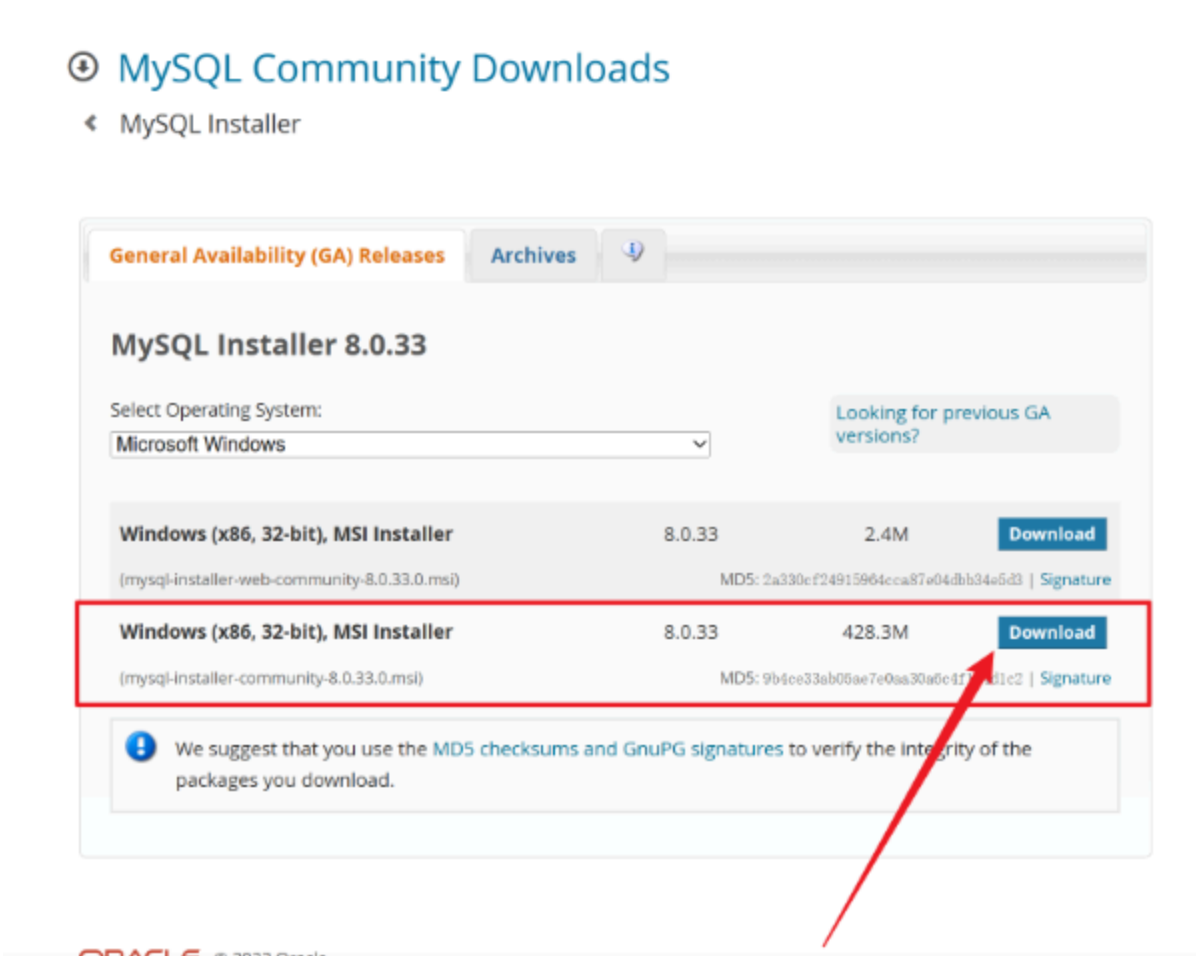
###1.1检查

在安装之前, 先确定一下, 电脑上之前有没有安装MySQL, 看看有没有这个路径。

