

建立新用户与建库建表操作

2025 年 3 月 14 日

一、主要任务：

第二节实验课的主要任务是配置环境并且在 workbench 中完成创建数据库，创建表，向表中添加数据等操作

二、实现方法：

配置环境：在 vs code 中配置 python 解释器，用 pip 命令下载 pymysql 包，在官网下载 mysql 及其图形化编辑页面 workbench

创建用户：在 workbench 页面找到 Users and Privileges，根据指引一步步完成创建用户的工作

创建数据库：使用 sql 语句 `create database if not exists dbsclab2025;` 即可创建一个叫做 dbsclab2025 的数据库

创建表：使用 sql 语句形如 `create table 表名 (`
`字段 1 字段 1 类型约束 1 comment 注释 1,`
`字段 2 字段 2 类型约束 2 comment 注释 2,`
`.....`

`字段 n 字段 n 类型约束 n comment 注释 n`
`)comment 表注释;`

进行建表

使用形如 `insert into 表名 values(字段 1 数据, 字段 2 数据....., 字段 n 数据)` 将数据插入表中

三、成果展示：

1. 配置环境, 在 vs code 中连接数据库并且检测版本号

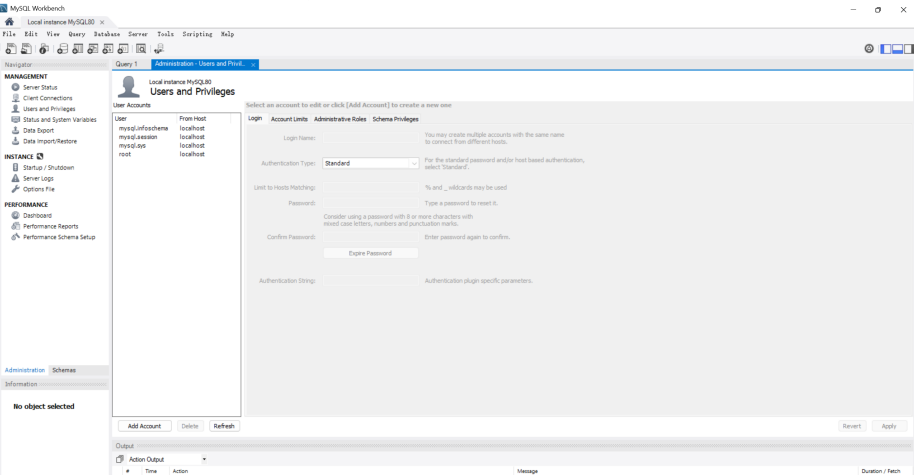
```
mysql> testsql.py> ...
1 import pymysql
2 conn=pymysql.connect(host='localhost',user='root',password='Aa17526909261',database='sys',charset='utf8')
3 cursor=conn.cursor()
4 sql = 'select version()'
5 cursor.execute(sql)
6 retdata = cursor.fetchall()
7 for row in retdata:
8     print(row[0])
9 cursor.close()
10 conn.close()

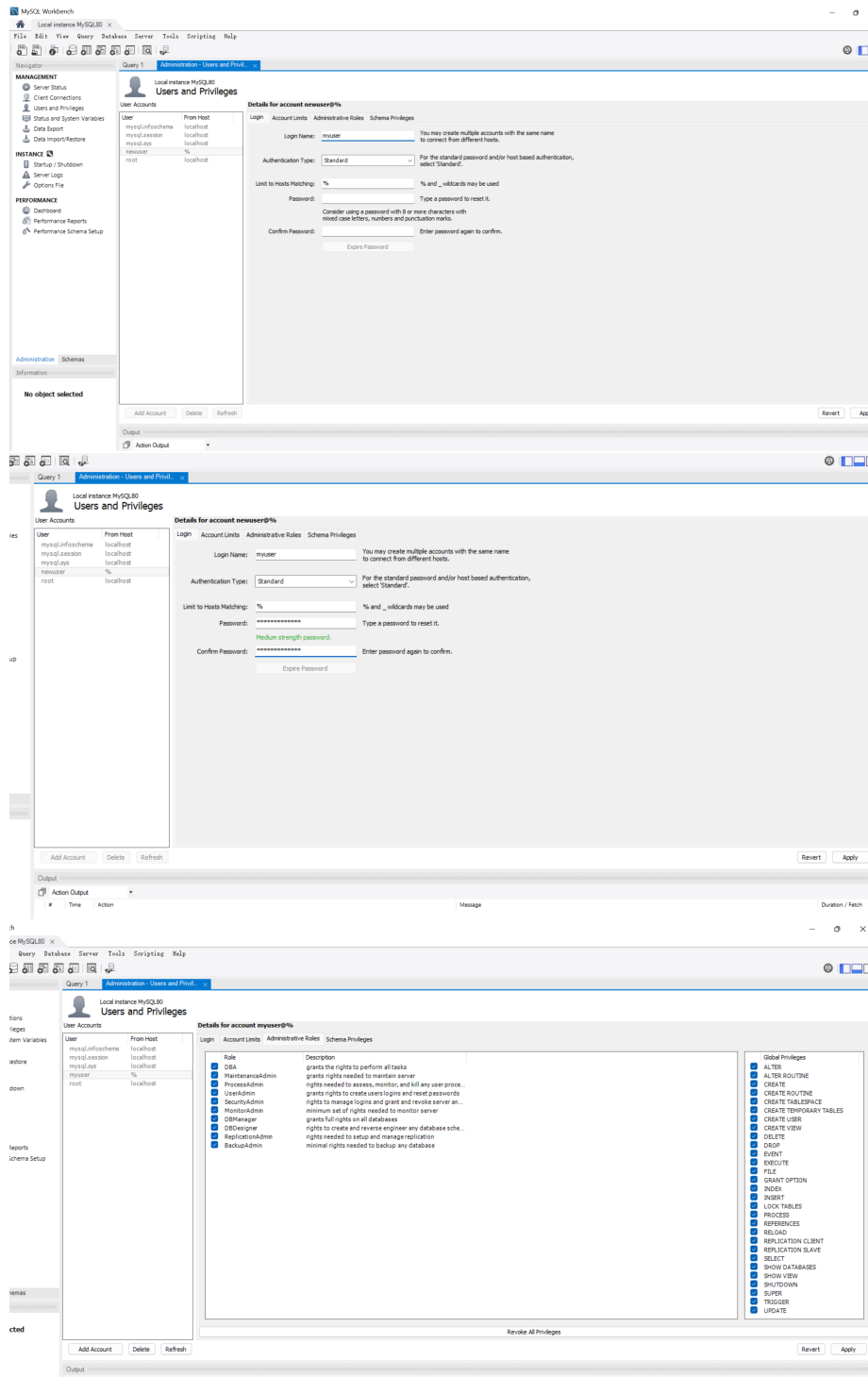
问题 输出 调试控制台 终端 窗口

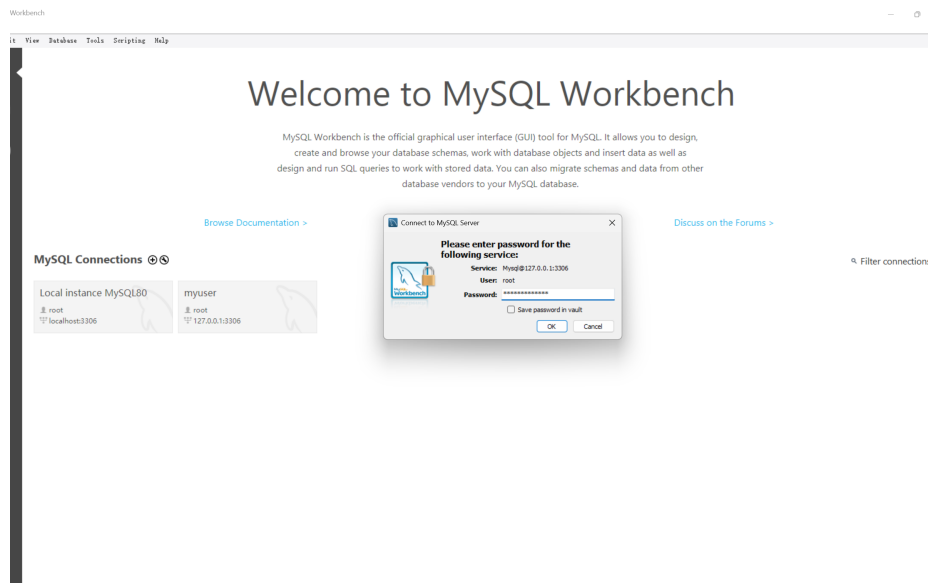
PS D:\code\python> & "C:/Users/the_dark/AppData/Local/Programs/Python/Python312/python.exe" d:/code/python/mysql/testsql.py
8.0.41
PS D:\code\python>
```

