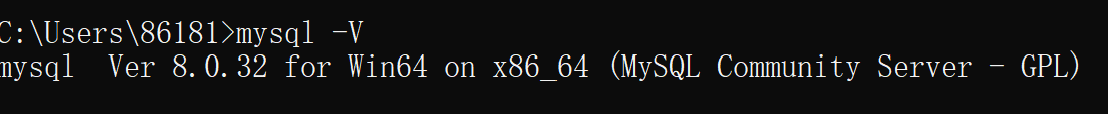
**SQL实验报告**

**姓名：陈鸿绪 学号：PB21000224 日期：5.1**

1. MySQL配置结果：版本显示

b) 数据库中基本表的建立：

create table course(CNO char(8) primary key,

NAME VARCHAR(40) NOT NULL,

TNO CHAR(7) NOT NULL)

create table student(SNO char(11) primary key,

NAME varchar(4) not null,

GENDER char(6) not null,

BIRTHDAY datetime not null,

DEPART int not null);create table teacher(TNO CHAR(7) primary key,

NAME VARCHAR(4) not null,

GENDER char(6) not null,

BIRTHDAY datetime not null,

POSITION char(25) not null,

DEPART INT not null)

create table score(SNO char(11) not null,

CNO char(8) not null,

DEGREE int)

create table score(SNO char(11),

CNO char(8) ,

DEGREE int,

foreign key(SNO) references student(SNO),

foreign key(CNO) references course(CNO))

c) 数据导入表中：

LOAD DATA INFILE 'Course.csv'

INTO TABLE course

FIELDS TERMINATED BY ',';

LOAD DATA INFILE 'Student.csv'

INTO TABLE student

FIELDS TERMINATED BY ',';

LOAD DATA INFILE 'Score.csv'

INTO TABLE score

FIELDS TERMINATED BY ',';

LOAD DATA INFILE 'Teacher.csv'

INTO TABLE teacher

FIELDS TERMINATED BY ',';

1. alter table lab01.student add AGE int not null;

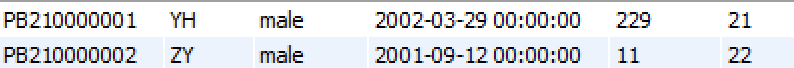




1. update student

set AGE=2023-year(BIRTHDAY);

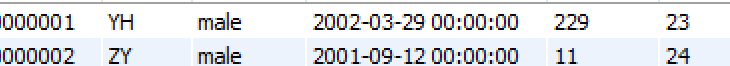




1. update student

set AGE=AGE+2;





1. alter table student

modify AGE CHAR(2);



1. alter table student

drop AGE;





1. create table teacher\_course(TNO CHAR(7),

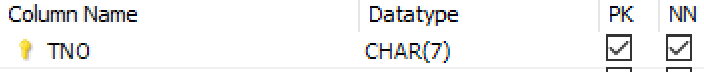
NUM\_COURSE INT);



1. alter table teacher\_course

add primary key(TNO);





1. insert into teacher\_course(TNO,NUM\_COURSE)

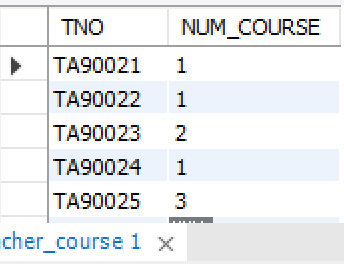
select teacher.TNO,temp.NUM\_COURSE

from teacher left outer join

(select TNO,count(CNO) from course group by TNO) as temp(TNO,NUM\_COURSE)

on teacher.TNO=temp.TNO;





1. delete from teacher\_course where NUM\_COURSE is null;



1. drop table teacher\_course;



1. insert into student(SNO,NAME,GENDER,BIRTHDAY,DEPART)

values('PB21000224','CHX','male','2004-2-24 00:00:00',229)

insert into student(SNO,NAME,GENDER,BIRTHDAY,DEPART)

values('PB21000227','jjy','male','2004-3-3 00:00:00',229)

insert into student(SNO,NAME,GENDER,BIRTHDAY,DEPART)

values('PB21000211','py','male','2004-12-24 00:00:00',229)

insert into score(SNO,CNO,DEGREE)

values('PB21000224','20230402',95);

insert into score(SNO,CNO,DEGREE)

values('PB21000224','20230410',97);

insert into score(SNO,CNO,DEGREE)

values('PB21000224','20230412',99);