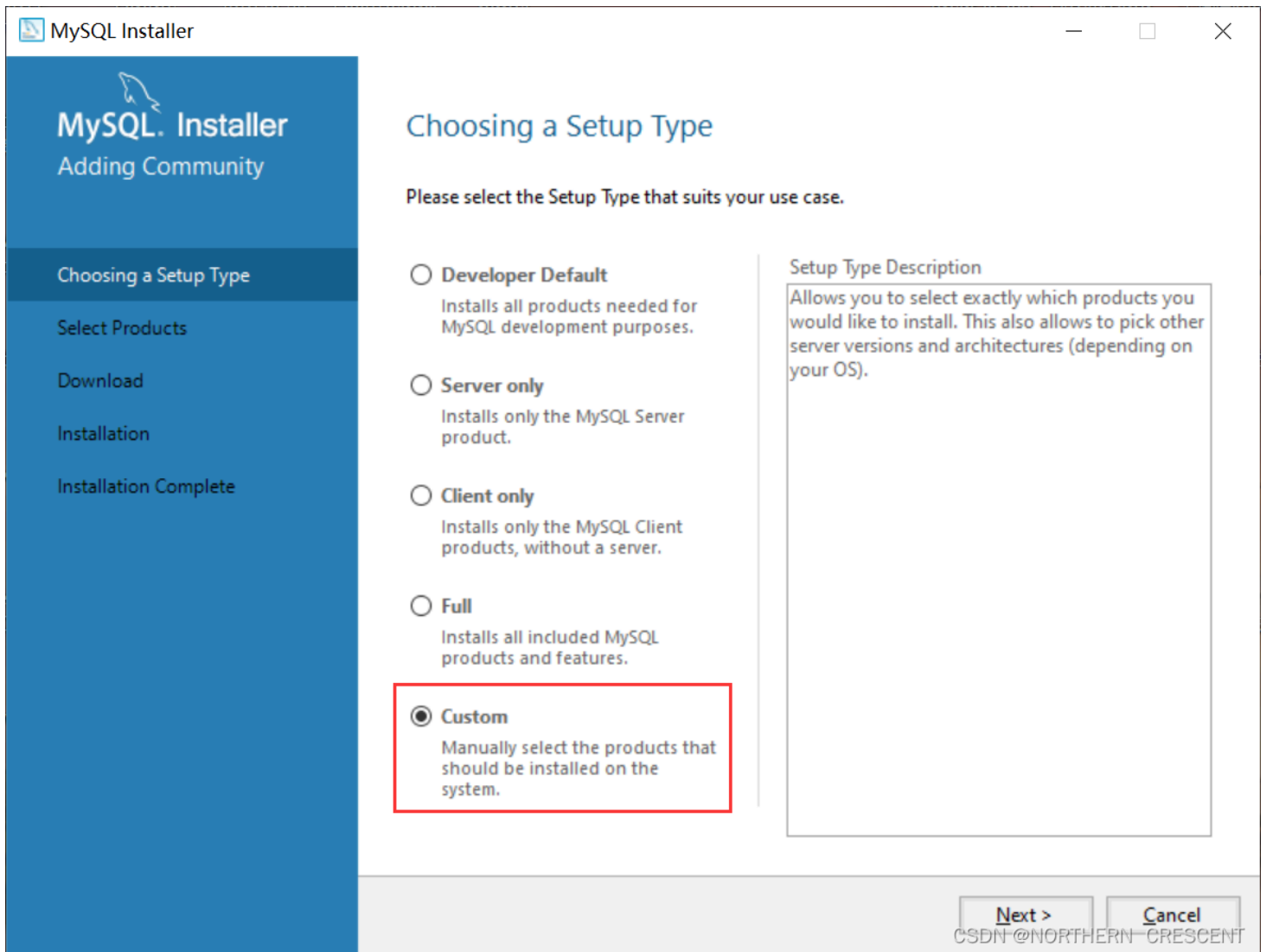


###1.2官网检查下载

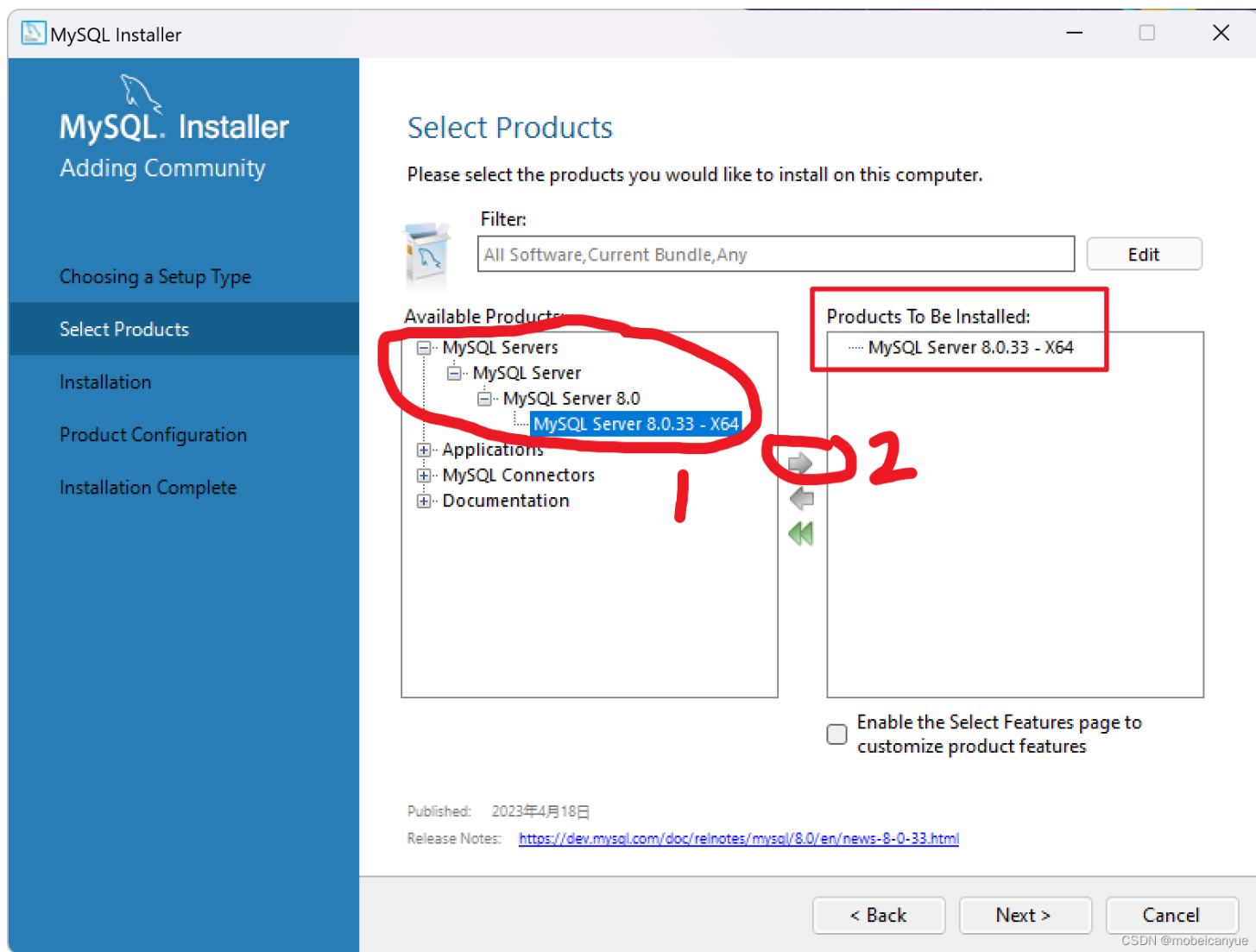
之后进入到官方网站进行下载



