# 学号班级姓名

2019212212113

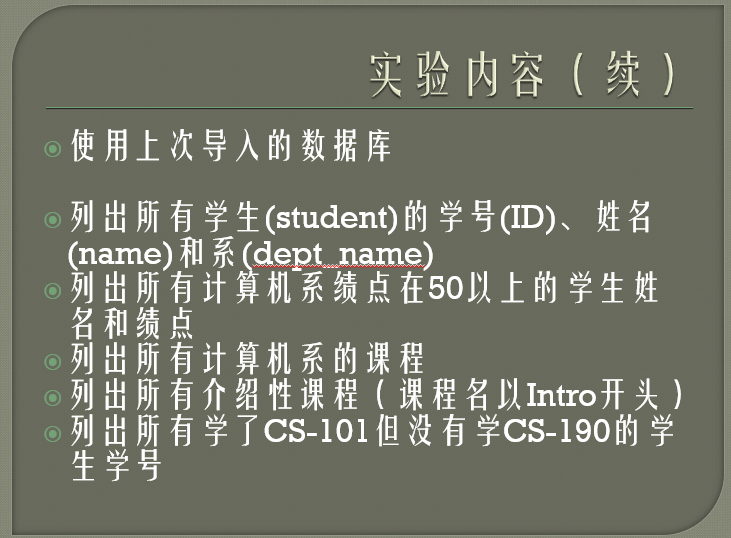
计算机193班

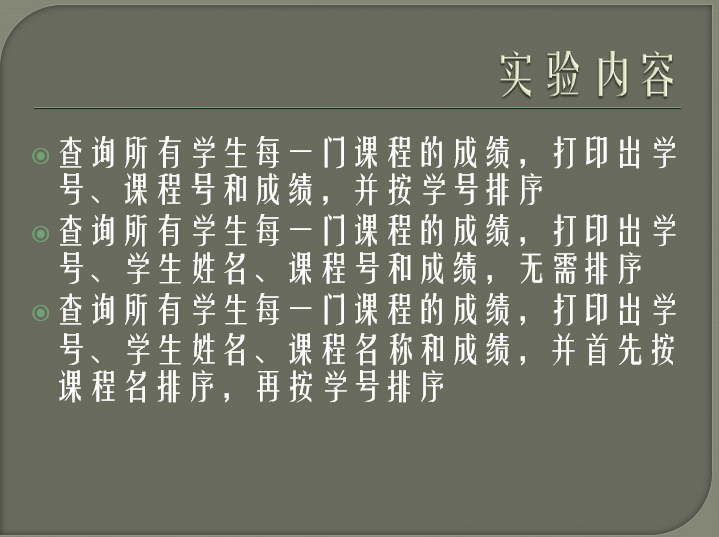
覃旭日

# 实验名称

SQL中的简单查询

# 实验内容





# 四、实验步骤

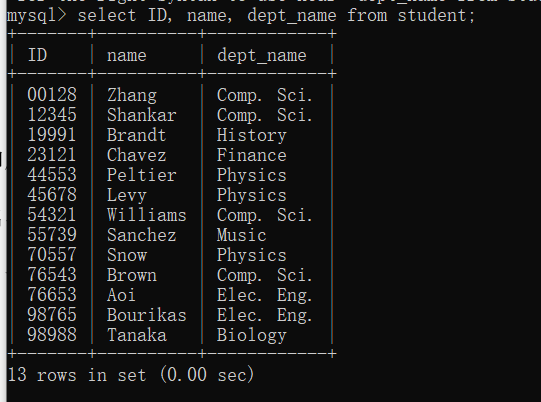
1、导入数据库

source DDL.sql

source smallRelationsInsertFile.sql

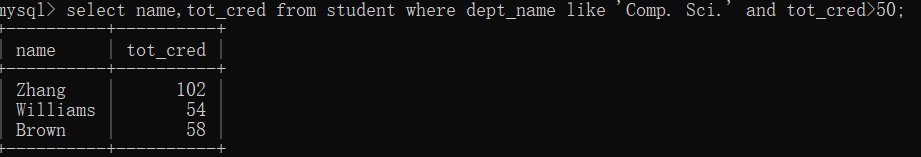
2、列出所有学生的的学号、姓名和系

select ID, name, dept\_name from student;

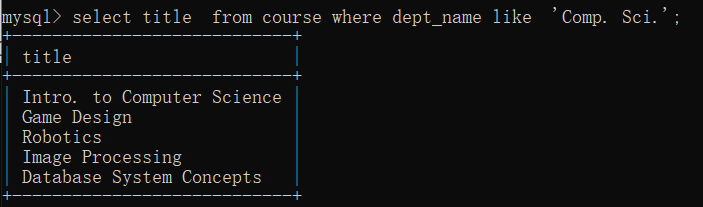


3、列出所有计算机系绩点在50以上的学生姓名和绩点

select name,tot\_cred from student where dept\_name like 'Comp. Sci.' and tot\_cred>50;

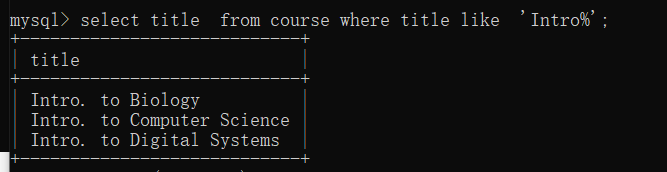


4、列出所有计算机系的课程

select title from course where dept\_name like 'Comp. Sci.'; 