1. delete from score

where score.SNO='PB21000224' and

score.DEGREE<=ALL(select temp.DEGREE

from (select DEGREE

from score

where SNO='PB21000224')

as temp(DEGREE));



1. create index NAME\_INDEX ON course(NAME);



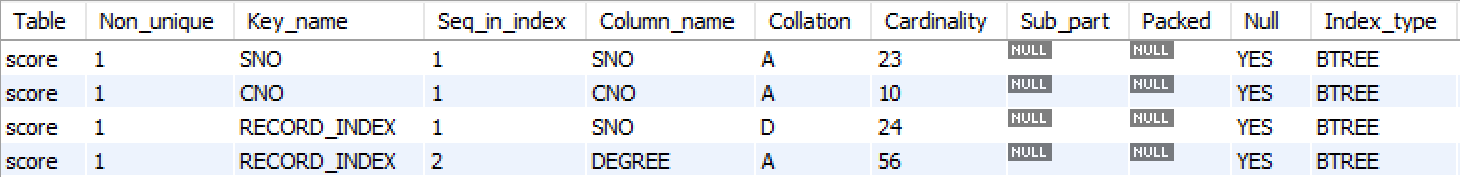
1. create unique index TNO\_INDEX ON teacher(TNO);



1. create index RECORD\_INDEX on score(SNO DESC,DEGREE ASC)



1. show index from score;



1. drop index TNO\_INDEX on teacher;



1. select SNO,NAME

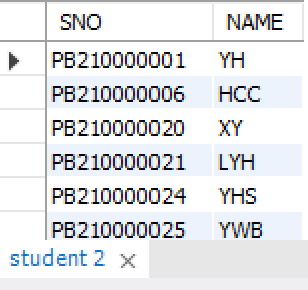
from student

where DEPART=

(select DEPART

from student

where SNO='PB21000224')



1. select SNO,NAME

from student

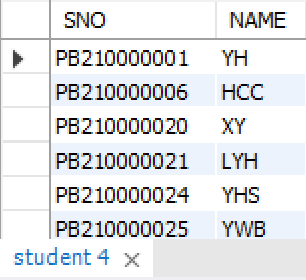
where DEPART=

(select DEPART

from student

where SNO='PB21000224')

and SNO<>'PB21000224';



1. select SNO,NAME

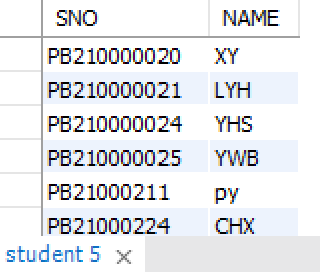
from student

where DEPART=

(select DEPART

from student

where NAME='jjy');



1. select SNO,NAME

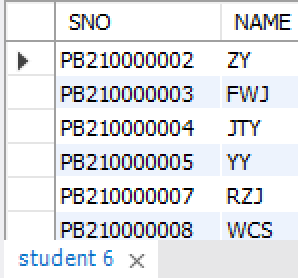
from student

where DEPART not in

(select DEPART

from student

where SNO='PB21000227' or SNO='PB21000211')



1. select TNO,NAME

from teacher

where TNO in

(select TNO

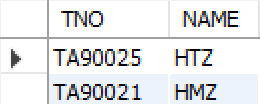
from course

where CNO in

(select CNO

from score

where SNO='PB21000224'));

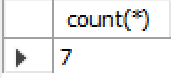


1. select count(\*)

from (select TNO

from teacher

where DEPART=11 or DEPART=229) as temp(TNO);



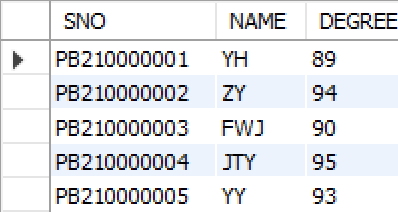
1. select score.SNO,NAME,DEGREE

from score,student

where CNO in

(select CNO from course where NAME='DB\_Design')

and DEGREE>=89 and score.SNO=student.SNO;



1. select distinct student.SNO,NAME

from student,score

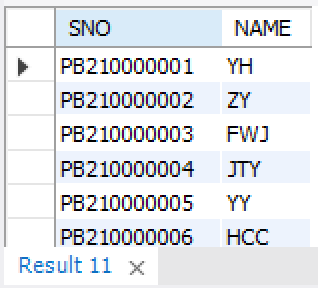
where (score.CNO in

(select CNO

from teacher

where NAME='ZDH')

and score.SNO=student.SNO);



