

数据库课程实验日志

学号	201726010211	姓名	陈汉轩	专业年级班级	数媒 1701
实验日期	2020. 5. 5	实验项目	数据库定义与操作语言		

目录

数据库课程实验日志.....	1
一、实验目的	1
二、实验过程&错误.....	2
内容（一）： 1.1 数据库定义	2
内容（二）： 1.2 数据基本查询.....	9
内容（三）： 1.3 数据库高级查询.....	13
内容（四）： 1.4 数据更新	15
内容（五）： 1.5 视图.....	20
内容（一）： 1.6 索引	22
三、实验重难点	27
四、实验心得体会	27

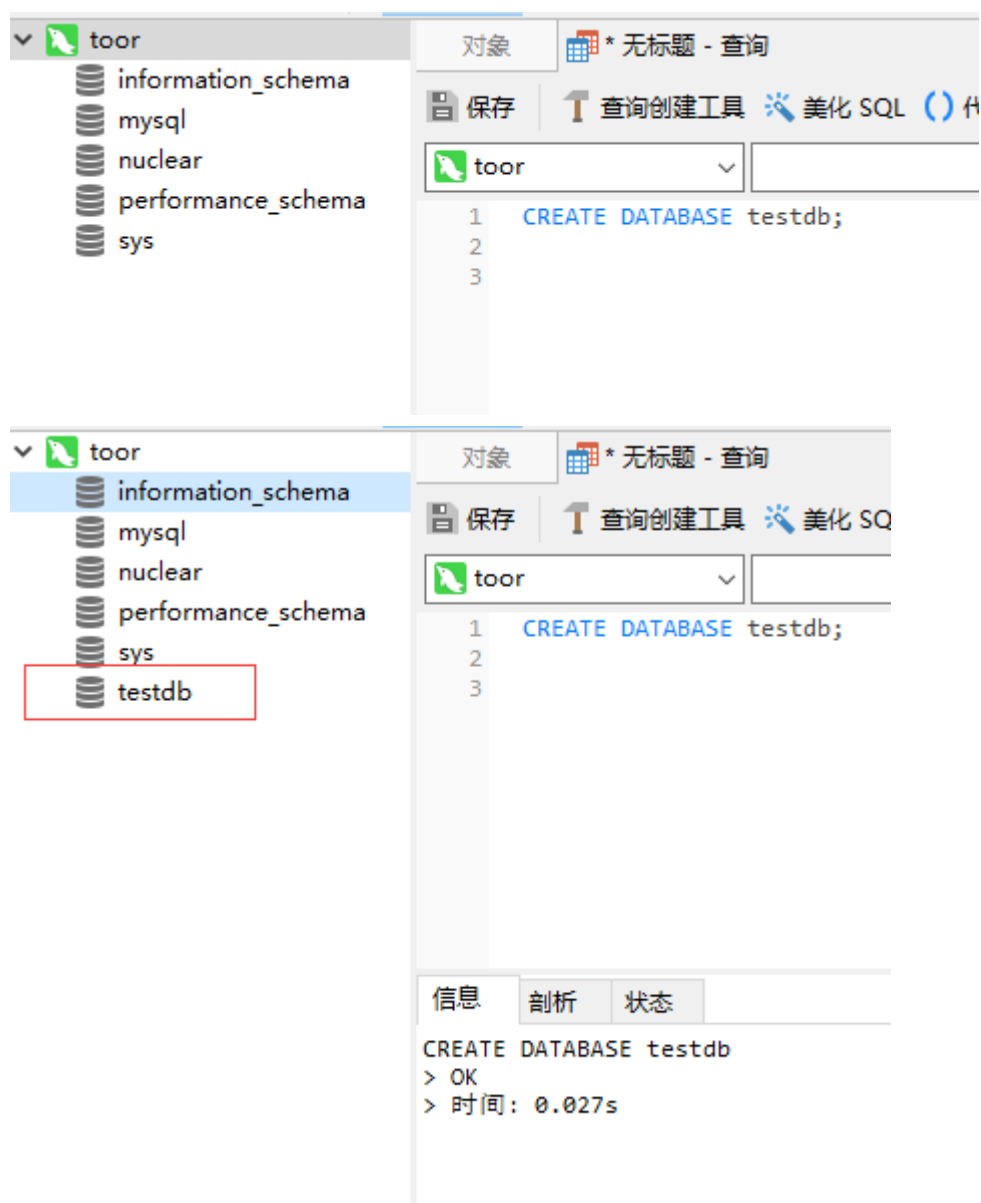
一、实验目的

数据库定义与操作语言实验包含 6 个实验项目，其中 5 个必修实验项目，1 个选修实验项目。其中实验项目 1.1 至 1.5 为设计型实验，1.6 为验证性实验项目。

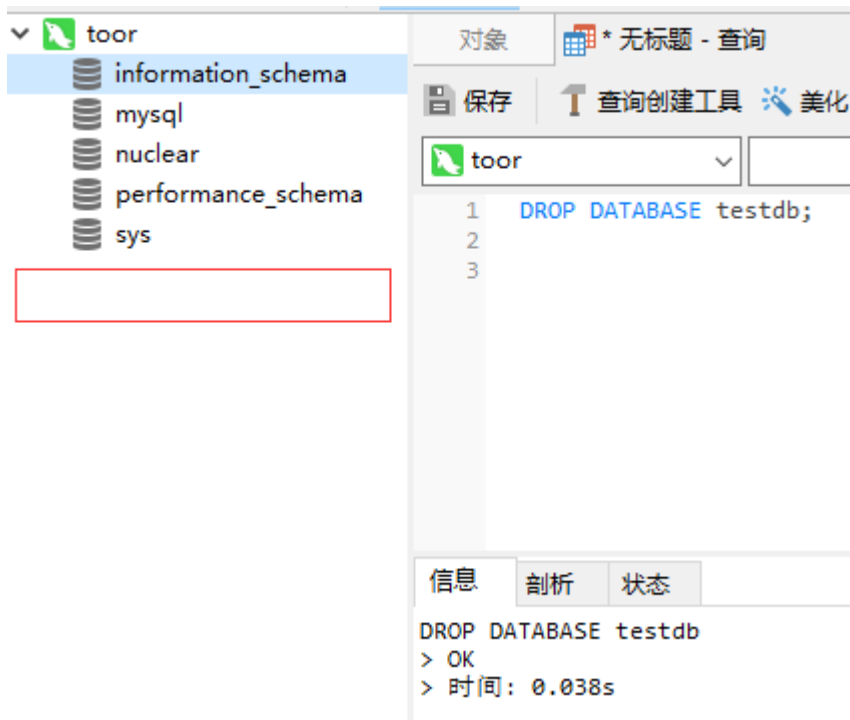
二、实验过程&错误

内容（一）：1.1 数据库定义

步骤 1：创建 testdb;



