

数据库实验一 用SQL进行数据操作

| 姓名 | 学号 | 邮箱 | 编辑时间 |
|-----|-----------|----------------------------|-----------|
| 刘国涛 | 181860055 | 181860055@smail.nju.edu.cn | 2020.11.2 |

实验环境

| | |
|-------|------------------|
| 操作系统 | Windows 10 20H2 |
| 数据库版本 | mysql Ver 8.0.21 |
| 代码编辑器 | vscode |

文件夹说明

| | |
|---|------------------------------------|
| 1 | └─ex1 |
| 2 | ex1.sql //本次实验的SQL代码 |
| 3 | exp1-manual.pdf //实验1的指导手册 |
| 4 | generateData.py //数据生成脚本 |
| 5 | README.md //实验报告的markdown文件 |
| 6 | exp1.pdf //实验报告 |
| 7 | updateSQL.bat //运行SQL代码的脚本 |

实验过程

使用SQL语句建立基本表

| | |
|----|--|
| 1 | CREATE TABLE IF NOT EXISTS `Course` (|
| 2 | `id` INT NOT NULL PRIMARY KEY COMMENT '课程号', |
| 3 | `title` CHAR(20) NOT NULL COMMENT '课程名', |
| 4 | `dept_name` CHAR(20) DEFAULT NULL COMMENT '院系名', |
| 5 | `credit` INT COMMENT '学分' |
| 6 |); |
| 7 | CREATE TABLE IF NOT EXISTS Student (|
| 8 | id INT NOT NULL PRIMARY KEY COMMENT '学号', |
| 9 | name CHAR(20) NOT NULL COMMENT '学生姓名', |
| 10 | dept_name CHAR(20) COMMENT '院系名', |
| 11 | major_name CHAR(20) COMMENT '专业名' |
| 12 |); |
| 13 | CREATE TABLE IF NOT EXISTS SC (|
| 14 | student_id INT NOT NULL COMMENT '学号', |
| 15 | course_id INT NOT NULL COMMENT '课程号', |
| 16 | year INT COMMENT '选课年份', |
| 17 | grade INT COMMENT '成绩', |

```

18     PRIMARY KEY(student_id,course_id)
19 );
20 CREATE TABLE IF NOT EXISTS Teacher (
21     id INT NOT NULL PRIMARY KEY COMMENT '教师编号',
22     name CHAR(20) NOT NULL COMMENT '教师姓名',
23     dept_name CHAR(20) COMMENT '院系名',
24     salary INT COMMENT '工资'
25 );

```

```

mysql> use ex1
Database changed
mysql> # 1 使用SQL语句建立基本表
mysql> CREATE TABLE IF NOT EXISTS `Course` (
->   `id` INT NOT NULL PRIMARY KEY COMMENT '课程号',
->   `title` CHAR(20) NOT NULL COMMENT '课程名',
->   `dept_name` CHAR(20) DEFAULT NULL COMMENT '院系名',
->   `credit` INT COMMENT '学分'
-> );
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE TABLE IF NOT EXISTS Student (
->   id INT NOT NULL PRIMARY KEY COMMENT '学号',
->   name CHAR(20) NOT NULL COMMENT '学生姓名',
->   dept_name CHAR(20) COMMENT '院系名',
->   major_name CHAR(20) COMMENT '专业名'
-> );
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE IF NOT EXISTS SC (
->   student_id INT NOT NULL COMMENT '学号',
->   course_id INT NOT NULL COMMENT '课程号',
->   year INT COMMENT '选课年份',
->   grade INT COMMENT '成绩',
->   PRIMARY KEY(student_id,course_id)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE IF NOT EXISTS Teacher (
->   id INT NOT NULL PRIMARY KEY COMMENT '教师编号',
->   name CHAR(20) NOT NULL COMMENT '教师姓名',
->   dept_name CHAR(20) COMMENT '院系名',
->   salary INT COMMENT '工资'
-> );
Query OK, 0 rows affected (0.04 sec)

```

使用SQL语句修改基本表

```

1 ALTER TABLE Student
2 ADD COLUMN age SMALLINT;
3 ALTER TABLE Student
4 CHANGE COLUMN age age INT;

```

```

mysql> # 2 使用SQL语句修改基本表
mysql> ALTER TABLE Student
-> ADD COLUMN age SMALLINT;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE Student
-> CHANGE COLUMN age age INT;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

使用SQL语句插入数据

首先编写数据生成的Python [脚本](#)，详见同目录下的 [generateData.py](#)

生成数据后通过 [INSERT](#) 语句插入数据

插入学生数据

