SQL 实验报告

姓名：陈鸿绪　学号：PB21000224　日期：5.1

a） MySQL 配置结果：版本显示

```
C:\Users\86181>mysql -V
mysql  Ver 8.0.32 for Win64 on x86_64 (MySQL Community Server - GPL)
```

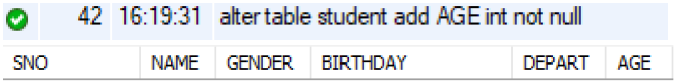
b) 数据库中基本表的建立：

```
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 ',';
```

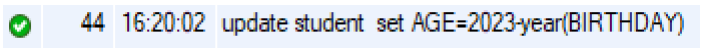
1. alter table lab01.student add AGE int not null;



42	16:19:31	alter table student add AGE int not null

SNO	NAME	GENDER	BIRTHDAY	DEPART	AGE

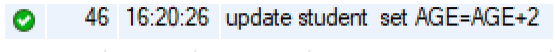
2. update student
   set AGE=2023-year(BIRTHDAY);



44	16:20:02	update student set AGE=2023-year(BIRTHDAY)

PB210000001	YH	male	2002-03-29 00:00:00	229	21
PB210000002	ZY	male	2001-09-12 00:00:00	11	22

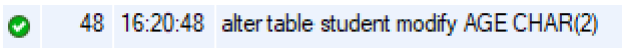
3. update student
   set AGE=AGE+2;



46	16:20:26	update student set AGE=AGE+2

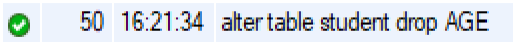
000001	YH	male	2002-03-29 00:00:00	229	23
000002	ZY	male	2001-09-12 00:00:00	11	24

4. alter table student
   modify AGE CHAR(2);



48	16:20:48	alter table student modify AGE CHAR(2)

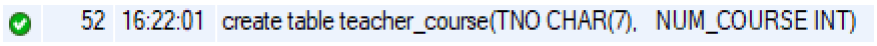
5. alter table student
   drop AGE;



50	16:21:34	alter table student drop AGE

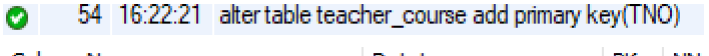
SNO	NAME	GENDER	BIRTHDAY	DEPART

6. create table teacher_course(TNO CHAR(7),
   NUM_COURSE INT);



52	16:22:01	create table teacher_course(TNO CHAR(7), NUM_COURSE INT)

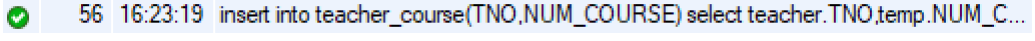
7. alter table teacher_course
   add primary key(TNO);



54	16:22:21	alter table teacher_course add primary key(TNO)

Column Name	Datatype	PK	NN
TNO	CHAR(7)	☑	☑

8. 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;



56	16:23:19	insert into teacher_course(TNO,NUM_COURSE) select teacher.TNO,temp.NUM_C...

	TNO	NUM_COURSE
▶	TA90021	1
	TA90022	1
	TA90023	2
	TA90024	1
	TA90025	3

cher_course 1 ×

9. delete from teacher_course where NUM_COURSE is null;

```
58  16:23:39  delete from teacher_course where NUM_COURSE is null
```

10. drop table teacher_course;

```
60  16:24:20  drop table teacher_course
```

11. 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);
```

```
62  16:27:07  insert into student(SNO,NAME,GENDER,BIRTHDAY,DEPART) values('PB210002...

