

**数据库系统实验报告**

|  |  |
| --- | --- |
| 作业名称： | SQL数据定义和操作 |
| 姓 名： | 刘韬 |
| 学 号： | 3220103422 |
| 电子邮箱： | xliutao17@163.com |
| 联系电话： | 13695835699 |
| 指导老师： | 孙建伶 |

2024年3月19日

实验名称

一、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。**

二、实验环境

操作系统： windows 11

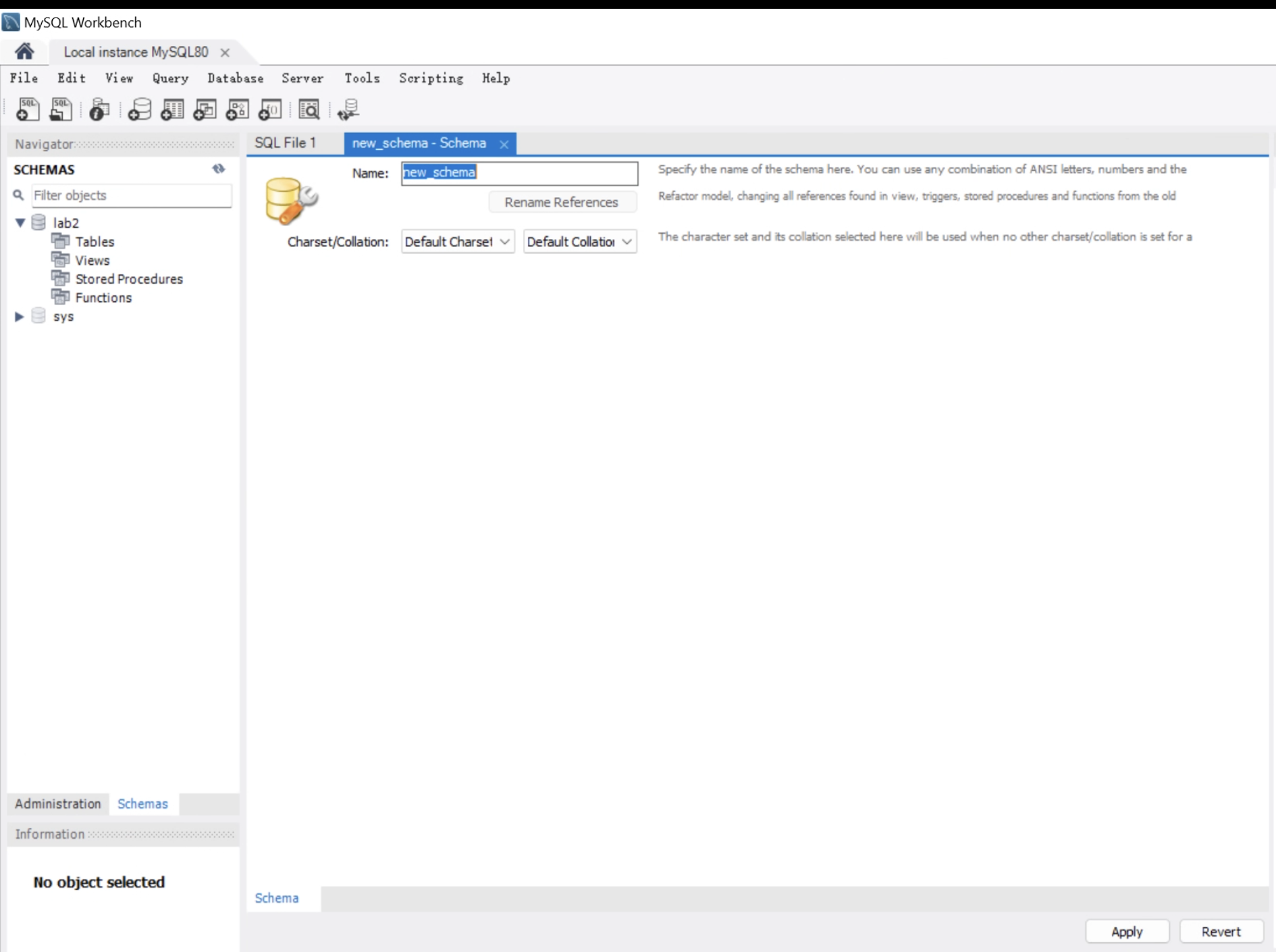
数据库管理系统： MySQL 8.0.36

图形化工具： MySQL Workbench

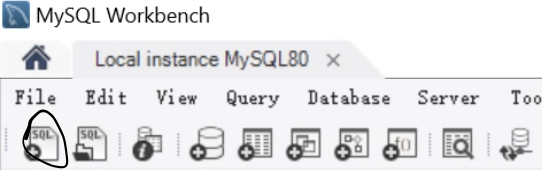
三、实验流程

3.1 建立数据库

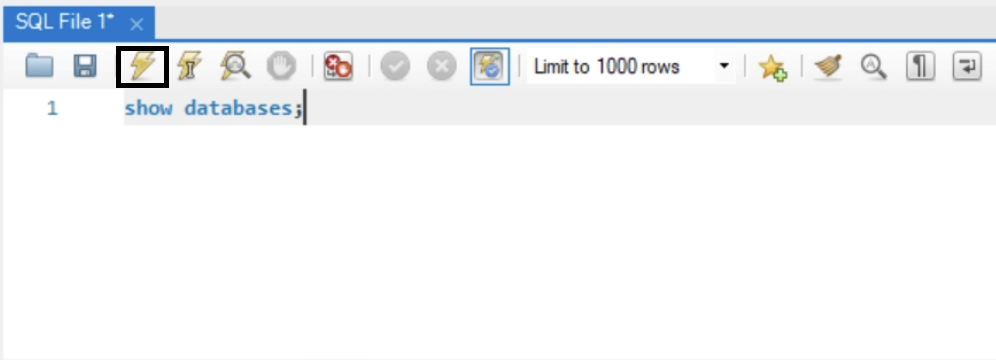
点击圈出的按钮，出现如下界面，自定义Name为lab2，然后一直apply直到完成创建schema，建立新架构即建立一个数据库。在下图的框内已经出现了新建的数据库lab2.



