

《数据库概论》实验一：用 SQL 进行数据操作 实验报告

姓名 吴紫航 学号 171860659 联系方式 邮箱: 401986905@qq.com

实验环境

[一句话介绍你使用的操作系统、软件版本]

操作系统: windows 10

软件版本: MySQL Workbench 8.0 CE

实验过程

[实验的详细过程, 必须包含所有 SQL 语句和要求的实验截图]

1.建立四个基本表

SQL 语句:

```
CREATE TABLE IF NOT EXISTS `Course`(  
  `id` Int NOT NULL,  
  `title` char(20) NOT NULL,  
  `dept_name` char(2),  
  `credit` INT,  
  PRIMARY KEY (`id`)  
);
```

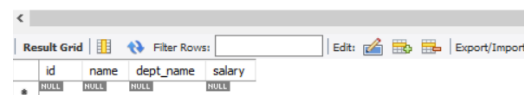
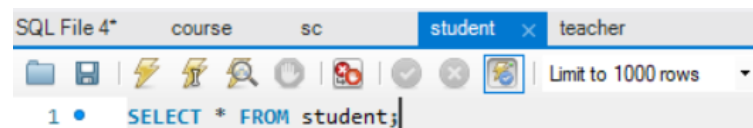
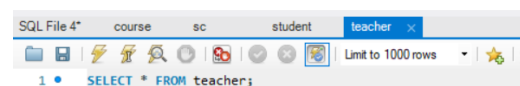
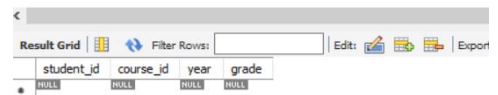
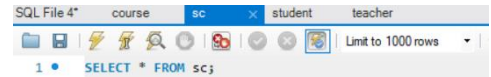
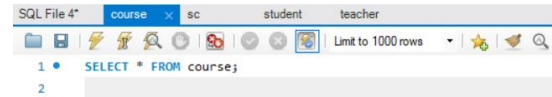
```
CREATE TABLE IF NOT EXISTS `Student`(  
  `id` INT NOT NULL,  
  `name` CHAR(20) NOT NULL,  
  `dept_name` CHAR(20),  
  PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE IF NOT EXISTS `SC` (  
  `student_id` INT NOT NULL,  
  `course_id` INT NOT NULL,  
  `year` INT,  
  `grade` INT,  
  PRIMARY KEY(`student_id`,`course_id`)  
);
```

```
CREATE TABLE IF NOT EXISTS `Teacher`(  
  `id` INT NOT NULL,  
  `name` CHAR(8) NOT NULL,  
  `dept_name` CHAR(2),  
  `salary` INT,  
  PRIMARY KEY(`id`));
```

截图：

4 个 table 如下



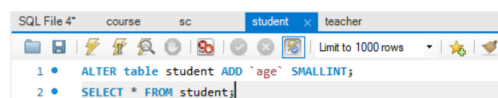
2.修改基本表

2.1. 在 Student 表加入属性 age (smallint 型)

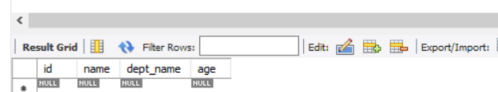
SQL 语句：

```
ALTER table student ADD `age` SMALLINT;
```

```
SELECT * FROM student;
```



截图：



SQL File 4*		course	sc	student	teacher	experiment1.student	
Info	Columns	Indexes	Triggers	Foreign keys	Partitions	Grants	DDL
Column	Type	Default Value	Nullable	Character S			
id	int(11)		NO				
name	char(20)		NO	latin1			
dept_name	char(20)		YES	latin1			
age	smallint(6)		YES				

增加了列 age smallint

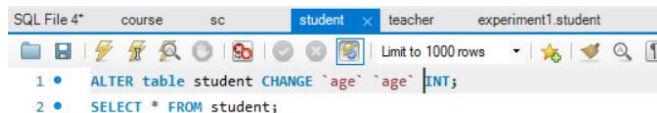
2.2. 将 Student 表中的属性 age 类型改为 int 型.