```

1 INSERT INTO Student
2 VALUES
3 (0,'米卡泰','社科','社科',20),
4 (1,'李电勇','人工智能','人工智能',20),
5 (2,'路世','社科','社科',20),
6 (3,'武琰','数理','数学',20),

```

```

7  (4, '晏楠', 'CS', '软件工程', 20),
8  (5, '滕枝', '人工智能', '人工智能', 20),
9  (6, '终力', '商院', '经济学', 20),
10 (7, '何忠雅', '工程管理', '工业工程', 20),
11 (8, '弘姬', '数理', '数学', 20),
12 (9, '衡友莉', 'CS', '计算机应用', 20),
13 (10, '姜加枝', '商院', '管理学', 20),
14 (11, '衡友群', '人工智能', '人工智能', 20),
15 (12, '芮宏', '商院', '保险学', 20),
16 (13, '督子智仪', '社科', '社科', 20),
17 (14, '劳婉', '工程管理', '工业工程', 20),
18 (15, '隗都瑞', '数理', '数学', 20),
19 (16, '庄宁', '商院', '保险学', 20),
20 (17, '凤莎', '商院', '管理学', 20),
21 (18, '孙无融', 'CS', '软件工程', 20),
22 (19, '焦妍', '工程管理', '工业工程', 20),
23 (20, '米莎', '社科', '社科', 20),
24 (21, '屈孝霄', 'CS', '软件工程', 20),
25 (22, '尚裕', '社科', '社科', 20),
26 (23, '干忠辉', '商院', '经济学', 20),
27 (24, '干桂', '数理', '物理', 20),
28 (25, '庞伯', 'CS', '软件工程', 20),
29 (26, '姚被昌', '工程管理', '金融工程', 20),
30 (27, '刁嫻', '数理', '天文', 20),
31 (28, '都电宁', '商院', '管理学', 20),
32 (29, '蒋明', '数理', '物理', 20);

```

```

mysql> # 3 使用SQL语句插入数据
mysql> INSERT INTO Student
-> VALUES
-> (0, '米卡泰', '社科', '社科', 20),
-> (1, '李电勇', '人工智能', '人工智能', 20),
-> (2, '路世', '社科', '社科', 20),
-> (3, '武琰', '数理', '数学', 20),
-> (4, '晏楠', 'CS', '软件工程', 20),
-> (5, '滕枝', '人工智能', '人工智能', 20),
-> (6, '终力', '商院', '经济学', 20),
-> (7, '何忠雅', '工程管理', '工业工程', 20),
-> (8, '弘姬', '数理', '数学', 20),
-> (9, '衡友莉', 'CS', '计算机应用', 20),
-> (10, '姜加枝', '商院', '管理学', 20),
-> (11, '衡友群', '人工智能', '人工智能', 20),
-> (12, '芮宏', '商院', '保险学', 20),
-> (13, '督子智仪', '社科', '社科', 20),
-> (14, '劳婉', '工程管理', '工业工程', 20),
-> (15, '隗都瑞', '数理', '数学', 20),
-> (16, '庄宁', '商院', '保险学', 20),
-> (17, '凤莎', '商院', '管理学', 20),
-> (18, '孙无融', 'CS', '软件工程', 20),
-> (19, '焦妍', '工程管理', '工业工程', 20),
-> (20, '米莎', '社科', '社科', 20),
-> (21, '屈孝霄', 'CS', '软件工程', 20),
-> (22, '尚裕', '社科', '社科', 20),
-> (23, '干忠辉', '商院', '经济学', 20),
-> (24, '干桂', '数理', '物理', 20),
-> (25, '庞伯', 'CS', '软件工程', 20),
-> (26, '姚被昌', '工程管理', '金融工程', 20),
-> (27, '刁嫻', '数理', '天文', 20),
-> (28, '都电宁', '商院', '管理学', 20),
-> (29, '蒋明', '数理', '物理', 20);
Query OK, 30 rows affected (0.01 sec)
Records: 30  Duplicates: 0  Warnings: 0

```

插入课程数据

```

1  INSERT INTO Course
2  VALUES
3  (0, '数据库', '工程管理', 4),
4  (1, '数据库', 'CS', 2),
5  (2, '数据库', '人工智能', 2),
6  (3, '操作系统', '工程管理', 3),
7  (4, '操作系统', 'CS', 4),

```

