
Prashant Bharti Roll No. 202251102

Indian Institute of Information Technology Vadodara

Database Management Systems Laboratory Assignment

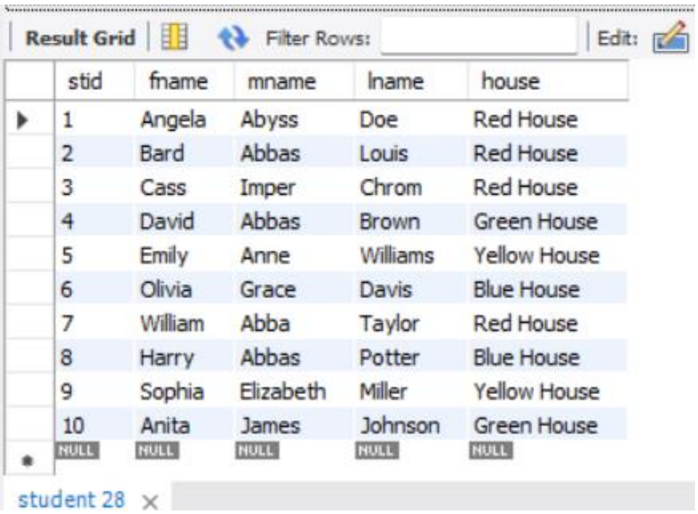
Week-02

ASSIGNMENT-02

-by PRASHANT BHARTI

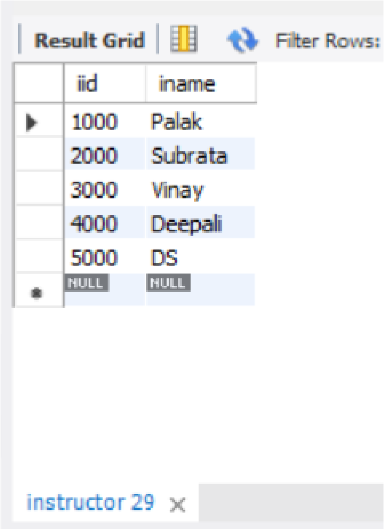
(202251102)

1. Students table: information about students.



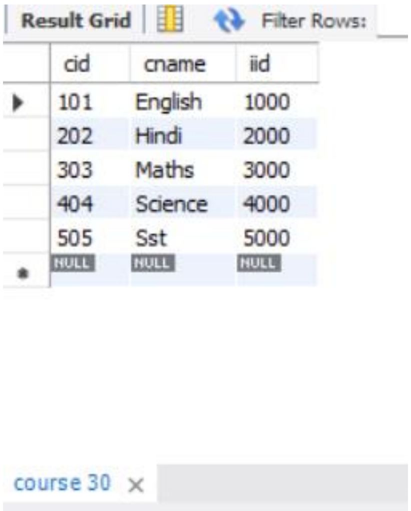
stid	fname	mname	lname	house
1	Angela	Abyss	Doe	Red House
2	Bard	Abbas	Louis	Red House
3	Cass	Imper	Chrom	Red House
4	David	Abbas	Brown	Green House
5	Emily	Anne	Williams	Yellow House
6	Olivia	Grace	Davis	Blue House
7	William	Abba	Taylor	Red House
8	Harry	Abbas	Potter	Blue House
9	Sophia	Elizabeth	Miller	Yellow House
10	Anita	James	Johnson	Green House
NULL	NULL	NULL	NULL	NULL