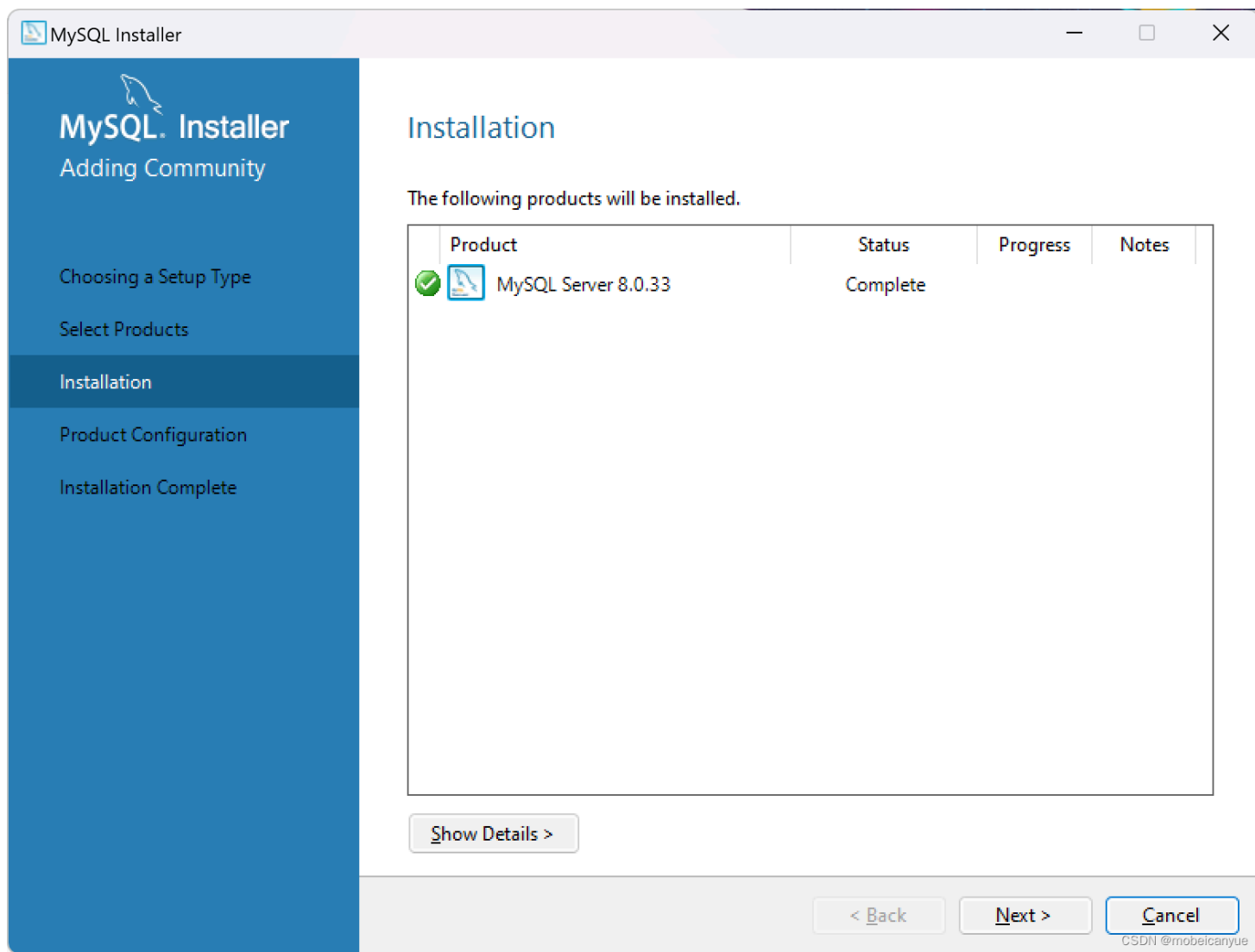
###1.3安装选择



选择custom模式进行自定义安装



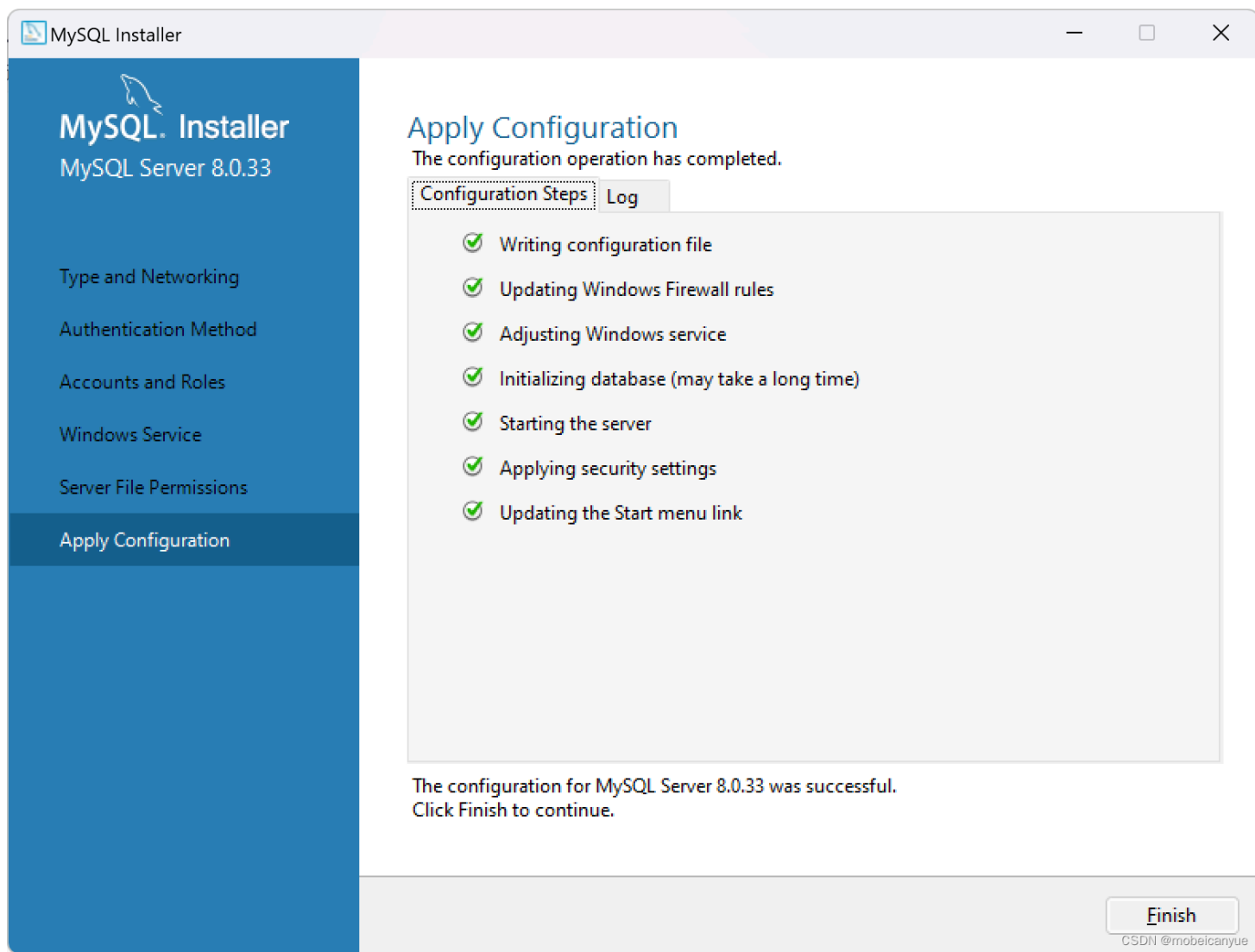
