

# 浙江大学



## 《数据库系统》 实验报告

作业名称 :	SQL数据定义和操作
姓 名 :	王晓宇
学 号 :	3220104364
电子邮箱 :	3220104364@zju.edu.cn
联系电话 :	19550222634
授课教师 :	孙建伶

2024 年 3 月 11 日

## 实验名称 SQL数据定义和操作

### 1 实验目的

1. 掌握关系数据库语言SQL的使用。
2. 使所有的SQL作业都能上机通过。

### 2 实验环境

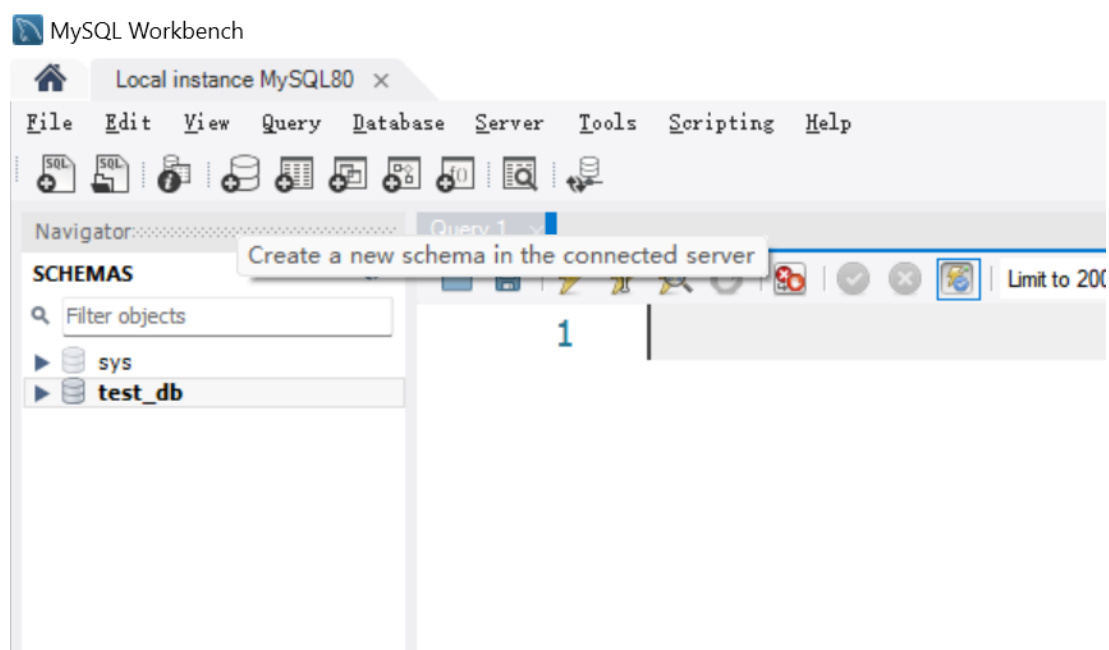
- 操作系统：Windows 11 23H2
- 数据库管理系统：MySQL 8.0.36
- 工具：MySQL workbench 8.0

### 3 实验流程

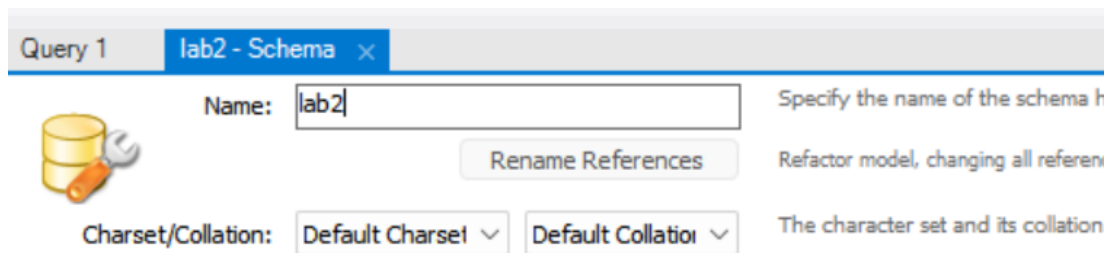
在实验过程中，截图所用命令和实验结果

#### 3.1 建立数据库

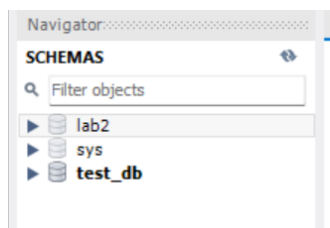
利用图形化界面新建数据库



如图所示，填入 lab2 作为新建数据库的名称。(由于SQL语法对大小写不敏感，这里使用 Lab2 也会被强制转为 lab2)