2. Instructors table: information about instructors.



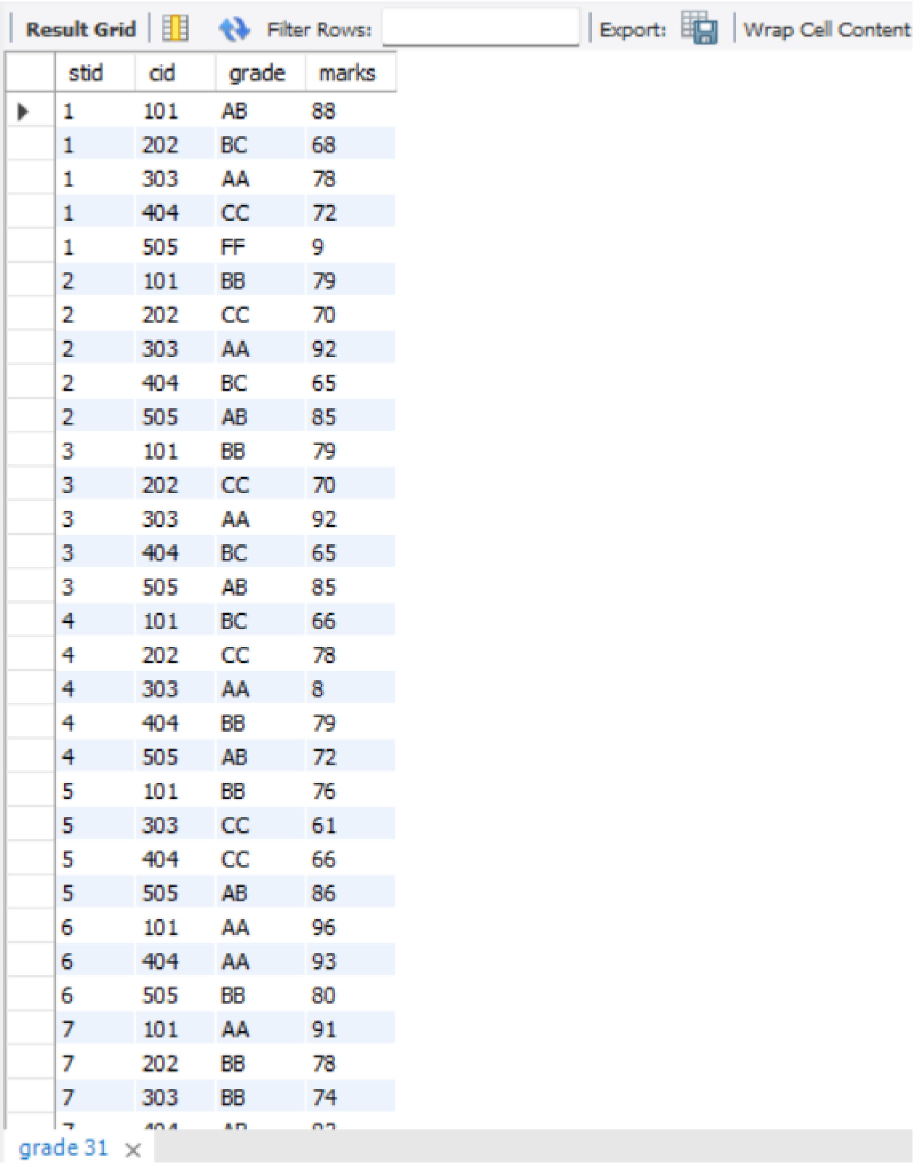
iid	iname
1000	Palak
2000	Subrata
3000	Vinay
4000	Deepali
5000	DS
NULL	NULL