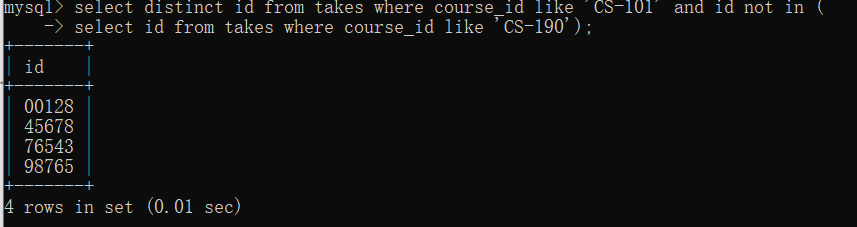
5、列出所有介绍性课程（课程名以Intro开头）

select title from course where title like 'Intro%';



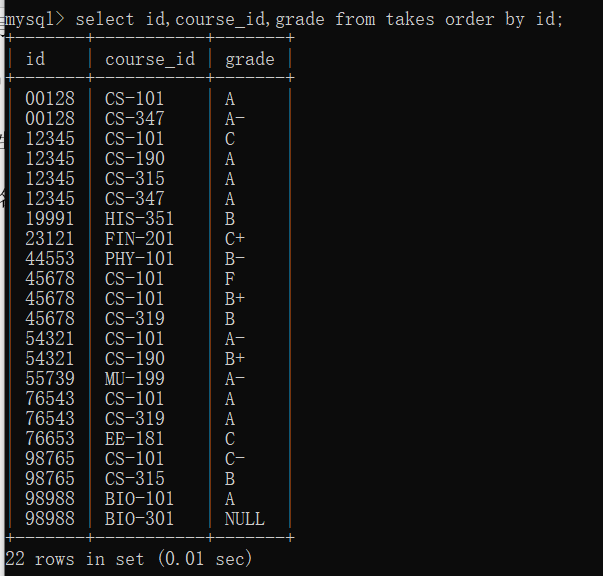
6、列出所有学了CS-101但没有学CS-190的学生学号

select distinct id from takes where course\_id like 'CS-101' and id not in (select id from takes where course\_id like 'CS-190');



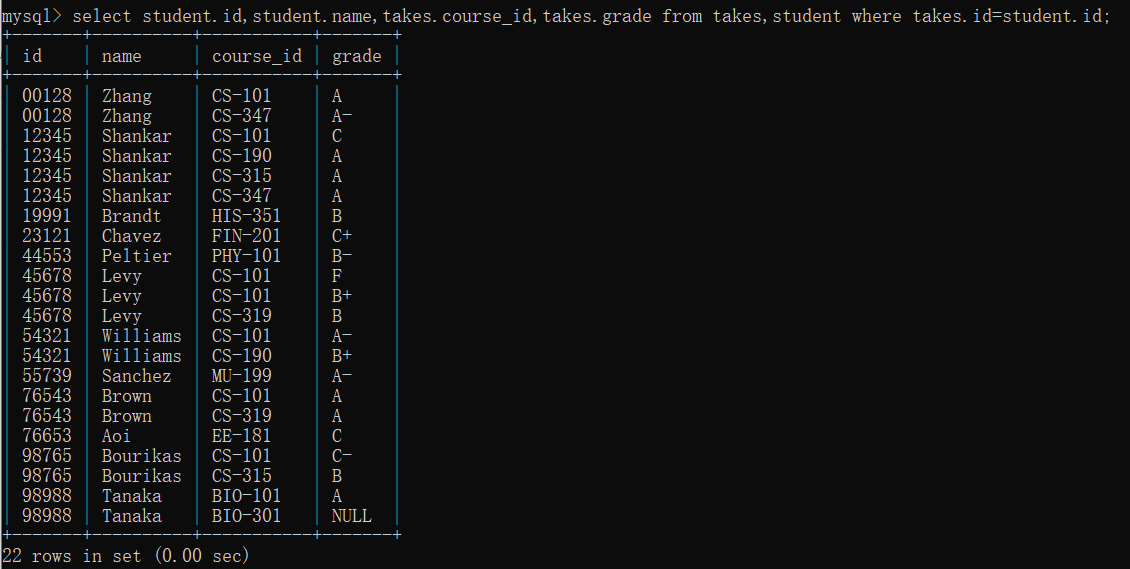
7、查询所有学生每一门课程的成绩，打印出学号、课程号和成绩，并按学号排序

select id,course\_id,grade from takes order by id;



8、查询所有学生每一门课程的成绩，打印出学号、学生姓名、课程号和成绩，无需排序

select student.id,student.name,takes.course\_id,takes.grade from takes,student where takes.id=student.id;