下图是建好的数据库，由于实验一也在本地操作过，这里也会有实验一建立的 test\_db 的数据库列出，本次实验在 lab2 schema上面操作。



## 3.2 数据定义

本次在网上找了一些数据库示例，将在此数据库上做查询操作。

### 3.2.1 表的建立/删除/修改

表的建立，以下是SQL语句

```
use lab2;#表明你使用的数据库
## 第一句是指明SQL语句作用的对象。
##后面的语句是创建了学生表、课程表和成绩表，并指明主键。
# 学生表
CREATE TABLE student(
s_id VARCHAR(20),
s_name VARCHAR(20),
s_birth VARCHAR(20),
s_sex VARCHAR(10) NOT NULL DEFAULT '',
```

```

PRIMARY KEY(s_id)
);

# 课程表

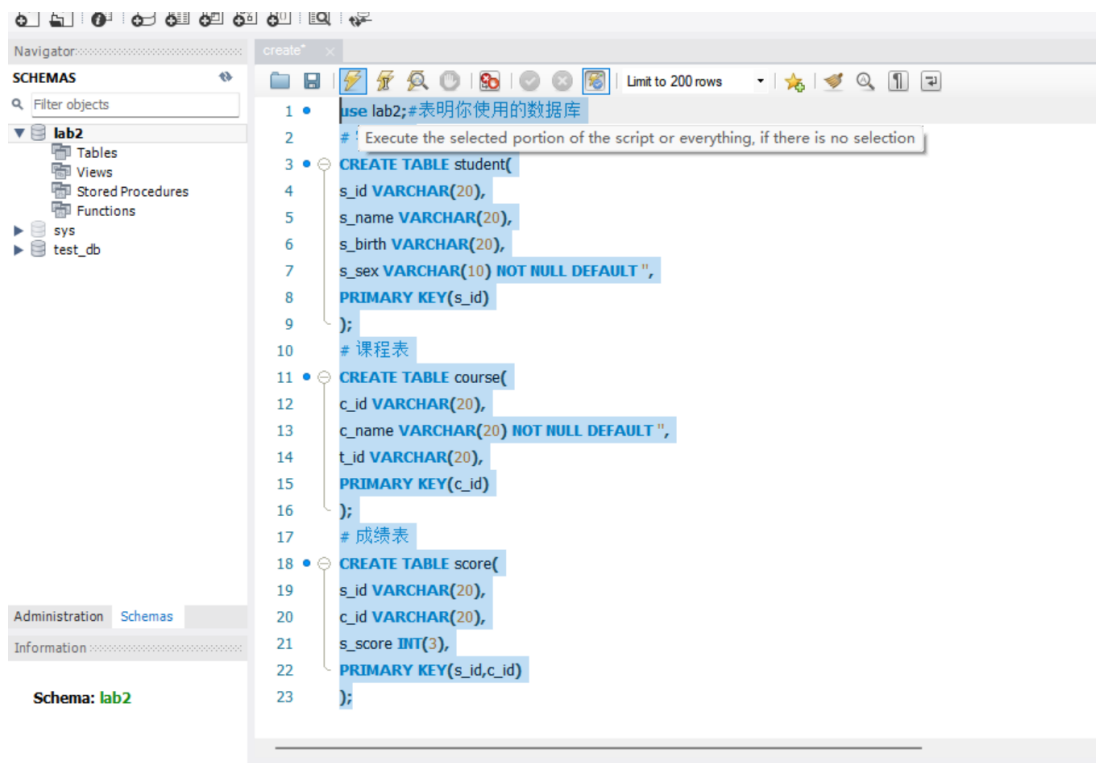
CREATE TABLE course(
c_id VARCHAR(20),
c_name VARCHAR(20) NOT NULL DEFAULT '',
t_id VARCHAR(20),
PRIMARY KEY(c_id)
);

# 成绩表

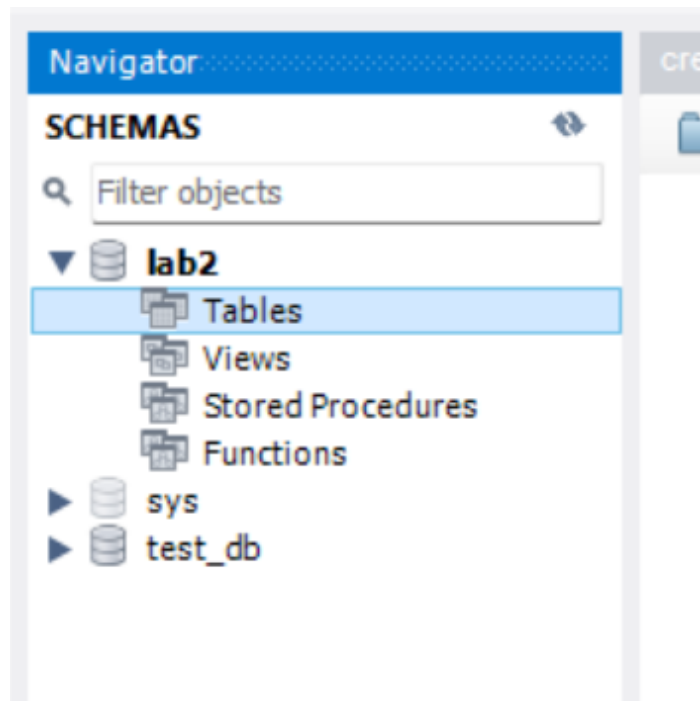
CREATE TABLE score(
s_id VARCHAR(20),
c_id VARCHAR(20),
s_score INT(3),
PRIMARY KEY(s_id,c_id)
);

