

实验 2 SQL 数据定义和操作

学号：3200102324

姓名：王晨雨

实验目的：


1. 掌握 MySQL 数据库的复杂 SQL 查询和 DML 语句的使用方法；
2. 掌握 MySQL 数据库的视图设计和使用方法。
3. 掌握 MySQL 数据库的索引设计和使用方法。

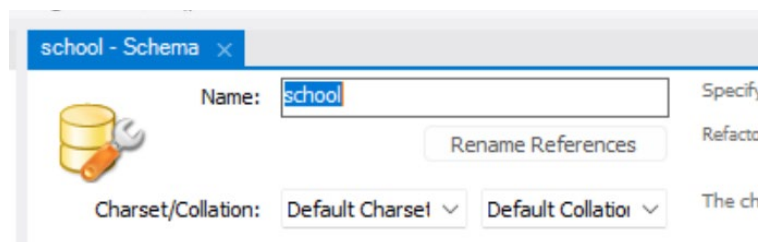
实验平台：

1. 数据库管理系统：MySQL

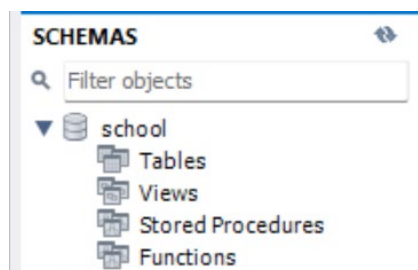
实验内容和要求：

1. 建立数据库 school；

- a) 点击图标 ，选择 Create a new schema in the connected server
- b) 填写新建数据库的名称



可以在左侧的 SCHEMAS 窗口中看到新创建的数据库：



2. 数据的定义：

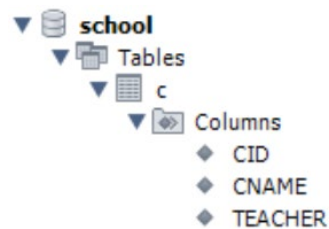
- a) 首先需要打开一个可以写 SQL 语句的文本框，点击
- b) 用 create 创建三张表



代码如下：

```
1. use school;
2. create table S
3. (
4.     SID char(2),
5.     SNAME varchar(20),
6.     AGE INT,
7.     SEX char(1)
8. );
```

点击运行后刷新可见：



同理可创建 S, SC, C 三张表：

S (SID, SNAME, AGE, SEX)

中文语义：学生（学号，姓名，年龄，性别）

SC (SID, CID, GRADE)

中文语义：学习（学号，课程号，成绩）

C (CID, CNAME, TEACHER)

中文语义：课程（课程号，课程名，任课教师）

c) 用 alter 增加主键和外键

为表添加外键约束的语法格式如下：

Alter table 表名 add primary key (列名);

```
1. use school;
2. ALTER table S add primary key (SID);
3. ALTER table C add primary key (CID);
```

d) 用 insert 添加如下数据：

S1	Wang feng	20	M
S2	Liu fang	19	M
S3	Chen yun	22	M
S4	Wu kai	19	M
S5	Liu li	21	F
S6	Dong qing	18	F

C2	maths	Ma
C4	physics	Shi
C3	chemistry	Zhou
C1	DB	Li
C5	OS	Wen

S1	C1	70
S2	C1	81
S3	C1	92
S4	C1	85
S1	C2	65
S2	C3	57
S3	C5	87

插入一行数据语法如下：

```
1. Insert into S values ('S1', 'Wang feng',  
20, 'M');
```

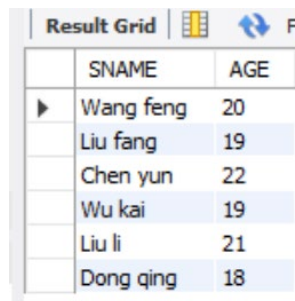
3. 数据查询：

单表查询，语法：

```
SELECT 字段 1, 字段 2... FROM 表名  
WHERE 条件  
GROUP BY field  
HAVING 筛选  
ORDER BY field  
LIMIT 限制条数
```

