

实验二

姓名：苏柏瑞
学号：201600301108
班级：泰山学堂
实验题目：检索查询

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。

实验题目

- 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)。

```
CREATE TABLE test2_01 AS SELECT sid, name FROM pub.student
WHERE sid NOT IN (SELECT sid FROM pub.student_course)
```

- 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。

```
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)
```

- 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。

```
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');
```

- 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。

```
--in的使用
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'));
```

- 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。通过下面方式实现列名定义:

```
create table test2_05 as select sid,name, (表达式) avg_score, (表达式) sum_score
from .....
```

```
--聚合函数的使用
-- 四舍五入函数的使用
CREATE TABLE test2_05 AS
SELECT sid, name, ROUND(avg(score), 0) avg_score, SUM(score) sum_score FR
OM pub.student NATURAL JOIN pub.STUDENT_course WHERE
age = 20 GROUP BY sid, name;
```

- 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。

```
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 sc1 WHERE sc1.score <
(SELECT max_score FROM score_max WHERE sc1.cid = score_max.cid) GROUP BY
cid),
tmp(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 B
Y cid, sid),
score_max_count(cid, max_score_count)
AS (SELECT cid, count(sid) FROM tmp GROUP BY cid)
SELECT cid, name, max_score, max_score2, max_score_count
FROM pub.course NATURAL LEFT JOIN score_max NATURAL FULL JOIN score_max2
NATURAL FULL JOIN score_max_count
```

- 查询所有不姓张、不姓李、也不姓王的学生的学号sid、姓名name

```
--not like
--正则匹配
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%';
```

- 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

```
-- with 语法建立新的table
-- substr的用法
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;
```

- 查询选修了300003号课程的学生们的sid、name、score

```
CREATE TABLE test2_09 AS
SELECT sid, name, score FROM pub.student NATURAL JOIN pub.student_course
WHERE cid = '300003'
```

- 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）。

```
-- having 的用法
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)
```