```

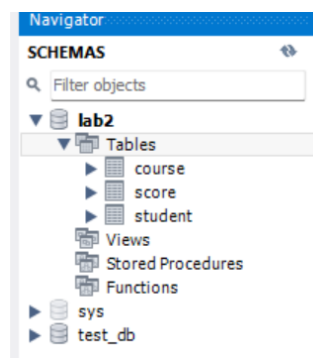
全部选中语句，点击 ⚡ 图标即可执行。



此时应该点击SCHEMA右上角的刷新按钮



此时显示出刚刚新建好的三个表

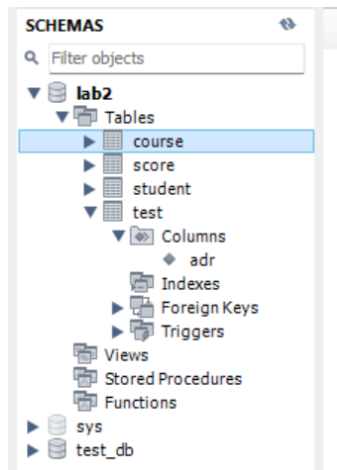
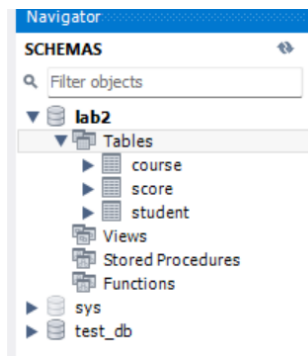


- 表的修改与删除

#为了不影响已建好的数据，此处新建一个test表来进行修改删除

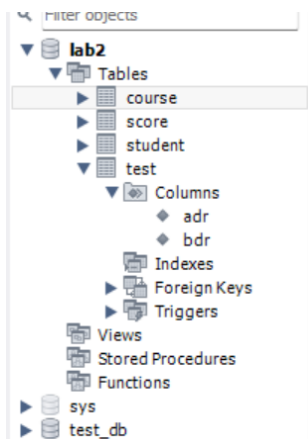
```
create table test (  
adr varchar(20)  
);  
alter table test add bdr varchar(20);  
drop table test;
```