步骤 2：删除 testdb



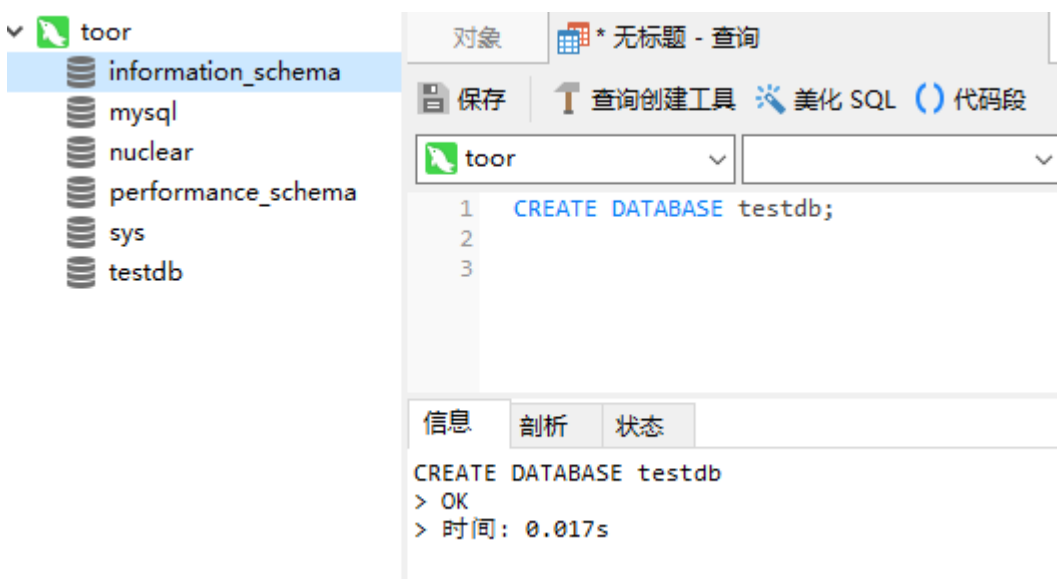
步骤 3: 创建模式

SQL 中创建模式的语句: `CRAETE SCHEMA <模式名> AUTHORIZATION <用户名>`;

SQL 中删除模式的语句: `DROP SCHEMA <模式名><CASCADE|RESTRICT>`

但是 MySQL 中没有模式，因此无法创建。

步骤 4: 创建表

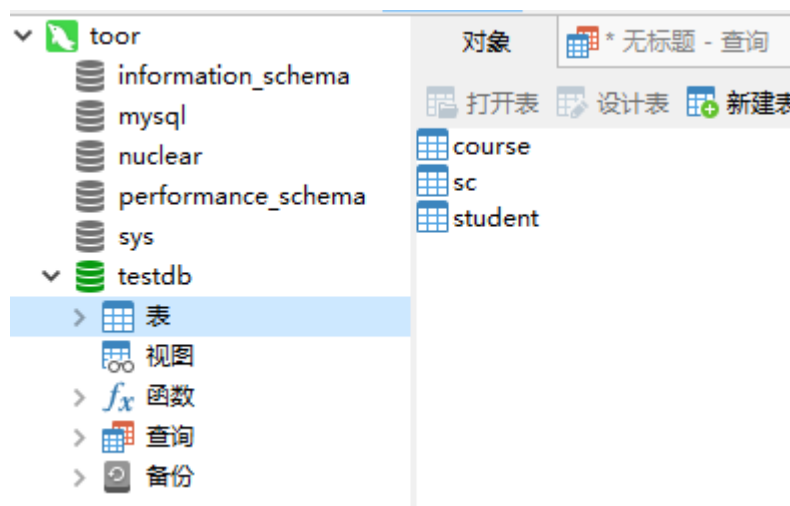


创建三张表，Student, Course, SC。



这里按照课本的 P80 基本表的定义、删除与修改部分。

其中：课程号(Cno) 课程名(Cname) 先行课(Cpno)



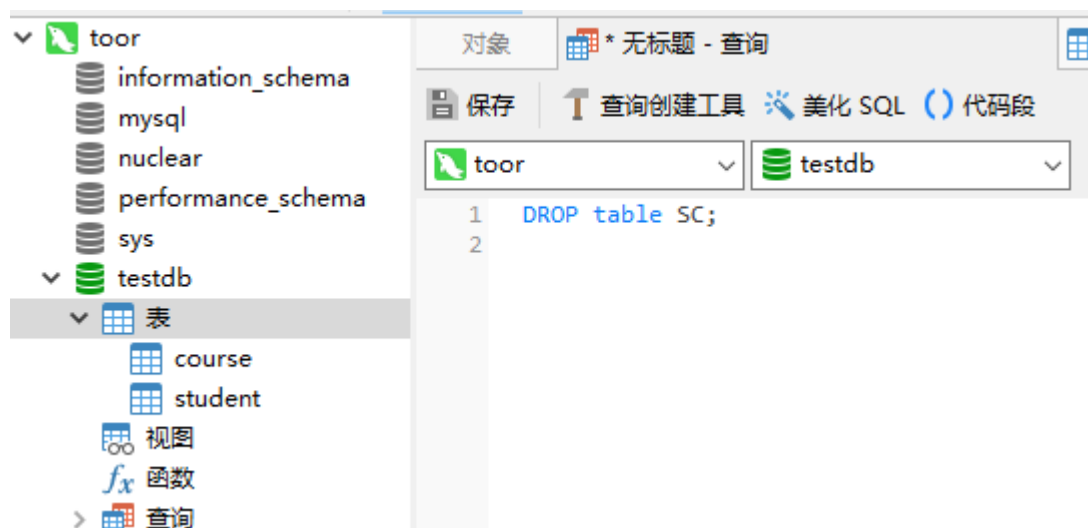
成功新建了 3 张表。

步骤 5：删除表

尝试删除表 student:



发现不能删除，因为 Student 的表中有一个外键指向了 SC 表。



但是删除 SC 表是可以的，因为没有其他的表指向它。

但是在尝试插入数据的时候，发现自己的学号 201726010211 有 13 位。

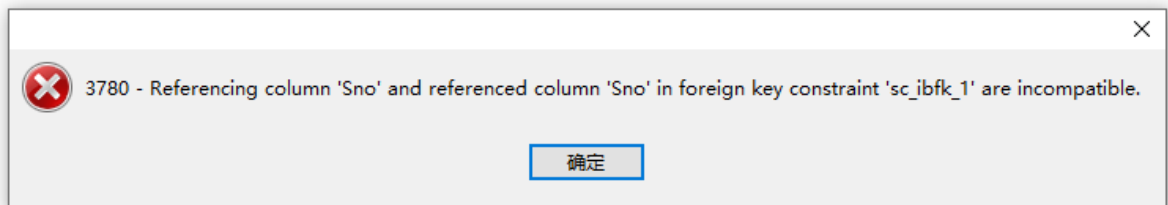
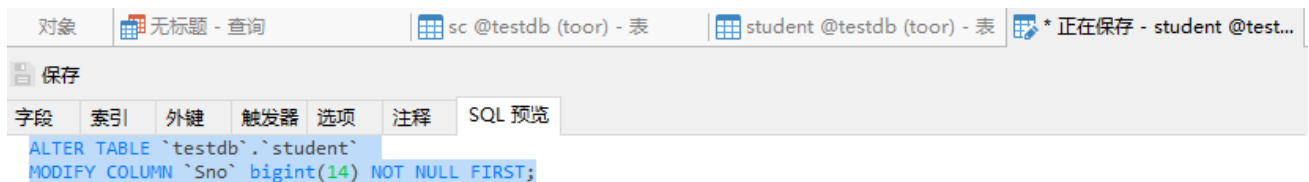
按照书上的定义 char(9)无法插入，所以对表的内容进行修改。

```
ALTER TABLE `testdb`.`student`
```

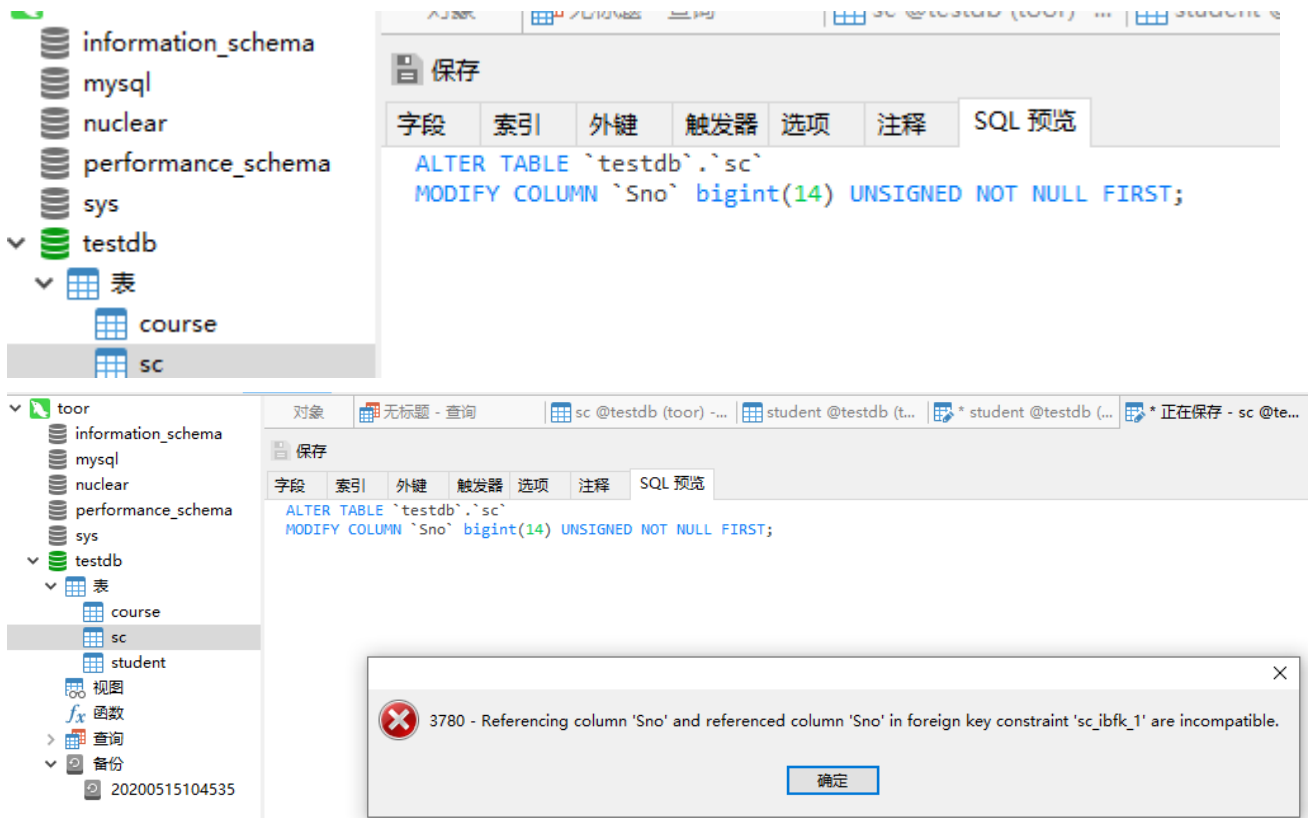
```
MODIFY COLUMN `Sno` bigint(14) NOT NULL FIRST;
```