9、查询所有学生每一门课程的成绩，打印出学号、学生姓名、课程名称和成绩，并首先按课程名排序，再按学号排序

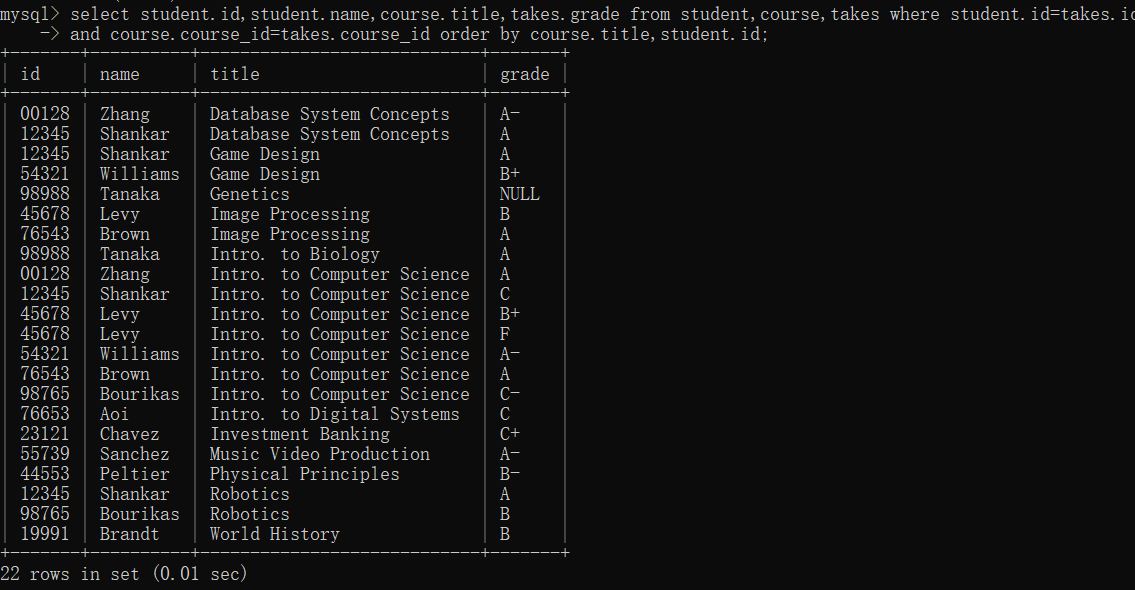
Select student.id,student.name,course.title,takes.grade

from student,course,takes

where student.id=takes.id

and course.course\_id=takes.course\_id

order by course.title,student.id;



