

# 《数据库概论》实验一：用 SQL 进行数据操作 实验报告

## 实验环境

Windows10, MySQL5.5

## 实验过程

### 1. 使用 SQL 语句建立基本表(Course, Student, SC, Teacher)

```
1 create table course(  
2 id int primary key not null,  
3 title varchar(20) not null,  
4 dept_name varchar(20),  
5 credit int  
6 );  
7  
8 create table student(  
9 id int primary key not null,  
10 name varchar(20) not null,  
11 dept_name varchar(20),  
12 major_name varchar(20)  
13 );  
14  
15 create table sc(  
16 student_id int,  
17 course_id int,  
18 year int,  
19 grade int,  
20 primary key(student_id, course_id),  
21 foreign key (student_id) references student,  
22 foreign key (course_id) references course  
23 )ENGINE=MyISAM;  
24  
25 create table teacher(  
26 id int primary key not null,  
27 name varchar(20) not null,  
28 dept_name varchar(20),  
29 salary int  
30 );
```

### 2. 使用 SQL 语句修改基本表

- 1) 在 Student 表中加入属性 age (smallint 型)。

```
1 alter table student add age smallint;
```

- 2) 将 Student 表中的属性 age 类型改为 int 型。

```
1 alter table student modify column age int;
```

### 3. 用 SQL 语句插入数据

- 1) 向 Student 表插入不少于 20 条数据。

```
1 insert into student values
2 (1,'S1','D1','M1',18),
3 (2,'S2','D1','M1',18),
4 (3,'S3','D1','M2',18),
5 (4,'S4','D1','M2',18),
6 (5,'S5','D1','M3',19),
7 (6,'S6','D2','M4',20),
8 (7,'S7','D2','M4',18),
9 (8,'S8','D2','M5',18),
10 (9,'S9','D2','M6',19),
11 (10,'S10','D2','M7',19),
12 (11,'S11','D2','M7',20),
13 (12,'S12','D3','M8',21),
14 (13,'S13','D3','M8',21),
15 (14,'S14','D3','M8',18),
16 (15,'S15','D3','M9',19),
17 (16,'S16','D3','M9',20),
18 (17,'S17','D4','M10',18),
19 (18,'S18','D4','M10',19),
20 (19,'S19','D4','M11',20),
21 (20,'S20','D4','M11',22);|
```

- 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。

```
1 insert into course values
2 (1,'database','CS',3),
3 (2,'C2','CS',2),
4 (3,'C3','CS',3),
5 (4,'C4','D3',2),
6 (5,'C5','D4',4);|
```

3) 向 SC 表插入不少于 50 条数据。注意，应该包含 2018 年前后课程。

```
1 insert into sc values
2 (1,1,2017,80),
3 (1,2,2018,90),
4 (1,5,2019,87),
5 (2,3,2019,92),
6 (2,4,2017,40),
7 (3,2,2019,60),
8 (3,1,2019,77),
9 (3,4,2019,95),
10 (4,2,2016,93),
11 (4,5,2020,82),
12 (5,1,2017,84),
13 (5,3,2018,98),
14 (5,4,2019,83),
15 (6,2,2019,89),
16 (6,4,2019,56),
17 (7,1,2017,73),
18 (7,2,2016,98),
19 (7,3,2018,29),
20 (8,3,2019,67),
21 (8,5,2019,98),
22 (9,1,2017,80),
23 (9,4,2018,78),
24 (9,2,2018,96),
25 (10,2,2017,97),
26 (10,3,2017,83),
27 (11,5,2018,84),
28 (11,1,2019,37),
29 (11,3,2019,84),
30 (12,1,2019,73),
31 (12,4,2018,97),
32 (13,3,2017,28),
33 (13,4,2017,98),
34 (13,2,2017,84),
35 (14,1,2019,59),
36 (14,2,2019,37).
```

```
37 (15,1,2019,84),
38 (15,4,2018,59),
39 (15,3,2019,37),
40 (16,3,2017,82),
41 (16,5,2019,28),
42 (17,1,2017,83),
43 (17,2,2019,97),
44 (17,4,2018,82),
45 (18,3,2019,83),
46 (18,4,2019,82),
47 (19,1,2017,94),
48 (19,2,2017,38),
49 (19,5,2017,74),
50 (20,2,2019,93),
51 (20,3,2019,38);
```

- 4) 向 Teacher 表插入不少于 5 条数据。注意，应该包含姓胡的教师。

```
1 insert into teacher values
2 (1, 'HU1', 'D1', 8000),
3 (2, 'HU2', 'CS', 8000),
4 (3, 'T3', 'CS', 9000),
5 (4, 'T4', 'D3', 7000),
6 (5, 'T5', 'D2', 6000);
```

#### 4. 查询

- 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。

```
1 select distinct student.name
2 from student, sc, course
3 where student.id=sc.student_id and sc.course_id=course.id
4 and course.dept_name='CS';
```

