数据库概论大作业

任务一：根据所给excel表（exp_data.xlsx）中的数据信息，在MySQL数据库中创建对应的关系表并将数据录入到数据库中。

1. 创建模式以及关系表

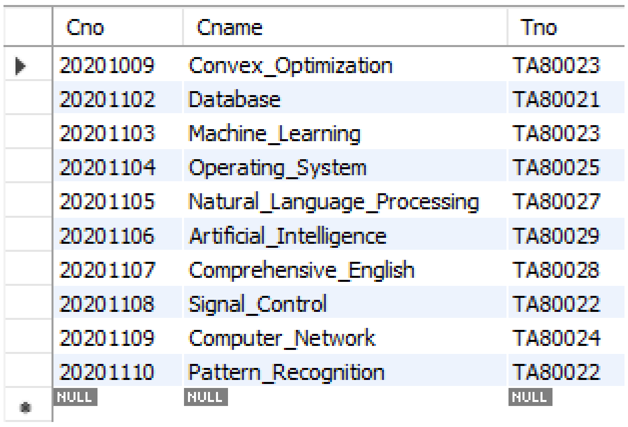
```sql
CREATE SCHEMA `2020exp1`;
Use 2020exp1;
CREATE TABLE STUDENT
(Sno CHAR(11) PRIMARY KEY, /*列级完整性约束条件*/
  Sname VARCHAR(5),
  Ssex CHAR(6),
  Sbirthday datetime,
  Sdepart SMALLINT
);
CREATE TABLE TEACHER
(Tno CHAR(7) PRIMARY KEY, /*列级完整性约束条件*/
  Tname VARCHAR(5),
  Tsex CHAR(6),
  Tbirthday datetime,
  Tprof VARCHAR(20),
  Tdepart SMALLINT
);
CREATE TABLE COURSE
(Cno CHAR(8) PRIMARY KEY, /*列级完整性约束条件*/
  Cname VARCHAR(35),
  Tno CHAR(7),
  FOREIGN KEY (Tno) REFERENCES TEACHER(Tno)
);
CREATE TABLE SCORE
(Sno CHAR(11),
  Cno CHAR(8),
  degree SMALLINT,
  PRIMARY KEY (Sno, Cno),/*表级完整性约束条件*/
  FOREIGN KEY (Sno) REFERENCES STUDENT(Sno),
  FOREIGN KEY (Cno) REFERENCES COURSE(Cno)
);
```

2. 将excel表另存为utf-8格式的csv数据后在服务端导入

```
CD C:\Program Files\MySQL\MySQL Server 8.0\bin
mysql --local-infile -u root -p
load data local INfile './teacher.csv' into table `2020exp1`.TEACHER fields terminated by ',' lines terminated by '\n';
load data local INfile './students.csv' into table `2020exp1`.STUDENT fields terminated by ',' lines terminated by '\n';
load data local INfile './course.csv' into table `2020exp1`.COURSE fields terminated by ',' lines terminated by '\n';
load data local INfile './score.csv' into table `2020exp1`.SCORE fields terminated by ',' lines terminated by '\n';
```

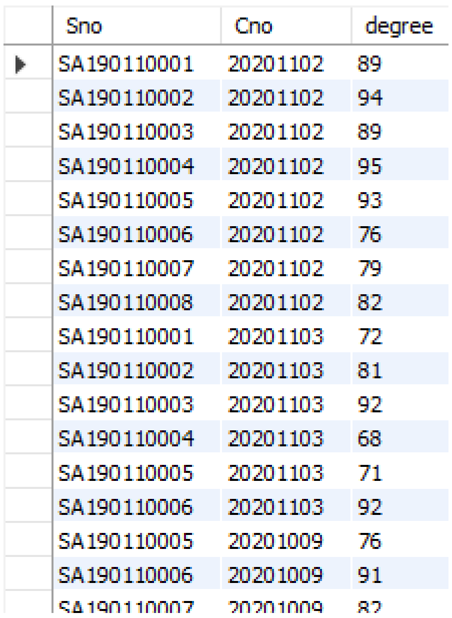
3. 导入后的截图

course表



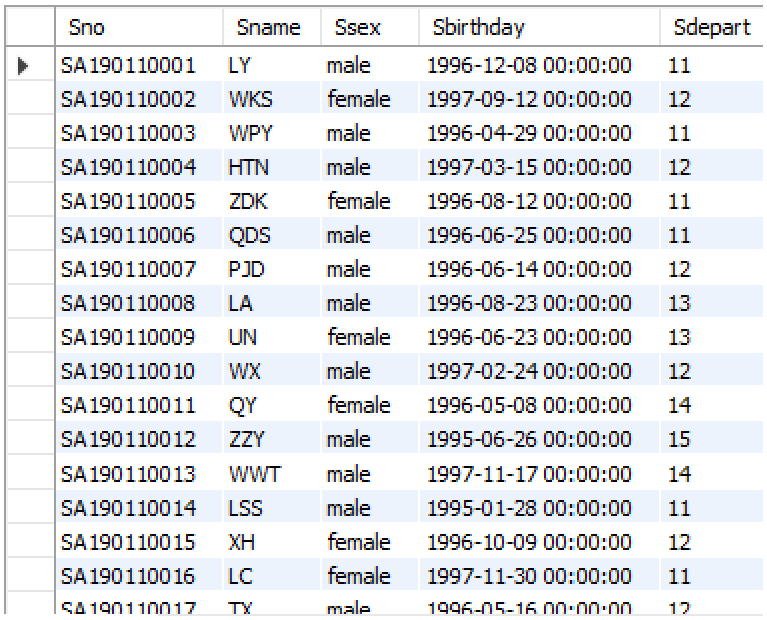
Cno	Cname	Tno
20201009	Convex_Optimization	TA80023
20201102	Database	TA80021
20201103	Machine_Learning	TA80023
20201104	Operating_System	TA80025
20201105	Natural_Language_Processing	TA80027
20201106	Artificial_Intelligence	TA80029
20201107	Comprehensive_English	TA80028
20201108	Signal_Control	TA80022
20201109	Computer_Network	TA80024
20201110	Pattern_Recognition	TA80022
NULL	NULL	NULL