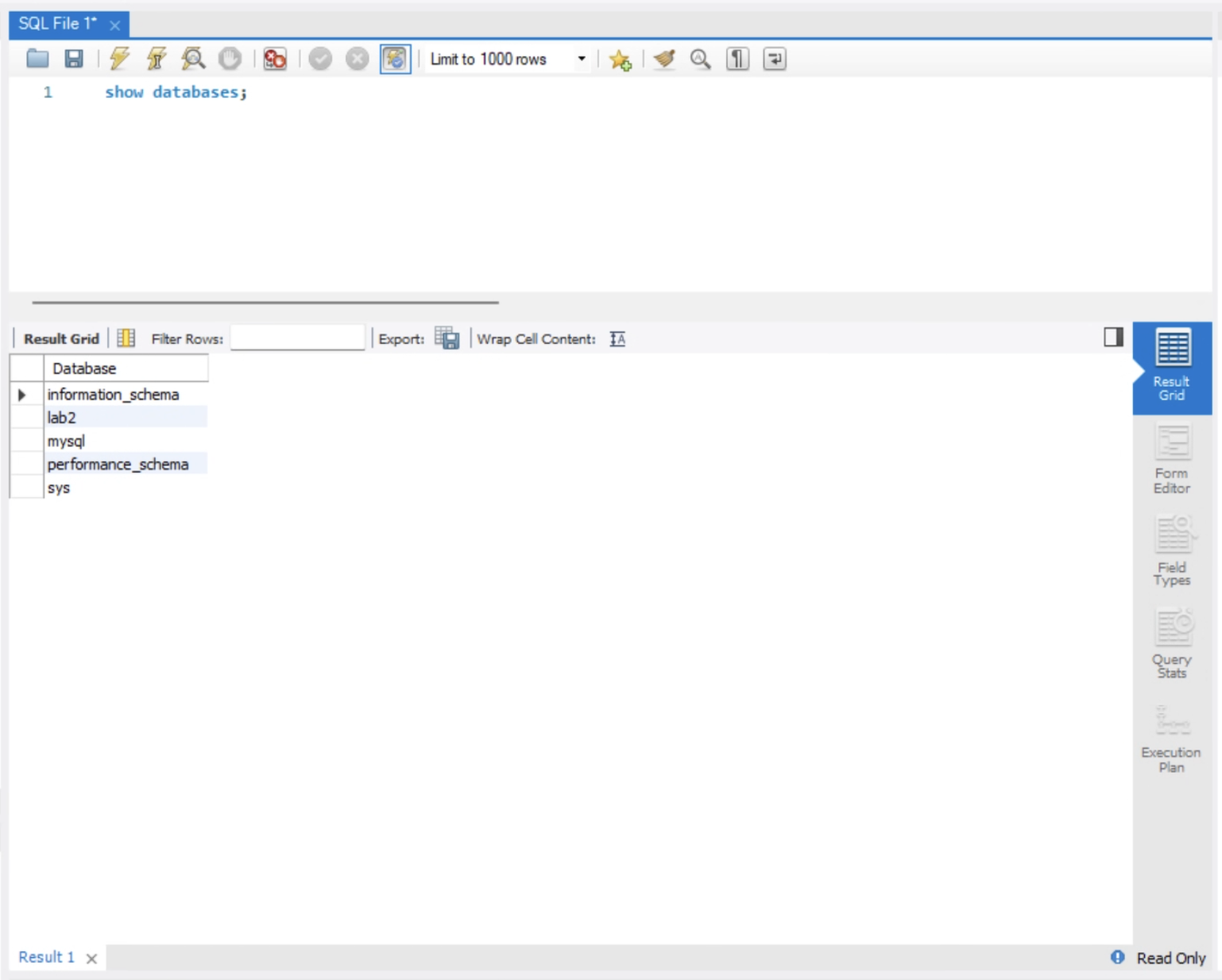
3.2 数据定义：**表的建立/删除/修改;** **索引的建立/删除；视图的建立/删除**



