

实验二-复制表, 删除数据

宋振华 201605301357

2019 年 4 月 8 日

1 实验内容

1. 找出没有选修任何课程的学生的学号, 姓名 (即没有选课记录的学生)

```
1 create table test2_01 as
2 select distinct stu.SID, stu.NAME
3 from pub.STUDENT stu
4 where stu.SID not in
5     (select distinct stucor.SID
6      from pub.STUDENT_COURSE stucor);
```

2. 找出至少选修了学号为'200900130417' 的学生所选修的一门课的学生的学号, 姓名

```
1 create table test2_02 as
2 select distinct stu1.SID, stu1.NAME
3 from pub.STUDENT_COURSE stucor,
4      pub.STUDENT_COURSE stucor1,
5      pub.STUDENT stu1
6 where stucor.SID='200900130417'
7      and stucor.CID=stucor1.CID
8      and stucor1.SID=stu1.SID;
```

3. 找出至少选修了一门其先行课程号为'300002' 号课程的学生的学号, 姓名

```

1 create table test2_03 as
2 select stu.sid, stu.name
3 from pub.STUDENT stu,
4      pub.STUDENT_COURSE stucor,
5      pub.COURSE course
6 where stu.SID=stucor.SID
7        and stucor.CID=course.CID
8        and course.FCID='300002';

```

4. 找出选修了'操作系统'并且也选修了'数据结构'的学生的学号, 姓名

```

1 create table test2_04 as
2 (select stu.SID,stu.NAME
3 from pub.COURSE course,
4      pub.STUDENT_COURSE stucor,
5      pub.STUDENT stu
6 where course.name='操作系统'
7        and course.CID=stucor.CID
8        and stu.SID=stucor.SID)
9 intersect
10 (select stu.SID, stu.NAME
11 from pub.COURSE course,
12      pub.STUDENT_COURSE stucor,
13      pub.STUDENT stu
14 where course.name='数据结构'
15        and course.CID=stucor.CID
16        and stu.SID=stucor.SID);

```

5. 查询 20 岁的所有有选课的学生的学号, 姓名, 平均成绩 (avg_score, 此为列名, 下同)(平均成绩四舍五入到个位), 总成绩 (sum_score) Test2_05 有四个列, 并且列名必须是: sid,name,avg_score,sum_score

```

1 create table test2_05 as
2 select stu.sid, stu.name,
3        cast(avg(score) as numeric(5,0)) avg_score,

```

```

4         sum(score) sum_score
5 from pub.STUDENT stu,
6      pub.STUDENT_COURSE stucor
7 where stu.sid=stucor.sid
8        and stu.age=20
9 group by stu.sid, stu.NAME;

```

6. 查询所有课的最高成绩, 次高成绩 (次高成绩一定小于最高成绩), 最高成绩人数, test2_06 有四个列: 课程号 cid, 课程名称 name, 最高成绩 max_score, 次高成绩 max_score2, 最高成绩人数 max_score_count (一个学生同一门课成绩都是第一, 只计一次). 如果没有学生选课, 则最高成绩为空值, 最高成绩人数为零. 如果没有次高成绩, 则次高成绩为空值.

```

1 create table test2_06 as
2 with
3     score_max(cid, max_score) as (
4         select cid, max(score)
5         from pub.student_course
6         group by cid),
7     score_max2(cid, max_score2) as (
8         select cid, max(score)
9         from pub.Student_course sc
10        where sc.score <
11              (select max_score
12               from score_max
13               where sc.cid=score_max.cid)
14        group by cid),
15     temp(cid, sid, max_score_count) as (
16         select cid, sid, count(score)
17         from pub.student_course sc
18         where score =
19              (select max_score
20               from score_max

```

```

21         where sc.cid=score_max.cid)
22         group by cid, sid),
23     score_max_count (cid, max_score_count) as
24         (select cid, count(sid)
25          from temp group by cid)
26 select cid, name, max_score,
27        max_score2, max_score_count
28 from pub.course
29     natural left outer join score_max
30     natural full outer join score_max2
31     natural full outer join score_max_count

```

思路: 首先使用 with 建立四个临时关系, 第一个关系是计算出每一门课对应的最大分数, 第二个关系计算出每门课的次最大关系, 通过选取小于最大分数的分数中最大的分数得到, 第三个关系是为了构建第四个关系, 而临时构建的一个关系, 第三个关系按照课程 id 和学生 id 分组, 我们计算每个课程 ID 对应的每个学生拿到最高分的次数, 第四个关系通过对第三个关系按照课程 id 分组, 计算每门课程 id 拿到最高分的学生数目, 即可得到最高成绩人数, 并且对一个学生在同一门课中成绩都是第一进行了去重, 最后我们左外连接这 course 和上面的第一、第二、第四个表即得到了最后的表格.

7. 查询所有不姓张, 不姓李, 也不姓王的学生的学号 sid, 姓名 name

```

1 create table test2_07 as
2 select stu.sid,
3        stu.name from pub.STUDENT stu
4 where stu.name not like '王%'
5        and stu.NAME not like '张%'
6        and stu.NAME not like '李%';

```

8. 查询学生表中每一个姓氏及其人数 (不考虑复姓), test2_08 有两个列: second_name, p_count

```

1 create table test2_08 as
2 select substr(name,1,1) second_name,

```

```

3         count(*) p_count from pub.student stu
4     group by substr(name,1,1);

```

9. 查询选修了 300003 号课程的学生们的 sid,name,score

```

1     create table test2_09 as
2     select stu.sid, stu.name,
3            stucor.score from pub.STUDENT stu,
4            pub.STUDENT_COURSE stucor
5     where stu.sid=stucor.sid
6            and stucor.cid='300003';

```

10. 找出同一个同学同一门课程有两次或以上不及格的所有学生的学号, 姓名 (即一门课程需要补考两次或以上的学生的学号, 姓名)

```

1     create table test2_10 as
2     select sid, name
3     from pub.student
4     where sid in
5     (
6         select sid from pub.student_course
7         where score < 60 group by sid, cid
8         having count(cid) >=2
9     )

```

2 收获

我对 select 查询语句有了更深刻的认识, 了解到课上没有讲过但很有用的操作, 比如 substr, round 等.