1. select SNO, DEGREE

from score

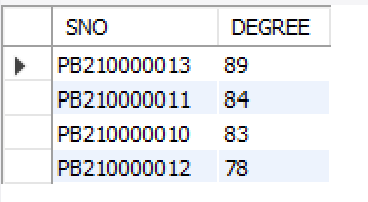
where CNO in

(select CNO

from course

where NAME='operating\_system')

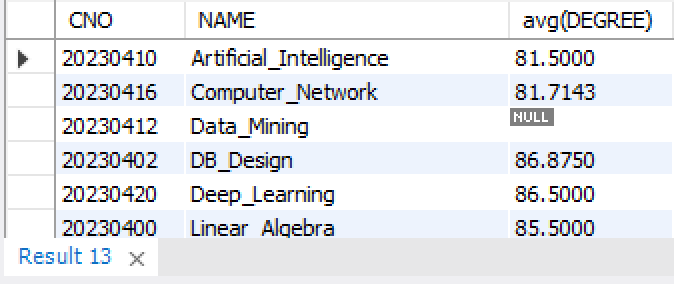
order by DEGREE desc;



1. select course.CNO,NAME,avg(DEGREE)

from course left join score on(course.CNO=score.CNO)

group by CNO;

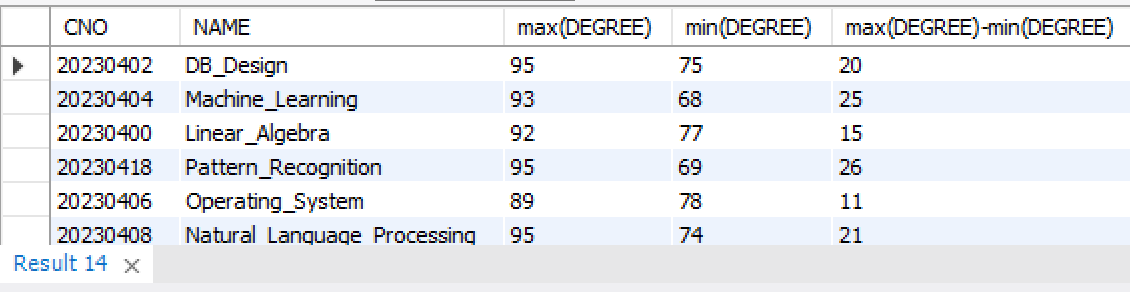


1. select course.CNO,NAME,max(DEGREE),min(DEGREE),max(DEGREE)-min(DEGREE)

from course,score

where course.CNO=score.CNO

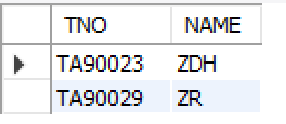
group by CNO;



1. select distinct teacher.TNO,teacher.NAME

from teacher,score,course

where course.CNO=score.CNO and DEGREE<72 and course.TNO=teacher.TNO;



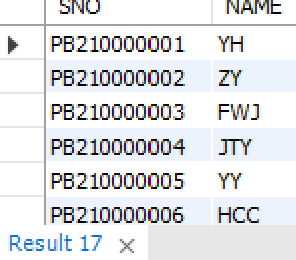
1. select student.SNO,NAME

from student,score

where student.SNO=score.SNO

group by SNO

having count(CNO)>=2

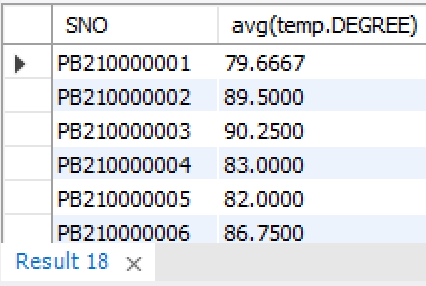


1. select temp.SNO,avg(temp.DEGREE)

from (select student.SNO,DEGREE

from student left join score on(student.SNO=score.SNO)) as temp(SNO,DEGREE)

group by temp.SNO;



1. select temps.DEPART,count(temps.SNO),avg(temps.DEGREE)