64  16:27:28  insert into student(SNO,NAME,GENDER,BIRTHDAY,DEPART) values('PB210002...
66  16:27:48  insert into student(SNO,NAME,GENDER,BIRTHDAY,DEPART) values('PB210002...
70  16:28:08  insert into score(SNO,CNO,DEGREE) values('PB21000224','20230412',99)  1 row(s) affected
```

12. 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));

```
72  16:28:36  delete from score where score.SNO='PB21000224' and  score.DEGREE<=ALL(...  1 row(s) affected
```

13. create index NAME_INDEX ON course(NAME);

```
74  16:29:01  create index NAME_INDEX ON course(NAME)
```

14. create unique index TNO_INDEX ON teacher(TNO);

```
76  16:29:19  create unique index TNO_INDEX ON teacher(TNO)
```

15. create index RECORD_INDEX on score(SNO DESC,DEGREE ASC)

```
78  16:29:45  create index RECORD_INDEX on score(SNO DESC,DEGREE ASC)  #show index .
```

16. show index from score;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
score	1	SNO	1	SNO	A	23	NULL	NULL	YES	BTREE
score	1	CNO	1	CNO	A	10	NULL	NULL	YES	BTREE
score	1	RECORD_INDEX	1	SNO	D	24	NULL	NULL	YES	BTREE
score	1	RECORD_INDEX	2	DEGREE	A	56	NULL	NULL	YES	BTREE

17. drop index TNO_INDEX on teacher;

```
82  16:37:22  drop index TNO_INDEX on teacher
```

18.
```
select SNO,NAME
from student
where DEPART=
      (select DEPART
       from student
       where SNO='PB21000224')
```

SNO	NAME
PB210000001	YH
PB210000006	HCC
PB210000020	XY
PB210000021	LYH
PB210000024	YHS
PB210000025	YWB

student 2

19.
```
select SNO,NAME
from student
where DEPART=
(select DEPART
from student
where SNO='PB21000224')
and SNO<>'PB21000224';
```

	SNO	NAME
▶	PB210000001	YH
	PB210000006	HCC
	PB210000020	XY
	PB210000021	LYH
	PB210000024	YHS
	PB210000025	YWB

student 4 ×

20. select SNO,NAME
    from student
    where DEPART=
    (select DEPART
    from student
    where NAME='jjy');

SNO	NAME
PB210000020	XY
PB210000021	LYH
PB210000024	YHS
PB210000025	YWB
PB21000211	py
PB21000224	CHX

student 5 ×

21. select SNO,NAME
    from student
    where DEPART not in
    (select DEPART
    from student
    where SNO='PB21000227' or SNO='PB21000211')

	SNO	NAME
▶	PB210000002	ZY
	PB210000003	FWJ
	PB210000004	JTY
	PB210000005	YY
	PB210000007	RZJ
	PB210000008	WCS

student 6 ×

22. select TNO,NAME
    from teacher
    where TNO in
    (select TNO
    from course
    where CNO in
    (select CNO
    from score

where SNO='PB21000224'));

	TNO	NAME
▶	TA90025	HTZ
	TA90021	HMZ

23. select count(*)
from (select TNO
from teacher
where DEPART=11 or DEPART=229) as temp(TNO);

	count(*)
▶	7

24. 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;

	SNO	NAME	DEGREE
▶	PB210000001	YH	89
	PB210000002	ZY	94
	PB210000003	FWJ	90
	PB210000004	JTY	95
	PB210000005	YY	93

25. select distinct student.SNO,NAME
from student,score
where (score.CNO in
(select CNO
from teacher
where NAME='ZDH')
and score.SNO=student.SNO);

	SNO	NAME
▶	PB210000001	YH
	PB210000002	ZY
	PB210000003	FWJ
	PB210000004	JTY
	PB210000005	YY
	PB210000006	HCC

Result 11 ×

26. select SNO, DEGREE
from score
where CNO in
(select CNO