3. Courses table: information about courses and instructors.

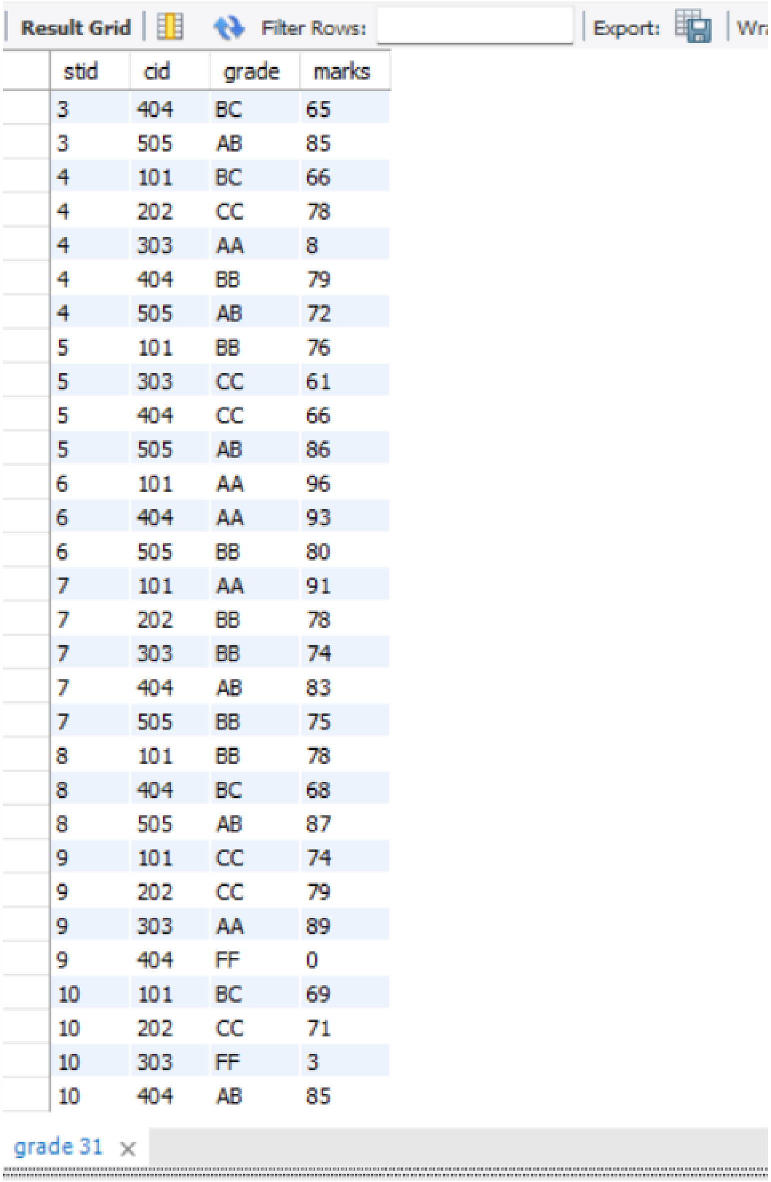


cid	cname	iid
101	English	1000
202	Hindi	2000
303	Maths	3000
404	Science	4000
505	Sst	5000
NULL	NULL	NULL

4. Grades table: information about enrolment and grades.

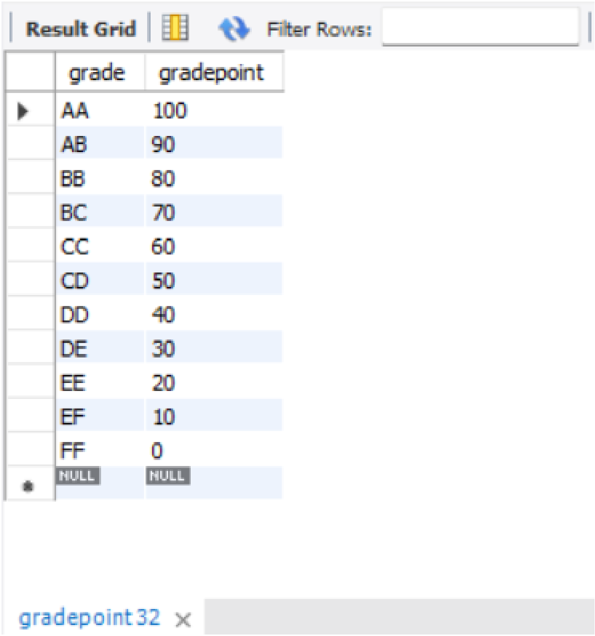


stid	cid	grade	marks
1	101	AB	88
1	202	BC	68
1	303	AA	78
1	404	CC	72
1	505	FF	9
2	101	BB	79
2	202	CC	70
2	303	AA	92
2	404	BC	65
2	505	AB	85
3	101	BB	79
3	202	CC	70
3	303	AA	92
3	404	BC	65
3	505	AB	85
4	101	BC	66
4	202	CC	78
4	303	AA	8
4	404	BB	79
4	505	AB	72
5	101	BB	76
5	303	CC	61
5	404	CC	66
5	505	AB	86
6	101	AA	96
6	404	AA	93
6	505	BB	80
7	101	AA	91
7	202	BB	78
7	303	BB	74