点击圈出的SQL+按钮，进入如下编辑界面

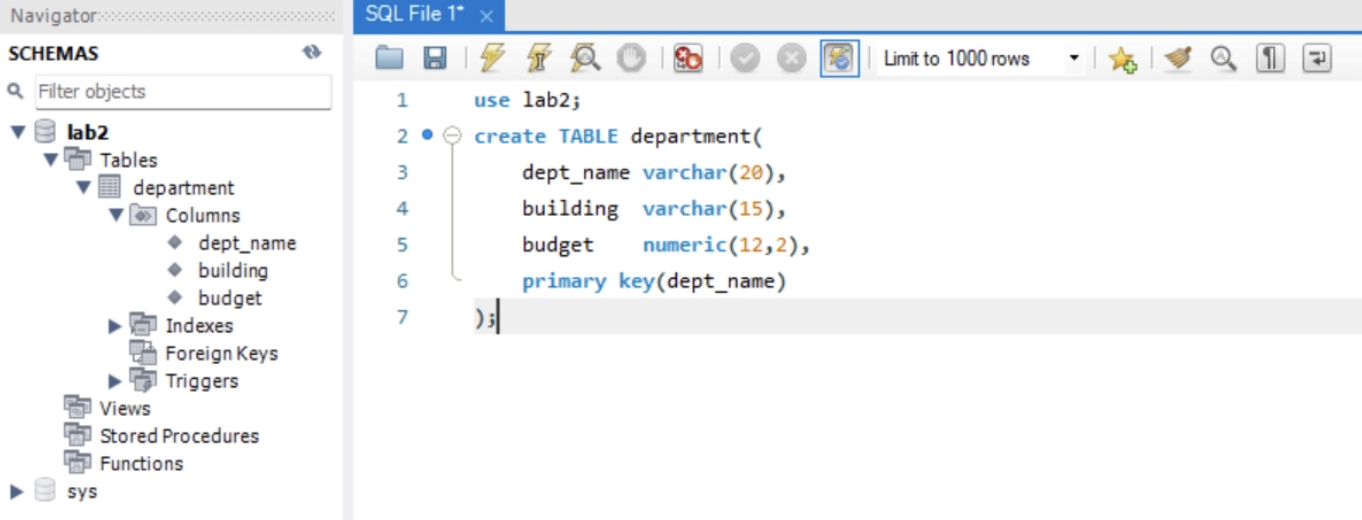


输入SQL语句，点击上方闪电按钮，刷新得到下面的显示：



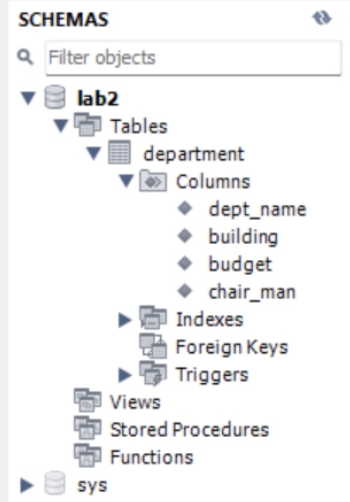
3.2.1 表的建立

输入下面的语句，点击执行，在刷新后可以看到多了一张表department



3.2.2 表的修改

运用 Alter 语句对表进行修改，输入alter table department add chair\_man varchar(20);执行语句后点击刷新就可以看到表内多了一个名为chair\_man的属性

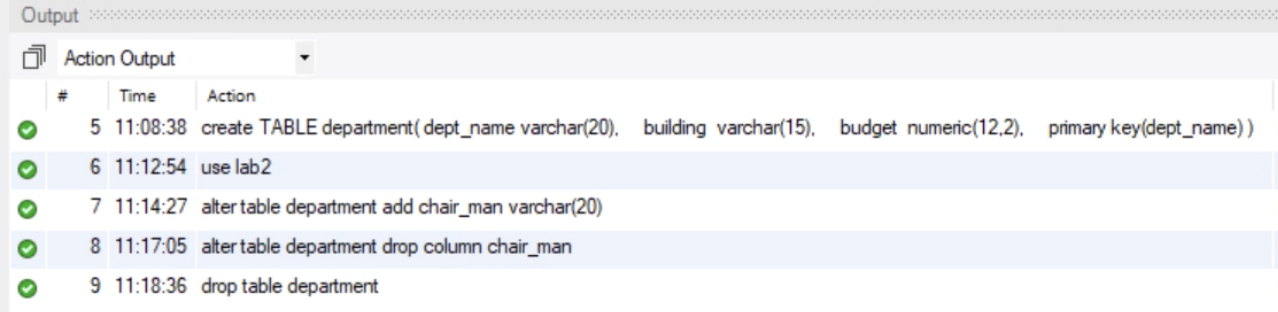


接下来执行：alter table department drop column chair\_man;就可以把刚才添加的属性删除。

3.2.3 表的删除

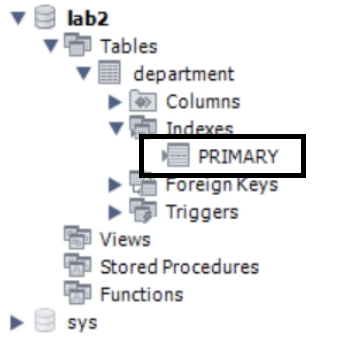
执行语句：drop table department;就可以把刚才建立的表删除。

以下是前述指令的执行情况，均执行成功。

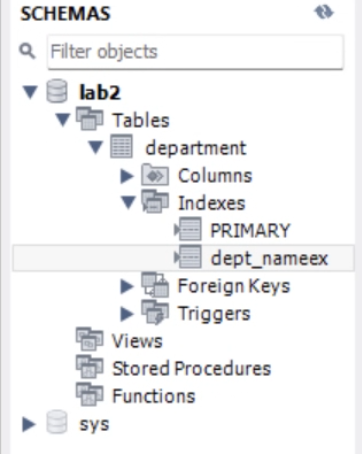


3.2.4 索引的建立

重新建立表以后，未手动建立索引时，只有系统自动为primary key建立的索引。



我们使用指令：create index dept\_nameex on department(dept\_name);