将学号字段的数据格式改为 bigint，设置为 14 位。

修改失败，因为和外键指向的内容类型不匹配：

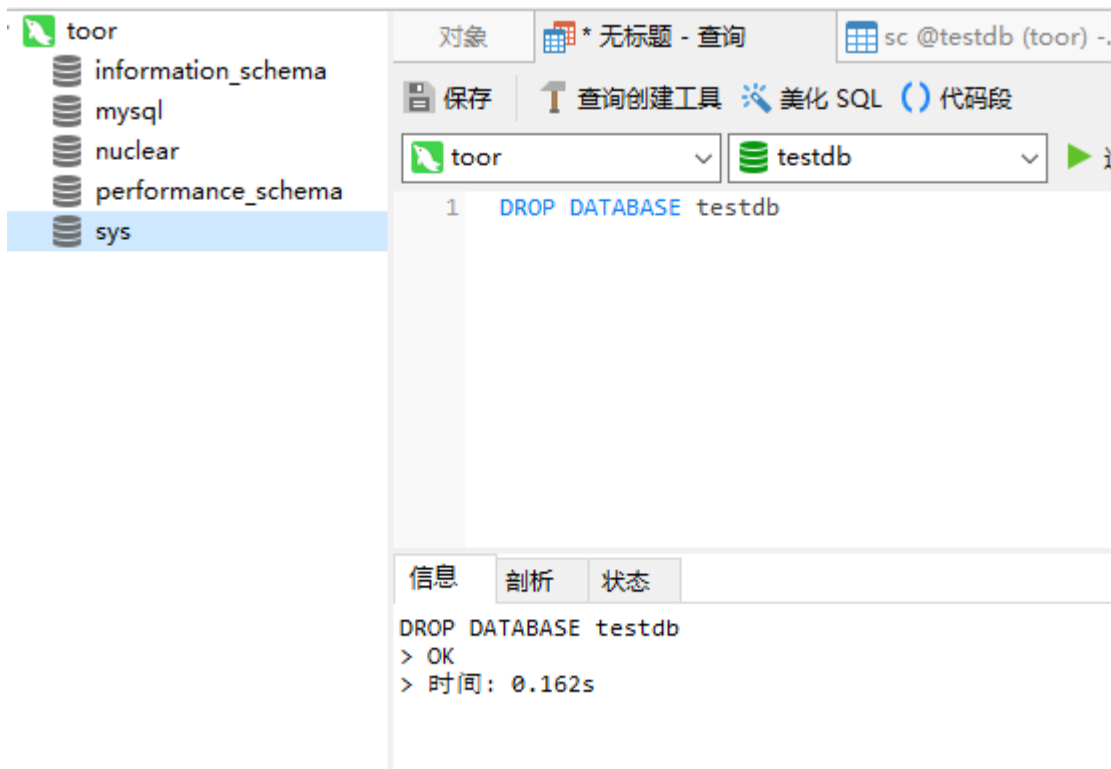


所以先修改外键指向的表 SC:



可是发现也不能修改，因为 SC 表引用了 Student 表，所以形成了一个循环的问题。

所以清空数据库，重新建立：

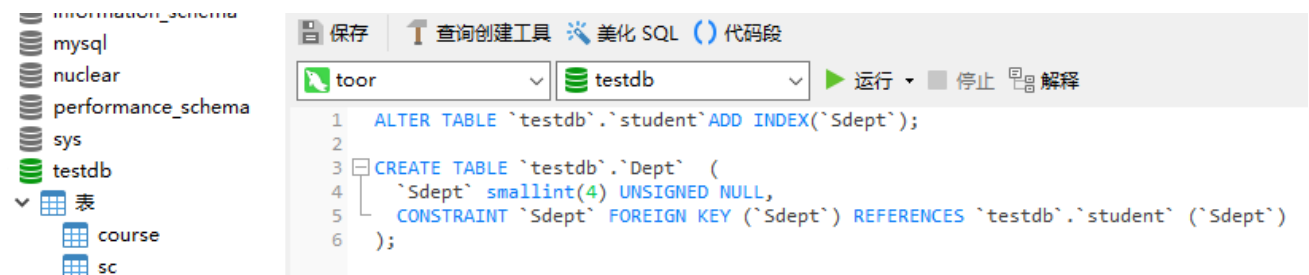


感觉 Student 表内的 Sdept 也非常浪费，使用一个 4 位的 smallint 代替：

ALTER TABLE `testdb`.`student`

MODIFY COLUMN `Sdept` smallint(4) NULL DEFAULT NULL AFTER `Sage`;

接着创建一个新的表，来存 Sdept 的名称：



ALTER TABLE `testdb`.`student` ADD INDEX(`Sdept`);

CREATE TABLE `testdb`.`Dept` (
 `Sdept` smallint(4) UNSIGNED NULL,

```

CONSTRAINT `Sdept` FOREIGN KEY (`Sdept`) REFERENCES `testdb`.`student`
(`Sdept`)
);

```

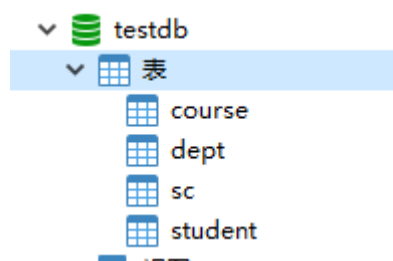
信息 剖析 状态

```

ALTER TABLE `testdb`.`student` ADD INDEX(`Sdept`)
> OK
> 时间: 0.089s

CREATE TABLE `testdb`.`Dept` (
  `Sdept` smallint(4) UNSIGNED NULL,
  CONSTRAINT `Sdept` FOREIGN KEY (`Sdept`) REFERENCES `testdb`.`student` (`Sdept`)
)
> OK
> 时间: 0.146s

