

# 《数据库概论》实验一：用 SQL 进行数据 操作 实验报告

汤聪 151220098 [1195882464@qq.com](mailto:1195882464@qq.com)

## 实验环境：

Windows 10 cmd ;

MYSQL57 ;

## 实验过程：

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

#### 1.1 Student 表

字段	类型	非空	主键	说明
SNO	int	是	是	学号
SNAME	Char(8)	是		姓名
SEX	Char(2)			性别
DEPTNO	int			所在系

```
mysql> create table student(  
-> SNO INT NOT NULL PRIMARY KEY,  
-> SNAME CHAR(8) NOT NULL,  
-> SEX CHAR(2) NULL,  
-> DEPTNO INT NULL  
-> );  
Query OK, 0 rows affected (0.20 sec)
```

#### 1.2 Course 表

字段	类型	非空	主键	说明
CNO	int	是	是	课程号
CNAME	Char(20)			课程名
TNO	Int	是	是	授课教师编号
CREDIT	int			学分

```
mysql> CREATE TABLE COURSE(
-> CNO INT NOT NULL,
-> CNAME CHAR(20),
-> TNO INT NOT NULL,
-> CREDIT INT,
-> PRIMARY KEY(CNO,TNO)
-> );
Query OK, 0 rows affected (0.32 sec)
```

### 1.3 SC 表

字段	类型	非空	主键	说明
SNO	int	是	是	学号
CNO	Int	是	是	课程号
GRADE	Int			成绩

```
mysql> create table sc(
-> SNO INT NOT NULL,
-> CNO INT NOT NULL,
-> GRADE INT
-> ,
-> PRIMARY KEY(SNO,CNO)
-> );
Query OK, 0 rows affected (0.23 sec)
```

### 1.4 Teacher 表

字段	类型	非空	主键	说明
TNO	int	是	是	教师编号
TNAME	Char(8)	是		教师姓名
DEPTNO	Int			所在系

```
mysql> CREATE TABLE TEACHER(
-> TNO INT NOT NULL,
-> TNAME CHAR(8) NOT NULL,
-> DEPTNO INT,
-> PRIMARY KEY(TNO,TNAME)
-> );
Query OK, 0 rows affected (0.29 sec)
```

### 1.5 Dept 表

字段	类型	非空	主键	说明
DEPTNO	int	是	是	系号
DNAME	Char(20)	是		系名

```
mysql> CREATE TABLE Dept(  
-> DEPTNO INT NOT NULL PRIMARY KEY,  
-> DNAME CHAR(20) NOT NULL,  
-> );
```

## 2. 使用 SQL 语句修改基本表

2.1. 在 Student 表中加入属性 AGE (SMALLINT 型)。

```
mysql> ALTER TABLE STUDENT ADD AGE SMALLINT;  
Query OK, 0 rows affected (0.54 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

2.2. 将 Student 表中的属性 AGE 类型改为 INT 型。

```
mysql> ALTER TABLE student MODIFY AGE INT;  
Query OK, 7 rows affected (0.12 sec)  
Records: 7 Duplicates: 0 Warnings: 0
```

## 3. 使用 SQL 语句插入数据

3.1. 向 Student 表插入下列数据：

(1001, 喵喵, m, 10, 20)、(1002, 汪汪, f, 10, 21)、  
(1003, 咩咩, m, 10, 21)、(1004, 哞哞, f, 20, 21)、  
(1005, 呱呱, m, 20, 22)、(1006, 嘎嘎, f, 20, 22)  
(1007, 咕咕哒, f, 30, 20)。