2. 创建新用户

根据以下流程创建新用户并连接



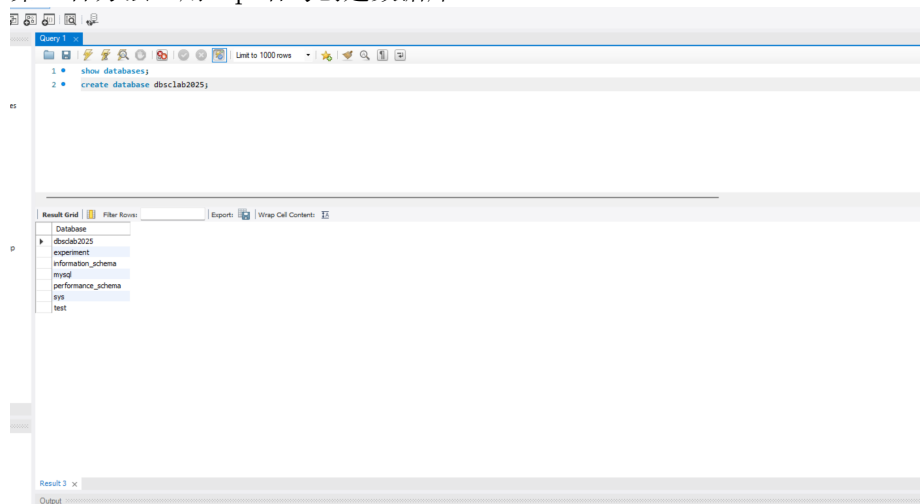




可以看到新用户已创建完毕、

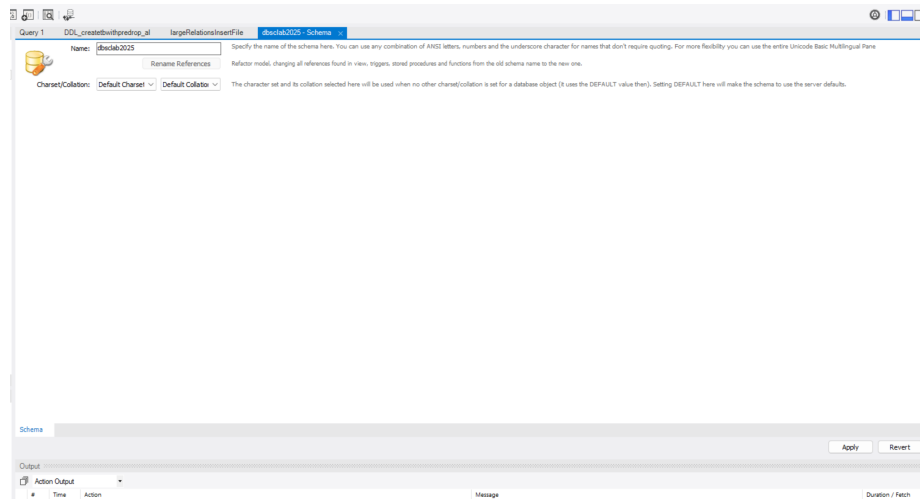
3. 创建数据库 dbslab2025

第一种方法：用 sql 语句创建数据库



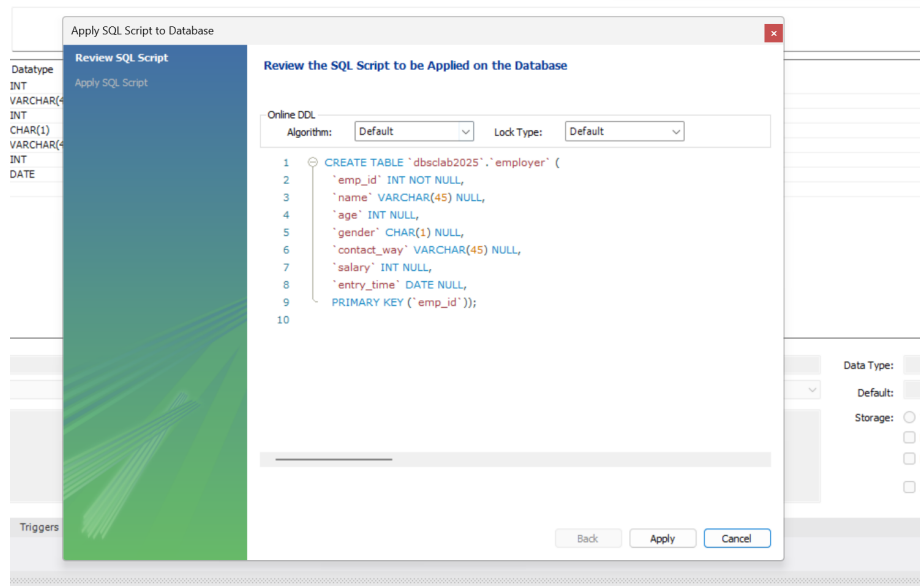
可以看到数据库已创建完毕

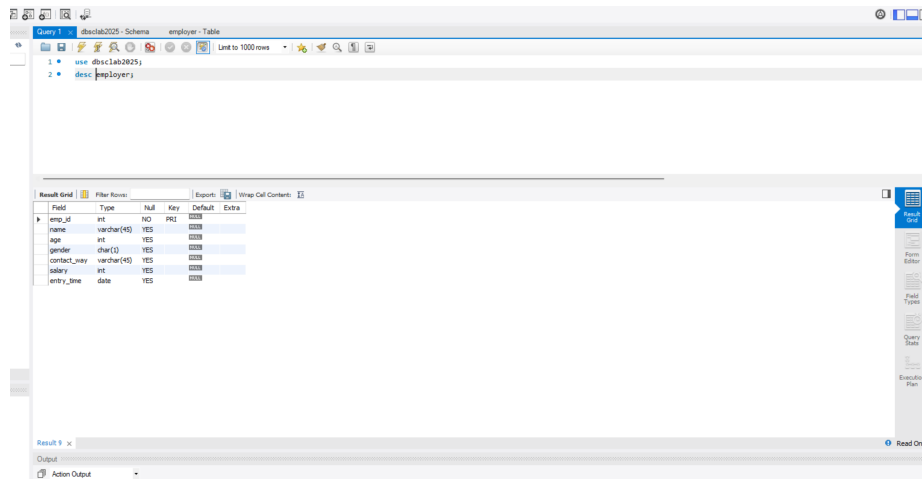
第二种方法：用 workbench 创建数据库



4. 建表

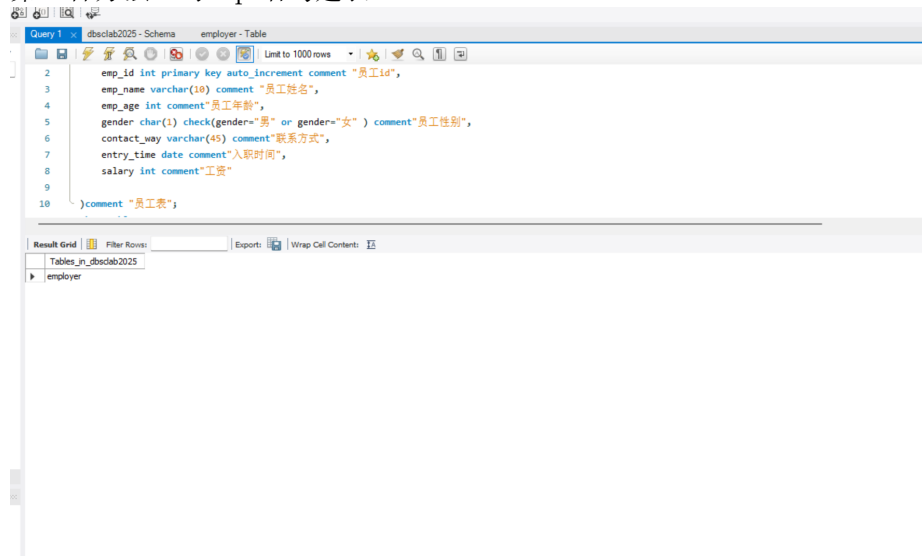
第一种方法：用 workbench 建表

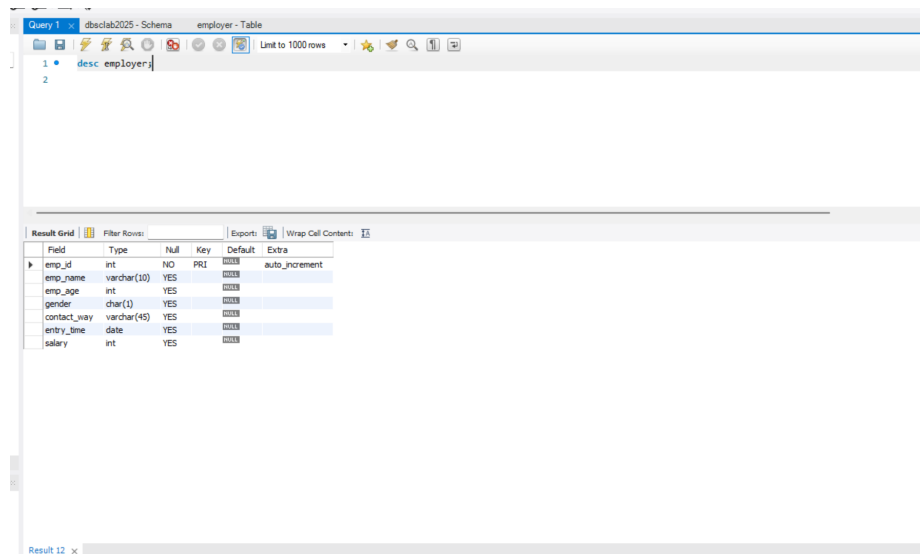




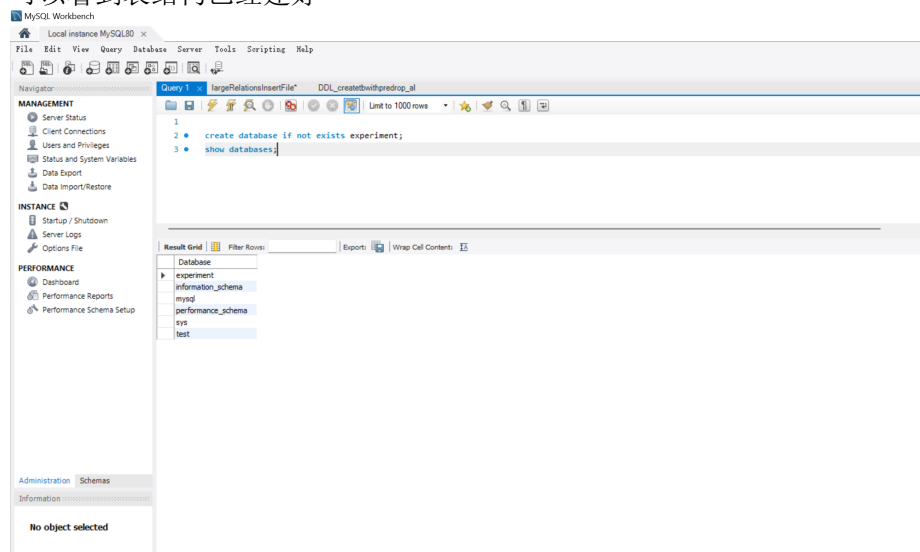
从这里可以看到表结构

第二种方法：写 sql 语句建表