逐条执行：



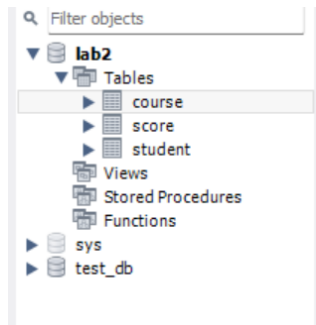
可以看到新建了 `test`，而且存在属性 `adr`

现在执行修改命令



可以看到新增属性 `bdr`

现在执行删除操作



可以看到test被删除

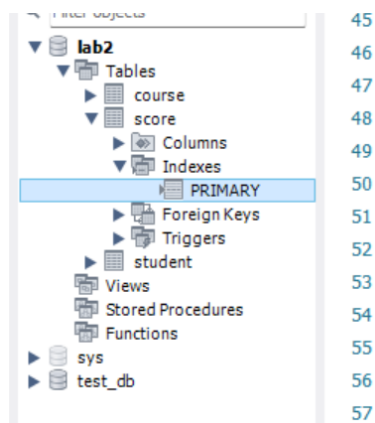
### 3.2.2 索引的建立/删除

#以下两个命令包含对索引的新建和删除，测试只针对`score`

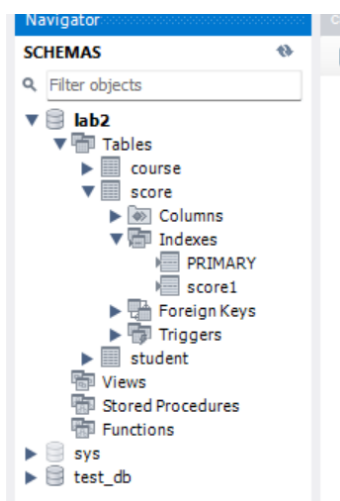
```
create index score1 on score(s_score);
```

```
drop index score1 on score;
```

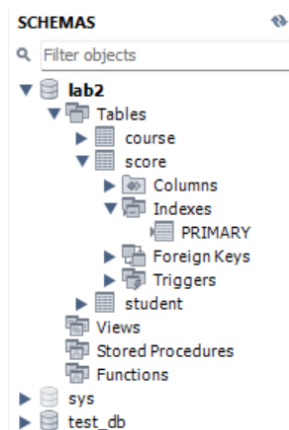
初始的index只有创建表时声明的 **primary key**，现新建 **score1** 索引



执行创建索引命令后



执行删除索引命令：

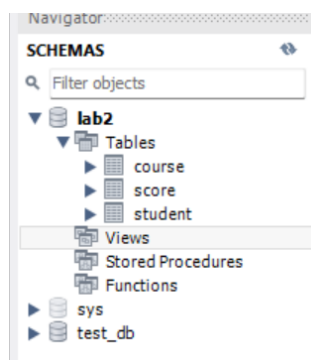


可以看到索引 `score1` 被删除

### 3.2.3 视图的建立/删除

#这里仅对 `student` 表进行视图展开，为了简便选择全选展开

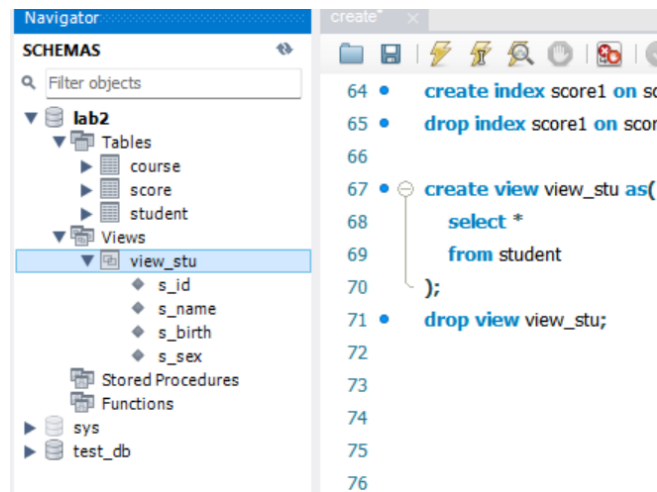
```
create view view_stu as(  
    select *  
    from student  
);  
##删除视图  
drop view view_stu
```