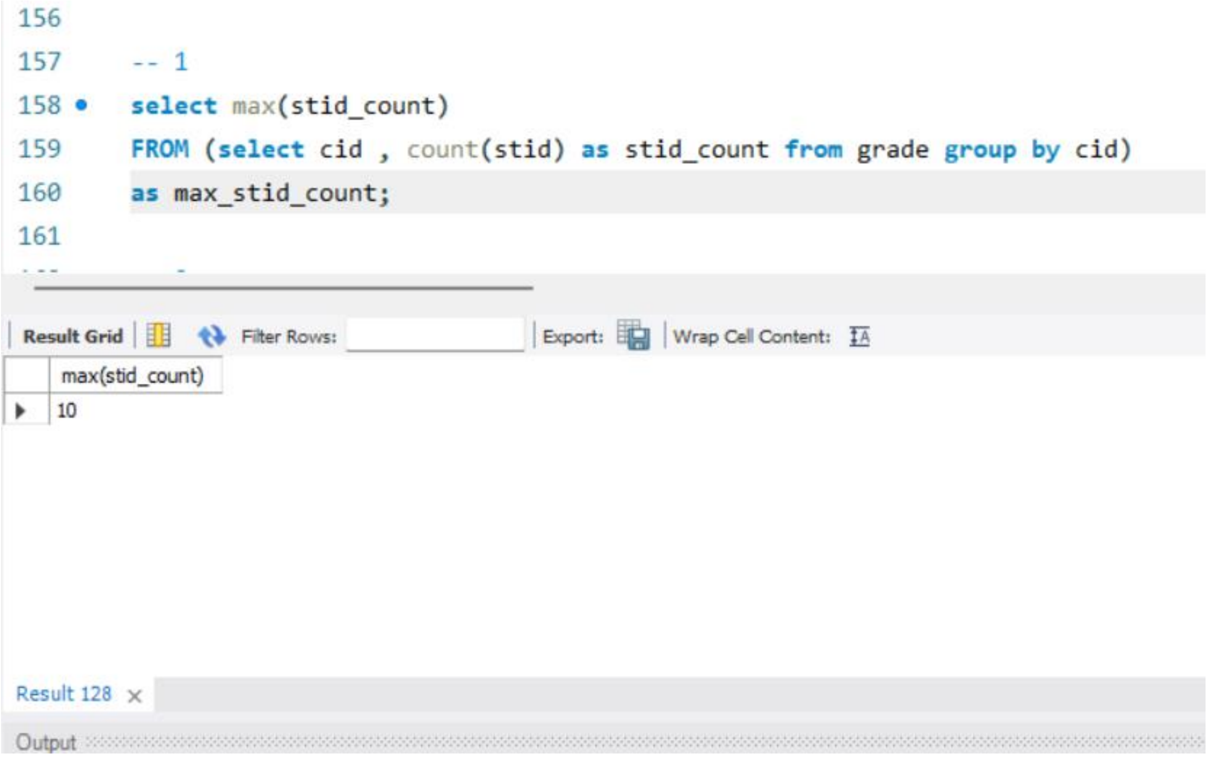
stid	cid	grade	marks
3	404	BC	65
3	505	AB	85
4	101	BC	66
4	202	CC	78
4	303	AA	8
4	404	BB	79
4	505	AB	72
5	101	BB	76
5	303	CC	61
5	404	CC	66
5	505	AB	86
6	101	AA	96
6	404	AA	93
6	505	BB	80
7	101	AA	91
7	202	BB	78
7	303	BB	74
7	404	AB	83
7	505	BB	75
8	101	BB	78
8	404	BC	68
8	505	AB	87
9	101	CC	74
9	202	CC	79
9	303	AA	89
9	404	FF	0
10	101	BC	69
10	202	CC	71
10	303	FF	3
10	404	AB	85

5. Gradepoint table: mapping from grades to grade points.



grade	gradepoint
AA	100
AB	90
BB	80
BC	70
CC	60
CD	50
DD	40
DE	30
EE	20
EF	10
FF	0
NULL	NULL