3.2.5 索引的删除

使用指令：drop index dept\_nameex on department;即可将刚刚建立的索引删除。

3.2.6 视图的建立

使用下面的指令建立视图：

create view Brighton\_department

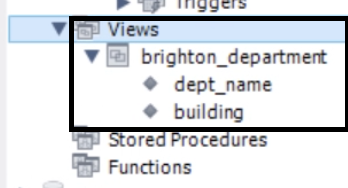
as

select dept\_name, building

from department

where dept\_name = 'Brighton'

;



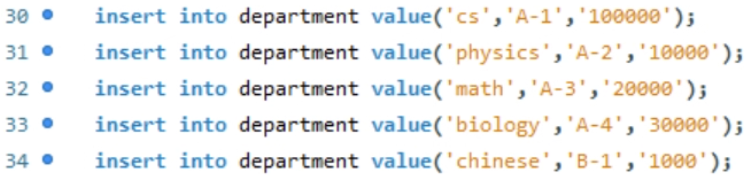
3.2.7 视图的删除

执行指令：drop view Brighton\_department;即可删除视图

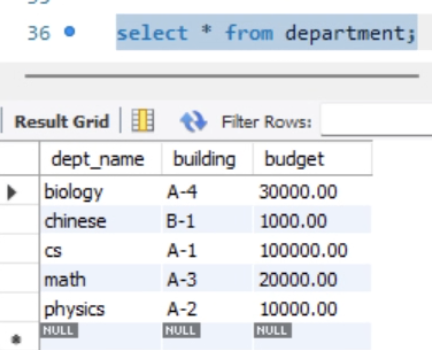
3.3 数据更新用 insert/delete/update命令插入/删除/修改表数据。

3.3.1 数据插入：

Insert



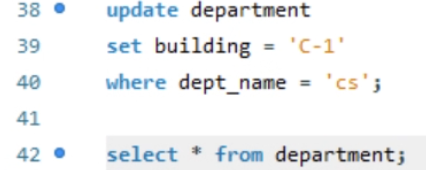
接着我们执行 select \* from department；



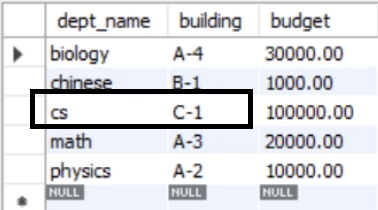
可以看到我们刚刚到五条插入都完成了。

3.3.2 数据修改

我们可以使用update指令修改数据



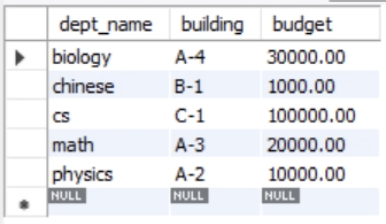
使用select \* from department 来查询



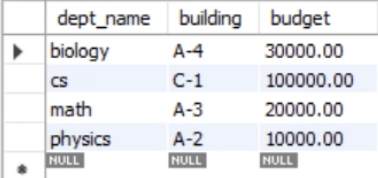
可以看到cs的building得到了修改由’A-1’变成了’C-1’