```

mysql> INSERT STUDENT VALUES(1002,'汪汪','f',10,21);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT STUDENT VALUES(1003,'咩咩','m',10,21);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT STUDENT VALUES(1004,'哞哞','f',20,21);
Query OK, 1 row affected (0.11 sec)

mysql> INSERT STUDENT VALUES(1005,'呱呱','m',20,22);
Query OK, 1 row affected (0.05 sec)

mysql> INSERT STUDENT VALUES(1006,'嘎嘎','f',20,22);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT STUDENT VALUES(1007,'咕咕哒','f',30,20);
Query OK, 1 row affected (0.09 sec)

mysql> select * from student;
+-----+-----+-----+-----+-----+
| SNO   | SNAME | SEX   | DEPTNO | AGE   |
+-----+-----+-----+-----+-----+
| 1001  | 喵喵  | m     | 10     | 20    |
| 1002  | 汪汪  | f     | 10     | 21    |
| 1003  | 咩咩  | m     | 10     | 21    |
| 1004  | 哞哞  | f     | 20     | 21    |
| 1005  | 呱呱  | m     | 20     | 22    |
| 1006  | 嘎嘎  | f     | 20     | 22    |
| 1007  | 咕咕哒| f     | 30     | 20    |
+-----+-----+-----+-----+-----+

```

3.2. 向 Course 表插入下列数据：

- (1, 数据结构, 101, 4)、(2, 数据库, 102, 4)、
- (3, 离散数学, 103, 4)、(4, C 语言程序设计, 101, 2)、
- (5, 高等量子力学, 105, 3)。

```

mysql> INSERT Course VALUES(1,'数据结构',101,4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT Course VALUES(2,'数据库',102,4);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT Course VALUES(2,'离散数学',103,4);
ERROR 1062 (23000): Duplicate entry '2' for key 'PRIMARY'
mysql> INSERT Course VALUES(3,'离散数学',103,4);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT Course VALUES(4,'C语言程序设计',101,2);
ERROR 1062 (23000): Duplicate entry '101' for key 'TNO'
mysql> INSERT Course VALUES(5,'高等量子力学',105,3);

```

```
mysql> INSERT Course VALUES(4,'C语言程序设计',101,2);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> SELECT * FROM Course;
```

CNO	CNAME	TNO	CREDIT
1	数据结构	101	4
2	数据库	102	4
3	离散数学	103	4
4	C语言程序设计	101	2
5	高等量子力学	105	3

5 rows in set (0.00 sec)

3.3. 向 SC 表插入下列数据：

(1001, 1, 80)、(1001, 2, 85)、(1001, 3, 78)、(1002, 1, 72)、  
(1002, 2, 82)、(1002, 3, 86)、(1003, 1, 92)、(1003, 3, 90)、  
(1004, 1, 87)、(1004, 4, 90)、(1005, 1, 85)、(1005, 4, 92)、  
(1006, 5, 99)、(1006, 2, 100)、(1007, 1, 80)、(1007, 3, 91)。

```

mysql> INSERT SC VALUES(1001,1,80);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT SC VALUES(1001,2,85);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT SC VALUES(1001,3,78);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT SC VALUES(1002,1,72);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT SC VALUES(1002,2,82);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT SC VALUES(1002,3,86);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT SC VALUES(1003,1,92);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT SC VALUES(1003,3,90);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT SC VALUES(1004,1,87);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT SC VALUES(1004,4,90);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT SC VALUES(1005,1,85);
Query OK, 1 row affected (0.11 sec)

mysql> INSERT SC VALUES(1005,4,92);
Query OK, 1 row affected (0.11 sec)

mysql> INSERT SC VALUES(1006,5,99);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT SC VALUES(1006,2,100);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT SC VALUES(1007,1,80);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT SC VALUES(1007,3,91);
Query OK, 1 row affected (0.08 sec)

```

```
mysql> SELECT * FROM SC;
```

SNO	CNO	GRADE
1001	1	80
1001	2	85
1001	3	78
1002	1	72
1002	2	82
1002	3	86
1003	1	92
1003	3	90
1004	1	87
1004	4	90
1005	1	85
1005	4	92

1005	4	92	
1006	5	99	
1006	2	100	
1007	1	80	
1007	3	91	
-----+-----+-----+			
16 rows in set (0.00 sec)			

3.4. 向 Teacher 表插入下列数据：

(101, 张小天, 10)、(102, 胡小伟, 10)、  
(103, 黄程, 10)、(104, 郭冰, 20)、(105, 钱祺, 30)。