score表



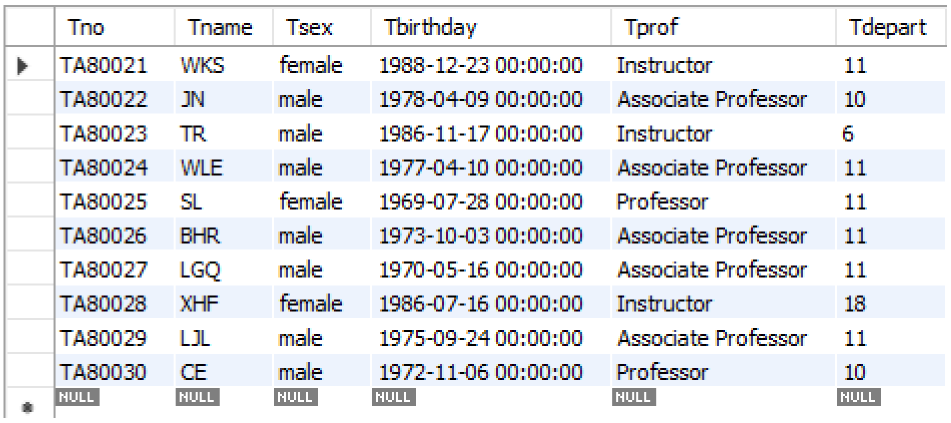
Sno	Cno	degree
SA190110001	20201102	89
SA190110002	20201102	94
SA190110003	20201102	89
SA190110004	20201102	95
SA190110005	20201102	93
SA190110006	20201102	76
SA190110007	20201102	79
SA190110008	20201102	82
SA190110001	20201103	72
SA190110002	20201103	81
SA190110003	20201103	92
SA190110004	20201103	68
SA190110005	20201103	71
SA190110006	20201103	92
SA190110005	20201009	76
SA190110006	20201009	91
SA190110007	20201009	82

student表



Sno	Sname	Ssex	Sbirthday	Sdepart
SA190110001	LY	male	1996-12-08 00:00:00	11
SA190110002	WKS	female	1997-09-12 00:00:00	12
SA190110003	WPY	male	1996-04-29 00:00:00	11
SA190110004	HTN	male	1997-03-15 00:00:00	12
SA190110005	ZDK	female	1996-08-12 00:00:00	11
SA190110006	QDS	male	1996-06-25 00:00:00	11
SA190110007	PJD	male	1996-06-14 00:00:00	12
SA190110008	LA	male	1996-08-23 00:00:00	13
SA190110009	UN	female	1996-06-23 00:00:00	13
SA190110010	WX	male	1997-02-24 00:00:00	12
SA190110011	QY	female	1996-05-08 00:00:00	14
SA190110012	ZZY	male	1995-06-26 00:00:00	15
SA190110013	WWT	male	1997-11-17 00:00:00	14
SA190110014	LSS	male	1995-01-28 00:00:00	11
SA190110015	XH	female	1996-10-09 00:00:00	12
SA190110016	LC	female	1997-11-30 00:00:00	11
SA190110017	TX	male	1996-05-16 00:00:00	12

teacher表



Tno	Tname	Tsex	Tbirthday	Tprof	Tdepart
TA80021	WKS	female	1988-12-23 00:00:00	Instructor	11
TA80022	JN	male	1978-04-09 00:00:00	Associate Professor	10
TA80023	TR	male	1986-11-17 00:00:00	Instructor	6
TA80024	WLE	male	1977-04-10 00:00:00	Associate Professor	11
TA80025	SL	female	1969-07-28 00:00:00	Professor	11
TA80026	BHR	male	1973-10-03 00:00:00	Associate Professor	11
TA80027	LGQ	male	1970-05-16 00:00:00	Associate Professor	11
TA80028	XHF	female	1986-07-16 00:00:00	Instructor	18
TA80029	LJL	male	1975-09-24 00:00:00	Associate Professor	11
TA80030	CE	male	1972-11-06 00:00:00	Professor	10
NULL	NULL	NULL	NULL	NULL	NULL

任务二：写出实现所给各题功能的SQL语句

修改基本表：

1、假设所有学生都是1班学生，在学生表 student 中增加一个新的属性列 CLASS(班)，类型为 char(20)；

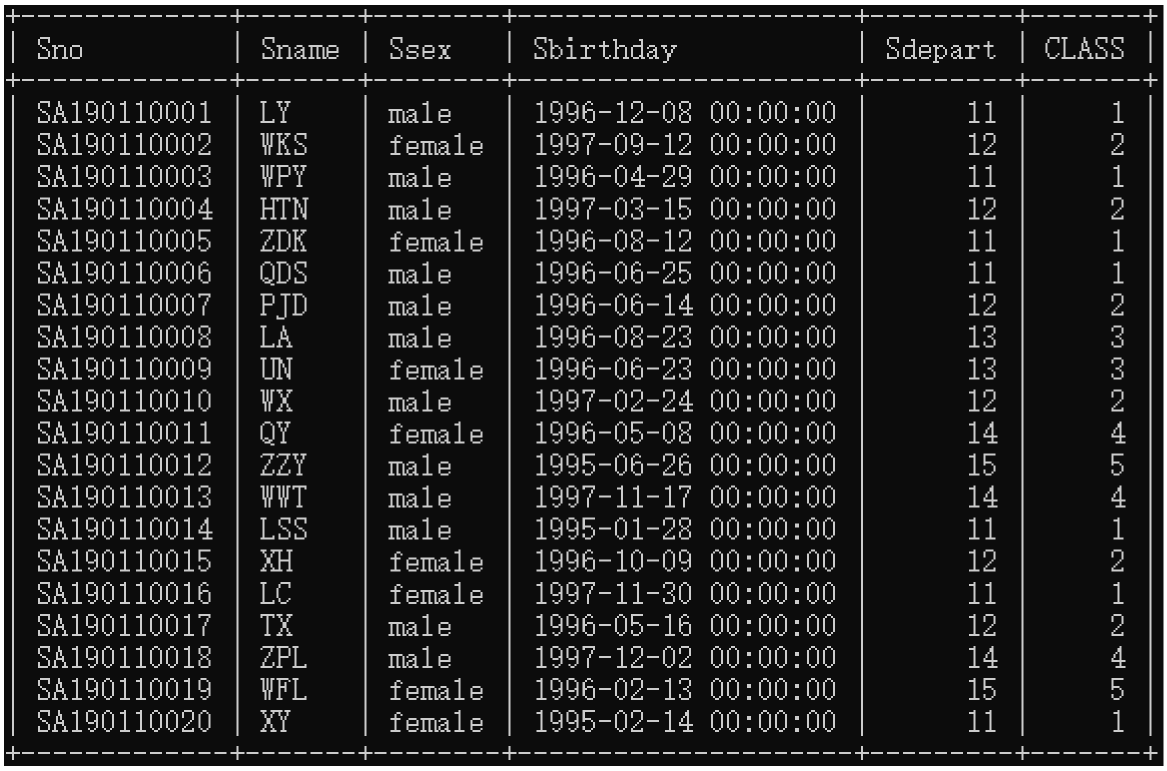
```
ALTER TABLE STUDENT ADD CLASS char(20);
```