a) 查询全体学生的姓名、年龄；

```
1. use school;  
2. select SNAME, AGE  
3. from S ;
```

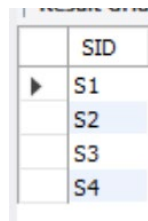


	SNAME	AGE
▶	Wang feng	20
	Liu fang	19
	Chen yun	22
	Wu kai	19
	Liu li	21
	Dong qing	18

b) 查询所有选修过课的学生的学号；

Sql 代码如下：

```
use school;  
select distinct SID From SC;  
结果如下：
```



	SID
▶	S1
	S2
	S3
	S4

c) 查询考试成绩低于 60 分的学生的学号；

```
use school;  
select distinct SID From SC  
where GRADE < 60;
```

结果如下：



	SID
▶	S2

- d) 查询年龄在 20 至 23 之间的学生姓名、性别和年龄；
代码如下：

```
use school;
select SNAME, SEX, AGE
from S
where AGE between 20 and 23;
```

	SNAME	SEX	AGE
▶	Wang feng	M	20
	Chen yun	M	22
	Liu li	F	21

- e) 查询所有姓 liu 的学生的学号、姓名和年龄；

```
use school;
select SNAME, SID, AGE from S
where SNAME like 'Liu %';
```

	SNAME	SID	AGE
▶	Liu fang	S2	19
	Liu li	S5	21

- f) 查询学习 C1 课程的学生最高分数；

```
use school;
select max(GRADE) from SC
where CID = 'C1';
```

	max(GRADE)
▶	92

- g) 查询各个课程号与相应的选课人数；

```
Select CID, COUNT(CID) from SC
group by CID
Order by (COUNT(CID)) DESC
```

	CID	COUNT(CID)
▶	C1	4
	C2	1
	C3	1
	C5	1

- h) 查询选修 C3 课程的学生的姓名；

```
Select SNAME from S, SC
where CID = 'C3' and S.SID = SC.SID
```

	SNAME
▶	Liu fang

- i) 查询选修 C1 课程且成绩在 90 分以上的所有学生姓名。

```
Select SNAME from S, SC
where CID = 'C1' and S.SID = SC.SID and
GRADE > 90;
```

	SNAME
►	Chen yun

j) 查询每一门课程的平均成绩

```
SELECT SC.CID, avg(GRADE) from sc
group by sc.cid;
```

	CID	avg(GRADE)
►	C1	82.0000
	C2	65.0000
	C3	57.0000
	C5	87.0000

k) 查询授课教师名字中包括 i 的课程名

```
SELECT CNAME from C
where TEACHER like '%i%';
```

	CNAME
►	DB
	physics

l) 查询课程成绩在 80~90 的学生编号、课程编号
多表查询，

```
select s.SID, c.CID from s,sc, C
where s.sid=sc.sid and sc.cid=C.cid and
grade>70;
```

	SID	CID
►	S2	C1
	S3	C1
	S4	C1
	S3	C5

m) 查询最高成绩为 80 以上的课程名、最高成绩
嵌套查询；

```
select C.CNAME, MAX(GRADE) from C, SC
having MAX(GRADE) > 80;
```

	CNAME	MAX(GRADE)
►	OS	92

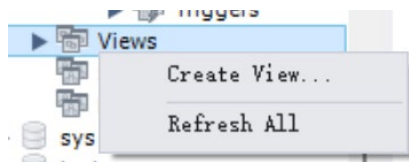
n) 查询 c1 课程在平均分以上的学生信息

```
select * from(select C.CNAME, MAX(GRADE) maxgrade
from C, SC where C.CID = SC.CID
group by sc.cid)a where maxgrade > 80;
```

	CNAME	maxgrade
►	DB	92
	OS	87

4. 视图使用：根据表 S, SC, C 创建视图 S_C_SC, 使其具有如下内容（学号，姓名，课程名，成绩）：

右击 SCHEMAS 中的 VIEWS 选项，选择 Create View



输入代码：

```
CREATE VIEW `S_C_SC` AS select S.SID, S.SNAME,
C.CNAME, SC.GRADE from C, SC, s
      where S.SID = SC.SID and C.CID = SC.CID;
```