之后选择对应的MySQL server进行下载安装



如上的界面就是安装好了

###1.4work bench安装

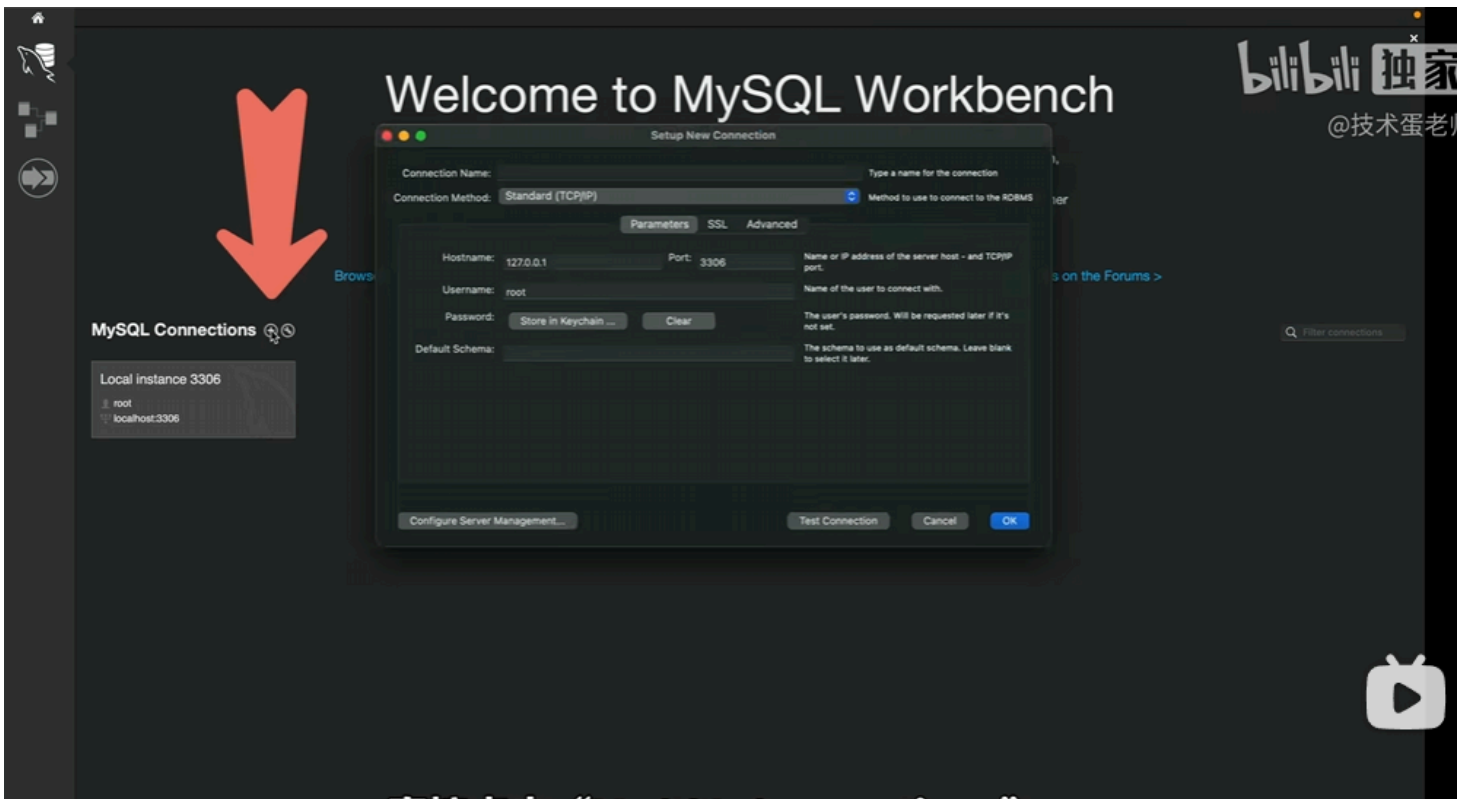
安装好扩展work bench



##2 创建与服务器的链接

###2.1 本地服务器创建

直接点击MySQL connection



###2.2 进入本地服务器

直接点击刚刚默认的本地服务器连接进入服务器

###3 MySQL的基本使用

###3.1 创建数据库

CREATE DATABASE 数据库名字

然后快捷执行--execute

就会发现有生成好的数据库

1 • CREATE DATABASE egg;