SQL 语句:

```
ALTER table student CHANGE `age` `age` INT;
```

```
SELECT * FROM student;
```

截图:



SQL File 4*		course	sc	student	teacher	experiment1.student	
Info	Columns	Indexes	Triggers	Foreign keys	Partitions	Grants	DOL
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Ext
id	int(11)		NO			select,insert,update,references	
name	char(20)		NO	latin1	latin1_swedish_d	select,insert,update,references	
dept_name	char(20)		YES	latin1	latin1_swedish_d	select,insert,update,references	
age	int(11)		YES			select,insert,update,references	

3.插入数据

3.1. 向 Student 表插入不少于 10 条数据

SQL 语句:

```
INSERT INTO student
```

```
('id','name','dept_name','age')
```

```
VALUES
```

```
(0000,'N0','D0',20),
```

```
(0001,'N1','D1',20),
```

```
(0002,'N2','D2',20),
```

```
(0003,'N3','D3',20),
```

```
(0004,'N4','D4',20),
```

```
(0005,'N5','D5',20),
```

```
(0006,'N6','D6',20),
```

```
(0007,'N7','D7',20),
```

```
(0008,'N8','D8',20),
```

```
(0009,'N9','D9',20)
```

```
;
```

```
SELECT * FROM student;
```

截图：

SQL File 4*

course

sc

teacher

student

experiment1.student

Limit to 1000 rows

```
1
2 • INSERT INTO student
3   ('id','name','dept_name','age')
4   VALUES
5   (0000,'N0','D0',20),
6   (0001,'N1','D1',20),
7   (0002,'N2','D2',20),
8   (0003,'N3','D3',20),
```

Result Grid

Filter Rows:

Edit:

Export/Import:

	id	name	dept_name	age
0	N0	D0	20	
1	N1	D1	20	
2	N2	D2	20	
3	N3	D3	20	
4	N4	D4	20	
5	N5	D5	20	
6	N6	D6	20	
7	N7	D7	20	
8	N8	D8	20	
9	N9	D9	20	
*	NULL	NULL	NULL	

3.2. 向 Course 表插入不少于 5 条数据。注意，应该包含计算机系（CS），包含数据库课（DB）

SQL 语句：

INSERT INTO course

('id','title','dept_name','credit')

VALUES

(000,'DataBase','CS',3),

(001,'C1','D1',2),

(002,'C2','D2',3),

(003,'C3','D3',4),

(004,'C4','D4',5),

(005,'C5','D5',2),

(006,'C6','D6',3)

;

SELECT * FROM course;

截图：

SQL File 4* course sc teacher student experiment1.student

Limit to 1000 rows

```
7   (002,'C2','D2',3),
8   (003,'C3','D3',4),
9   (004,'C4','D4',5),
10  (005,'C5','D5',2),
11  (006,'C6','D6',3)
12
13 • SELECT * FROM course;
14
```

<

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell

	id	title	dept_name	credit
▶	0	DataBase	CS	3
1	C1	D1	2	
2	C2	D2	3	
3	C3	D3	4	
4	C4	D4	5	
5	C5	D5	2	
6	C6	D6	3	
*	NULL	NULL	NULL	NULL

3.3. 向 SC 表插入不少于 20 条数据。注意，应该包含 2018 年前后课程

SQL 语句：

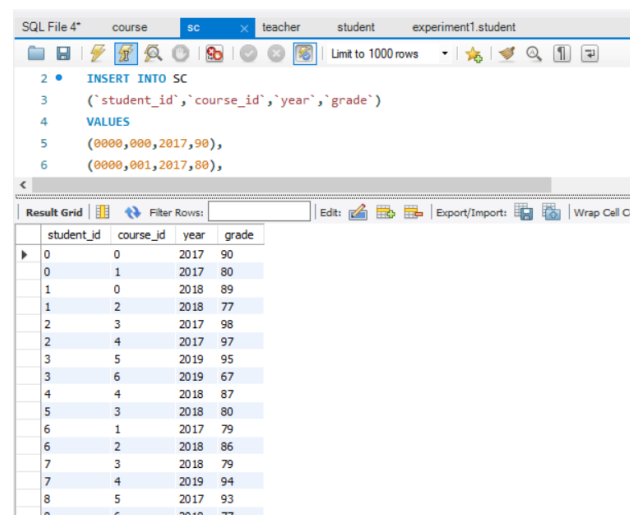
```

INSERT INTO SC
('student_id','course_id','year','grade')
VALUES
(0000,000,2017,90),
(0000,001,2017,80),
(0001,000,2018,89),
(0001,002,2018,77),
(0002,003,2017,98),
(0002,004,2017,97),
(0003,005,2019,95),
(0003,006,2019,67),
(0004,004,2018,87),
(0005,003,2018,80),
(0006,001,2017,79),
(0006,002,2018,86),
(0007,003,2018,79),
(0007,004,2019,94),
(0008,005,2017,93),
(0008,006,2018,77),
(0009,003,2018,76),
(0009,001,2018,88),
(0009,000,2018,67)
;

SELECT * FROM SC;