3.3.3**删除数据**

执行指令：delete from department where budget = 1000



结果如下：主键名为’chinese’的元组被删除了；



在删除的时候遇到问题在章节四中写出。

4 数据查询

4.1 单表查询

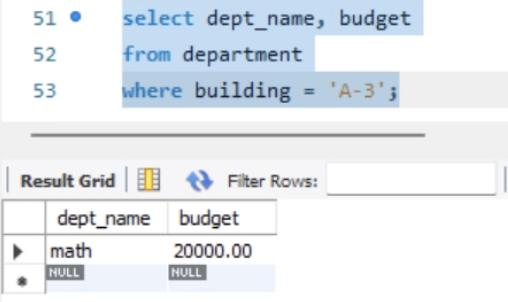
运行如下的查询语句：

select dept\_name, budget

from department

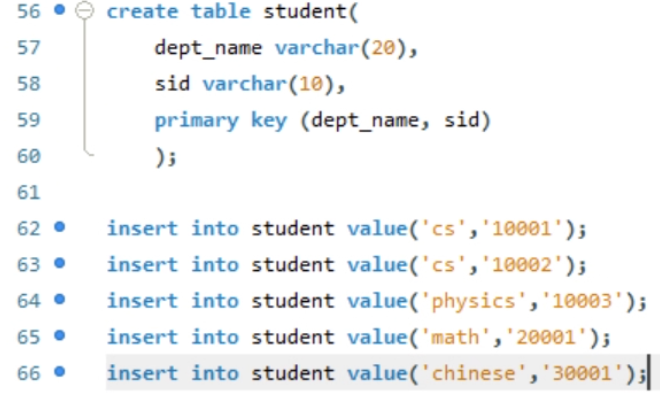
where building = 'A-3';

可以看到结果显示了 building = ‘A-3’的数据条目



4.2 多表查询

多表查询，我们再建立几个表

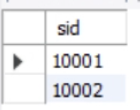


然后执行多表查询：

select sid

from student, department

where student.dept\_name = department.dept\_name AND department.budget = 100000;



结果与插入相符合，budget为100000的department为cs，查询到两个结果。

4.3 **嵌套子查询：**

嵌套子查询实际上就是把内部的查询结果作为外层的查询条件，执行下面语句

select max(budget) as bud\_max

from (

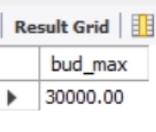
select budget, dept\_name

from department as d

where d.dept\_name <> 'cs'

) as no\_cs;

我们找到了除了cs以外的具有最大budget的部门的budget。



5. 视图操作：通过视图的数据查询和数据修改

重新建立视图

create view cs\_department

as

select dept\_name, building

from department

where dept\_name = 'cs'

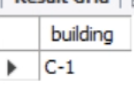
;

输入下面的语句

select building

from cs\_department;

这条命令是在视图cs\_department中查询building，结果如下



数据修改：

update cs\_department

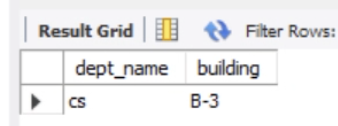
SET building = 'B-3'

where dept\_name = 'cs';

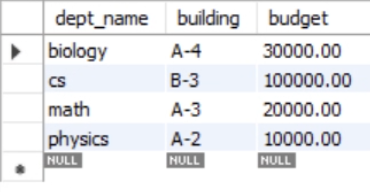
查询结果

select \* from cs\_department;

从view的角度看，属性得到修改了



select \* from department;



从另一个表中也可以看到cs.building值也发生了改变。

四、遇到的问题及解决方法

[error 1146]在执行索引删除的指令时出现报错，原因是指令最后的分号误输入为中文分号，导致错误。检查输入以后完成指令。

[error code: 1175]在执行数据删除的时候出现报错，原因是mysql运行在safe-update模式下，导致非主键条件下无法执行删除操作；随后我使用指令SET SQL\_SAFE\_UPDATES = 0;关闭了safe-update模式，再次执行指令就可以通过了。

[Error Code: 1248]. Every derived table must have its own alias。在使用嵌套子查询的过程中出现错误，开始查询时并未添加as no\_cs，这导致了这个查询的表没有自己的名字。在多级查询中要求给每个表都有一个别名才能完成搜索。

五、总结

本次实验中，我们完成了新建数据库、数据定义、数据更新、数据查询、以及视图操作，基本完成了表的建立/删除/修改；索引的建立/删除；视图的建立/ 删除；能用insert/delete/update 命令插入/删除/修改数据；以及单表/多表/嵌套子查询。圆满地完成了实验内容和操作，也在一些操作中发现了一些规范性的操作问题，在实践中得到了学习。