# 一、班级、学号、姓名

班级：计科193

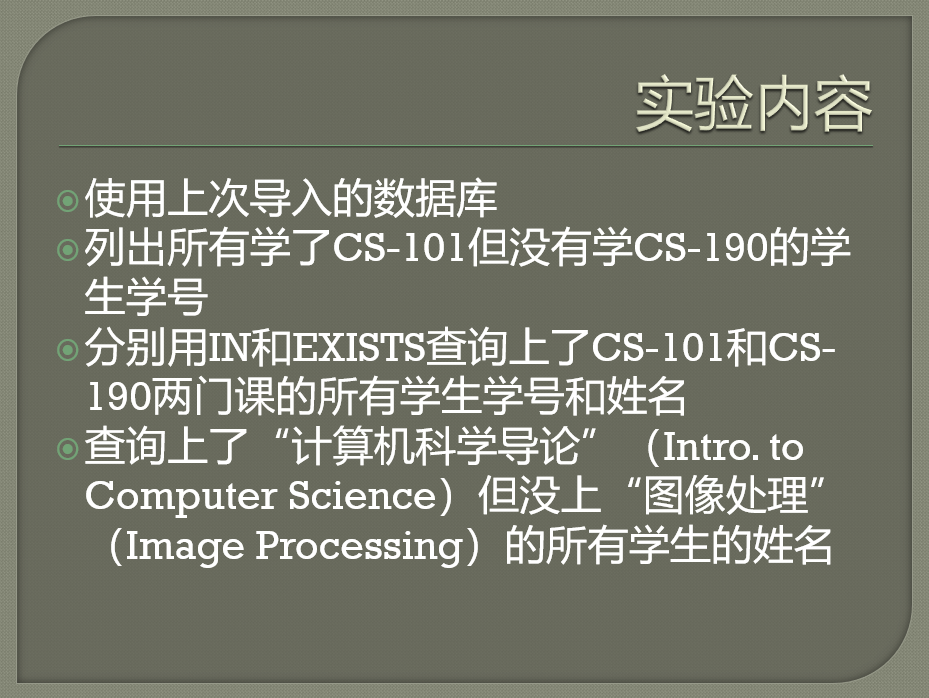
学号：2019212212112

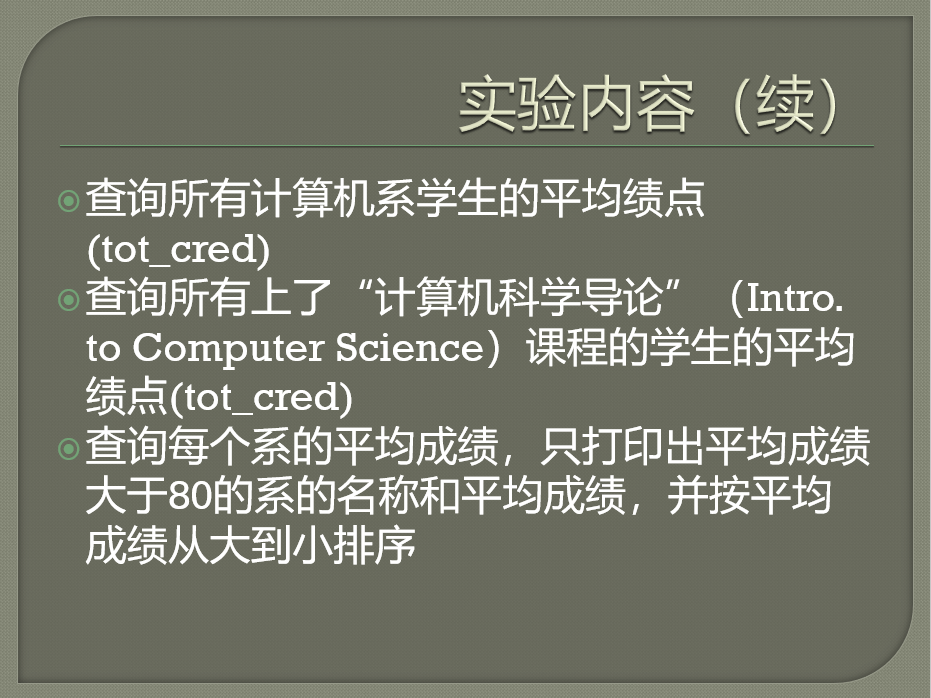
姓名：罗千尧

# 二、实验名称

实验名称：多表查询与聚集函数的应用

# 三、实验内容





# 四、实验记录

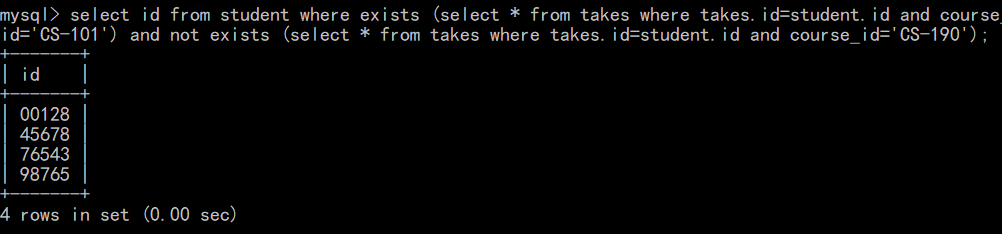
**1、使用上次导入的数据库**

命令：use dblab;



2、列出所有学了CS-101但没有学CS-190的学生学号

命令：select id from student where exists (select \* from takes where takes.id=student.id and course\_id='CS-101') and not exists (select \* from takes where takes.id=student.id and course\_id='CS-190');

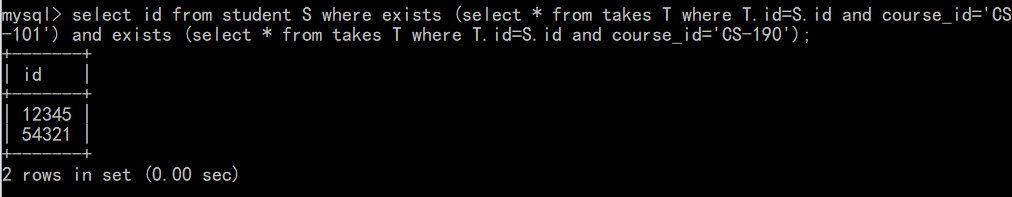


3、分别用IN和EXISTS查询上了CS-101和CS-190两门课的所有学生学号和姓名

IN查询命令：select id from student S where id in (select id from takes T where T.id=S.id and course\_id='CS-101') and id in (select id from takes T where T.id=S.id and course\_id='CS-190');

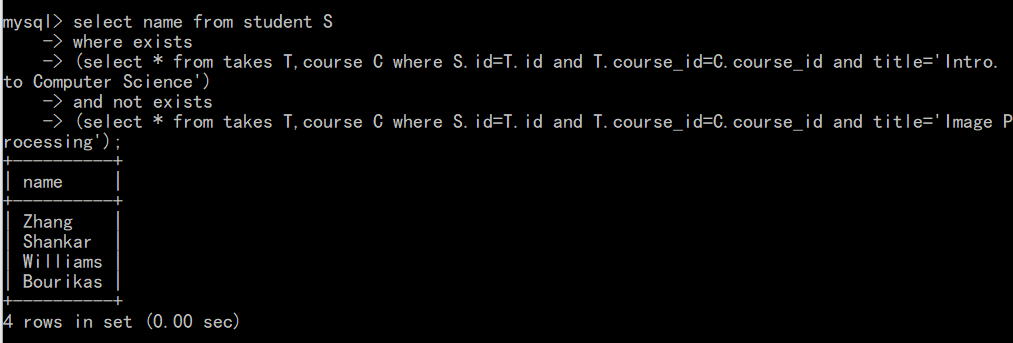


EXISTS查询命令：select id from student S where exists (select \* from takes T where T.id=S.id and course\_id='CS-101') and exists (select \* from takes T where T.id=S.id and course\_id='CS-190');



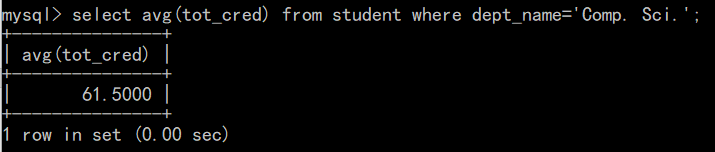
4、查询上了“计算机科学导论”（Intro. to Computer Science）但没有上“图像处理”（Image Processing）的所有学生的姓名

命令：select name from student S where exists (select \* from takes T,course C where S.id=T.id and T.course\_id=C.course\_id and title='Intro. to Computer Science') and not exists (select \* from takes T,course C where S.id=T.id and T.course\_id=C.course\_id and title='Image Processing');



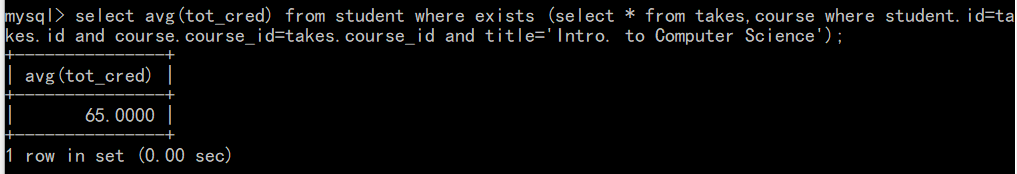
5、查询所有计算机系学生的平均绩点（tot\_cred）

命令：select avg(tot\_cred) from student where dept\_name='Comp. Sci.';