```

8      (5, '操作系统', '人工智能', 3),
9      (6, '数字电路', '工程管理', 4),
10     (7, '数字电路', 'CS', 2),
11     (8, '数字电路', '人工智能', 3),
12     (9, '线代', '工程管理', 5),
13     (10, '线代', 'CS', 5),
14     (11, '线代', '人工智能', 2),
15     (12, '微积分', '工程管理', 1),
16     (13, '微积分', 'CS', 5),
17     (14, '微积分', '人工智能', 4);

```

```

mysql>
mysql> INSERT INTO Course
-> VALUES
-> (0, '数据库', '工程管理', 4),
-> (1, '数据库', 'CS', 2),
-> (2, '数据库', '人工智能', 2),
-> (3, '操作系统', '工程管理', 3),
-> (4, '操作系统', 'CS', 4),
-> (5, '操作系统', '人工智能', 3),
-> (6, '数字电路', '工程管理', 4),
-> (7, '数字电路', 'CS', 2),
-> (8, '数字电路', '人工智能', 3),
-> (9, '线代', '工程管理', 5),
-> (10, '线代', 'CS', 5),
-> (11, '线代', '人工智能', 2),
-> (12, '微积分', '工程管理', 1),
-> (13, '微积分', 'CS', 5),
-> (14, '微积分', '人工智能', 4);
Query OK, 15 rows affected (0.02 sec)
Records: 15  Duplicates: 0  Warnings: 0

```

插入选课数据

```

1      INSERT INTO SC
2      VALUES
3      (0, 6, 2020, 87),
4      (0, 7, 2019, 94),
5      (0, 8, 2018, 90),
6      (1, 7, 2020, 91),
7      (1, 10, 2020, 81),
8      (1, 14, 2018, 90),
9      (2, 8, 2020, 99),
10     (2, 10, 2018, 92),
11     (3, 2, 2019, 79),
12     (3, 6, 2017, 80),
13     (3, 7, 2018, 99),
14     (3, 9, 2017, 74),
15     (4, 9, 2018, 79),
16     (5, 8, 2019, 80),
17     (6, 4, 2017, 82),
18     (6, 6, 2019, 71),
19     (6, 8, 2018, 83),
20     (6, 9, 2017, 84),
21     (7, 10, 2018, 95),
22     (7, 12, 2018, 94),
23     (7, 13, 2017, 90),
24     (8, 10, 2017, 90),
25     (9, 1, 2020, 89),
26     (9, 9, 2018, 75),
27     (10, 4, 2020, 93),
28     (10, 6, 2017, 95),
29     (11, 13, 2019, 86),
30     (12, 0, 2019, 76),
31     (12, 3, 2018, 73),

```

32 (13,1,2018,88),
33 (13,2,2019,86),
34 (13,3,2019,80),
35 (13,11,2017,72),
36 (14,2,2019,98),
37 (14,4,2018,73),
38 (14,8,2019,88),
39 (14,11,2020,79),
40 (16,9,2017,73),
41 (16,11,2019,90),
42 (17,3,2018,84),
43 (17,6,2018,96),
44 (17,11,2019,91),
45 (18,1,2019,82),
46 (18,5,2019,70),
47 (19,0,2017,86),
48 (20,0,2018,100),
49 (20,9,2018,81),
50 (21,1,2017,70),
51 (22,1,2019,91),
52 (22,2,2017,96),
53 (22,9,2019,100),
54 (24,1,2018,98),
55 (24,5,2020,97),
56 (24,11,2019,74),
57 (25,6,2018,100),
58 (27,8,2020,87),
59 (27,11,2019,93),
60 (28,4,2017,87),
61 (28,5,2019,98),
62 (29,2,2018,96),
63 (29,14,2018,70);

```

-> (12,'微积分','工程管理',1),
-> (13,'微积分','CS',5),
-> (14,'微积分','人工智能',4);
Query OK, 15 rows affected (0.02 sec)
Records: 15 Duplicates: 0 Warnings: 0

```