2、 将上述新增的 CLASS属性的数据类型修改为 int。

```
ALTER TABLE STUDENT MODIFY CLASS INT;
```

3、更改每个学生的班级(CLASS)为其系(DEPART)的值减去10。注意，此操作可能需要关闭安全更新模式。

```
SET SQL_SAFE_UPDATES = 0;/*安全更新模式关闭*/
UPDATE STUDENT SET CLASS=Sdepart-10;
SET SQL_SAFE_UPDATES = 1;/*安全更新模式开启*/
```



```
+-------------+-------+--------+---------------------+---------+-------+
| Sno         | Sname | Ssex   | Sbirthday           | Sdepart | CLASS |
+-------------+-------+--------+---------------------+---------+-------+
| SA190110001 | LY    | male   | 1996-12-08 00:00:00 |      11 |     1 |
| SA190110002 | WKS   | female | 1997-09-12 00:00:00 |      12 |     2 |
| SA190110003 | WPY   | male   | 1996-04-29 00:00:00 |      11 |     1 |
| SA190110004 | HTN   | male   | 1997-03-15 00:00:00 |      12 |     2 |
| SA190110005 | ZDK   | female | 1996-08-12 00:00:00 |      11 |     1 |
| SA190110006 | QDS   | male   | 1996-06-25 00:00:00 |      11 |     1 |
| SA190110007 | PJD   | male   | 1996-06-14 00:00:00 |      12 |     2 |
| SA190110008 | LA    | male   | 1996-08-23 00:00:00 |      13 |     3 |
| SA190110009 | UN    | female | 1996-06-23 00:00:00 |      13 |     3 |
| SA190110010 | WX    | male   | 1997-02-24 00:00:00 |      12 |     2 |
| SA190110011 | QY    | female | 1996-05-08 00:00:00 |      14 |     4 |
| SA190110012 | ZZY   | male   | 1995-06-26 00:00:00 |      15 |     5 |
| SA190110013 | WWT   | male   | 1997-11-17 00:00:00 |      14 |     4 |
| SA190110014 | LSS   | male   | 1995-01-28 00:00:00 |      11 |     1 |
| SA190110015 | XH    | female | 1996-10-09 00:00:00 |      12 |     2 |
| SA190110016 | LC    | female | 1997-11-30 00:00:00 |      11 |     1 |
| SA190110017 | TX    | male   | 1996-05-16 00:00:00 |      12 |     2 |
| SA190110018 | ZPL   | male   | 1997-12-02 00:00:00 |      14 |     4 |
| SA190110019 | WFL   | female | 1996-02-13 00:00:00 |      15 |     5 |
| SA190110020 | XY    | female | 1995-02-14 00:00:00 |      11 |     1 |
+-------------+-------+--------+---------------------+---------+-------+
```

4、为student表中的字段SEX实现用户自定义约束（注：MySQL中check语句是不起作用的），并说明作用；

```sql
ALTER TABLE STUDENT MODIFY Ssex enum('male','female');
/*作用是使得插入或修改数据时，该属性值只能为'male'或'female'*/
```

5、创建一个课程平均成绩表：course_ave(CNO,TNO,AVE_SCORE)，三个属性分别表示课程号，授课教师工号，课程平均成绩，类型自定义；

```sql
CREATE TABLE course_ave
(CNO CHAR(8) ,
  TNO VARCHAR(7),
  AVE_SCORE FLOAT
);
```

6、 为表 course_ave 添加主键(CNO);

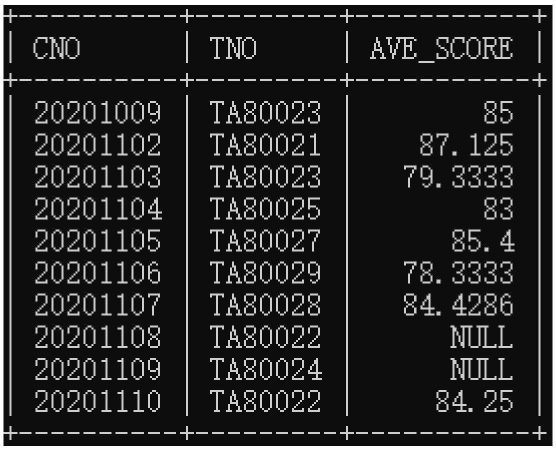
```sql
ALTER TABLE course_ave ADD PRIMARY KEY (CNO);
```

7、为表course_ave 字段TNO添加非空约束;

```sql
ALTER TABLE course_ave MODIFY TNO VARCHAR(7) NOT NULL;
```

8、用一条语句，结合表score记录，为表course中所有课程，在表course_ave添加对应记录（若是表score中未出现的课程，则平均成绩记为空）；

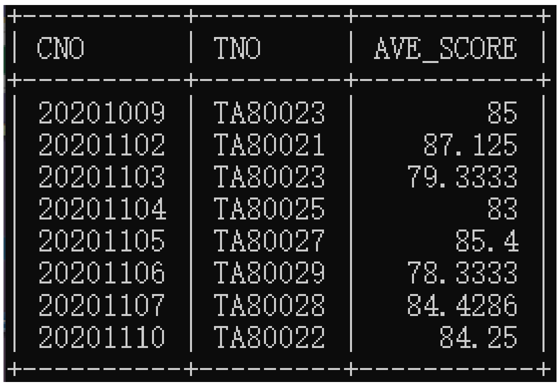
```sql
INSERT INTO course_ave(CNO, TNO, AVE_SCORE)
SELECT COURSE.CNO, COURSE.TNO, AVG(SCORE.degree)
FROM SCORE SCORE RIGHT JOIN COURSE ON(SCORE.CNO=COURSE.CNO)
GROUP BY COURSE.CNO;
```



```
+----------+---------+-----------+
| CNO      | TNO     | AVE_SCORE |
+----------+---------+-----------+
| 20201009 | TA80023 |        85 |
| 20201102 | TA80021 |    87.125 |
| 20201103 | TA80023 |   79.3333 |
| 20201104 | TA80025 |        83 |
| 20201105 | TA80027 |      85.4 |
| 20201106 | TA80029 |   78.3333 |
| 20201107 | TA80028 |   84.4286 |
| 20201108 | TA80022 |      NULL |
| 20201109 | TA80024 |      NULL |
| 20201110 | TA80022 |     84.25 |
+----------+---------+-----------+
```