Output

Action Output

#	Time	Action	Message
✓ 1	13:37:47	CREATE DATABASE egg	1 row(s) affected

SCHEMAS

Filter objects

- egg
- sys

如果再次执行同样的创造数据库就会出错，原因是数据库的名字不能一样

###3.2 指定数据库

USE 数据库名字

###3.3 创造表格

CREATE TABLE 表格名字

{

数据名字 类型....

}

典型类型:

INT 整形数据

VARCHAR() 字符串

DATE 时间

```
create table eggs_record(  
    num INT,  
    egg_name VARCHAR(10),  
    egg_time DATE,  
);
```

数据类型有空和非空两种属性, NULL和NOT NULL

然后可以设置primary key, 主键, 必须非空

```
create table eggs_record(  
    id INT primary key,  
    egg_name VARCHAR(10) not null,  
    egg_time DATE null,  
);
```

###3.4 插入数据

INSERT INTO 数据库名字.表格名字

(列名...)

VALUES

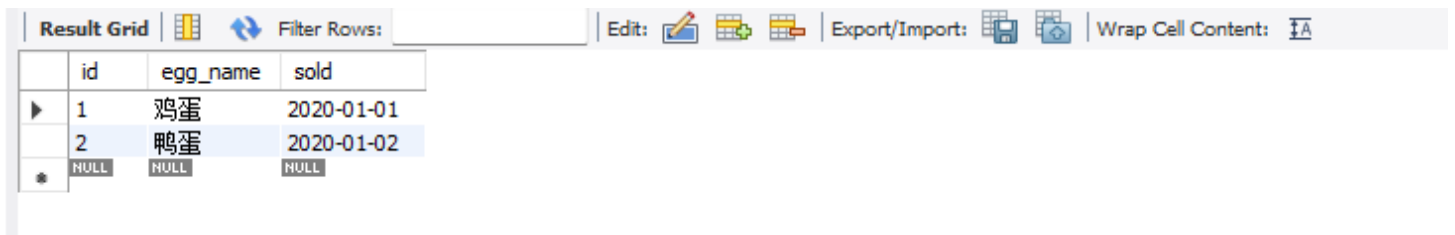
(数值....)

USE egg;

```
create table eggs_record(  
    id INT primary key auto_increment eggs_record,  
    egg_name VARCHAR(10) not null,  
    sold DATE null  
);
```

```
insert into egg.eggs_record(id,egg_name,sold)  
values(1,'鸡蛋','2020-01-01');
```

```
insert into egg.eggs_record(id,egg_name,sold)  
values(2,'鸭蛋','2020-01-02');
```



	id	egg_name	sold
▶	1	鸡蛋	2020-01-01
	2	鸭蛋	2020-01-02
*	NULL	NULL	NULL

###3.5 改变表格

####3.5.1 添加新数据

ALTER TABLE 表格名字

ADD 列名 数据类型

就可以改变表格

####3.5.2 改变旧数据

UPDATE 数据库名.表格名字

SET 值

WHERE 定位id

```
alter table egg.eggs_record  
add stock int null;
```

```
update egg.eggs_record  
set sold='2022-06-06'  
where id=2;
```

修改结果如下

Result Grid				
	id	egg_name	sold	stock
▶	1	鸡蛋	2020-01-01	NULL
	2	鸭蛋	2022-06-06	NULL
*	NULL	NULL	NULL	NULL