```
from course
where NAME='operating_system')
order by DEGREE desc;
```

SNO	DEGREE
PB210000013	89
PB210000011	84
PB210000010	83
PB210000012	78

27.
```
select course.CNO,NAME,avg(DEGREE)
from course left join score on(course.CNO=score.CNO)
group by CNO;
```

CNO	NAME	avg(DEGREE)
20230410	Artificial_Intelligence	81.5000
20230416	Computer_Network	81.7143
20230412	Data_Mining	NULL
20230402	DB_Design	86.8750
20230420	Deep_Learning	86.5000
20230400	Linear_Algebra	85.5000

Result 13

28.
```
select course.CNO,NAME,max(DEGREE),min(DEGREE),max(DEGREE)-min(DEGREE)
from course,score
where course.CNO=score.CNO
group by CNO;
```

CNO	NAME	max(DEGREE)	min(DEGREE)	max(DEGREE)-min(DEGREE)
20230402	DB_Design	95	75	20
20230404	Machine_Learning	93	68	25
20230400	Linear_Algebra	92	77	15
20230418	Pattern_Recognition	95	69	26
20230406	Operating_System	89	78	11
20230408	Natural_Language_Processing	95	74	21

Result 14

29.
```
select distinct teacher.TNO,teacher.NAME
from teacher,score,course
where course.CNO=score.CNO and DEGREE<72 and course.TNO=teacher.TNO;
```

TNO	NAME
TA90023	ZDH
TA90029	ZR

30.
```
select student.SNO,NAME
from student,score
where student.SNO=score.SNO
group by SNO
```

having count(CNO)>=2

SNO	NAME
PB210000001	YH
PB210000002	ZY
PB210000003	FWJ
PB210000004	JTY
PB210000005	YY
PB210000006	HCC

Result 17

31. 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;

SNO	avg(temp.DEGREE)
PB210000001	79.6667
PB210000002	89.5000
PB210000003	90.2500
PB210000004	83.0000
PB210000005	82.0000
PB210000006	86.7500

Result 18

32. 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

DEPART	count(temps.SNO)	avg(temps.DEGREE)
229	9	85.80238571
11	6	80.80000000
12	6	84.25000000
13	2	83.37500000
14	3	86.16666667
10	2	80.16665000

Result 19

33. 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'))

NAME
ZY
FWJ
JTY
HCC
RZJ
ZMS

student 20 ×

34. 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

NAME	avg(2023-year(student.BIRTHDAY))
Artificial_Intelligence	20.5000
Computer_Network	21.4286
Data_Mining	20.7500
DB_Design	21.2500
Deep_Learning	20.8333
Linear_Algebra	20.7500

Result 23 ×

35. 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%");

SNO	NAME
PB210000019	MT
PB210000020	XY
PB210000001	YH
PB210000002	ZY
PB210000003	FWJ
PB210000004	JTY

Result 24 ×

36. 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) ;

	SNO	CNO	DEGREE
▶	PB210000006	20230404	93
	PB210000015	20230412	98
	PB210000003	20230416	97
	PB21000224	20230410	97
	PB21000224	20230412	99

score 25 ×

37. create view db_female_student(SNO,NAME,GENDER,BIRTHDAY,DEPART)
    as select * from student where GENDER='female'
    with check option

✓ 134 16:59:55 create view db_female_student(SNO,NAME,GEND...

38. update db_female_student
    set NAME='CHX'
    where SNO='PB210000016'

✓ 136 17:00:30 update db_female_student set NAME='CHX' where...

39. select SNO,NAME
    from db_female_student
    where 2023-year(BIRTHDAY)<21;

	SNO	NAME
▶	PB210000016	CHX
	PB210000024	YHS

40. insert into db_female_student
    values('SA210110021',' QXY','female','1997-07-27 00:00:00','12');
    select * from db_female_student

SNO	NAME	GENDER	BIRTHDAY	DEPART
PB210000016	CHX	female	2003-11-30 00:00:00	11
PB210000019	MT	female	2002-02-13 00:00:00	10
PB210000020	XY	female	2001-02-14 00:00:00	229
PB210000024	YHS	female	2003-04-01 00:00:00	229
SA210110021	QXY	female	1997-07-27 00:00:00	12

db_female_student 27 ×

41. insert into db_female_student
    values('SA210110023',' DPC','male','1997-04-27 00:00:00','11');

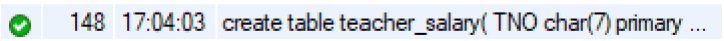
✗ 143 17:02:25 insert into db_female_student values('SA2101100... Error Code: 1369. CHECK OPTION failed 'new_sc...

42. drop view db_female_student

✓ 146 17:03:28 drop view db_female_student -- create table tea...

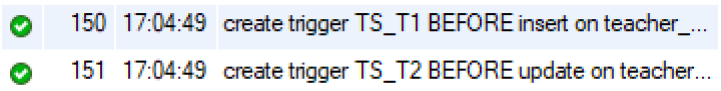
43. create table teacher_salary(

TNO char(7) primary key,
SAL float)

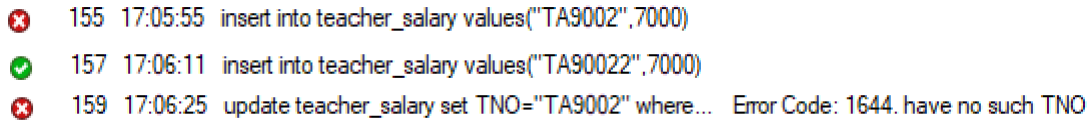


44. DELIMITER //
create trigger TS_T1
BEFORE insert on teacher_salary
for each row
if new.TNO not in (select TNO from teacher) then
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = "have no such TNO";
END IF//
DELIMITER ;

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"



45. 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 ;
```

```
161  17:08:08  create trigger TS_T3 BEFORE update on teacher...
162  17:08:08  create trigger TS_T4 BEFORE insert on teacher_...
```

```
insert into teacher_salary
values("TA90023",2000)
```

```
164  17:08:53  insert into teacher_salary values("TA90023",2000...  Error Code: 1644. the salary is not correct
```

