

山东大学 泰山学堂计算机取向

数据库系统 课程实验报告

学号：201600262022	姓名：张童	班级：16 泰山学堂
实验题目：检索查询	实验学时：2	
实验内容：		
针对公共用户 pub 下的表，完成实验内容要求 10 个查询。		
实验过程记录：		
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)。		
sql 语句：		
<pre>create table test2_01 as select sid, name from pub.STUDENT where sid not in (select sid from pub.STUDENT_COURSE)</pre>		
结果：		
201600262022 张童	2 - 1	查询1
2019-04-14	按时完成	(NULL)
2019-03-30 11:29:27	2.0 / 2.0	
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。		
sql 语句：		
<pre>create table test2_02 as select sid, name from pub.STUDENT where sid in (select sc2.SID from pub.STUDENT_COURSE sc1, pub.STUDENT_COURSE sc2 where sc1.SID='200900130417'and sc1.CID=sc2.CID)</pre>		
结果：		
201600262022 张童	2 - 2	查询2
2019-04-14	按时完成	(NULL)
2019-03-30 11:49:21	2.0 / 2.0	
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。		
sql 语句：		
<pre>create table test2_03 as select sid, name from pub.STUDENT where sid in (select sid from pub.STUDENT_COURSE natural join pub.COURSE where fcid='300002')</pre>		
结果：		
201600262022 张童	2 - 3	查询3
2019-04-14	按时完成	(NULL)
2019-03-30 15:11:19	2.0 / 2.0	
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。		
sql 语句：		
<pre>create table test2_04 as select sid, name from pub.STUDENT where sid in (select sc1.SID from pub.STUDENT_COURSE sc1, pub.STUDENT_COURSE sc2 where sc1.SID= sc2.SID and sc1.CID in (select cid from pub.COURSE where name='数据结构') and sc2.CID in (select cid from pub.COURSE where name='操作系统'))</pre>		
结果：		
201600262022 张童	2 - 5	查询5
2019-04-14	按时完成	(NULL)
2019-03-30 17:00:45	2.0 / 2.0	
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		
sql 语句：		

```
create table test2_05 as↓
select sid, name, round(avg(score),0) avg_score, sum(score) sum_score
from pub.STUDENT natural join pub.STUDENT_COURSE↓
where age=20 group by sid, name↓
```

结果:

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。

sql 语句:

```
create table test2_06 as↓
with score_max(cid, max_score) ↓
as(select cid, max(score) from pub.STUDENT_COURSE group by cid),↓
score_max2(cid, max_score2) ↓
as (select cid, max(score) from pub.STUDENT_COURSE sc where sc.score<(select max_score from score_max where sc.cid=score_max.cid) group by cid ),↓
temp(cid,sid, max_score_count) ↓
as (select cid, sid, count(score) from pub.STUDENT_COURSE sc where score=(select max_score from score_max where sc.cid=score_max.cid)group by cid,sid),
score_max_count(cid, max_score_count) ↓
as (select cid, count(sid) from temp group by cid)↓
select cid, name, max_score, max_score2, max_score_count ↓
from pub.COURSE natural left outer join score_max natural full outer join score_max2 natural full outer join score_max_count
```

这是比较繁琐的一道题，我觉得有必要分析一下我的思路。

我首先使用 with 建立了四个临时关系，第一个关系是计算出每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的最大分数得到，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，计算每个课程 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

结果:

201600262022	张童	2 - 6	查询6	2019-04-14	按时完成	(NULL)	2019-03-30 21:23:44	2.0 / 2.0
--------------	----	-------	-----	------------	------	--------	---------------------	-----------

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

sql 语句:

```
create table test2_07 as↓
select sid,name from pub.STUDENT ↓
where name not like '李%' and name not like '王%' and name not like '张%'
```

结果:

201600262022	张童	2 - 7	查询7	2019-04-14	按时完成	(NULL)	2019-03-30 19:11:00	2.0 / 2.0
--------------	----	-------	-----	------------	------	--------	---------------------	-----------

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

sql 语句:

```
create table test2_08 as↓
with second_name_table(second_name) as(select substr(name,1,1)from pub.STUDENT)↓
select second_name, count(second_name) p_count from second_name_table group by second_name
```

结果:

201600262022	张童	2 - 8	查询8	2019-04-14	按时完成	(NULL)	2019-03-30 20:07:44	2.0 / 2.0
--------------	----	-------	-----	------------	------	--------	---------------------	-----------

9. 查询选修了 300003 号课程的学生的 sid、name、score

sql 语句:

```
create table test2_09 as↓
select sid, name, score from pub.STUDENT natural join pub.STUDENT_COURSE where cid='300003'
```

结果:

201600262022	张童	2 - 9	查询9	2019-04-14	按时完成	(NULL)	2019-03-30 19:20:20	2.0 / 2.0
--------------	----	-------	-----	------------	------	--------	---------------------	-----------

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）

sql 语句:

```
create table test2_10 as
select sid,name from pub.STUDENT where sid in
(select sid from pub.STUDENT_COURSE where score<60 group by sid,cid having count(cid)>=2)
```

结果:

201600262022	张童	2 - 10	查询10	2019-04-14	按时完成	(NULL)	2019-03-30 19:52:52	2.0 / 2.0
--------------	----	--------	------	------------	------	--------	---------------------	-----------

收获:

通过实验中的是个查询语句的联系,我对查询语句有了更深刻的认识,了解了许多课上没有讲过但很有用的操作,比如 substr、round 等。