```



之后填入数据库 P77-P78 的数据:

开始事务	文本	筛选	排序	导入	导出
Cno	Cname	Cpno	Ccredit		
1	数据库	5	4		
2	数学	(Null)	2		
3	信息系统	1	4		
4	操作系统	6	3		
5	数据结构	(Null)	4		
6	数据处理	7	2		
7	PASCAL语言	6	4		

开始事务	文本	筛选	排序	导入	导出
Sdept	DName				
0	CS				
1	MA				
2	IS				

开始事务 文本 筛选 排序 导入 导出				
Sno	Sname	Ssex	Sage	Sdept
201215121	李勇	男	20	0
201215122	刘晨	女	19	0
201215123	王敏	女	18	1
201215125	张立	男	19	2

内容（二）：1.2 数据基本查询

步骤 1：查询指定的列

对象
* 无标题 - 查询
student @testdb (t...

保存
查询创建工具
美化 SQL
代码段
文本

toor
testdb
运行

```

1 select sno,sname from student;
2

```

信息	结果 1	剖析	状态
	sno	sname	
	201215122	刘晨	
	201215125	张立	
	201215121	李勇	
	201215123	王敏	

步骤 2：查询并计算

```
1 select sname,2020-sage from student;
2
```

信息	结果 1	剖析	状态										
	<table border="1"> <thead> <tr> <th>sname</th> <th>2020-sage</th> </tr> </thead> <tbody> <tr> <td>李勇</td> <td>2000</td> </tr> <tr> <td>刘晨</td> <td>2001</td> </tr> <tr> <td>王敏</td> <td>2002</td> </tr> <tr> <td>张立</td> <td>2001</td> </tr> </tbody> </table>	sname	2020-sage	李勇	2000	刘晨	2001	王敏	2002	张立	2001		
sname	2020-sage												
李勇	2000												
刘晨	2001												
王敏	2002												
张立	2001												

步骤 3: 带条件的查询

toor

testdb

运行

停止

解释

1

select sname from student where sdept='CS' and sage<22;

2

信息

结果 1

剖析

状态

sname

▶ 李勇

刘晨

步骤 4: 分组查询每一个系的平均分数

toor
testdb
运行
停止
解释

```

1 select sdept, avg(sage) from student group by sdept;
2

```

信息
结果 1
剖析
状态

sdept	avg(sage)
0	19.5000
1	18.0000
2	19.0000

步骤 5: 多表链接查询

toor
testdb
运行
停止
解释

```

1 select student.*,sc.* from student,sc where student.sno=sc.sno;
2

```

信息
结果 1
剖析
状态

Sno	Sname	Ssex	Sage	Sdept	Sno(1)	Cno	Grade
201215121	李勇	男	20	0	201215121	1	92
201215121	李勇	男	20	0	201215121	2	85
201215121	李勇	男	20	0	201215121	3	88
201215122	刘晨	女	19	0	201215122	2	90
201215122	刘晨	女	19	0	201215122	3	80

步骤 6: 自身链接查询

toor

testdb

运行

停止

解释

```
1 select first.cno,second.cjno from course first,course second where first.cjno=second.cno;
2
```

信息

结果 1

剖析

状态

cno	cjno
3	5
1	(Null)
4	7
7	7
6	6

步骤 7: 查询先修课的先修课

toor

testdb

运行

停止

解释

```
1 select first.cno,second.cjno from course first,course second where first.cjno=second.cno;
2
```

信息

结果 1

剖析

状态

cno	cjno
3	5
1	(Null)
4	7
7	7
6	6

步骤 8: 外连接

toor
testdb
运行
停止
解释

```

1 select student.*,cno,grade from student left outer join sc on (student.sno=sc.sno);
2

```

信息
结果 1
剖析
状态

Sno	Sname	Ssex	Sage	Sdept	cno	grade
201215121	李勇	男	20	0	1	92
201215121	李勇	男	20	0	2	85
201215121	李勇	男	20	0	3	88
201215122	刘晨	女	19	0	2	90
201215122	刘晨	女	19	0	3	80
201215123	王敏	女	18	1	(Null)	(Null)
201215125	张立	男	19	2	(Null)	(Null)

内容（三）：1.3 数据库高级查询

步骤 1：不相关子查询

toor
testdb
运行
停止
解释

```

1 select sname from student where sno in (select sno from sc where cno='1');
2 |

```

信息
结果 1
剖析
状态

sname
李勇

现象 2：相关子查询

toor testdb 运行 停止 解释

```

1 select sno,cno from sc x where grade>=(select avg(grade) from sc y where y.sno=x.sno);
2

```

信息 结果 1 剖析 状态

sno	cno
201215121	1
201215122	2

步骤 3: 带有 any 的查询

toor testdb 运行 停止 解释

```

1 select sname,sage from student where sage<any(select sage from student where sdept='cs') and sdept<>'cs';
2 |

```

信息 结果 1 剖析 状态

sname	sage
王敏	18
张立	19

步骤 4: 带有 exists 的子查询

toor testdb 运行 停止 解释

```

1 select sname from student where exists (select * from sc where sno=student.sno and cno='1');
2 |

```

信息 结果 1 剖析 状态

sname
李勇

步骤 5: 查询选修了全部课程的学生的姓名 (查询结果为空)

toor
testdb
运行
停止
解释

```

1 select sname from student where not exists (select * from course where not exists (select * from sc where sno
2 |
=student.sno and cno=course.cno));

```

信息
结果 1
剖析
状态

sname
(N/A)

步骤 6：集合查询

查询计算机科学系的学生与年龄不大于 19 岁学生的交集

toor
testdb
运行
停止
解释

```

1 select * from student where sdept='cs' union select * from student where sage<=19;
2

```

信息
结果 1
剖析
状态

Sno	Sname	Ssex	Sage	Sdept
201215121	李勇	男	20	0
201215122	刘晨	女	19	0
201215123	王敏	女	18	1
201215125	张立	男	19	2