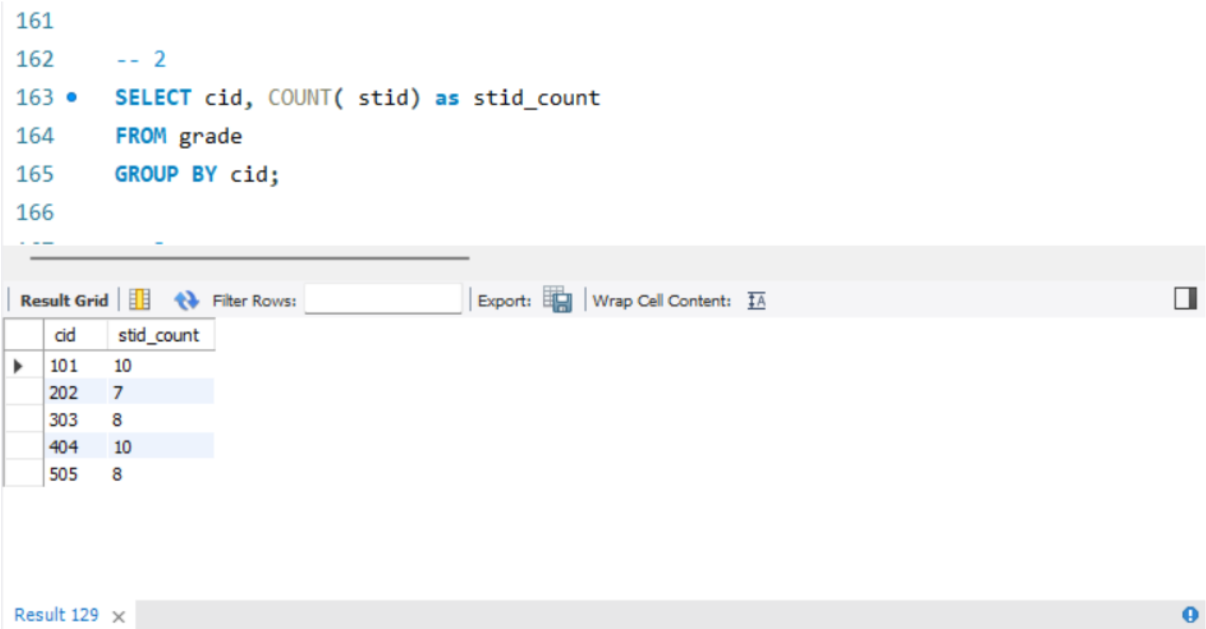
1. Course name(s) with highest enrolment: (course name, number of students enrolled).



```
-- 1
select max(stid_count)
FROM (select cid , count(stid) as stid_count from grade group by cid)
as max_stid_count;
```

max(stid_count)
10

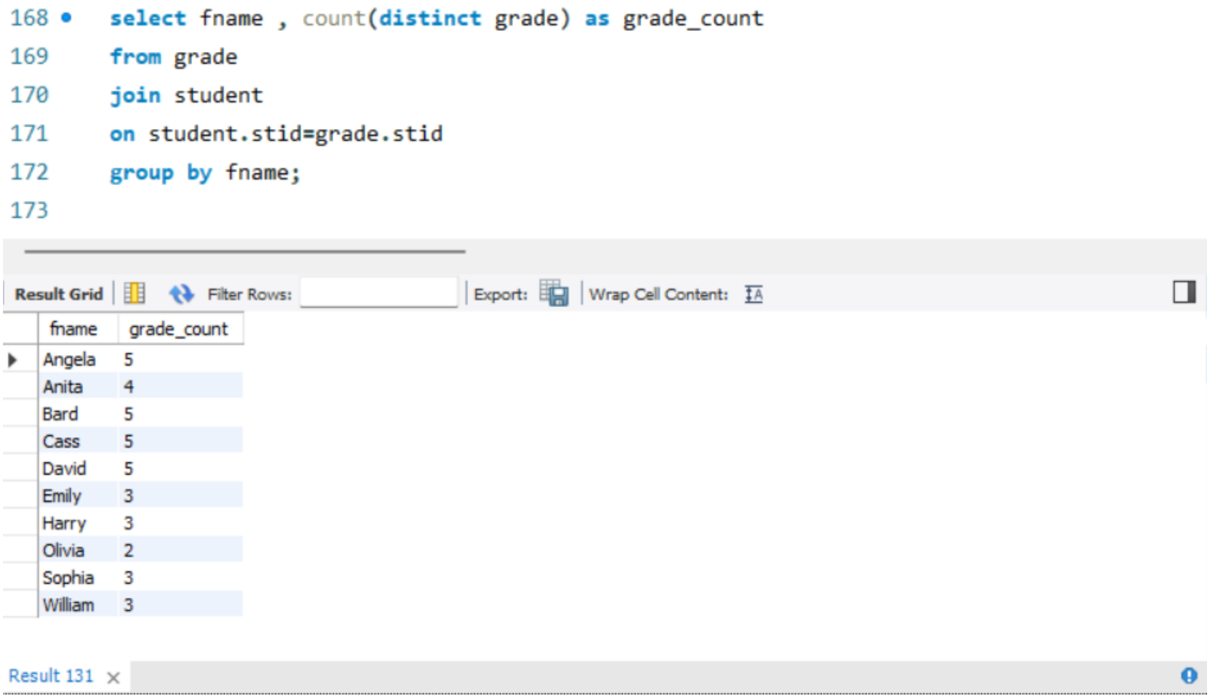
2. Show the total number of students who are enrolled in multiple courses.



```
-- 2
SELECT cid, COUNT( stid) as stid_count
FROM grade
GROUP BY cid;
```