```

截图：



student_id	course_id	year	grade
0	0	2017	90
0	1	2017	80
1	0	2018	89
1	2	2018	77
2	3	2017	98
2	4	2017	97
3	5	2019	95
3	6	2019	67
4	4	2018	87
5	3	2018	80
6	1	2017	79
6	2	2018	86
7	3	2018	79
7	4	2019	94
8	5	2017	93
8	6	2018	77

3.4. 向 Teacher 表插入不少于 3 条数据。注意，应该包含姓胡的教师

SQL 语句：

```

INSERT INTO teacher
('id','name','dept_name','salary')
VALUES

```

```
(0000,'Zhangsan','D1',30000),
(0001,'Lisi','D2',20000),
(0002,'Hutao','D3',24000),
(0003,'Huwei','CS',99999),
(0004,'Wangwu','CS',50000)
;
```

```
SELECT * FROM teacher;
```

截图：

The screenshot shows the SQL Developer interface with a script editor containing an INSERT statement and a SELECT statement. Below the editor, the 'Result Grid' displays the data inserted into the 'teacher' table.

id	name	dept_name	salary
0	Zhangsan	D1	30000
1	Lisi	D2	20000
2	Hutao	D3	24000
3	Huwei	CS	99999
4	Wangwu	CS	50000

4.1 找出所有至少选修了一门计算机系课程的学生姓名，保证结果中没有重复的姓名。

SQL 语句：

```
select distinct student.`name`
from student,course,sc
where (sc.course_id=course.id) and (course.dept_name='CS')and(sc.student_id=student.id)
;
```

截图：

The screenshot shows the SQL Developer interface with a query editor containing a SELECT statement. Below the editor, the 'Result Grid' displays the results of the query.

name
N0
N1
N9

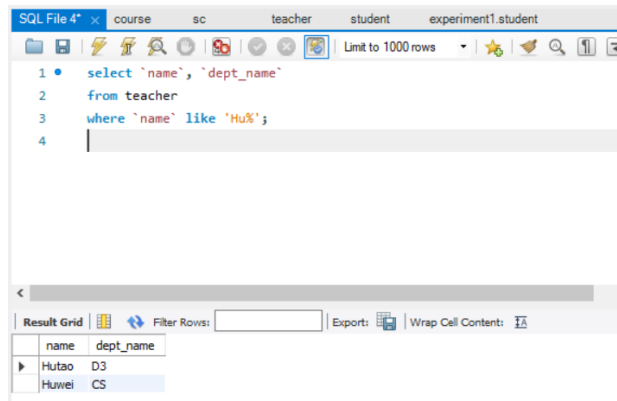
4.2 找出所有姓胡的教师的姓名和院系

SQL 语句：

```
select `name`,`dept_name`
from teacher
```

where `name` like 'Hu%';

截图：

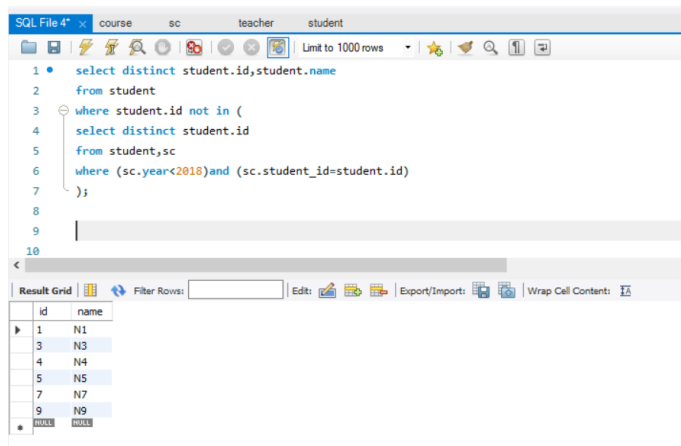


4.3 找出所有没有选修在 2018 年之前（不含 2018 年）开设的任何课程的学生 ID 和姓名

SQL 语句：

```
select distinct student.id, student.name
from student
where student.id not in (
select distinct student.id
from student, sc
where (sc.year < 2018) and (sc.student_id = student.id)
);
```