###3.6 删除数据

####3.6.1删除一条记录

DELETE FROM 数据库名.表格名
WHERE 索引

```
delete from egg.egg_record
where id=1;
```

Result Grid				
	id	egg_name	sold	stock
▶	2	鸭蛋	2022-06-06	NULL
*	NULL	NULL	NULL	NULL

####3.6.2删除表格

DROP TABLE 数据库名.表格名

####3.6.3删除数据库

DROP DATABASE 数据库名

###3.7 查找数据

####3.7.1 查看表格全部内容

```
select *
from 表格名
```

```
select *
from eggs_record;
```

```
1 • select *
2   from eggs_record;
```

Result Grid			
Filter Rows: <input type="text"/>			
Edit:			
	id	egg_name	sold
▶	1	鸡蛋	2020-01-01
	2	鸭蛋	2020-01-02
	3	鹅蛋	2020-01-03
*	NULL	NULL	NULL

####3.7.2 查看表格某列内容

select 列名...

from 表格名

```
select id,sold
from eggs_record;
```

```
1 • select id,sold
2   from eggs_record;
```

Result Grid		
Filter Rows: <input type="text"/>		
Edit:		
	id	sold
▶	1	2020-01-01
	2	2020-01-02
	3	2020-01-03
*	NULL	NULL

####3.7.3 特殊限制

#####3.7.3.1 去重

select distinct *

from 表格名

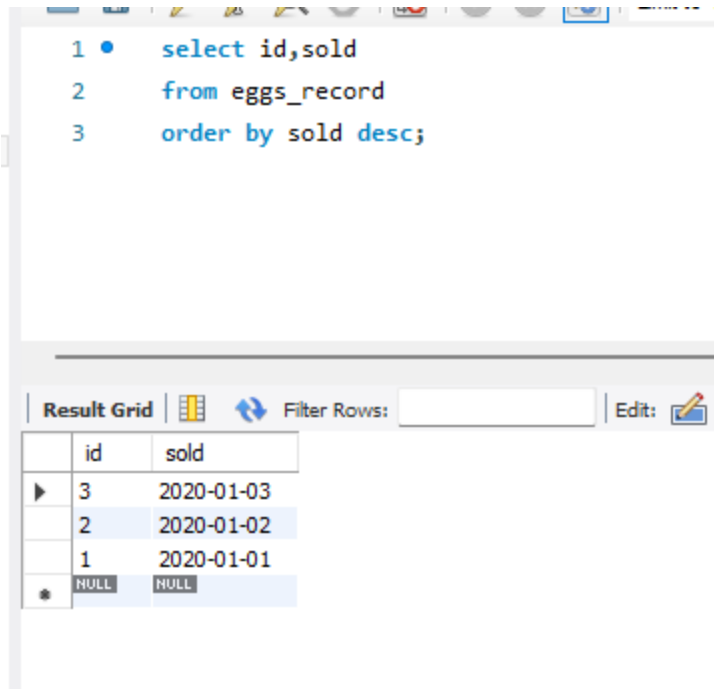
#####3.7.3.2 排序

select *

from 表格名

order by 列名 asc/desc

就是以列名的 升序/降序 排列



#####3.7.4 过滤

select *

from 表格名

where 条件

order by 列名 asc/desc

有如下的限制内容

比较运算符

=	等于
!= 或 <>	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于
BETWEEN	两值之间
IN	一组值里
LIKE	相似匹配

逻辑运算符

AND	与
OR	或
NOT 或 !	非

可以进行如下限制

```
1 • select id,sold
2   from eggs_record
3  where id<=2
4  order by sold desc;
```

Result Grid		
Filter Rows:		
	id	sold
▶	2	2020-01-02
	1	2020-01-01
*	HULL	HULL

####3.7.5 合并

```
select *
from 表格名
inner join 表格名
on 条件
```

用条件来取两个表格交集--inner 并集--union

保留上表格并且加入下层表格符合条件的--left

保留下表格并且加入上层表格符合条件的--right