9、删除course_ave表中所有平均成绩为空的课程记录。

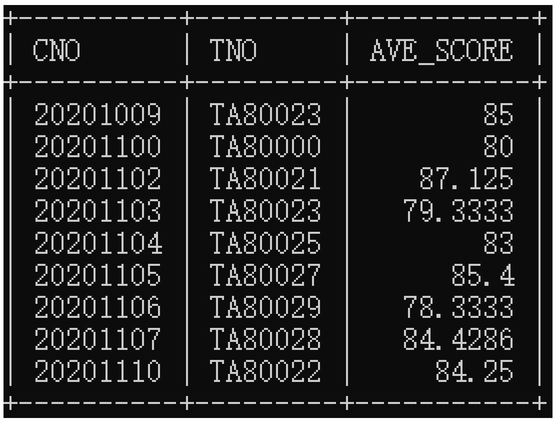
```
SET SQL_SAFE_UPDATES = 0;/*安全更新模式关闭*/
DELETE FROM `course_ave` WHERE(`AVE_SCORE` IS NULL);
SET SQL_SAFE_UPDATES = 1;/*安全更新模式开启*/
```



```
+----------+---------+-----------+
| CNO      | TNO     | AVE_SCORE |
+----------+---------+-----------+
| 20201009 | TA80023 |        85 |
| 20201102 | TA80021 |    87.125 |
| 20201103 | TA80023 |   79.3333 |
| 20201104 | TA80025 |        83 |
| 20201105 | TA80027 |      85.4 |
| 20201106 | TA80029 |   78.3333 |
| 20201107 | TA80028 |   84.4286 |
| 20201110 | TA80022 |     84.25 |
+----------+---------+-----------+
```

10、为表course_ave添加课程号为“20201100”，授课教师号为“TA80000”，平均成绩为80的记录。

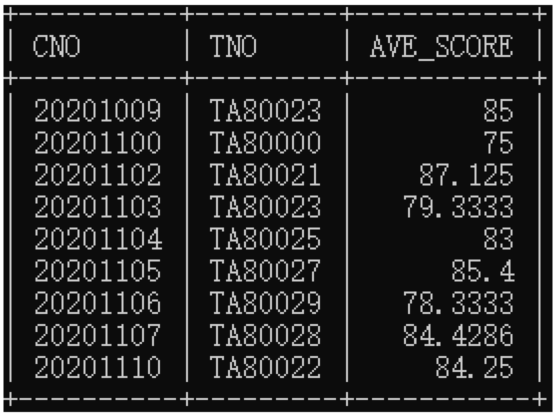
```
INSERT INTO course_ave(CNO, TNO, AVE_SCORE)
VALUE ('20201100', 'TA80000', 80);
```



```
+----------+---------+-----------+
| CNO      | TNO     | AVE_SCORE |
+----------+---------+-----------+
| 20201009 | TA80023 |        85 |
| 20201100 | TA80000 |        80 |
| 20201102 | TA80021 |    87.125 |
| 20201103 | TA80023 |   79.3333 |
| 20201104 | TA80025 |        83 |
| 20201105 | TA80027 |      85.4 |
| 20201106 | TA80029 |   78.3333 |
| 20201107 | TA80028 |   84.4286 |
| 20201110 | TA80022 |     84.25 |
+----------+---------+-----------+
```

11、修改课程号为“20201100”的课程的平均成绩为75.

```
UPDATE course_ave SET `AVE_SCORE`=75 WHERE (`CNO`='20201100');
```



```
+----------+---------+-----------+
| CNO      | TNO     | AVE_SCORE |
+----------+---------+-----------+
| 20201009 | TA80023 |        85 |
| 20201100 | TA80000 |        75 |
| 20201102 | TA80021 |    87.125 |
| 20201103 | TA80023 |   79.3333 |
| 20201104 | TA80025 |        83 |
| 20201105 | TA80027 |      85.4 |
| 20201106 | TA80029 |   78.3333 |
| 20201107 | TA80028 |   84.4286 |
| 20201110 | TA80022 |     84.25 |
+----------+---------+-----------+
```

索引：

12、 用 create 语句在 course 的 名称CNAME上建立普通索引 CNAME_INDEX；

```sql
CREATE INDEX CNAME_INDEX ON COURSE(Cname);
```

13、用 create 语句在 student 的学号 NO 上建立唯一索引 NO_INDEX；

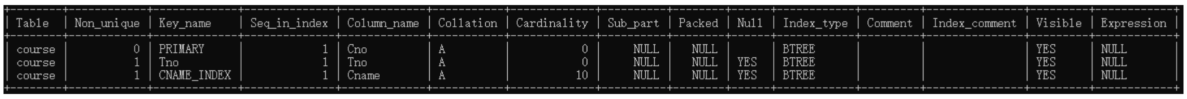
```sql
CREATE UNIQUE INDEX NO_INDEX ON STUDENT(Sno);
```

14、用 create 语句在 Score 表上的学号 NO、成绩 Degree 上建立复合索引 NODE_INDEX，要求学号为升序，学号相同时成绩为降序。

```sql
CREATE INDEX NODE_INDEX ON SCORE(Sno ASC, Cno DESC);
```

15、用show语句查询表course的索引

```sql
SHOW INDEX FROM COURSE;
```



```
| Table  | Non_unique | Key_name    | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
| course |          0 | PRIMARY     |            1 | Cno         | A         |           0 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| course |          1 | Tno         |            1 | Tno         | A         |           0 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| course |          1 | CNAME_INDEX |            1 | Cname       | A         |          10 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
```

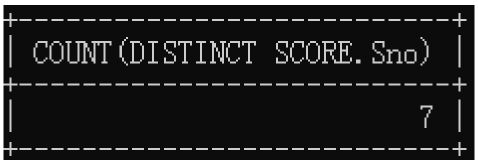
16、删除course表字段CNAME上的普通索引CNAME_INDEX；

```sql
DROP INDEX CNAME_INDEX ON COURSE;
```

查询：

17、查询选过18系老师课程的学生的人数；

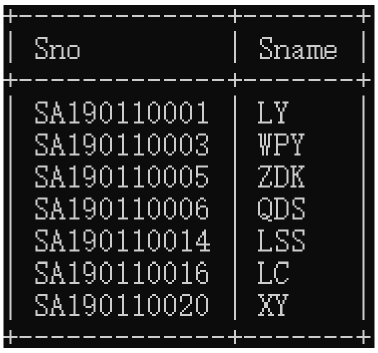
```sql
SELECT COUNT(DISTINCT SCORE.Sno)
FROM COURSE, TEACHER, SCORE
WHERE COURSE.Tno=TEACHER.Tno AND COURSE.Cno=SCORE.Cno AND TEACHER.Tdepart=18;
```



```
+---------------------------+
| COUNT(DISTINCT SCORE.Sno) |
+---------------------------+
|                         7 |
+---------------------------+
```