```

mysql>
mysql> INSERT INTO SC
-> VALUES
-> (0,6,2020,87),
-> (0,7,2019,94),
-> (0,8,2018,90),
-> (1,7,2020,91),
-> (1,10,2020,81),
-> (1,14,2018,90),
-> (2,8,2020,99),
-> (2,10,2018,92),
-> (3,2,2019,79),
-> (3,6,2017,80),
-> (3,7,2018,99),
-> (3,9,2017,74),
-> (4,9,2018,79),
-> (5,8,2019,80),
-> (6,4,2017,82),
-> (6,6,2019,71),
-> (6,8,2018,83),
-> (6,9,2017,84),
-> (7,10,2018,95),
-> (7,12,2018,94),
-> (7,13,2017,90),
-> (8,10,2017,90),
-> (9,1,2020,89),
-> (9,9,2018,75),
-> (10,4,2020,93),
-> (10,6,2017,95),
-> (11,13,2019,86),
-> (12,0,2019,76),
-> (12,3,2018,73),
-> (13,1,2018,88),
-> (13,2,2019,86),
-> (13,3,2019,80),
-> (13,11,2017,72),
-> (14,2,2019,98),
-> (14,4,2018,73),
-> (14,8,2019,88),
-> (14,11,2020,79),
-> (16,9,2017,73),
-> (16,11,2019,90),
-> (17,3,2018,84),
-> (17,6,2018,96),
-> (17,11,2019,91),
-> (18,1,2019,82),
-> (18,5,2019,70),
-> (19,0,2017,86),
-> (20,0,2018,100),
-> (20,9,2018,81),
-> (21,1,2017,70),
-> (22,1,2019,91),
-> (22,2,2017,96),
-> (22,9,2019,100),
-> (24,1,2018,98),
-> (24,5,2020,97),
-> (24,11,2019,74),
-> (25,6,2018,100),
-> (27,8,2020,87),
-> (27,11,2019,93),
-> (28,4,2017,87),
-> (28,5,2019,98),
-> (29,2,2018,96),
-> (29,14,2018,70);
Query OK, 61 rows affected (0.01 sec)
Records: 61 Duplicates: 0 Warnings: 0

```

插入老师数据

```

1  INSERT INTO Teacher
2  VALUES
3  (0,'胡伟','CS',261522),
4  (1,'胡伟大','人工智能',849588),
5  (2,'王胜利','CS',675038),
6  (3,'李其芳','人工智能',894765),
7  (4,'王曦','工程管理',146217),
8  (5,'张帅','工程管理',525091);

```

```
mysql> INSERT INTO Teacher
-> VALUES
-> (0,'胡伟','CS',261522),
-> (1,'胡伟大','人工智能',849588),
-> (2,'王胜利','CS',675038),
-> (3,'李其芳','人工智能',894765),
-> (4,'王曦','工程管理',146217),
-> (5,'张帅','工程管理',525091);
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

查询

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

```
1 SELECT DISTINCT Student.name
2 FROM Student,SC,Course
3 WHERE Student.id = SC.student_id AND SC.course_id = Course.id AND Course.dept_name
   = 'CS';
```

mysql> # 4 查询

```
mysql> SELECT DISTINCT Student.name
-> FROM Student,SC,Course
-> WHERE Student.id = SC.student_id AND SC.course_id = Course.id AND Course.dept_name = 'CS';
```

| name |
|------|
| 米卡泰 |
| 李电勇 |
| 路世 |
| 武琰 |
| 终力 |
| 何忠雅 |
| 弘姬 |
| 衡友莉 |
| 姜加枝 |
| 衡友群 |
| 管子智仪 |
| 劳婉 |
| 孙无融 |
| 屈孝霄 |
| 尚裕 |
| 干桂 |
| 都电宁 |

17 rows in set (0.00 sec)

4.2 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 Teacher.name LIKE '胡%';
```

```
mysql> SELECT name,dept_name
-> FROM Teacher
-> WHERE Teacher.name LIKE '胡%';
```

| name | dept_name |
|------|-----------|
| 胡伟 | CS |
| 胡伟大 | 人工智能 |

2 rows in set (0.00 sec)

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的ID和姓名

```

1  SELECT DISTINCT id,name
2  FROM Student
3  WHERE id NOT IN (
4      SELECT student_id FROM SC
5      WHERE year >= 2018
6  );

```

```

mysql> SELECT DISTINCT id,name
-> FROM Student
-> WHERE id NOT IN (
->     SELECT student_id FROM SC
->     WHERE year >= 2018
-> );

```

| id | name |
|----|------|
| 8 | 弘姬 |
| 15 | 陶都瑞 |
| 19 | 熊妍 |
| 21 | 屈孝霄 |
| 23 | 于忠辉 |
| 26 | 姚被昌 |

6 rows in set (0.00 sec)

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