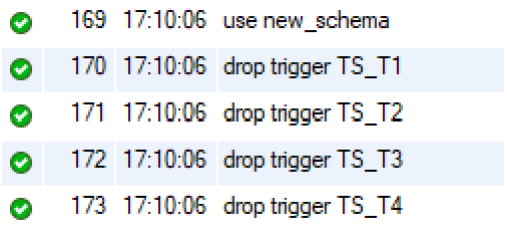
```
insert into teacher_salary
values("TA90023",5000);
```

```
166  17:09:16  insert into teacher_salary values("TA90023",5000)
```

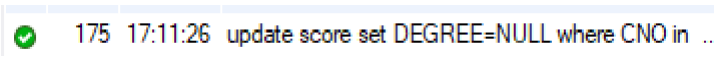
```
update teacher_salary
set SAL="1000"
where TNO="TA90023"
```

```
168  17:09:36  update teacher_salary set SAL="1000" where TN...  Error Code: 1644. the salary is not correct
```

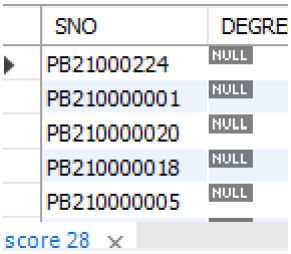
46.
```
drop trigger TS_T1;
drop trigger TS_T2;
drop trigger TS_T3;
drop trigger TS_T4;
```



47. 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;

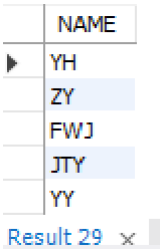


SNO	DEGRE
PB21000224	NULL
PB210000001	NULL
PB210000020	NULL
PB210000018	NULL
PB210000005	NULL

score 28

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

48. 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));

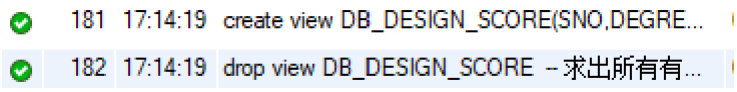


NAME
YH
ZY
FWJ
JTY
YY

Result 29

49. 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



50. 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

	SNO	avg_degree
▶	PB21000224	97.0000
	PB210000018	95.0000
	PB210000003	90.2500
	PB210000002	89.5000
	PB210000013	89.0000

Result 30