```
mysql> INSERT Teacher VALUES(101,'张小天',10);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT Teacher VALUES(102,'胡小伟',10);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT Teacher VALUES(103,'黄程',10);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT Teacher VALUES(104,'郭冰',20);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT Teacher VALUES(105,'钱祺',30);
Query OK, 1 row affected (0.01 sec)

mysql> selecct * from Teacher;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual
s to your MySQL server version for the right syntax to use near 'selecct'
t line 1
mysql> SELECT * FROM Teacher;
+-----+-----+-----+
| TNO | TNAME | DEPTNO |
+-----+-----+-----+
| 101 | 张小天 | 10 |
| 102 | 胡小伟 | 10 |
| 103 | 黄程 | 10 |
| 104 | 郭冰 | 20 |
| 105 | 钱祺 | 30 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

3.5. 向 Dept 表插入下列数据：

(10, 计算机)、(20, 信管)、(30, 物理)。

```

s to your MySQL server version for the right synt
mysql> INSERT Dept VALUES(10,'计算机');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT Dept VALUES(20,'信管');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT Dept VALUES(30,'物理');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Dept;
+-----+-----+
| DEPTNO | DNAME |
+-----+-----+
|      10 | 计算机 |
|      20 | 信管 |
|      30 | 物理 |
+-----+-----+
3 rows in set (0.00 sec)

```

#### 4. 单表查询

4.1. 查询所有女生的学生信息。

```

mysql> SELECT * FROM student WHERE SEX='f';
+-----+-----+-----+-----+-----+
| SNO | SNAME | SEX | DEPTNO | AGE |
+-----+-----+-----+-----+-----+
| 1002 | 汪汪 | f | 10 | 21 |
| 1004 | 哞哞 | f | 20 | 21 |
| 1006 | 嘎嘎 | f | 20 | 22 |
| 1007 | 咕咕哒 | f | 30 | 20 |
+-----+-----+-----+-----+-----+
4 rows in set (0.08 sec)

```

4.2. 查询成绩在 80 到 89 之间的所有学生的选课记录，查询结果按成绩的升序排列  
[将查询结果截图保留在实验报告中]



```
mysql> SELECT * FROM SC WHERE GRADE BETWEEN 80 AND 89 ORDER BY GRADE ASC;
+-----+-----+-----+
| SNO   | CNO   | GRADE |
+-----+-----+-----+
| 1001  | 1     | 80    |
| 1007  | 1     | 80    |
| 1002  | 2     | 82    |
| 1005  | 1     | 85    |
| 1001  | 2     | 85    |
| 1002  | 3     | 86    |
| 1004  | 1     | 87    |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

4.3. 查询各个系的学生人数。

[将查询结果截图保留在实验报告中]

```
mysql> SELECT DEPTNO,COUNT(SNO) FROM STUDENT GROUP BY DEPTNO;
+-----+-----+
| DEPTNO | COUNT(SNO) |
+-----+-----+
| 10     | 3          |
| 20     | 3          |
| 30     | 1          |
+-----+-----+
3 rows in set (0.11 sec)
```

## 5. 连接查询

5.1. 查询信管系年龄在 21 岁以下（含 21 岁）的女生姓名及其年龄。

```
mysql> SELECT STUDENT.SNAME,STUDENT.AGE FROM STUDENT,DEPT WHERE STUDENT.DEPTNO=DEPT.DEPTNO
AND STUDENT.SEX='f' AND STUDENT.AGE<=21 AND DEPT.DNAME='信管';
+-----+-----+
| SNAME | AGE |
+-----+-----+
| 晖晖  | 21  |
+-----+-----+
1 row in set (0.14 sec)
```

## 6. 嵌套查询

6.1. 查询在计算机方向总学分不足 5 分的学生姓名。

（计算机方向总学分 指 该学生选修的计算机系老师授课的课程总学分）

[将查询结果截图保留在实验报告中]