```

1  SELECT dept_name,max(salary) FROM Teacher
2  GROUP BY dept_name;

```

```

mysql>
mysql> SELECT dept_name,max(salary) FROM Teacher
-> GROUP BY dept_name;

```

| dept_name | max(salary) |
|-----------|-------------|
| CS | 675038 |
| 人工智能 | 894765 |
| 工程管理 | 525091 |

3 rows in set (0.00 sec)

4.5找出被所有学生选修过的课程

```

1  SELECT * FROM Course
2  WHERE id in (
3      SELECT course_id FROM SC
4      GROUP BY course_id
5      HAVING COUNT(*) = (SELECT COUNT(*) FROM Course)
6  );

```

```

mysql> SELECT * FROM Course
-> WHERE id in (
->     SELECT course_id FROM SC
->     GROUP BY course_id
->     HAVING COUNT(*) = (SELECT COUNT(*) FROM Course)
-> );
Empty set (0.00 sec)

```


修改数据

```
1  UPDATE SC
2  SET grade = grade + 2
3  WHERE course_id IN (
4      SELECT id FROM Course
5      WHERE title = "数据库"
6  );
```

修改前

```
mysql> # 5 修改数据
mysql>
mysql> # 输出修改前的数据
mysql> SELECT * FROM SC
-> WHERE course_id IN (
->   SELECT id FROM Course
->   WHERE title = "数据库"
-> );
```

| student_id | course_id | year | grade |
|------------|-----------|------|-------|
| 3 | 2 | 2019 | 79 |
| 9 | 1 | 2020 | 89 |
| 12 | 0 | 2019 | 76 |
| 13 | 1 | 2018 | 88 |
| 13 | 2 | 2019 | 86 |
| 14 | 2 | 2019 | 98 |
| 18 | 1 | 2019 | 82 |
| 19 | 0 | 2017 | 86 |
| 20 | 0 | 2018 | 100 |
| 21 | 1 | 2017 | 70 |
| 22 | 1 | 2019 | 91 |
| 22 | 2 | 2017 | 96 |
| 24 | 1 | 2018 | 98 |
| 29 | 2 | 2018 | 96 |

14 rows in set (0.00 sec)

修改

```
mysql> # 修改
mysql> UPDATE SC
-> SET grade = grade + 2
-> WHERE course_id IN (
->   SELECT id FROM Course
->   WHERE title = "数据库"
-> );
Query OK, 14 rows affected (0.01 sec)
Rows matched: 14  Changed: 14  Warnings: 0
```

修改后

```
mysql> # 输出修改后的数据
mysql> SELECT * FROM SC
-> WHERE course_id IN (
->   SELECT id FROM Course
->   WHERE title = "数据库"
-> );
```

| student_id | course_id | year | grade |
|------------|-----------|------|-------|
| 3 | 2 | 2019 | 81 |
| 9 | 1 | 2020 | 91 |
| 12 | 0 | 2019 | 78 |
| 13 | 1 | 2018 | 90 |
| 13 | 2 | 2019 | 88 |
| 14 | 2 | 2019 | 100 |
| 18 | 1 | 2019 | 84 |
| 19 | 0 | 2017 | 88 |
| 20 | 0 | 2018 | 102 |
| 21 | 1 | 2017 | 72 |
| 22 | 1 | 2019 | 93 |
| 22 | 2 | 2017 | 98 |
| 24 | 1 | 2018 | 100 |
| 29 | 2 | 2018 | 98 |

14 rows in set (0.00 sec)

删除数据

```
1 DELETE FROM SC
2 WHERE student_id IN (
3     SELECT tmp.student_id FROM (
4         SELECT student_id FROM SC
5         GROUP BY student_id
6         HAVING AVG(grade) < 80
7     ) AS tmp
8 );
```

删除前

```
mysql> # 6 删除数据
mysql>
mysql> # 输出删除前的数据
mysql> SELECT * FROM SC
      -> GROUP BY student_id
      -> HAVING AVG(grade) < 80;
+-----+-----+-----+-----+
| student_id | course_id | year | grade |
+-----+-----+-----+-----+
|         4 |         9 | 2018 |    79 |
|        12 |         0 | 2019 |    78 |
|        18 |         1 | 2019 |    84 |
|        21 |         1 | 2017 |    72 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