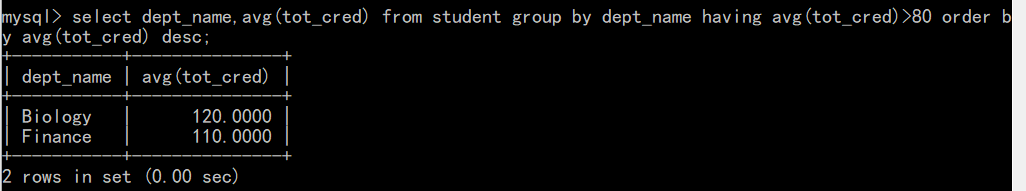
6、查询所有上了“计算机科学导论”（Intro. to Computer Science）课程的学生的平均绩点（tot\_cred）

命令：select avg(tot\_cred) from student where exists (select \* from takes,course where student.id=takes.id and course.course\_id=takes.course\_id and title='Intro. to Computer Science');



7、查询每个系的平均成绩，只打印出平均成绩大于80的系的名称和平均成绩，并按平均成绩从大到小排序

命令：select dept\_name,avg(tot\_cred) from student group by dept\_name having avg(tot\_cred)>80 order by avg(tot\_cred) desc;



# 一、班级、学号、姓名

班级：计科193

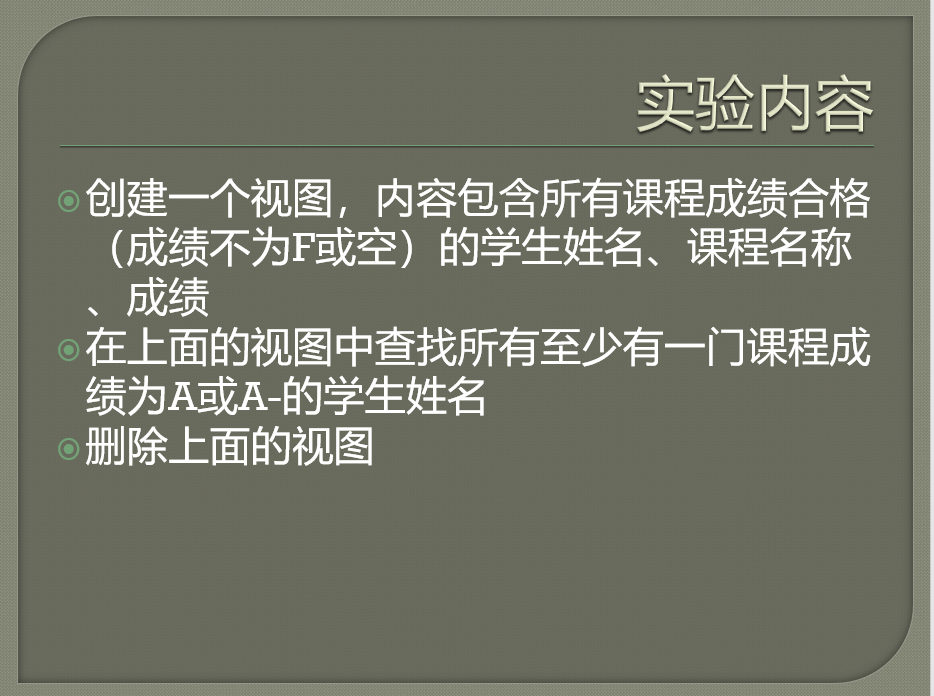
学号：2019212212113

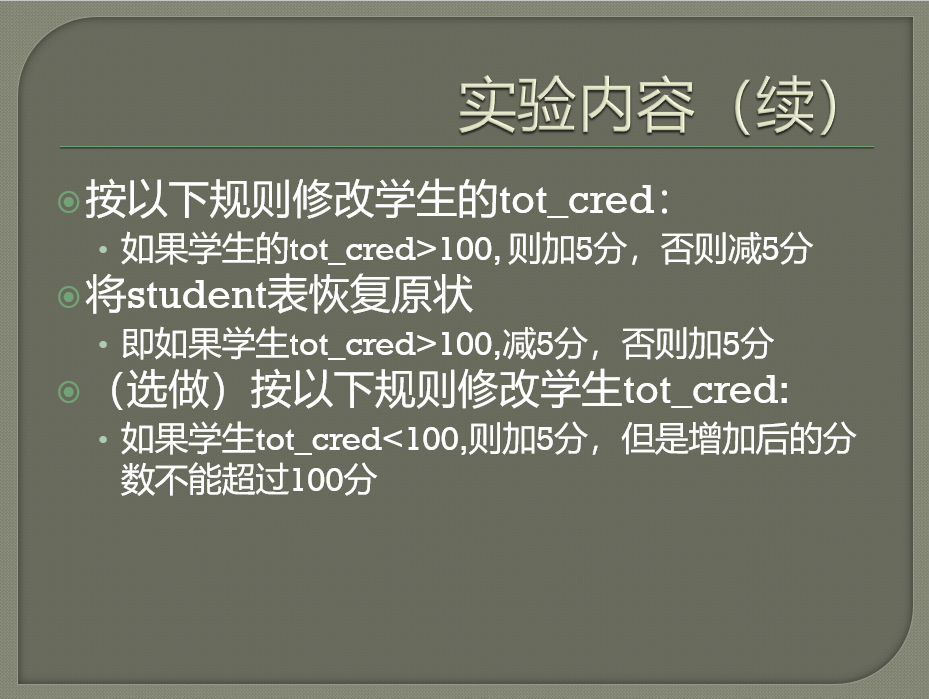
姓名：覃旭日

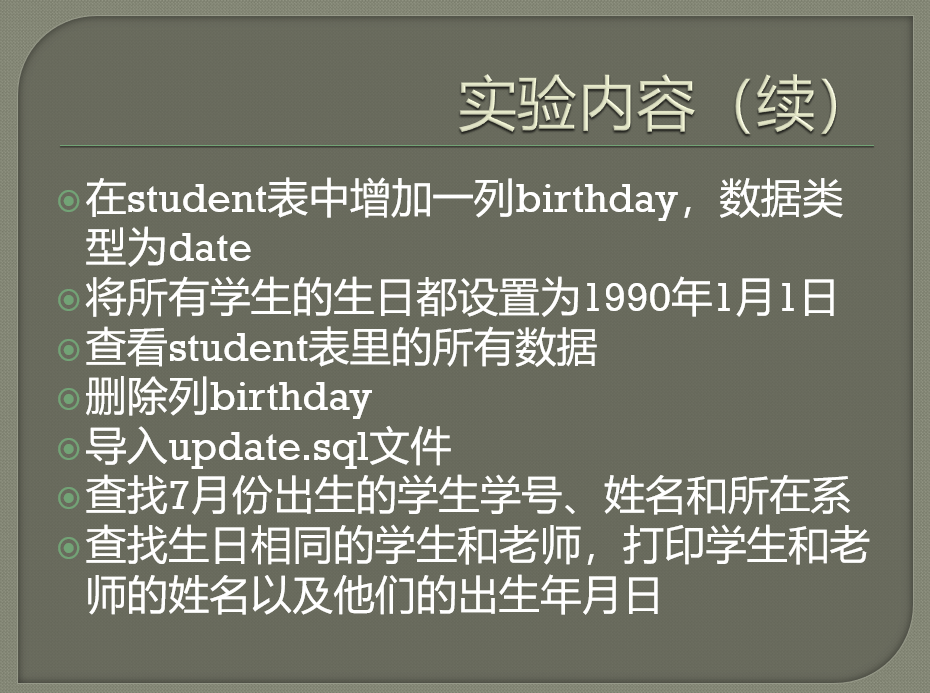
# 二、实验名称

实验名称：多表连接查询与数据库修改

# 三、实验内容







# 四、实验记录

**1、创建一个视图，内容包含所有课程成绩合格（成绩不为F或空）的学生姓名、课程名称、成绩**

CREATE VIEW VIEW AS (

SELECT

name,

title,

grade

FROM

student,

takes,