```
mysql> select sname FROM STUDENT WHERE SNO IN
-> (SELECT SNO FROM
-> (SELECT CNO,CREDIT FROM COURSE WHERE COURSE.TNO IN
-> (SELECT TNO FROM TEACHER WHERE DEPTNO IN
-> (SELECT DEPTNO FROM DEPT WHERE DNAME='计算机'))
-> )T,SC
-> WHERE SC.CNO=T.CNO
-> GROUP BY SNO
-> HAVING SUM(T.CREDIT)<5);
+-----+
| sname |
+-----+
| 嘎嘎 |
+-----+
1 row in set (0.02 sec)
```

6.2. 查询各门课程取得最高成绩的学生姓名及其成绩。

[将查询结果截图保留在实验报告中]

```
mysql> select student.sname,newsc.maxgrade from
-> (select cno,max(grade) as maxgrade from sc group by cno)newsc,
-> sc,student
-> where sc.cno=newsc.cno and sc.grade=newsc.maxgrade and sc.sno=student.sno;
+-----+-----+
| sname | maxgrade |
+-----+-----+
| 咩咩 | 92 |
| 嘎嘎 | 99 |
| 呱呱 | 92 |
| 嘎嘎 | 100 |
| 咕咕哒 | 91 |
+-----+-----+
5 rows in set (0.07 sec)
```

6.3. 查询选修了 1007 学生选课的全部课程的学生学号。

[将查询结果截图保留在实验报告中]

```
mysql> SELECT DISTINCT SNO FROM SC WHERE NOT EXISTS
-> (SELECT CNO FROM SC A WHERE A.SNO=1007 AND CNO NOT IN (SELECT CNO FROM SC B WHERE B.SNO=SC.SNO));
+-----+
| SNO |
+-----+
| 1001 |
| 1002 |
| 1003 |
| 1007 |
+-----+
4 rows in set (0.13 sec)
```

6.4. 查询没有选修任何 1006 学生选修的课程 的学生姓名。

[将查询结果截图保留在实验报告中]

```
mysql> select sname from student where sno in
-> (select distinct sno from sc where not exists
-> (select cno from sc a where a.sno=1006 and cno in (select cno from sc b where b.sno
=sno)));
+-----+
| sname |
+-----+
| 咩咩 |
| 哞哞 |
| 呱呱 |
| 咕咕哒 |
+-----+
4 rows in set (0.00 sec)

mysql>
```

## 7. 修改数据

7.1. 将数据结构课的学生成绩全部加 2 分。

[将修改后的 sc 表 (select \* from sc;) 截图保留在实验报告中]

```
mysql> UPDATE SC SET grade=grade+2 WHERE CNO=(SELECT CNO FROM COURSE WHERE CNAME='数据结构');
Query OK, 6 rows affected (0.00 sec)
Rows matched: 6 Changed: 6 Warnings: 0

mysql> select * from sc;
+-----+-----+-----+
| SNO | CNO | GRADE |
+-----+-----+-----+
| 1001 | 1 | 82 |
| 1002 | 1 | 74 |
| 1003 | 1 | 94 |
| 1004 | 1 | 89 |
| 1005 | 1 | 87 |
| 1006 | 5 | 99 |
| 1007 | 1 | 82 |
| 1001 | 2 | 85 |
| 1001 | 3 | 78 |
| 1002 | 2 | 82 |
| 1002 | 3 | 86 |
| 1003 | 3 | 90 |
| 1004 | 4 | 90 |
| 1005 | 4 | 92 |
| 1006 | 2 | 100 |
| 1007 | 3 | 91 |
+-----+-----+-----+
16 rows in set (0.00 sec)
```

## 8. 删除数据

8.1. 删除成绩不足 80 分的所有女生的选课记录。

(和 7.1 中学生成绩加 2 分相对独立，互不影响)