截图：



4.4 找出每个系教师的最高工资值。可以假设每个系至少有一位教师

SQL 语句：

```
select teacher.dept_name, max(teacher.salary)
from teacher
group by teacher.dept_name
```

截图：

SQL File 4* course sc teacher student

```

1 • select teacher.dept_name, max(teacher.salary)
2   from teacher
3  group by teacher.dept_name
4
5
6

```

Result Grid

dept_name	max(teacher.salary)
CS	99999
D1	30000
D2	20000
D3	24000

注：因为假设每个院系都有老师，所以对 `teacher` table 操作，按院系分类 `group` 即可

5. 修改数据

5.1. 将数据库课的学生成绩全部加 2 分

SET SQL_SAFE_UPDATES=0;

SQL 语句：

```
update sc set sc.grade=sc.grade+2
```

```
where sc.course_id in(
```

```
select distinct course.id
```

```
from course
```

```
where course.title='DataBase')
```

```
;
```

```
select *
```

```
from sc,course
```

```
where (course.title='DataBase')and(sc.course_id=course.id)
```

```
;
```

截图：

SQL File 4* course sc teacher student

```

1 • select *
2   from sc,course
3  where (course.title='DataBase')and(sc.course_id=course.id)
4
5
6
7
8

```

Result Grid

student_id	course_id	year	grade	id	title	dept_name	credit
0	0	2017	90	0	DataBase	CS	3
1	0	2018	89	0	DataBase	CS	3
9	0	2018	67	0	DataBase	CS	3

增加前


```

1 • SET SQL_SAFE_UPDATES=0;
2
3 • update sc set sc.grade=sc.grade+2
4   where sc.course_id in(
5     select distinct course.id
6     from course
7     where course.title='DataBase')
8   ;
9
10 • select *
11   from sc,course
12  where (course.title='DataBase')and(sc.course_id=course.id)
13   ;
14

```

student_id	course_id	year	grade	id	title	dept_name	credit
0	0	2017	92	0	DataBase	CS	3
1	0	2018	91	0	DataBase	CS	3
9	0	2018	69	0	DataBase	CS	3

增加 2 分后

这里注意第一行代码，先修改一下安全设置模式，否则更新会报错

6. 删除数据

6.1. 删除均分不足 80 分的所有学生的选课记录

SQL 语句：

```

delete from sc
where sc.student_id in(
select * from(
select sc.student_id
from sc
group by sc.student_id
having avg(sc.grade)<80
)a)
;

```

select * from sc;

截图：

```

1 • select sc.student_id,avg(sc.grade)
2   from sc
3  group by sc.student_id
4   ;
5
6

```

student_id	avg(sc.grade)
0	86.0000
1	84.0000
2	97.5000
3	81.0000
4	87.0000
5	80.0000
6	82.5000
7	86.5000
8	85.0000
9	77.6667

所有学生的均分

```

1 • select sc.student_id,avg(sc.grade)
2   from sc
3  group by sc.student_id
4 having avg(sc.grade)<80
5
6
7

```

student_id	avg(sc.grade)
9	77.6667

不足 80 分学生的均分

```

1 • select * from sc;
2
3

```

student_id	course_id	year	grade
2	3	2017	98
2	4	2017	97
3	5	2019	95
3	6	2019	67
4	4	2018	87
5	3	2018	80
6	1	2017	79
6	2	2018	86
7	3	2018	79
7	4	2019	94
8	5	2017	93
8	6	2018	77
9	0	2018	69
9	1	2018	88
9	3	2018	76
NULL	NULL	NULL	NULL

删除前选课记录详情

```

1 • delete from sc
2   where sc.student_id in(
3     select * from(
4       select sc.student_id
5       from sc

```

student_id	course_id	year	grade
1	0	2018	91
1	2	2018	77
2	3	2017	98
2	4	2017	97
3	5	2019	95
3	6	2019	67
4	4	2018	87
5	3	2018	80
6	1	2017	79
6	2	2018	86
7	3	2018	79
7	4	2019	94
8	5	2017	93
8	6	2018	77
NULL	NULL	NULL	NULL