18、查询与学生“XY”属于同一个系的学生学号和姓名；

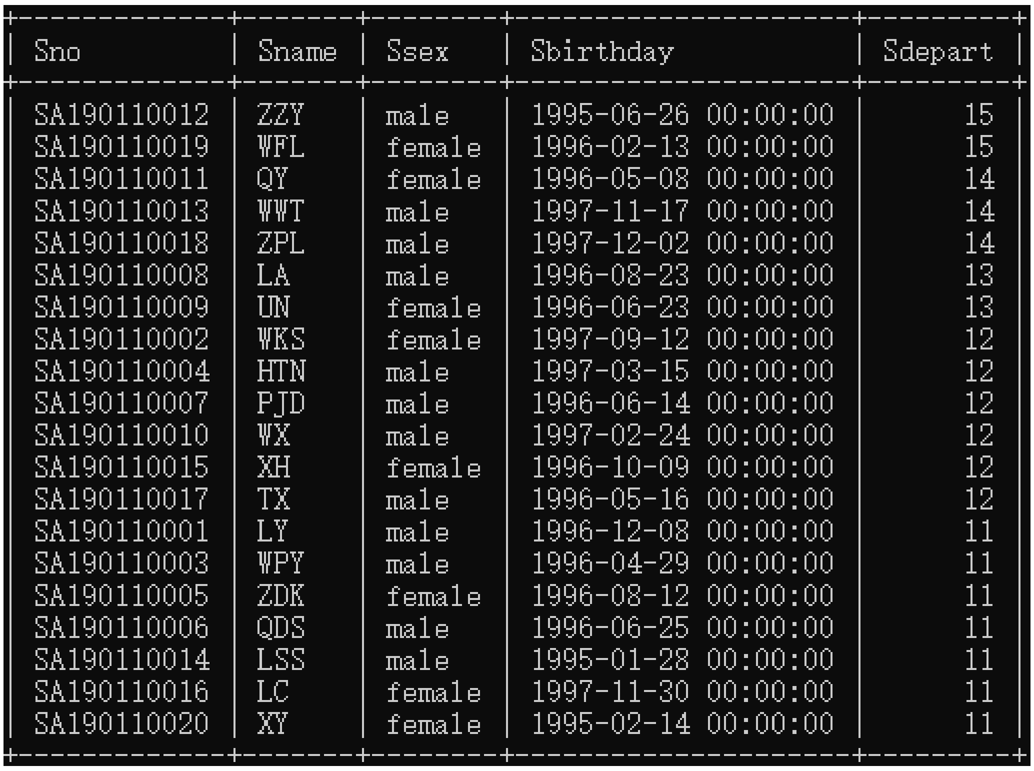
```sql
SELECT Sno, Sname
FROM STUDENT
WHERE Sdepart= (SELECT Sdepart FROM STUDENT WHERE Sname='XY');
```



```
+-------------+-------+
| Sno         | Sname |
+-------------+-------+
| SA190110001 | LY    |
| SA190110003 | WPY   |
| SA190110005 | ZDK   |
| SA190110006 | QDS   |
| SA190110014 | LSS   |
| SA190110016 | LC    |
| SA190110020 | XY    |
+-------------+-------+
```

19、查询全体学生的情况，查询结果按所在系降序排列，对同一系中的学生按学号升序排列；

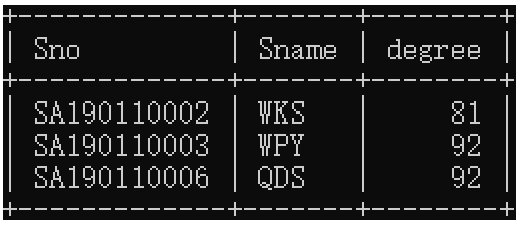
```sql
SELECT *
FROM STUDENT
ORDER BY Sdepart DESC, Sno ASC;
```



```
+-------------+-------+--------+---------------------+---------+
| Sno         | Sname | Ssex   | Sbirthday           | Sdepart |
+-------------+-------+--------+---------------------+---------+
| SA190110012 | ZZY   | male   | 1995-06-26 00:00:00 |      15 |
| SA190110019 | WFL   | female | 1996-02-13 00:00:00 |      15 |
| SA190110011 | QY    | female | 1996-05-08 00:00:00 |      14 |
| SA190110013 | WWT   | male   | 1997-11-17 00:00:00 |      14 |
| SA190110018 | ZPL   | male   | 1997-12-02 00:00:00 |      14 |
| SA190110008 | LA    | male   | 1996-08-23 00:00:00 |      13 |
| SA190110009 | UN    | female | 1996-06-23 00:00:00 |      13 |
| SA190110002 | WKS   | female | 1997-09-12 00:00:00 |      12 |
| SA190110004 | HTN   | male   | 1997-03-15 00:00:00 |      12 |
| SA190110007 | PJD   | male   | 1996-06-14 00:00:00 |      12 |
| SA190110010 | WX    | male   | 1997-02-24 00:00:00 |      12 |
| SA190110015 | XH    | female | 1996-10-09 00:00:00 |      12 |
| SA190110017 | TX    | male   | 1996-05-16 00:00:00 |      12 |
| SA190110001 | LY    | male   | 1996-12-08 00:00:00 |      11 |
| SA190110003 | WPY   | male   | 1996-04-29 00:00:00 |      11 |
| SA190110005 | ZDK   | female | 1996-08-12 00:00:00 |      11 |
| SA190110006 | QDS   | male   | 1996-06-25 00:00:00 |      11 |
| SA190110014 | LSS   | male   | 1995-01-28 00:00:00 |      11 |
| SA190110016 | LC    | female | 1997-11-30 00:00:00 |      11 |
| SA190110020 | XY    | female | 1995-02-14 00:00:00 |      11 |
+-------------+-------+--------+---------------------+---------+
```

20、查询选修“Machine_Learning”课程且成绩在80分及以上的学生的学号、姓名和分数；

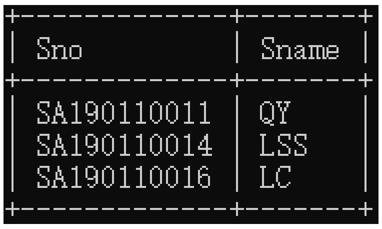
```sql
SELECT STUDENT.Sno, STUDENT.Sname, degree
FROM STUDENT, COURSE, SCORE
WHERE STUDENT.Sno=SCORE.Sno AND SCORE.Cno=COURSE.Cno AND Cname='Machine_Learning' AND degree > 80;
```



```
+-------------+-------+--------+
| Sno         | Sname | degree |
+-------------+-------+--------+
| SA190110002 | WKS   |     81 |
| SA190110003 | WPY   |     92 |
| SA190110006 | QDS   |     92 |
+-------------+-------+--------+
```