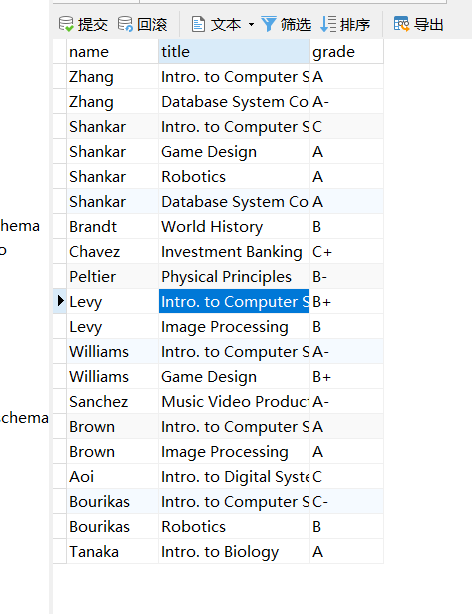
course

WHERE

student.id = takes.id

AND course.course\_id = takes.course\_id

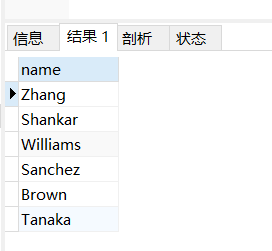
AND NOT ( grade = 'F' OR grade IS NULL ));



**2、在上面的视图中查找所有至少一门课程成绩为A或A-的学生姓名**

select distinct name from view

where grade like "A%";



**3、删除上面的视图**

drop view view;



4、按以下规则修改学生的tot\_cred: 如果学生的tot\_cred>100，则加5分，否则减5分

update student

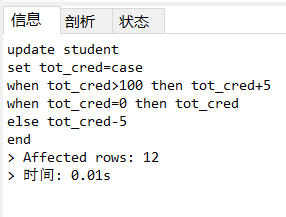
set tot\_cred=case

when tot\_cred>100 then tot\_cred+5

when tot\_cred=0 then tot\_cred

else tot\_cred-5

end;





5、将student表恢复原状（即如果学生tot\_cred>100，则减5分，否则加5分）

update student

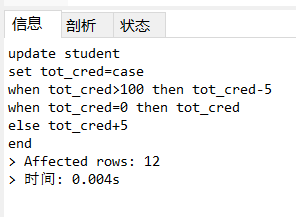
set tot\_cred=case

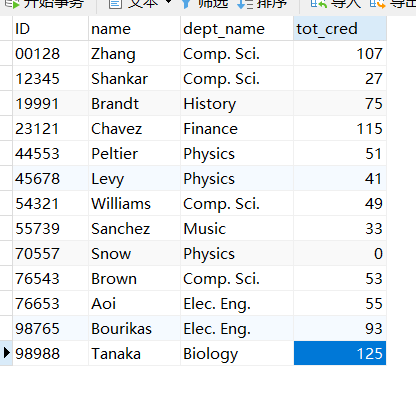
when tot\_cred>100 then tot\_cred-5

when tot\_cred=0 then tot\_cred

else tot\_cred+5

end;