删除均分不足 80 学生的所有选课记录后

7. 视图操作

7.1. 创建一个视图，记录每位学生已修课程的总学分数

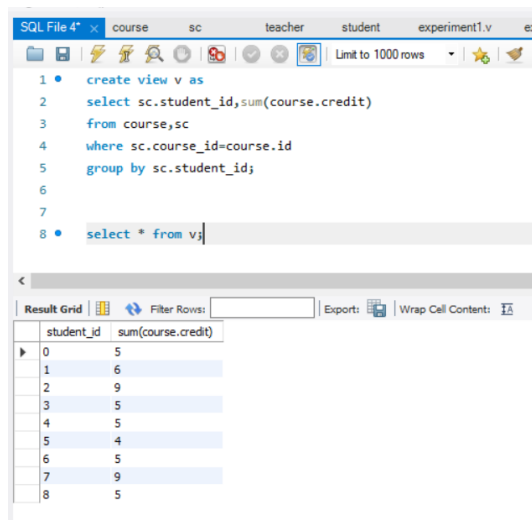
SQL 语句：

create view v as

```
select sc.student_id,sum(course.credit)
from course,sc
where sc.course_id=course.id
group by sc.student_id;
```

```
select * from v;
```

截图：



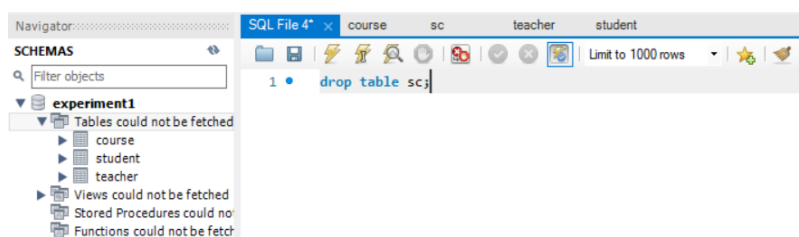
8. 删除基本表

8.1. 删除 SC 表

SQL 语句：

```
drop table sc;
```

截图：



实验中遇到的困难及解决办法

[详细说明你认为本次实验中比较困难的地方，也可以对实验设计提出建议]

(1)刚开始使用 mysql 时，发现修改表后，要 refresh 一下才能刷新出新状态

解决方法：和室友讨论得出解决方案

(2)一些关键词的使用语法和课本讲述并不完全一致比如 create

解决方法：搜索引擎搜索、重复尝试

(3)第六题的代码

```
delete from sc
where sc.student_id in(
select * from(
select sc.student_id
from sc
group by sc.student_id
having avg(sc.grade)<80
)a)
;
```

如果改成直观的

```
delete from sc
where sc.student_id in(
select sc.student_id
from sc
group by sc.student_id
having avg(sc.grade)<80
)
;
```

在 mysql 中会报错 **You can't specify target table for update in FROM clause**，查询搜索引擎后了解到需要再加一层封装，mysql 中特有的问题，比如 oracle 中就没有

(4)关于均分少于 80 的学生那题，所谓均分是否考虑加权均分，即每门课的 credit 学分数目是否需要考虑，如果需要考虑，则情况变得复杂，按题目本意应该是直接调用 AVG 函数

解决方案：和胡伟老师讨论，得出不需要考虑加权

(5)第五题需要先修改安全设置模式，否则会报错：无法修改

解决方案：搜索引擎

参考文献及致谢

[如果你参考了任何书籍、网页，或与他人进行了讨论，请在此注明]

(1)参考书籍：

数据库系统实用教程/徐浩磐，柏文阳，刘奇志. 北京：高等教育出版社，2006. 6 ISBN, 7-04-019584-4

(2)参考网址：

<https://www.runoob.com/mysql/mysql-tutorial.html> 菜鸟教程 MYSQL

<https://blog.csdn.net/yihanzhi/article/details/81065573> IN 的用法

<https://www.cnblogs.com/hhe0/p/9556070.html> GROUP BY 的用法

<https://blog.csdn.net/qxxyy99/article/details/79980005> MAX 的用法

<https://www.jianshu.com/p/d69a269d8018> UPDATE 的用法

<https://blog.csdn.net/a837201942/article/details/78713775> 安全更新模式的设置问题

<https://blog.csdn.net/fdipzone/article/details/52695371> You can't specify target table for update in FROM clause

错误的解决方法

(3)致谢讨论人：感谢舍友徐浩、老师胡伟