cid	stid_count
101	10
202	7
303	8
404	10
505	8

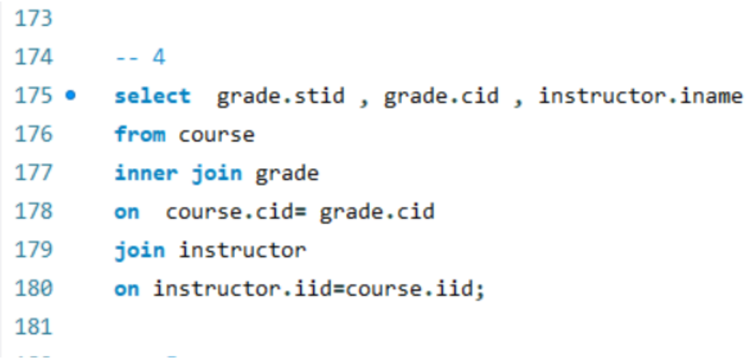
3. Student(s) with maximum number of distinct grades across courses: (student name, number of distinct grades).



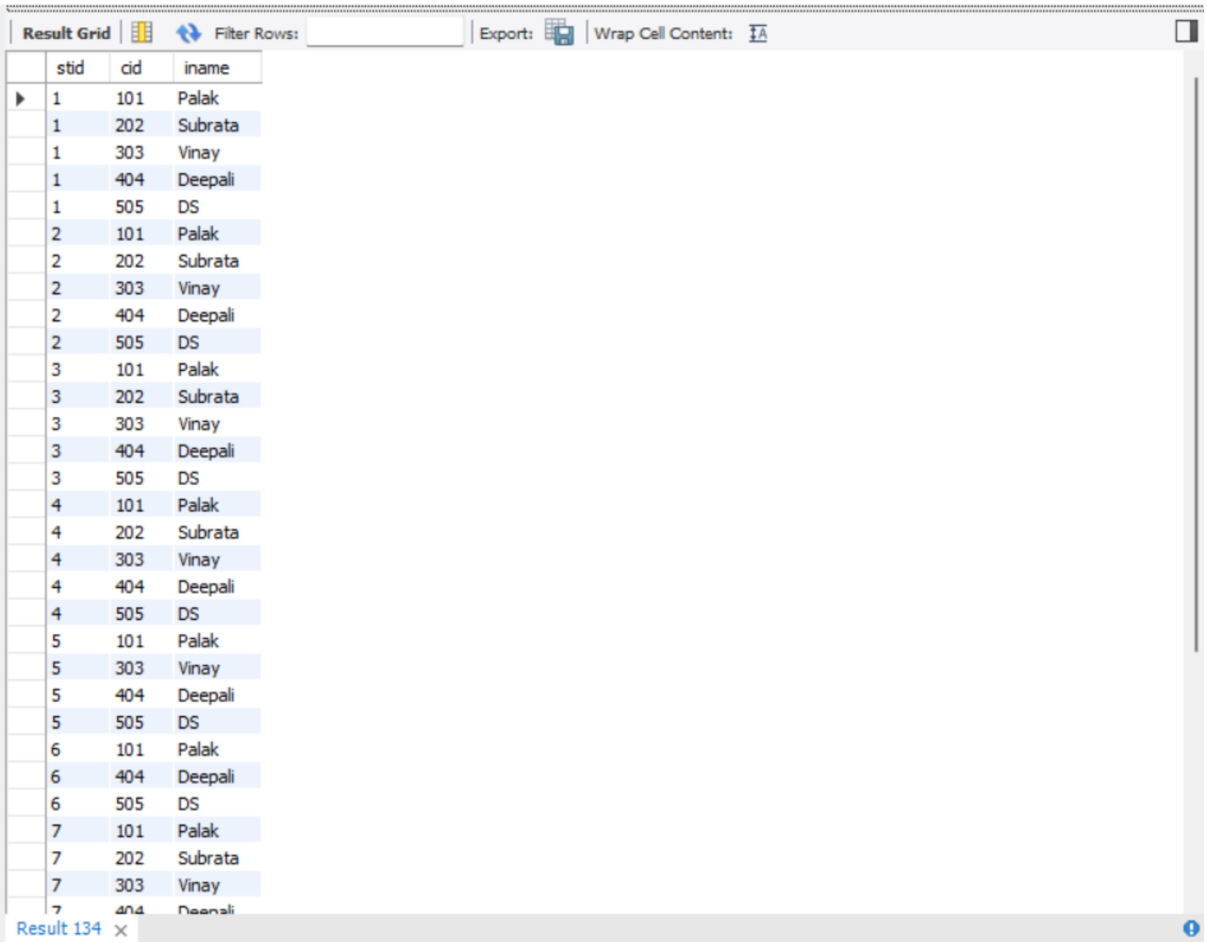
```
select fname , count(distinct grade) as grade_count
from grade
join student
on student.stid=grade.stid
group by fname;
```

fname	grade_count