上图是创建视图前

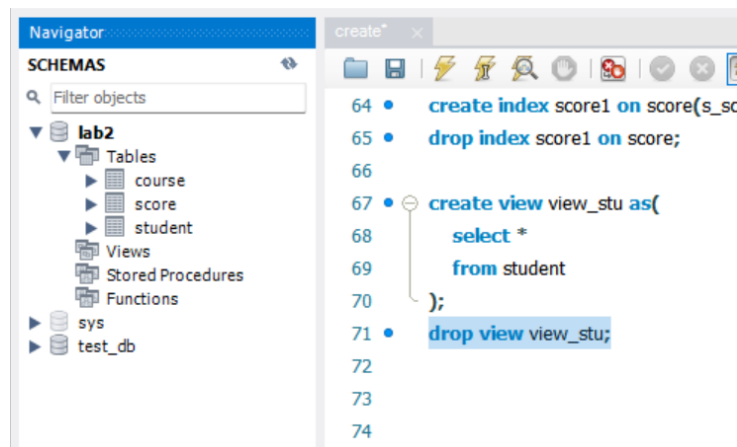
执行创建视图命令：





可以看到增加了 `view_stu` 视图

执行删除命令：



### 3.3 数据更新： 用 insert/delete/update命令插入/删除/修改表数据。

- `Insert` 插入表数据，完善table

# 插入学生表测试数据

```
insert into student values('01' , '赵雷' , '1990-01-01' , '男');
insert into student values('02' , '钱电' , '1990-12-21' , '男');
insert into student values('03' , '孙凤' , '1990-05-20' , '男');
insert into student values('04' , '李云' , '1990-08-06' , '男');
insert into student values('05' , '周梅' , '1991-12-01' , '女');
insert into student values('06' , '吴兰' , '1992-03-01' , '女');
insert into student values('07' , '郑竹' , '1989-07-01' , '女');
insert into student values('08' , '王菊' , '1990-01-20' , '女');
```

# 课程表测试数据

```
insert into course values('01' , '语文' , '02');
insert into course values('02' , '数学' , '01');
insert into course values('03' , '英语' , '03');
# 成绩表测试数据
insert into score values('01' , '01' , 80);
insert into score values('01' , '02' , 90);
insert into score values('01' , '03' , 99);
insert into score values('02' , '01' , 70);
insert into score values('02' , '02' , 60);
insert into score values('02' , '03' , 80);
insert into score values('03' , '01' , 80);
insert into score values('03' , '02' , 80);
insert into score values('03' , '03' , 80);
insert into score values('04' , '01' , 50);
insert into score values('04' , '02' , 30);
insert into score values('04' , '03' , 20);
insert into score values('05' , '01' , 76);
insert into score values('05' , '02' , 87);
insert into score values('06' , '01' , 31);
insert into score values('06' , '03' , 34);
insert into score values('07' , '02' , 89);
insert into score values('07' , '03' , 98);
```

出现下图说明插入成功（最下方的message），稍后我们会查看

**CHEMAS**

Filter objects

- lab2
  - Tables
    - course
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
    - score
    - student
  - Views
  - Stored Procedures
  - Functions
- sys
- test\_db

Administration Schemas

Information .....

**Table: course**

**Columns:**

- c\_id varchar(20) PK
- c\_name varchar(20)
- t\_id varchar(20)

```
31 • insert into student values('06', '吴兰', '1992-03-01', '女');
32 • insert into student values('07', '郑竹', '1989-07-01', '女');
33 • insert into student values('08', '王菊', '1990-01-20', '女');
34 • # 课程表测试数据
35 • insert into course values('01', '语文', '02');
36 • insert into course values('02', '数学', '01');
37 • insert into course values('03', '英语', '03');
38 • # 成绩表测试数据
39 • insert into score values('01', '01', 80);
40 • insert into score values('01', '02', 90);
41 • insert into score values('01', '03', 99);
42 • insert into score values('02', '01', 70);
43 • insert into score values('02', '02', 60);
44 • insert into score values('02', '03', 80);
45 • insert into score values('03', '01', 80);
46 • insert into score values('03', '02', 80);
47 • insert into score values('03', '03', 80);
48 • insert into score values('04', '01', 50);
49 • insert into score values('04', '02', 30);
50 • insert into score values('04', '03', 20);
51 • insert into score values('05', '01', 76);
52 • insert into score values('05', '02', 87);
53 • insert into score values('06', '01', 31);
54 • insert into score values('06', '03', 34);
```