内容（四）：1.4 数据更新

步骤 1：插入一个学生信息

保存

查询创建工具

美化 SQL

代码段

toor

testdb

运行

停止

解释

```
1 insert into student values('201215126','张成民','男','18','CS');
2 |
```

信息

状态

```
insert into student values('201215126','张成民','男','18','CS')
> 1366 - Incorrect integer value: 'CS' for column 'Sdept' at row 1
> 时间: 0s
```

步骤 2: 插入一条课程记录

toor

testdb

运行

停止

解释

```
1 insert into course values('1','数据库',null,4);
2 |
```

信息

状态

```
insert into course values('1','数据库',null,4)
> 1062 - Duplicate entry '1' for key 'course.PRIMARY'
> 时间: 0.001s
```

步骤 3: 插入一条选课记录

toor

testdb

运行

停止

解释

```
1 insert into sc values('201215126','1',null);
2
```

信息

状态

```
insert into sc values('201215126','1',null)
> 1452 - Cannot add or update a child row: a foreign key constraint fails (`testdb`.`sc`, CONSTRAINT `sc_ibfk_1`
FOREIGN KEY (`Sno`) REFERENCES `student` (`Sno`))
> 时间: 0.003s
```

步骤 4: 插入子查询结果

toor

testdb

运行

停止

解释

```
1 create table dept_age(sdept char(15),avg_age smallint);
2 insert into dept_age(sdept,avg_age) select sdept,avg(sage) from student group by sdept;
3
```

信息

剖析

状态

```
create table dept_age(sdept char(15),avg_age smallint)
> OK
> 时间: 0.166s

insert into dept_age(sdept,avg_age) select sdept,avg(sage) from student group by sdept
> Affected rows: 3
> 时间: 0.02s
```

开始事务

文本

筛选

排序

导入

导出

sdept	avg_age
0	20
1	18
2	19

步骤 5: 更新数据

toor

testdb

运行

停止

解释

1

update student set sage=22 where sno='201215126';

2

|

信息

剖析

状态

```
update student set sage=22 where sno='201215126'
> Affected rows: 0
> 时间: 0.001s
```

步骤 6: 将所有学生的年龄加一

toor

testdb

运行

停止

解释

1

update student set sage=sage+1;

2

信息

剖析

状态

```
update student set sage=sage+1
> Affected rows: 4
> 时间: 0.012s
```

步骤 7: 将所有的计算机系的学生成绩置为 90

```
1 update sc set grade=90 where sno in (select sno from student,dept where student.Sdept = dept.Sdept AND dept.DName = 'CS');
2
```

信息 剖析 状态

```
update sc set grade=90 where sno in (select sno from student,dept where student.Sdept = dept.Sdept AND dept.DName = 'CS')
> Affected rows: 4
> 时间: 0.012s
```

步骤 8: 删除一条数据

toor testdb 运行 停止 解释

```
1 delete from student where sno='201215128';
2
```

信息 剖析 状态

```
delete from student where sno='201215128'
> Affected rows: 0
> 时间: 0.004s
```

步骤 9: 带子查询的删除

toor testdb 运行 停止 解释

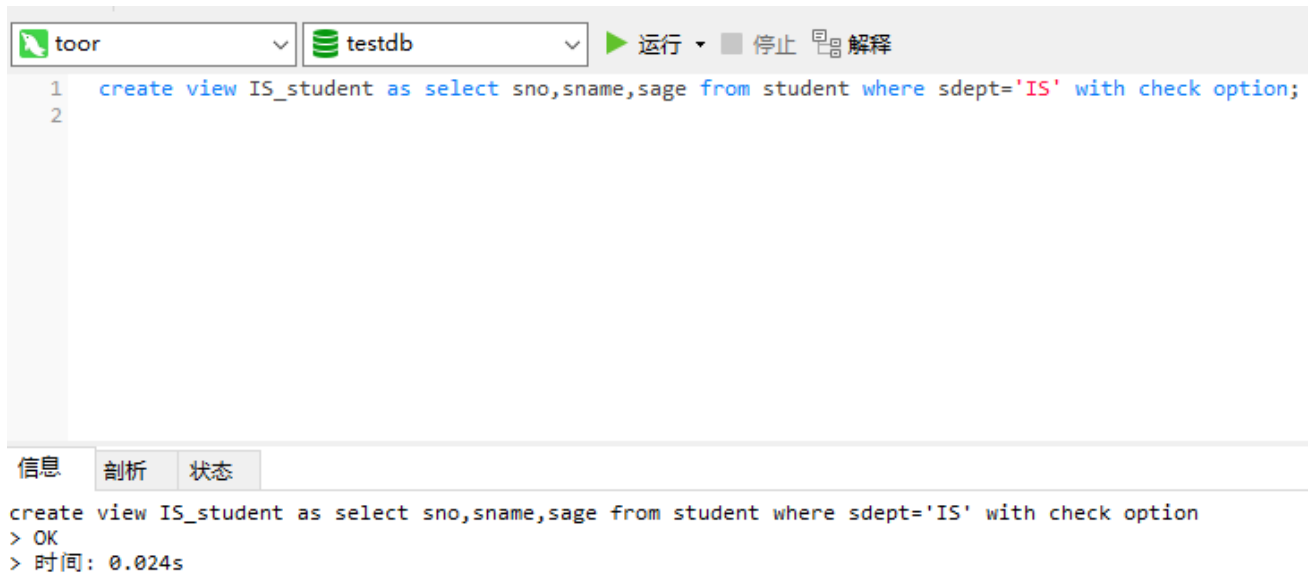
```
1 delete from sc where sno in (select sno from student, dept where student.Sdept=dept.Sdept and dept.DName='EE');
2
```

信息 剖析 状态

```
delete from sc where sno in (select sno from student, dept where student.Sdept=dept.Sdept and dept.DName='EE')
> Affected rows: 0
> 时间: 0.001s
```