可以看到表结构已经建好



可以看到 experiment 数据库已经被建好

第三种方法：用 python 连接数据库建表

```
import pymysql
```

```
# 建立数据库连接（建议直接指定目标数据库）
```

```
conn = pymysql.connect(
```

```

        host='localhost',
        user='root',
        password='yourpassword',
        database='dbslab2025', # 直接指定目标数据库
        charset='utf8mb4'      # 保持和表定义一致
    )

    cursor = conn.cursor()

    try:
        # 创建表前先删除可能存在的旧表
        cursor.execute("DROP TABLE IF EXISTS emp")

        # 优化后的建表语句
        sql = """
        CREATE TABLE 'emp' (
            'emp_id' int NOT NULL AUTO_INCREMENT COMMENT '员工id',
            'emp_name' varchar(10) DEFAULT NULL COMMENT '员工姓名',
            'emp_age' int DEFAULT NULL COMMENT '员工年龄',
            'gender' char(1) DEFAULT NULL COMMENT '员工性别',
            'contact_way' varchar(45) DEFAULT NULL COMMENT '联系方式',
            'entry_time' date DEFAULT NULL COMMENT '入职时间',
            'salary' int DEFAULT NULL COMMENT '工资',
            PRIMARY KEY ('emp_id'),
            CONSTRAINT 'employer_chk_gender' CHECK ('gender' IN ('男', '女'))
        ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='员工表'
        """
        cursor.execute(sql)

        # 提交事务
        conn.commit()

        # 查看表结构

```



```

print("表结构：")
cursor.execute("DESC emp")
for row in cursor.fetchall():
    print(row[0], row[1], row[2])

# 查看所有表
print("\n数据库中的表：")
cursor.execute("SHOW TABLES")
for row in cursor.fetchall():
    print(row[0])

finally:
    # 清理资源
    cursor.close()
    conn.close()