21、查询选修过“LJL”老师课程的学生学号和姓名；

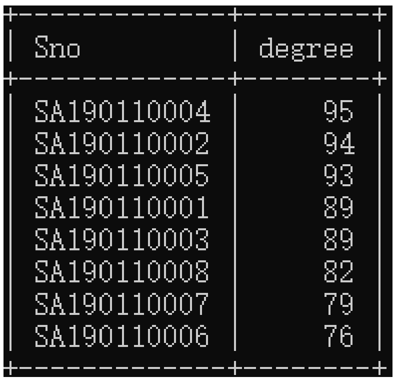
```sql
SELECT DISTINCT STUDENT.Sno, STUDENT.Sname
FROM STUDENT, COURSE, TEACHER, SCORE
WHERE STUDENT.Sno=SCORE.Sno AND COURSE.Cno=SCORE.Cno AND
COURSE.Tno=TEACHER.Tno AND Tname='LJL';
```



```
+-------------+--------+
| Sno         | Sname  |
+-------------+--------+
| SA190110011 | QY     |
| SA190110014 | LSS    |
| SA190110016 | LC     |
+-------------+--------+
```

22、查询选过Database课程的学生学号和分数，并按分数降序展示；

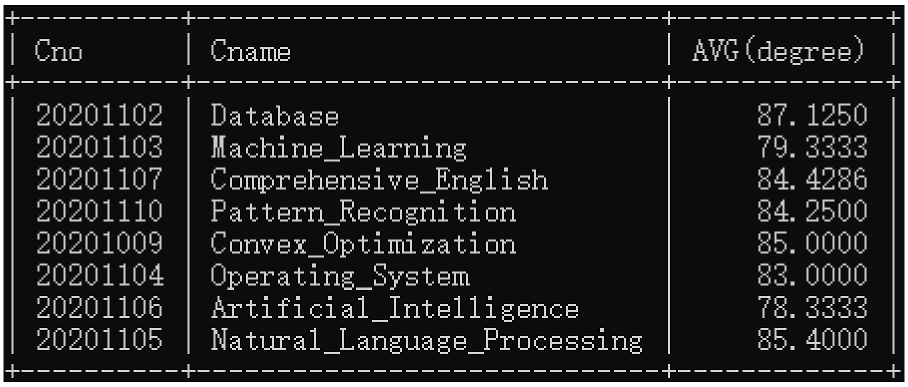
```sql
SELECT STUDENT.Sno, degree
FROM STUDENT, COURSE, SCORE
WHERE STUDENT.Sno=SCORE.Sno AND SCORE.Cno=COURSE.Cno AND Cname='Database'
ORDER BY degree DESC;
```



```
+-------------+--------+
| Sno         | degree |
+-------------+--------+
| SA190110004 |     95 |
| SA190110002 |     94 |
| SA190110005 |     93 |
| SA190110001 |     89 |
| SA190110003 |     89 |
| SA190110008 |     82 |
| SA190110007 |     79 |
| SA190110006 |     76 |
+-------------+--------+
```

23、查询每门课的平均成绩，其中每行包含课程号、课程名和平均成绩；

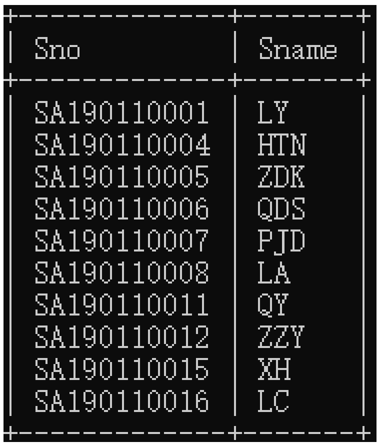
```sql
SELECT DISTINCT COURSE.Cno, COURSE.Cname, AVG(degree)
FROM COURSE, SCORE
WHERE COURSE.Cno=SCORE.Cno
GROUP BY COURSE.Cno;
```



```
+----------+-----------------------------+-------------+
| Cno      | Cname                       | AVG(degree) |
+----------+-----------------------------+-------------+
| 20201102 | Database                    |     87.1250 |
| 20201103 | Machine_Learning            |     79.3333 |
| 20201107 | Comprehensive_English       |     84.4286 |
| 20201110 | Pattern_Recognition         |     84.2500 |
| 20201009 | Convex_Optimization         |     85.0000 |
| 20201104 | Operating_System            |     83.0000 |
| 20201106 | Artificial_Intelligence     |     78.3333 |
| 20201105 | Natural_Language_Processing |     85.4000 |
+----------+-----------------------------+-------------+
```

24、查询考试成绩有低于80分情况的学生学号和姓名（去掉重复行）；

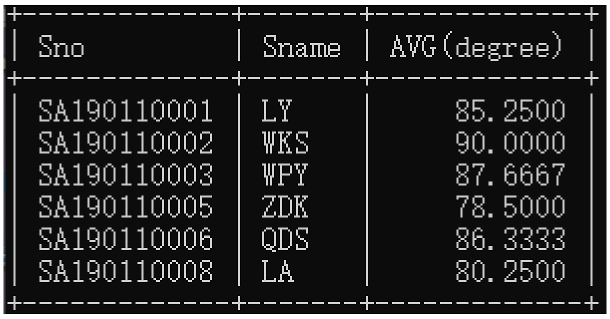
```sql
SELECT DISTINCT STUDENT.Sno, STUDENT.Sname
FROM STUDENT, SCORE
WHERE STUDENT.Sno=SCORE.Sno AND degree < 80;
```



```
+-------------+-------+
| Sno         | Sname |
+-------------+-------+
| SA190110001 | LY    |
| SA190110004 | HTN   |
| SA190110005 | ZDK   |
| SA190110006 | QDS   |
| SA190110007 | PJD   |
| SA190110008 | LA    |
| SA190110011 | QY    |
| SA190110012 | ZZY   |
| SA190110015 | XH    |
| SA190110016 | LC    |
+-------------+-------+
```

25、查询选修了3门课程及以上的学生的学号、姓名和平均成绩；

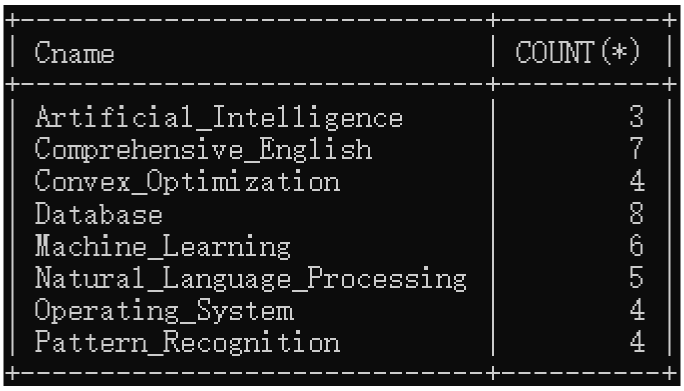
```
SELECT DISTINCT STUDENT.Sno, STUDENT.Sname, AVG(degree)
FROM STUDENT, SCORE
WHERE STUDENT.Sno=SCORE.Sno
GROUP BY STUDENT.Sno
HAVING COUNT(*)>=3;
```



```
+-------------+-------+-------------+
| Sno         | Sname | AVG(degree) |
+-------------+-------+-------------+
| SA190110001 | LY    |     85.2500 |
| SA190110002 | WKS   |     90.0000 |
| SA190110003 | WPY   |     87.6667 |
| SA190110005 | ZDK   |     78.5000 |
| SA190110006 | QDS   |     86.3333 |
| SA190110008 | LA    |     80.2500 |
+-------------+-------+-------------+
```