内容（五）：1.5 视图

步骤 1：建立信息系学生的视图



The screenshot shows a SQL IDE interface. At the top, there are dropdown menus for 'toor' and 'testdb', followed by buttons for '运行' (Run), '停止' (Stop), and '解释' (Explain). The main text area contains the following SQL code:

```
1 create view IS_student as select sno,sname,sage from student where sdept='IS' with check option;  
2
```

Below the code area, there are tabs for '信息' (Info), '剖析' (Analyze), and '状态' (Status). The '信息' tab is selected, showing the following output:

```
create view IS_student as select sno,sname,sage from student where sdept='IS' with check option  
> OK  
> 时间: 0.024s
```

步骤 2：创建分组视图：



The screenshot shows a SQL IDE interface. At the top, there are dropdown menus for 'toor' and 'testdb', followed by buttons for '运行' (Run), '停止' (Stop), and '解释' (Explain). The main text area contains the following SQL code:

```
1 create view s_g(sno,gavg) as select sno,avg(grade) from sc group by sno;  
2
```

Below the code area, there are tabs for '信息' (Info), '剖析' (Analyze), and '状态' (Status). The '信息' tab is selected, showing the following output:

```
create view s_g(sno,gavg) as select sno,avg(grade) from sc group by sno  
> OK  
> 时间: 0.017s
```

可以看到创建了的视图。

对象	...	student @t...	dept @tes...	sc @testd...	course @t..
开始事务	文本	筛选	排序	导出	
sno	gavg				
201215121	90.0000				
201215122	90.0000				

步骤 3: 查询视图

保存

查询创建工具

美化 SQL

代码段

文本

导出结果

toor

testdb

运行

停止

解释

```

1 select sno,sage from is_student where sage<=20;
2 |

```

信息	结果 1	剖析	状态
sno	sage		
201215122	20		

步骤 4: with check option 验证

toor

testdb

运行

停止

解释

```
1 create view seniorstu(sno,sage) as select sno,sage from student where sage>='19' with check option;
2 insert into seniorstu values('201215127',18);
3 insert into seniorstu values('201215127',21);
4 |
```

信息

状态

```
create view seniorstu(sno,sage) as select sno,sage from student where sage>='19' with check option
> OK
> 时间: 0.019s

insert into seniorstu values('201215127',18)
> 1423 - Field of view 'testdb.seniorstu' underlying table doesn't have a default value
> 时间: 0.001s
```

步骤 5: 更新不可更新的视图（请求被拒绝）

toor

testdb

运行

停止

解释

```
1 update s_g set gavg=90 where sno='201215121';
2 |
```

信息

状态

```
update s_g set gavg=90 where sno='201215121'
> 1288 - The target table s_g of the UPDATE is not updatable
> 时间: 0.001s
```

内容（一）：1.6 索引

步骤 1: 在 dept 表的 sdept 字段上创建唯一索引 Idx_dept_sdept。

toor

testdb

运行

停止

解释

```
1 CREATE UNIQUE INDEX Idx_dept_sdept ON dept(sdept);
```

信息

剖析

状态

CREATE UNIQUE INDEX Idx_dept_sdept ON dept(sdept)
> OK
> 时间: 0.102s

步骤 2: 验证索引的有效性 (注入 10w 条数据)

保存

查询创建工具

美化 SQL

代码段

toor

testdb

运行

停止

解释

```
1 CREATE TABLE test_index(num char(10), id char(10));
```

信息

剖析

状态

CREATE TABLE test_index(num char(10), id char(10))
> OK
> 时间: 0.119s

创建一个表 test_index 来进行测试。

设置 id 为主键:

对象

* 无标题 - 查询

sc @testdb (toor) - 表

test_index @testdb (toor) ...

* test_ind

保存

字段

索引

外键

触发器

选项

注释

SQL 预览

```
ALTER TABLE `testdb`.`test_index`  
MODIFY COLUMN `num` char(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL FIRST,  
ADD PRIMARY KEY (`num`);
```

首先插入 10w 条数据:

```

import MySQLdb # 用于Mysql数据库操作

def add_new_job():
    # connect to the database with the host \ username \ password \ schema name
    db = MySQLdb.connect(host="localhost", user="root", password="123456", db="testdb", charset='utf8')
    cur = db.cursor()

    # console 输出提示信息
    print("数据库连接成功")

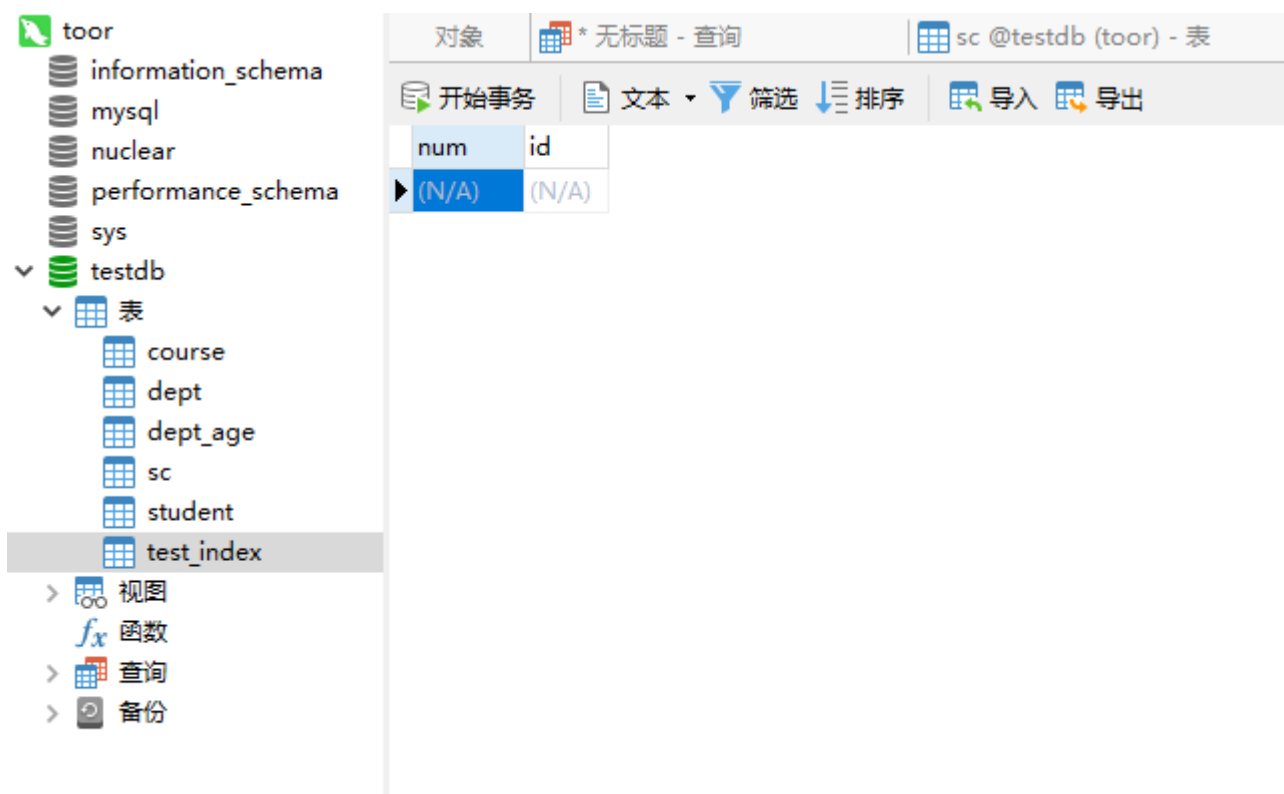
    for i in range(100000):
        cur.execute('''
            INSERT INTO test_index (num, id) VALUES (%s, %s)
            ''', (42, i))

    # 向数据库提交执行
    db.commit()
    db.close()

if __name__ == "__main__":
    add_new_job()