```

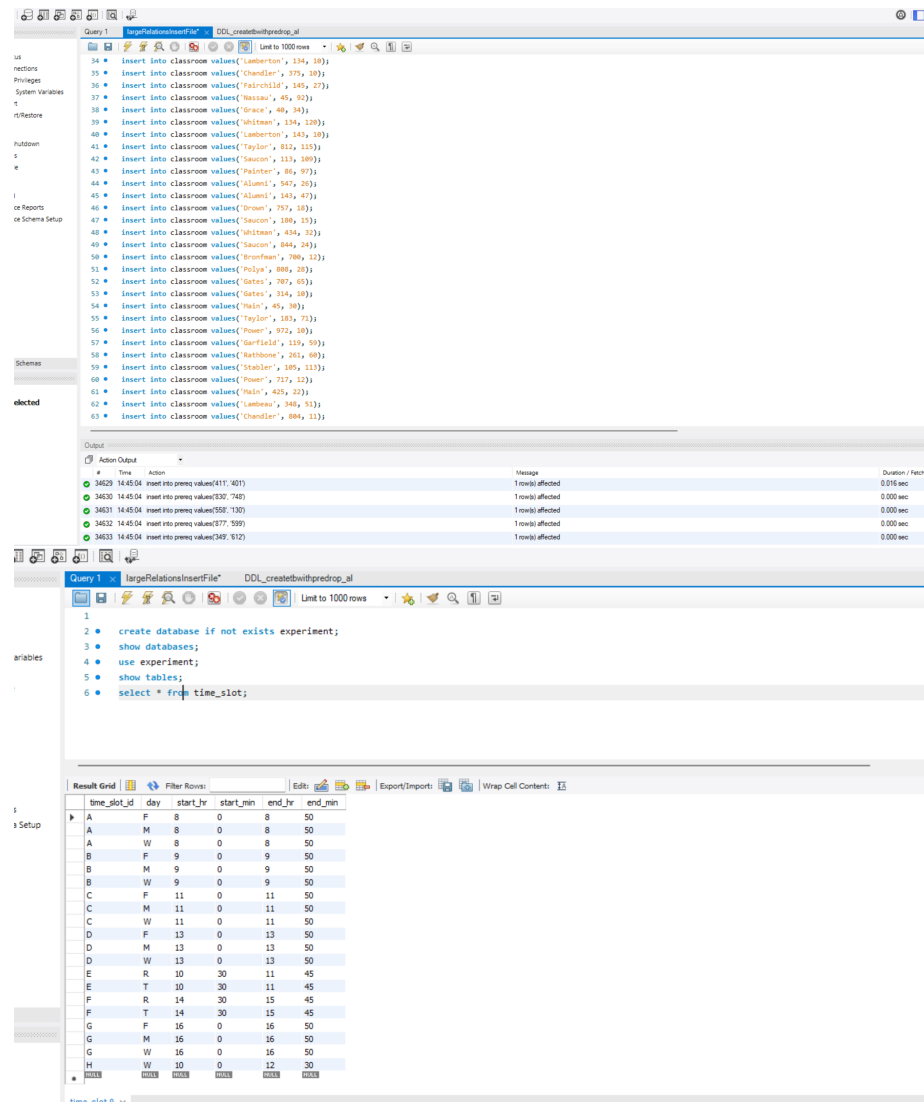
```

20 SQL > ""
21 CREATE TABLE 'emp' (
22   'emp_id' int NOT NULL AUTO_INCREMENT COMMENT '员工id',
23   'emp_name' varchar(10) DEFAULT NULL COMMENT '员工姓名',
24   'emp_age' int DEFAULT NULL COMMENT '员工年龄'
25 )
26
27 > & "C:/Users/the_dark/AppData/Local/Programs/Python/Python312/python.exe" d:/code/python/mysql/createtable.py
28
29 表结构:
30 emp_id int NO
31 emp_name varchar(10) YES
32 emp_age int YES
33 gender char(1) YES
34 contact_way varchar(45) YES
35 entry_time date YES
36 salary int YES
37
38 数据库中的表:
39 advice
40 classroom
41 course
42 department
43 emp
44 employer
45 instructor
46 prereq
47 section
48 student
49 takes
50 teaches
51 time_slot
52
53 PS D:\code\python\mysql>

```

输出结果

5. 添加数据



用 select 语句可看表中数据，与 time_slot 的添加数据语句完全一致

四、结论：

用 sql 语句可以很便捷地执行创建数据库，创建表结构，添加数据等操作

本次建表用到 int, varchar 字符串, char 字符串, 使用 check 约束性别使其必须是男或者女, 将 id 声明为主键并且会自动增加

1