6、按照以下规则修改学生tot\_cred: 如果学生tot\_cred<100，则加5分，但是增加后的分数不能超过100分

update student

set tot\_cred=case

when tot\_cred>100 and tot\_cred+5>100

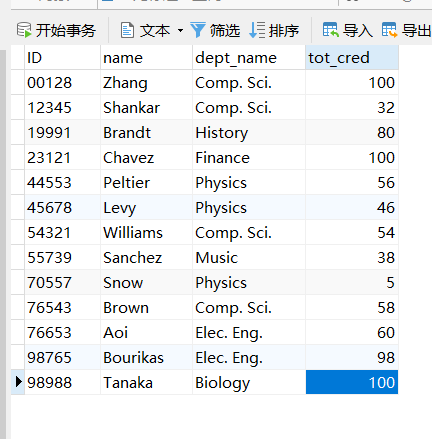
then 100

when tot\_cred<100 then tot\_cred+5

else tot\_cred

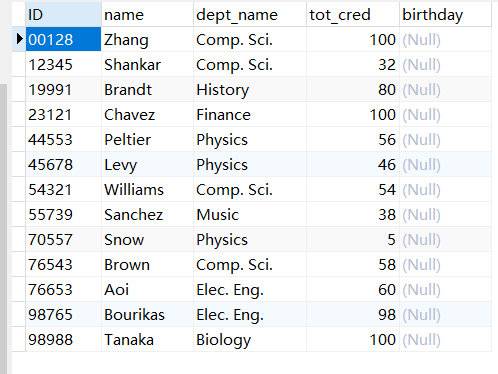
end;





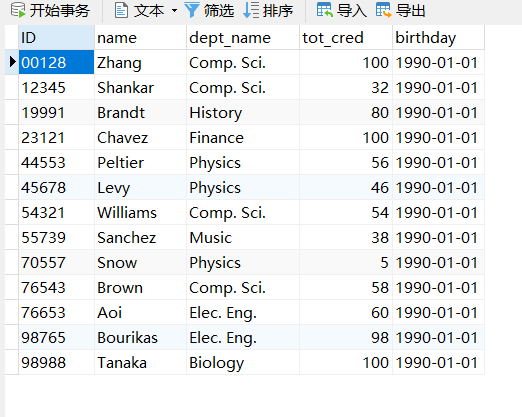
**7、在student表中增加一列birthday，数据类型为date**

ALTER TABLE student ADD COLUMN birthday date;



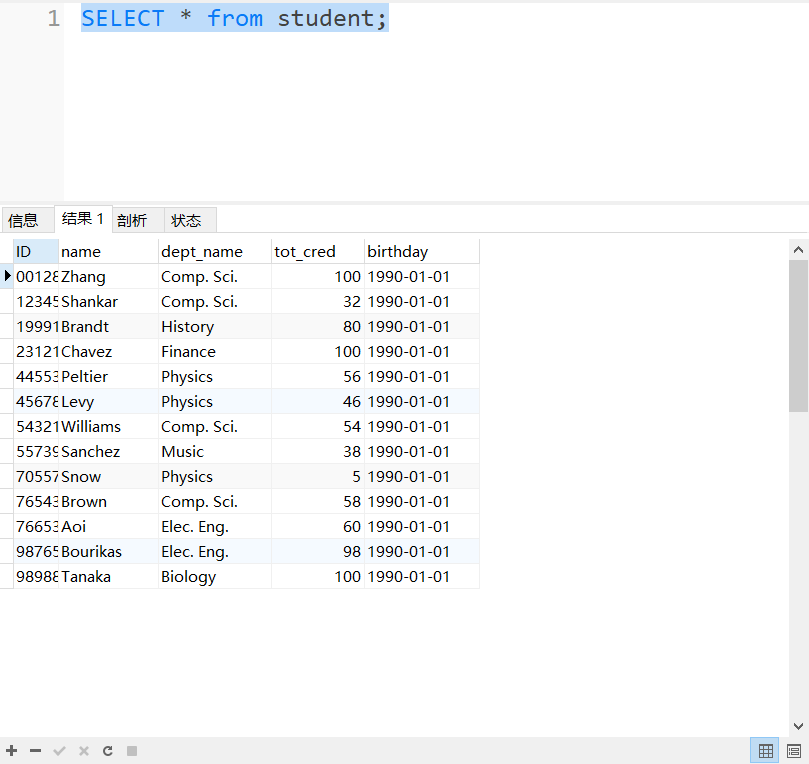
8、将所有学生的生日都设置为1990年1月1日

UPDATE student set birthday='1990-1-1';



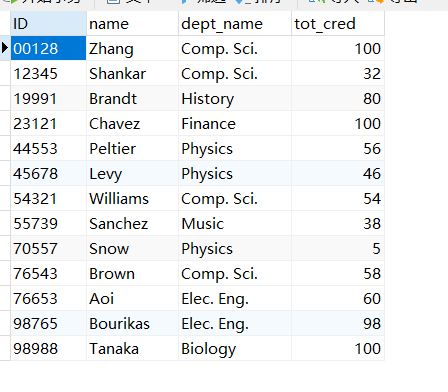
9、查看student表里的所有数据

SELECT \* from student;



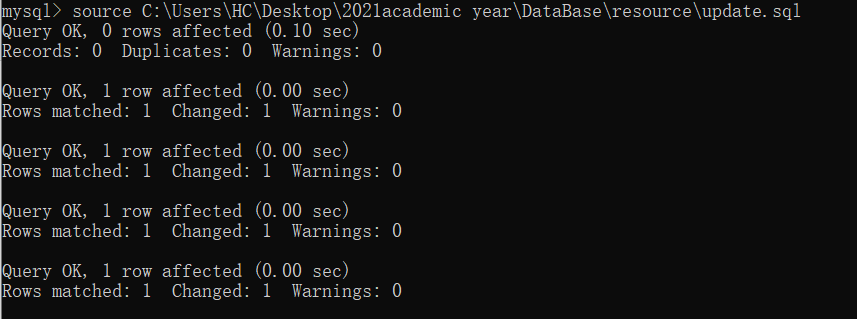
10、删除列birthday

alter table student drop COLUMN birthday;