```
mysql> delete from sc where grade<=80 and 'f'=(select sex from student where student.sno=
sc.sno);
Query OK, 2 rows affected (0.20 sec)

mysql> select * from sc;
+-----+-----+-----+
| SNO | CNO | GRADE |
+-----+-----+-----+
| 1001 | 1 | 80 |
| 1003 | 1 | 92 |
| 1004 | 1 | 87 |
| 1005 | 1 | 85 |
| 1006 | 5 | 99 |
| 1001 | 2 | 85 |
| 1001 | 3 | 78 |
| 1002 | 2 | 82 |
| 1002 | 3 | 86 |
| 1003 | 3 | 90 |
| 1004 | 4 | 90 |
| 1005 | 4 | 92 |
| 1006 | 2 | 100 |
| 1007 | 3 | 91 |
+-----+-----+-----+
14 rows in set (0.00 sec)
```

## 9. 视图操作

9.1. 在 Student 表上为计算机系的学生记录建立一个视图 CS\_STUDENT。

[将 CS\_STUDENT 的全部内容截图保留在实验报告中]

```
mysql> create view cs_s as select * from student where deptno in
-> (select deptno from dept where dname='计算机');
Query OK, 0 rows affected (0.14 sec)

mysql> select * from cs_s;
+-----+-----+-----+-----+-----+
| SNO | SNAME | SEX | DEPTNO | AGE |
+-----+-----+-----+-----+-----+
| 1001 | 喵喵 | m | 10 | 20 |
| 1002 | 汪汪 | f | 10 | 21 |
| 1003 | 咩咩 | m | 10 | 21 |
+-----+-----+-----+-----+-----+
3 rows in set (0.15 sec)
```

9.2. 删除视图 CS\_STUDENT。

```
mysql> drop view cs_s;
Query OK, 0 rows affected (0.00 sec)
```

## 10. 删除基本表

10.1. 删除 Student 表。

10.2. 删除 Course 表。

- 10.3. 删除 SC 表。
- 10.4. 删除 Teacher 表。
- 10.5. 删除 Dept 表。

```
mysql> drop table student;
Query OK, 0 rows affected (0.28 sec)

mysql> select * from student;
ERROR 1146 (42S02): Table 't1.student' doesn't exist
mysql> select * from cs_s;
ERROR 1146 (42S02): Table 't1.cs_s' doesn't exist
mysql> drop table course;
Query OK, 0 rows affected (0.15 sec)

mysql> drop table sc;
Query OK, 0 rows affected (0.12 sec)

mysql> drop table teacher;
Query OK, 0 rows affected (0.06 sec)

mysql> drop table dept;
Query OK, 0 rows affected (0.03 sec)

mysql> select * from course;
ERROR 1146 (42S02): Table 't1.course' doesn't exist
mysql> select * from sc;
ERROR 1146 (42S02): Table 't1.sc' doesn't exist
mysql> select * from teacher;
ERROR 1146 (42S02): Table 't1.teacher' doesn't exist
mysql> select * from dept;
ERROR 1146 (42S02): Table 't1.dept' doesn't exist
```

## 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. 嵌套查询部分对我来说比较困难，主要是不知道能够建一个新的表来存放查询的数据，并且用 IN 时后面一个嵌套语句只能 select 一个参数，就会觉得很烦很不方便，然后就是一步一步往外推看哪部出错，然后查一些资料，询问参考一下，最后得到解决，其他部分还是比较顺利。
2. 我觉得 primary key 还是一个表运用一个比较好。

## 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]

声明：因为之前学过一些 mysql，然后实在 cmd 中进行操作，后来装课上提供的那个软件总是装不上，所以选择在 cmd 中操作，不会影响实验过程。

参考资料：

课件，书，

<http://www.runoob.com/>

[www.imooc.com/](http://www.imooc.com/)