点击 Apply，新建 view 如下

	SID	SNAME	CNAME	GRADE
▶	S1	Wang feng	DB	70
	S2	Liu fang	DB	81
	S3	Chen yun	DB	92
	S4	Wu kai	DB	85
	S1	Wang feng	maths	65
	S2	Liu fang	chemistry	57
	S3	Chen yun	OS	87

使用视图查询姓 Wu 的各门课程成绩：

```
SELECT SNAME, CNAME, GRADE FROM school.s_c_sc
where SNAME like "Wu%";
```

	SNAME	CNAME	GRADE
▶	Wu kai	DB	85

修改视图中的数据观察视图和原关系表中的变化：

例如：将 Wukai 的分数改为 100：

```
update s_c_sc set GRADE = 100
      where SNAME = 'Wu kai';
```

查询视图：

	SID	SNAME	CNAME	GRADE
▶	S1	Wang feng	DB	70
	S2	Liu fang	DB	81
	S3	Chen yun	DB	92
	S4	Wu kai	DB	100
	S1	Wang feng	maths	65
	S2	Liu fang	chemistry	57
	S3	Chen yun	OS	87

查询原关系表：

	SID	CID	GRADE
▶	S1	C1	70
	S2	C1	81
	S3	C1	92
	S4	C1	100
	S1	C2	65
	S2	C3	57
	S3	C5	87

发现 Wukai 的分数相应的发生了改变。

删除视图

```
Drop view s_c_sc;
```

5. 索引的使用:以课程编号为索引为C创建索引 INDEX_CID 使用查询语句,查看使用索引后数据的变化,删除索引。

```
create index INDEX_CID on C(CID);
```

查看索引:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
c	0	PRIMARY	1	CID	A	5				BTREE			YES	
c	1	INDEX_CID	1	CID	A	5				BTREE			YES	

删除索引:

```
drop index INDEX_CID on c;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation
c	0	PRIMARY	1	CID	A

6. 数据更新:

- a) 用 update 命令修改 c 表中课程号为 C3 的课程课程号修改为 C6

```
update c set CID = 'C6'
where CID = 'C3';
```

	CID	CNAME	TEACHER
►	C1	DB	Li
	C2	maths	Ma
	C4	physics	Shi
	C5	OS	Wen
	C6	chemistry	Zhou
*	NULL	NULL	NULL

- b) 用 update 命令修改 sc 表中课由 “Li” 任教的课程号为由 “Wen” 任教的课程号

```
update sc, c set sc.CID = c.CID
where sc.cid = (select cid from c where teacher
= 'Li') and c.teacher = 'wen';
```

	SID	CID	GRADE
►	S1	C5	70
	S2	C5	81
	S3	C5	92
	S4	C5	100
	S1	C2	65
	S2	C3	57
	S3	C5	87

- c) 用 delete 命令删除表 sc 中分数在 60 以下的记录

```
delete from sc where grade < 60;
```

	SID	CID	GRADE
▶	S1	C5	70
	S2	C5	81
	S3	C5	92
	S4	C5	100
	S1	C2	65
	S3	C5	87

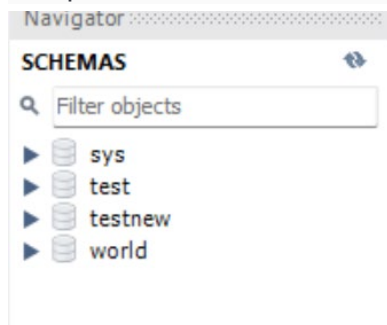
d) 用 delete 命令删除表 sc 中课程名为 “OS” 的记录。

```
delete from sc
  where sc.cid = (select cid from c where cname
= 'OS');
```

	SID	CID	GRADE
▶	S1	C2	65

7. 删除三张关系表和数据库

```
drop table s;
drop table sc;
drop table c;
drop database school;
```



可以看到右侧 SCHEMAS 窗口中所创建的 school 数据库已经消失。