26、查询各个课程名与相应的选课人数；

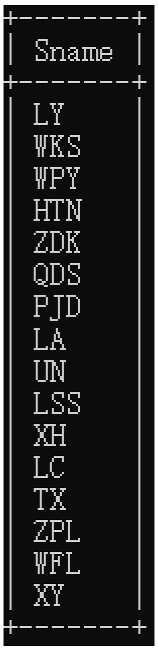
```
SELECT DISTINCT COURSE.Cname, COUNT(*)
FROM COURSE, SCORE
WHERE COURSE.Cno=SCORE.Cno
GROUP BY COURSE.Cno;
```



```
+-----------------------------+----------+
| Cname                       | COUNT(*) |
+-----------------------------+----------+
| Artificial_Intelligence     |        3 |
| Comprehensive_English       |        7 |
| Convex_Optimization         |        4 |
| Database                    |        8 |
| Machine_Learning            |        6 |
| Natural_Language_Processing |        5 |
| Operating_System            |        4 |
| Pattern_Recognition         |        4 |
+-----------------------------+----------+
```

27、查询所有未选修“Operating_System”课程的学生姓名；

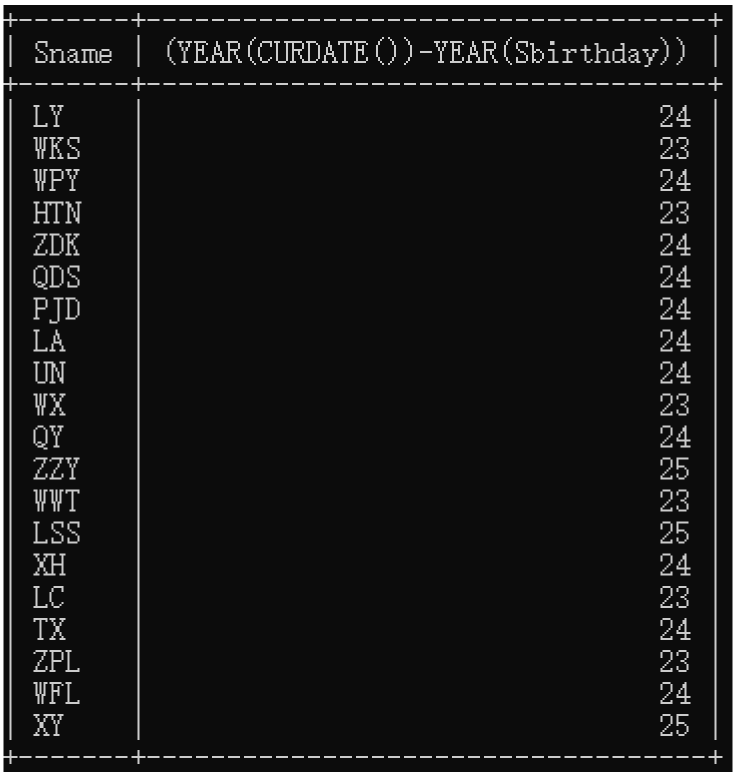
```sql
SELECT DISTINCT STUDENT.Sname
FROM STUDENT
WHERE NOT EXISTS
    (SELECT *
    FROM COURSE, SCORE
    WHERE STUDENT.Sno=SCORE.Sno AND COURSE.Cname='Operating_System'AND
COURSE.Cno=SCORE.Cno);
```



```
+-------+
| Sname |
+-------+
| LY    |
| WKS   |
| WPY   |
| HTN   |
| ZDK   |
| QDS   |
| PJD   |
| LA    |
| UN    |
| LSS   |
| XH    |
| LC    |
| TX    |
| ZPL   |
| WFL   |
| XY    |
+-------+
```

28、查询年龄在 24 岁及以下的学生姓名和年龄（只考虑年）；

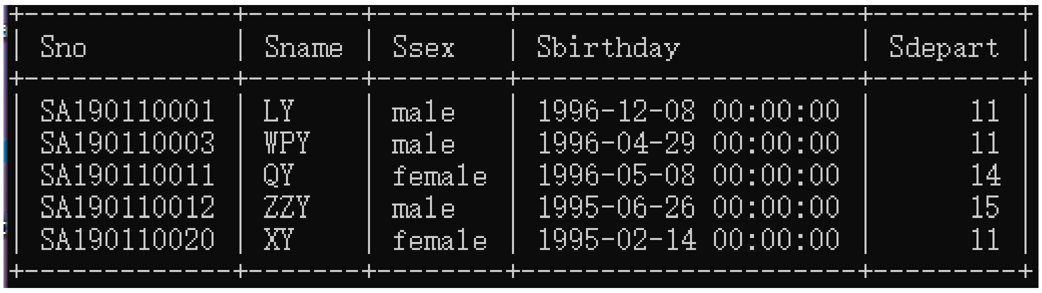
```sql
SELECT Sname, (YEAR(CURDATE())-YEAR(Sbirthday))
FROM STUDENT;
```



```
+-------+-----------------------------------+
| Sname | (YEAR(CURDATE())-YEAR(Sbirthday)) |
+-------+-----------------------------------+
| LY    |                                24 |
| WKS   |                                23 |
| WPY   |                                24 |
| HTN   |                                23 |
| ZDK   |                                24 |
| QDS   |                                24 |
| PJD   |                                24 |
| LA    |                                24 |
| UN    |                                24 |
| WX    |                                23 |
| QY    |                                24 |
| ZZY   |                                25 |
| WWT   |                                23 |
| LSS   |                                25 |
| XH    |                                24 |
| LC    |                                23 |
| TX    |                                24 |
| ZPL   |                                23 |
| WFL   |                                24 |
| XY    |                                25 |
+-------+-----------------------------------+
```