删除

```
mysql> # 删除
mysql> DELETE FROM SC
      -> WHERE student_id IN (
      ->     SELECT tmp.student_id FROM (
      ->         SELECT student_id FROM SC
      ->         GROUP BY student_id
      ->         HAVING AVG(grade) < 80
      ->     ) AS tmp
      -> );
Query OK, 6 rows affected (0.01 sec)
```

删除后

```
mysql> # 输出删除后的数据
mysql> SELECT * FROM SC
      -> GROUP BY student_id
      -> HAVING AVG(grade) < 80;
Empty set (0.00 sec)

mysql>
```

视图操作

```
1 # 创建视图
2 CREATE VIEW myview(id,name,course_id,credit)
3 AS SELECT SC.student_id,Student.name,SC.course_id,Course.credit
4 FROM SC,Course,Student
5 WHERE SC.course_id = Course.id AND Student.id = SC.student_id;
6 CREATE VIEW TotalCredit(id,name,total_credit)
7 AS SELECT id,name,SUM(credit)
8 FROM myview
9 GROUP BY id;
10 # 输出视图
11 SELECT * FROM TotalCredit;
```

```
mysql> # 7 视图操作
mysql> CREATE VIEW myview(id,name,course_id,credit)
-> AS SELECT SC.student_id,Student.name,SC.course_id,Course.credit
-> FROM SC,Course,Student
-> WHERE SC.course_id = Course.id AND Student.id = SC.student_id;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE VIEW TotalCredit(id,name,total_credit)
-> AS SELECT id,name,SUM(credit)
-> FROM myview
-> GROUP BY id;
Query OK, 0 rows affected (0.01 sec)

mysql> # 输出视图
mysql> SELECT * FROM TotalCredit;
```

| id | name | total_credit |
|----|------|--------------|
| 0 | 米卡泰 | 9 |
| 1 | 李电勇 | 11 |
| 2 | 路世 | 8 |
| 3 | 武琰 | 13 |
| 5 | 滕枝 | 3 |
| 6 | 终力 | 16 |
| 7 | 何忠雅 | 11 |
| 8 | 弘姬 | 5 |
| 9 | 衡友莉 | 7 |
| 10 | 姜加枝 | 8 |
| 11 | 衡友群 | 5 |
| 13 | 管子智仪 | 9 |
| 14 | 劳婉 | 11 |
| 16 | 庄宁 | 7 |
| 17 | 凤莎 | 9 |
| 19 | 焦妍 | 4 |
| 20 | 米莎 | 9 |
| 22 | 尚裕 | 9 |
| 24 | 干桂 | 7 |
| 25 | 庞伯 | 4 |
| 27 | 刁嫫 | 5 |
| 28 | 都电宁 | 7 |
| 29 | 蒋明 | 6 |

```
23 rows in set (0.00 sec)
```

删除基本表

题目只要求删除SC表，但是为了脚本运行的方便，这里全删完了

```
1 DROP TABLE SC;
2 DROP TABLE Course;
3 DROP TABLE Teacher;
4 DROP TABLE Student;
5 DROP VIEW myview;
6 DROP VIEW TotalCredit;
```

```
mysql> # 8 删除基本表
mysql> DROP TABLE SC;
Query OK, 0 rows affected (0.03 sec)

mysql> DROP TABLE Course;
Query OK, 0 rows affected (0.02 sec)

mysql> DROP TABLE Teacher;
Query OK, 0 rows affected (0.02 sec)

mysql> DROP TABLE Student;
Query OK, 0 rows affected (0.03 sec)

mysql> DROP VIEW myview;
Query OK, 0 rows affected (0.01 sec)

mysql> DROP VIEW TotalCredit;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> █
```

实验中遇到的麻烦及解决方法

1、本次实验要生成符合要求的学生数据、课程数据、选课数据和教室数据较为麻烦

我通过编写Python代码自动生成需要的数据，其中随机生成姓名的部分直接调用了 [网上的代码](#)

2、SQL语句在命令行中编写没有自动补全

这对代码编写体验来说十分糟糕，但是我发现vscode创建sql后缀的文件可以完美解决这一问题。并且通过vscode的mysql插件，还可以得到图形化的数据库界面，大大提升了完成实验的效率

致谢

- 1、[mysql的语句使用教程](#)
- 2、[python生成随机姓名](#)