```

最初为空表：



The screenshot shows a database management interface. On the left, a tree view displays the database structure, including schemas like 'information_schema', 'mysql', 'nuclear', 'performance_schema', 'sys', and 'testdb'. Under 'testdb', a list of tables is shown, with 'test_index' selected. On the right, a table viewer displays the 'test_index' table. The table has two columns: 'num' and 'id'. The data row shows '(N/A)' for both columns, indicating the table is empty.

num	id
(N/A)	(N/A)

插入 10w 数据之后：


```
dbinject x
"C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\envs\final_project\python.
数据库连接成功

Process finished with exit code 0
```

对象 | * 无标题 - 查询 | sc @testdb (toor) - 表 | test_index @testdb (toor) ... | * test_index @testdb (toor) ...

开始事务 | 文本 | 筛选 | 排序 | 导入 | 导出

num	id
42	0
42	1
42	2
42	3
42	4
42	5
42	6
42	7
42	8
42	9
42	10
42	11
42	12
42	13
42	14
42	15
42	16
42	17
42	18
42	19
42	20
42	21
42	22
42	23

+ - ✓ ✕ ↺ ■ | ⏮ ⏪ 1 ⏩ ⏭ | ⚙ | 第 1 页

SELECT * FROM `testdb`.`test_index` LIMIT 0,1000

然后验证索引的有效性。

步骤 3：在 test 表种没有索引的情况下测试查询速度

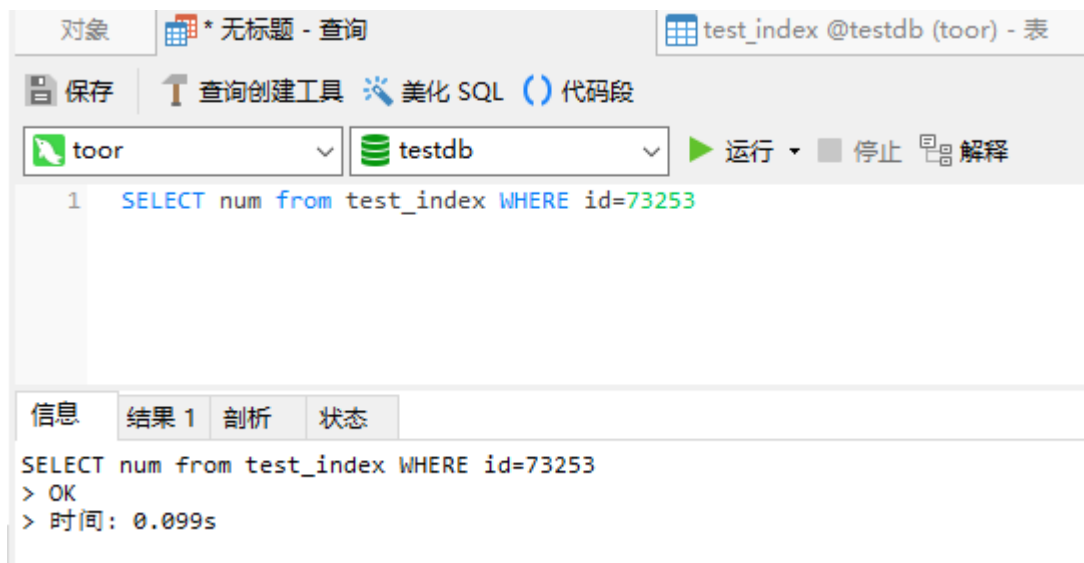


查询一个 id 为 73253 的 num，用时 0.113s。

步骤 4：在 test 表中对 id 创建索引，再次测试查询速度



再次查询：



相同的查询语句，用时从 0.113s 降低到 0.099s。

三、实验重难点

我遇到的一个问题就是在有外键的情况下的增删改查的问题，尤其是在自身引用自己的时候，例如课程的先修课程，最开始第一个课程就无法插入，因为此时外键的值在主键种并不存在，这种情况下需要首先将外键设置为 `null`，在所有数据全部插入之后再进行 `update` 先修课程列的值。

然后验证索引有效性的时候设置自身的前后对照是一个有趣的创新思路。

四、实验心得体会

学习了如何使用 SQL 语句实现数据库定义、查询、更新、删除、索引提高查询速度。