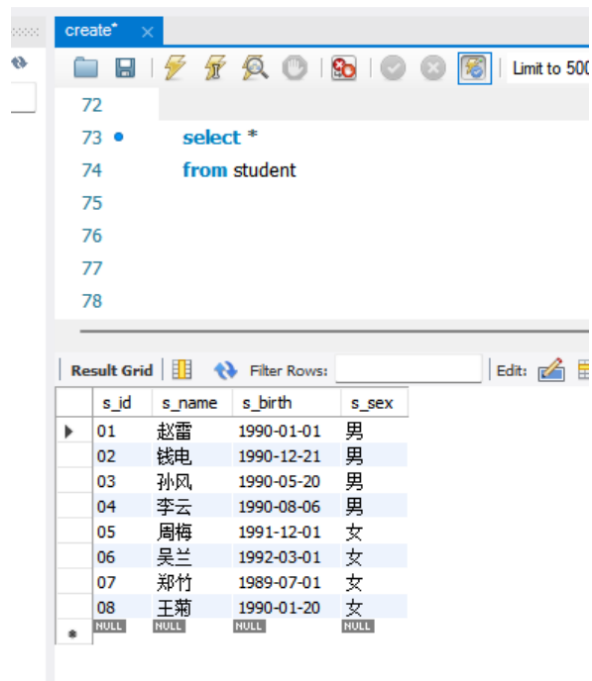
Output .....

Action Output

#	Time	Action
✓ 31	11:01:22	insert into Score values('05', '02', 87)
✓ 32	11:01:22	insert into Score values('06', '01', 31)
✓ 33	11:01:22	insert into Score values('06', '03', 34)
✓ 34	11:01:22	insert into Score values('07', '02', 89)
✓ 35	11:01:22	insert into Score values('07', '03', 98)

执行以下查询命令可视化查看 `student` 表

```
select *
from student;
```



- `update` 命令更新数据

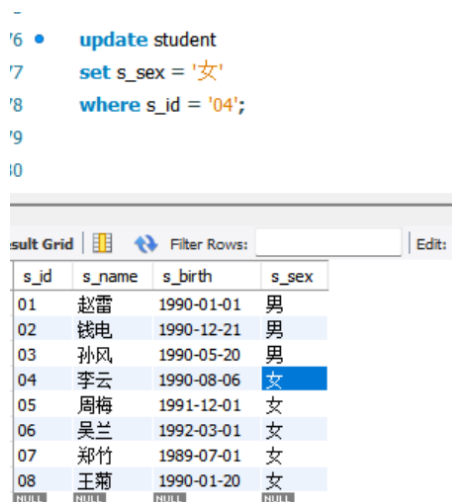
# 这里做一个性别的更改，其他位置更新同理

```
update student
set s_sex = '女'
where s_id = '04';
```

# 执行查询命令

```
select *
from student;
```

如图：



- 利用 `delete` 删除数据

# 这里依然对student表做删除

```
delete
```

```
from student
```

```
where s_id = '04';
```

#执行查询命令

```
select *
```

```
from student;
```

```
79
80 • delete
81 from student
82 where s_id = '04'
```

Result Grid				
Filter Rows:				
Edit:				
	s_id	s_name	s_birth	s_sex
▶	01	赵雷	1990-01-01	男
	02	钱电	1990-12-21	男
	03	孙风	1990-05-20	男
	05	周梅	1991-12-01	女
	06	吴兰	1992-03-01	女
	07	郑竹	1989-07-01	女
	08	王菊	1990-01-20	女
*	NULL	NULL	NULL	NULL

可以看到 `s_id = 04` 的数据被删除

## 3.4 数据查询

### 3.4.1 单表查询

#做一个简单的筛选性别的查询

#查询所有的男同学的出生年月

```
select s_birth
```

```
from student
```

```
where s_sex = '男';
```

```
79
80 • delete
81 from student
82 where s_id = '04';
83
```

Result Grid	
Filter Rows:	
Export:	
Wrap Cell Content:	
	s_birth
▶	1990-01-01
	1990-12-21
	1990-05-20

### 3.4.2 多表查询

#将学生表和分数表连接，查询所有在课程2得分超过80分的同学的名字

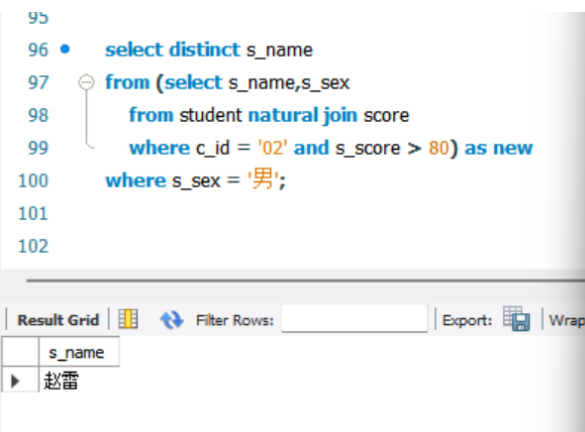
```
select distinct s_name
from student natural join score
where c_id = '02' and s_score > 80;
```