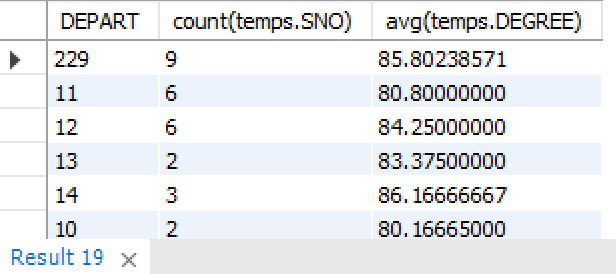
from (select temp.DEPART,temp.SNO,avg(temp.DEGREE)

from (select DEPART,student.SNO,DEGREE

from student left join score on(student.SNO=score.SNO)) as temp(DEPART,SNO,DEGREE)

group by temp.SNO) as temps(DEPART,SNO,DEGREE)

group by DEPART



1. select NAME

from student x

where not exists(

select \*

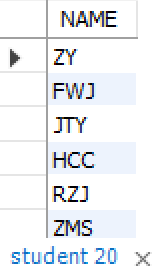
from score

where score.SNO=x.SNO and score.CNO in(

select CNO

from course

where NAME='Data\_Mining'))

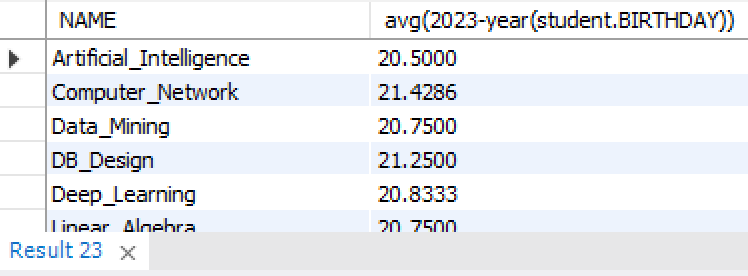


1. select course.NAME,avg(2023-year(student.BIRTHDAY))

from course left join score on(score.CNO=course.CNO)

left join student on(score.SNO=student.SNO)

group by course.NAME



1. select student.SNO,student.NAME

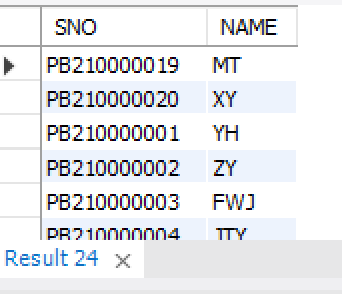
from student,score

where student.SNO=score.SNO and score.CNO in (

select CNO

from course

where NAME like "%Computer%");

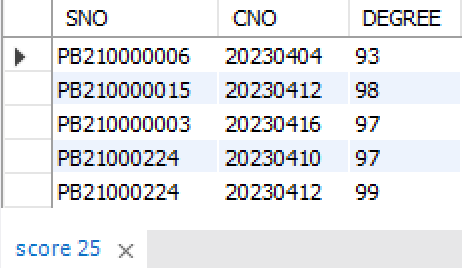


1. select SNO,CNO,DEGREE

from score x

where x.DEGREE-12 >

(select avg(DEGREE) from score y where y.CNO=x.CNO group by CNO) ;



1. create view db\_female\_student(SNO,NAME,GENDER,BIRTHDAY,DEPART)

as select \* from student where GENDER='female'

with check option



1. update db\_female\_student

set NAME='CHX'

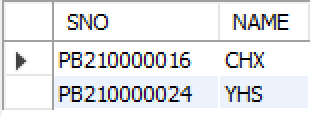
where SNO='PB210000016'



1. select SNO,NAME

from db\_female\_student

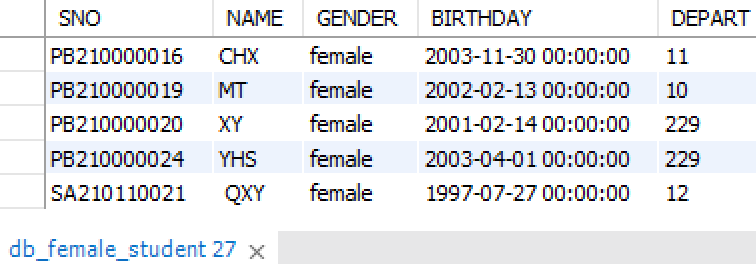
where 2023-year(BIRTHDAY)<21;



1. insert into db\_female\_student

values('SA210110021',' QXY','female','1997-07-27 00:00:00','12');

select \* from db\_female\_student



1. insert into db\_female\_student

values('SA210110023',' DPC','male','1997-04-27 00:00:00','11');



1. drop view db\_female\_student



1. create table teacher\_salary(

TNO char(7) primary key,

SAL float)



1. DELIMITER //

create trigger TS\_T1

BEFORE insert on teacher\_salary

for each row

if new.TNO not in (select TNO from teacher) then

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "have no such TNO";