11、导入update.sql文件

source update.sql



12、查找7月份出生的学生学号、姓名和所在系

SELECT

id,

NAME,

dept\_name

FROM

student

WHERE

MONTH ( birthday )= 7;



# 一、班级、学号、姓名

班级：计科193

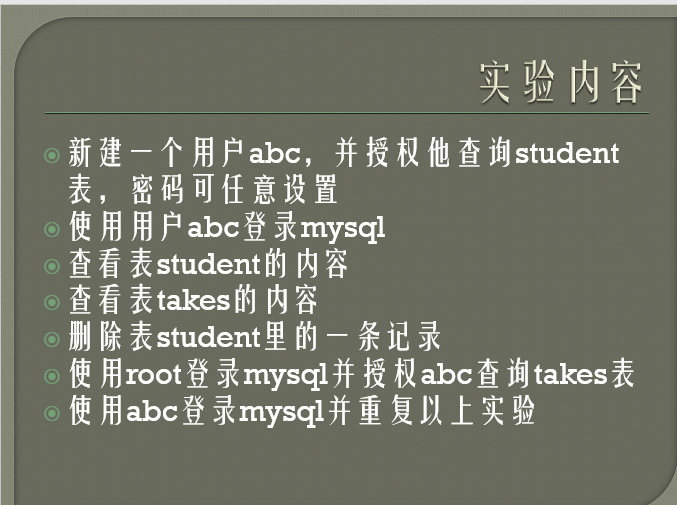
学号：2019212212113

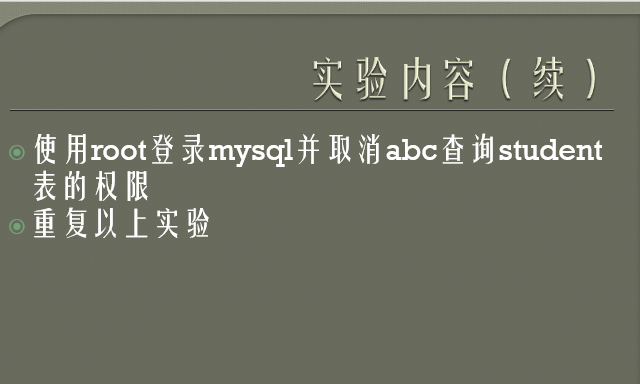
姓名：覃旭日

# 二、实验名称

实验名称：权限管理

# 三、实验内容

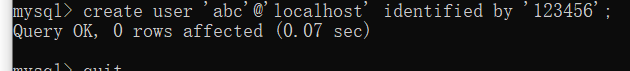




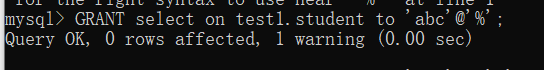
# 四、实验记录

1、新建一个用户abc，并授权他查询student表，密码可任意设置

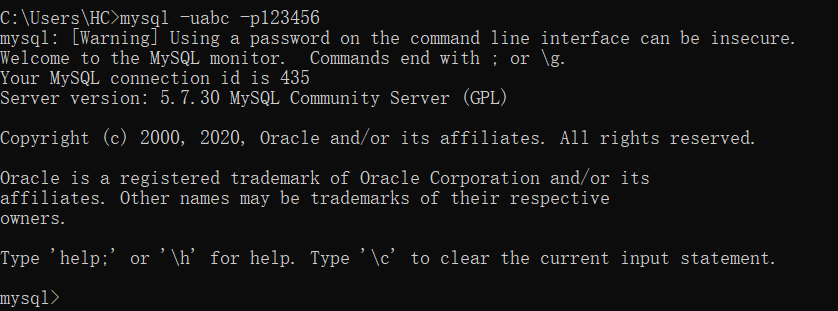
create user 'abc'@'localhost' identified by '123456';



GRANT select on test1.student to 'abc'@'%';

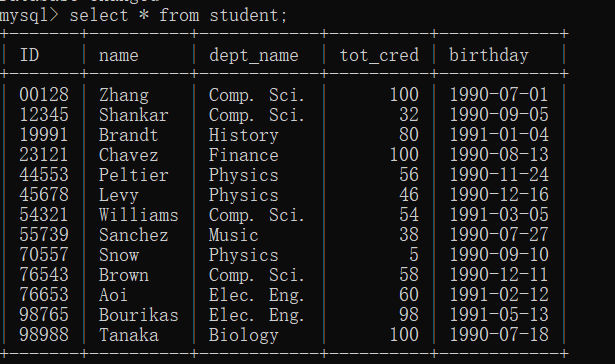


2、使用用户abc登录mysql

mysql -uabc -p123456

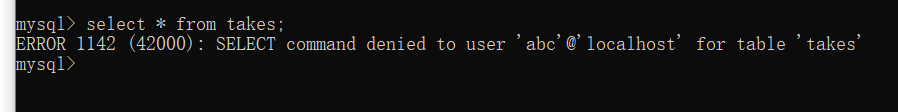
3、查看表student的内容

select \* from student;



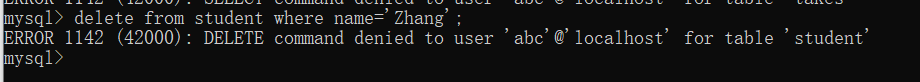
4、查看表takes的内容

select \* from takes;



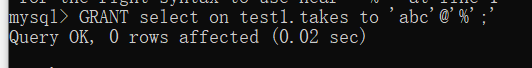
5、删除表student里的一条记录

delete from student where name='Zhang';



6、使用root登录mysql并授权abc查询takes表

GRANT select on test1.takes to 'abc'@'%';



7、使用root登录mysql并取消abc查询student表的权限

revoke select on test1.student from 'abc'@'%';