### 3.4.3 嵌套子查询

#查询所有在课程2得分超过80分的男同学的名字

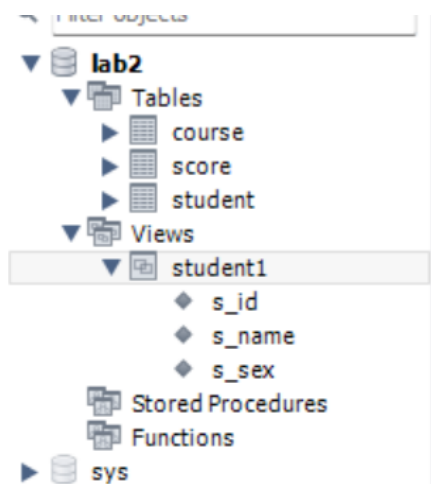
```
select distinct s_name
from (select s_name,s_sex
      from student natural join score
      where c_id = '02' and s_score > 80) as new
where s_sex = '男';
```



### 3.5 视图操作

首先关于 `student` 创造视图

```
create view student1 as(  
select s_id,s_name,s_sex  
from student  
);
```



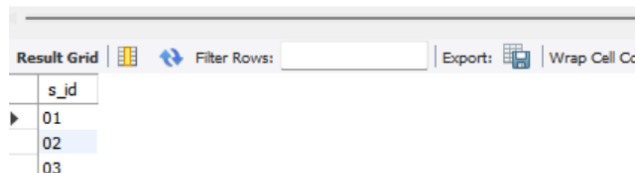
```
100 where s_sex = '男';  
101  
102 • create view student1 as(  
103 select s_id,s_name,s_sex  
104 from student  
105 );  
106  
107  
108  
109  
110
```

#### 3.5.1 数据查询

#查询视图中所有男同学的id

```
select s_id  
from student1  
where s_sex = '男';
```

```
106  
107 • select s_id  
108 from student1  
109 where s_sex = '男';  
110  
111  
112  
113
```



s_id
01
02
03

### 3.5.2 数据修改

#修改id为01的同学的性别为女

```
update student1
```

```
set s_sex = '女'
```

```
where s_id = '01';
```

#查询视图、原数据表全部内容，发现数据同步更新

```
select * from student;
```

```
select * from student1;
```

```
111 #修改id为01的同学的性别为女
112 • update student1
113 set s_sex = '女'
114 where s_id = '01';
115 #查询视图、原数据表全部内容，发现数据同步更新
116 • select * from student;
117 • select * from student1;
118
119
```

	s_id	s_name	s_sex
▶	01	赵雷	女
	02	钱电	男
	03	孙风	男
	05	周梅	女
	06	吴兰	女
	07	郑竹	女
	08	王菊	女

```
10
11 #修改id为01的同学的性别为女
12 • update student1
13 set s_sex = '女'
14 where s_id = '01';
15 #查询视图、原数据表全部内容，发现数据同步更新
16 • select * from student;
17 • select * from student1;
18
19
```

s_id	s_name	s_birth	s_sex
01	赵雷	1990-01-01	女
02	钱电	1990-12-21	男
03	孙风	1990-05-20	男
05	周梅	1991-12-01	女
06	吴兰	1992-03-01	女
07	郑竹	1989-07-01	女
08	王菊	1990-01-20	女
NULL	NULL	NULL	NULL

数据同步正常，说明视图只是将原数据表进行引用，对视图的操作将同步到原table



## 4 遇到的问题及解决方法

```
warning(s): 1681 Integer display width is deprecated and will be removed  
in a future release.
```

似乎 `INT` 类型会在不久后删除，所以为了未来版本能够兼容，尽量少用 `INT` 即可

```
Warnings: every derived table must have its own alias
```

解决方式：[Every derived table must have its own alias \(sql语句错误解决方法\)-CSDN 博客](#)

结论：如果存在嵌套子查询，记得将新产生的表命名，否则会报错。

## 5 总结

本次的实验较为基础，主要是对第三周学到的SQL语言进行学习回顾，熟悉并使用SQL语言进行查询，借助于MySQL的Workbench可视化界面可以实时的查看数据库内容（好像命令行也是可以的）。在撰写查询操作时可以发现自己的习惯的语法错误，例如最简单最末尾没有分号，或者是嵌套子查询没有重命名新表等等。