END IF//

DELIMITER ;

DELIMITER //

create trigger TS\_T2

BEFORE update on teacher\_salary

for each row

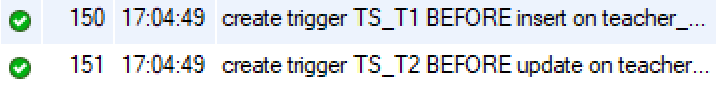
if new.TNO not in (select TNO from teacher) then

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "have no such TNO";

END IF//

DELIMITER ;



insert into teacher\_salary

values("TA9002",7000);

insert into teacher\_salary

values("TA90022",7000);

update teacher\_salary

set TNO="TA9002"

where TNO="TA90022"







1. DELIMITER //

create trigger TS\_T3

BEFORE update on teacher\_salary

for each row

if (new.SAL < 4000 and (select position from teacher where new.TNO=teacher.TNO)="Instructor") or

(new.SAL < 5000 and (select position from teacher where new.TNO=teacher.TNO)="Associate Professor") or

(new.SAL < 6000 and (select position from teacher where new.TNO=teacher.TNO)="Professor") then

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "the salary is not correct";

END IF//

DELIMITER ;

DELIMITER //

create trigger TS\_T4

BEFORE insert on teacher\_salary

for each row

if (new.SAL < 4000 and (select position from teacher where new.TNO=teacher.TNO)="Instructor") or

(new.SAL < 5000 and (select position from teacher where new.TNO=teacher.TNO)="Associate Professor") or

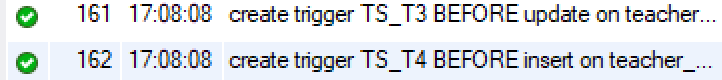
(new.SAL < 6000 and (select position from teacher where new.TNO=teacher.TNO)="Professor") then

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "the salary is not correct";

END IF//

DELIMITER ;



insert into teacher\_salary

values("TA90023",2000)



insert into teacher\_salary

values("TA90023",5000);



update teacher\_salary

set SAL="1000"

where TNO="TA90023"

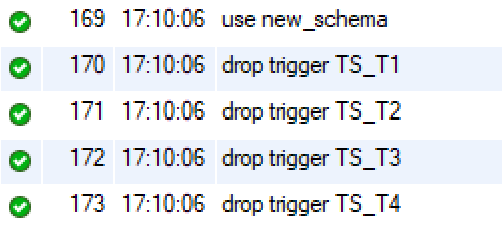


1. drop trigger TS\_T1;

drop trigger TS\_T2;

drop trigger TS\_T3;

drop trigger TS\_T4;



1. update score

set DEGREE=NULL

where CNO in

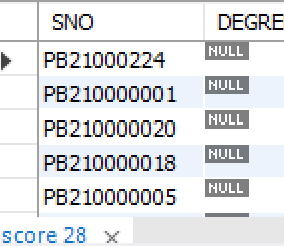
(select CNO from course where score.CNO=course.CNO and course.NAME="Data\_Mining");



select SNO,DEGREE

from score

order by DEGREE ASC;



null不参与比较大小，但是在数据排序的时候默认排在最小值的前面

1. 查看比DB\_DESIGN课程平均分高的人

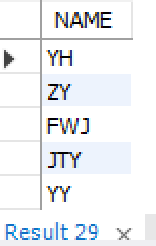
select student.NAME

from score x,student

where student.SNO=x.SNO and

(x.DEGREE > (select avg(DEGREE) from score y,course where course.NAME="DB\_DESIGN" and course.CNO=y.CNO))

and (x.CNO in (select course.CNO from score z,course where course.NAME="DB\_DESIGN" and course.CNO=z.CNO));



1. 将DB\_DESIGN课程的成绩由高到低排出一个试图，随后删除

create view DB\_DESIGN\_SCORE(SNO,DEGREE)

as

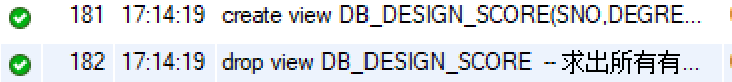
select score.SNO, score.DEGREE

from score,course

where score.CNO=course.CNO and course.NAME="DB\_DESIGN"

order by score.DEGREE desc;

drop view DB\_DESIGN\_SCORE



1. 求出所有有成绩的同学平均成绩排序,只显示学号和平均成绩

select SNO,avg(score.DEGREE) avg\_degree

from score

group by SNO

order by avg\_degree desc