X	name
	S1
	S2
	S3
	S4
	S5
	S6
	S7
	S8
	S9
	S10
	S11
	S12
	S13
	S14
	S15
	S16
	S17
	S18
	S19
	S20

- 2) 找出所有姓胡的教师的姓名和院系。

```
1 select name, dept_name
2 from teacher
3 where name like 'HU%';
```

x	name	dept_name
	HU1	D1
	HU2	CS

- 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的 ID 和姓名。

```
1 select student.id, student.name
2 from student
3 where student.id not in (select sc.student_id
4                           from sc
5                           where sc.year < 2018);
```

x	id	name
	3	S3
	6	S6
	8	S8
	11	S11
	12	S12
	14	S14
	15	S15
	18	S18
	20	S20

- 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。

```
1 select dept_name, max(salary)
2 from teacher
3 group by dept_name;
```

x	dept_name	max(salary)
	CS	9000
	D1	8000
	D2	6000
	D3	7000

5) 找出被所有学生选修过的课程。

原来自己插的数据并没有符合要求的课：

<pre>1 select * 2 from course 3 where not exists(select * 4                 from student 5                 where not exists(select * 6                                 from sc 7                                 where sc.student_id=student.id and sc.course_id=course.id));</pre>				
x	id	title	dept_name	credit
			<NULL>	<NULL>

更改数据后，让所有学生都选修数据库，以下是查询结果：

<pre>1 select * 2 from course 3 where not exists(select * 4                 from student 5                 where not exists(select * 6                                 from sc 7                                 where sc.student_id=student.id and sc.course_id=course.id));</pre>				
x	id	title	dept_name	credit
	1	database	CS	3

## 5. 修改数据

- 1) 将数据库课的学生成绩全部加 2 分。

修改前:

```
1 select sc.student_id, sc.grade
2 from sc, course
3 where sc.course_id=course.id and course.title='database';|
```

×	student_id	grade
	1	80
	2	92
	3	77
	4	93
	5	84
	6	89
	7	73
	8	67
	9	80
	10	97
	11	37
	12	73
	13	84
	14	59
	15	84
	16	82
	17	83
	18	83
	19	94
	20	93

修改后:

```
1 update sc
2 set grade=grade+2
3 where course_id in(select id
4                     from course
5                     where course.title='database');
6
7 select sc.student_id, sc.grade
8 from sc, course
9 where sc.course_id=course.id and course.title='database';
```

×	student_id	grade
	1	82
	2	94
	3	79
	4	95
	5	86
	6	91
	7	75
	8	69
	9	82
	10	99
	11	39
	12	75
	13	86
	14	61
	15	86
	16	84
	17	85
	18	85
	19	96
	20	95



## 6. 删除数据

1) 删除均分不足 80 分的所有学生的选课记录。

均分不足 80 的学生：

```
1 select student_id
2 from sc
3 group by student_id
4 having avg(grade)<80;
```

×	student_id
	2
	3
	6
	7
	11
	13
	14
	15
	16
	19
	20

删除：

```
1 delete from sc
2 where student_id in(select *
3                       from(select student_id
4                              from sc
5                              group by student_id
6                              having avg(grade)<80) as to_delete);
7
```

再看 sc 表，均分小于 80 的学生的所有选课记录已经被删除了：

exp1	student_id	course_id	year	grade
course	1	1	2017	82
主索引	1	2	2018	90
id	1	5	2019	87
title	4	1	2016	95
dept_name	4	5	2020	82
credit	5	1	2017	86
sc	5	3	2018	98
主索引	5	4	2019	83
course_id	8	1	2019	69
student_id	8	5	2019	98
course_id	9	1	2017	82
year	9	2	2018	96
grade	9	4	2018	78
student	10	1	2017	99
teacher	10	3	2017	83
主索引	12	1	2019	75
id	12	4	2018	97
name	17	1	2017	85
dept_name	17	2	2019	97
salary	17	4	2018	82
information_schema	18	1	2019	85
performance_schem	18	4	2019	82
进程				

## 7. 视图操作

- 1) 创建一个视图，记录每位学生已修课程的总学分数。

```
1 create view student_credit
2 as
3 select student.id, sum(course.credit)
4 from student
5 left join sc
6 on student.id=sc.student_id
7 left join course
8 on sc.course_id=course.id
9 group by student.id;
```

```
1 select * from student_credit;
```

x	id	sum(course.credit)
	1	9
	2	<NULL>
	3	<NULL>
	4	7
	5	8
	6	<NULL>
	7	<NULL>
	8	7
	9	7
	10	6
	11	<NULL>
	12	5
	13	<NULL>
	14	<NULL>
	15	<NULL>
	16	<NULL>
	17	7
	18	5
	19	<NULL>
	20	<NULL>

## 8. 删除基本表

- 1) 删除 SC 表。

```
1 drop table sc;
```

## 实验中遇到的困难及解决办法

编数据真的痛苦...

## 参考文献及致谢