29、查询名字以'Y'结尾的同学记录；

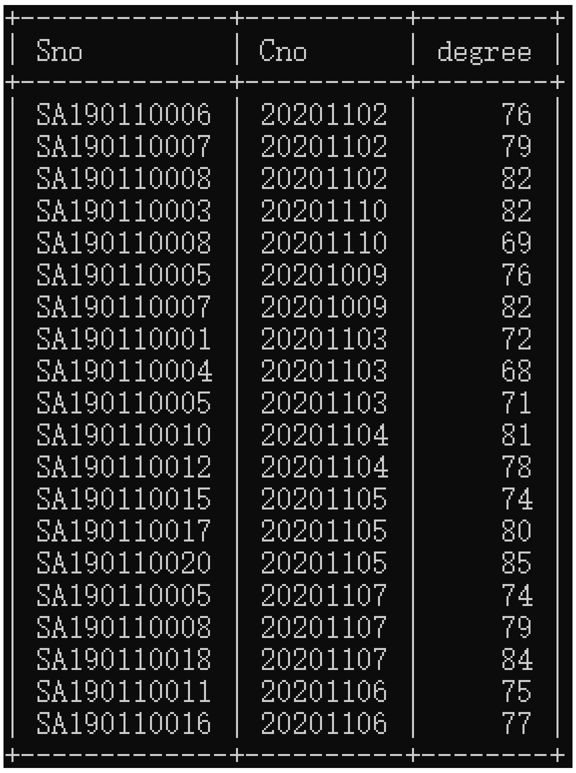
```sql
SELECT *
FROM STUDENT
WHERE Sname LIKE '%Y';
```



```
+-------------+-------+--------+---------------------+---------+
| Sno         | Sname | Ssex   | Sbirthday           | Sdepart |
+-------------+-------+--------+---------------------+---------+
| SA190110001 | LY    | male   | 1996-12-08 00:00:00 |      11 |
| SA190110003 | WPY   | male   | 1996-04-29 00:00:00 |      11 |
| SA190110011 | QY    | female | 1996-05-08 00:00:00 |      14 |
| SA190110012 | ZZY   | male   | 1995-06-26 00:00:00 |      15 |
| SA190110020 | XY    | female | 1995-02-14 00:00:00 |      11 |
+-------------+-------+--------+---------------------+---------+
```

30、查询成绩比该课程平均成绩低的同学的成绩表；

```sql
SELECT S1.*
FROM COURSE, SCORE S1
WHERE COURSE.Cno=S1.Cno AND S1.degree< ANY
(SELECT AVG(degree)
FROM SCORE S2
WHERE COURSE.Cno=S2.Cno
GROUP BY S2.Cno
);
```

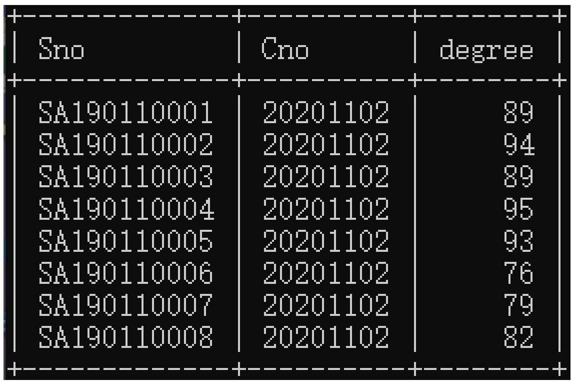


```
+-------------+----------+--------+
| Sno         | Cno      | degree |
+-------------+----------+--------+
| SA190110006 | 20201102 |     76 |
| SA190110007 | 20201102 |     79 |
| SA190110008 | 20201102 |     82 |
| SA190110003 | 20201110 |     82 |
| SA190110008 | 20201110 |     69 |
| SA190110005 | 20201009 |     76 |
| SA190110007 | 20201009 |     82 |
| SA190110001 | 20201103 |     72 |
| SA190110004 | 20201103 |     68 |
| SA190110005 | 20201103 |     71 |
| SA190110010 | 20201104 |     81 |
| SA190110012 | 20201104 |     78 |
| SA190110015 | 20201105 |     74 |
| SA190110017 | 20201105 |     80 |
| SA190110020 | 20201105 |     85 |
| SA190110005 | 20201107 |     74 |
| SA190110008 | 20201107 |     79 |
| SA190110018 | 20201107 |     84 |
| SA190110011 | 20201106 |     75 |
| SA190110016 | 20201106 |     77 |
+-------------+----------+--------+
```

视图：

31、建立选过Database课程的学生的成绩视图（db_student_score），属性与score表一样，并要求对该视图进行修改和插入操作时仍需保证该视图只有选过 Database课程的学生；

```
CREATE VIEW db_student_score
AS
SELECT SCORE.Sno, SCORE.Cno, SCORE.degree
FROM SCORE, COURSE
WHERE COURSE.Cname = 'Database' AND COURSE.Cno=SCORE.Cno
WITH CHECK OPTION;
```



```
+-------------+----------+--------+
| Sno         | Cno      | degree |
+-------------+----------+--------+
| SA190110001 | 20201102 |     89 |
| SA190110002 | 20201102 |     94 |
| SA190110003 | 20201102 |     89 |
| SA190110004 | 20201102 |     95 |
| SA190110005 | 20201102 |     93 |
| SA190110006 | 20201102 |     76 |
| SA190110007 | 20201102 |     79 |
| SA190110008 | 20201102 |     82 |
+-------------+----------+--------+
```

32、删除视图 db_student_score。

```
DROP VIEW db_student_score;
```

触发器：

33、为数据库创建触发器。目标：维持学生数量更新的一致性。
a) 创建关系表：Gender(SEX, s_count)。其中 SEX为性别，非空，主键；s_count是对应性别统计得到的人数。

```
CREATE TABLE Gender
(
    CHAR(6) SEX NOT NULL PRIMARY KEY,
    INT s_count
);
```

b) 根据 student 关系表，补全 Gender中 s_count 的数据。

```
INSERT
    INTO Gender (SEX, s_count)
    SELECT Ssex, COUNT(*)
    FROM STUDENT
    GROUP BY Ssex;
```

c) 为关系表 student 创建两个触发器，使得每插入（删除）一条学生记录，Gender 表中对应性别的人数加1（减1）。

```
DELIMITER ||
(1)
CREATE TRIGGER update_Gende_plus
AFTER INSERT ON STUDENT
/*触发事件是插入操作*/
FOR EACH ROW /*行级触发器*/
BEGIN /*定义触发动作体，是PL/SQL过程块*/
UPDATE Gender
SET s_count=s_count+1
```

```
WHERE Gender.SEX=new.Ssex;
END||
(2)
CREATE TRIGGER update_Gende_minus
AFTER DELETE ON STUDENT
/*触发事件是删除操作*/
FOR EACH ROW /*行级触发器*/
BEGIN /*定义触发动作体，是PL/SQL过程块*/
UPDATE Gender
SET s_count=s_count-1
WHERE Gender.SEX=old.Ssex;
END||
```

d) 检验触发器是否工作。

```
DELIMITER ;

INSERT INTO STUDENT
VALUES ('SA200110001','DYX','male','1999-01-29 00:00:00',16,6);

SELECT * FROM 2020exp1.gender;

DELETE
FROM STUDENT
WHERE Sno='SA200110001';

SELECT * FROM 2020exp1.gender;
```

经检验确实工作。