Angela	5
Anita	4
Bard	5
Cass	5
David	5
Emily	3
Harry	3
Olivia	2
Sophia	3
William	3

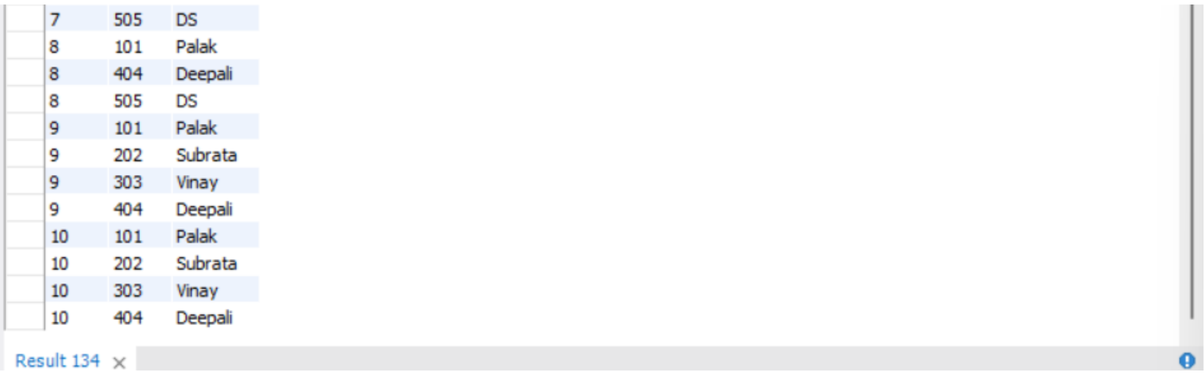
4. Instructor name for the students enrolled in different courses. (Hint: Use join operator).



```
-- 4
select  grade.stid , grade.cid , instructor.iname
from course
inner join grade
on  course.cid= grade.cid
join instructor
on instructor.iid=course.iid;
```

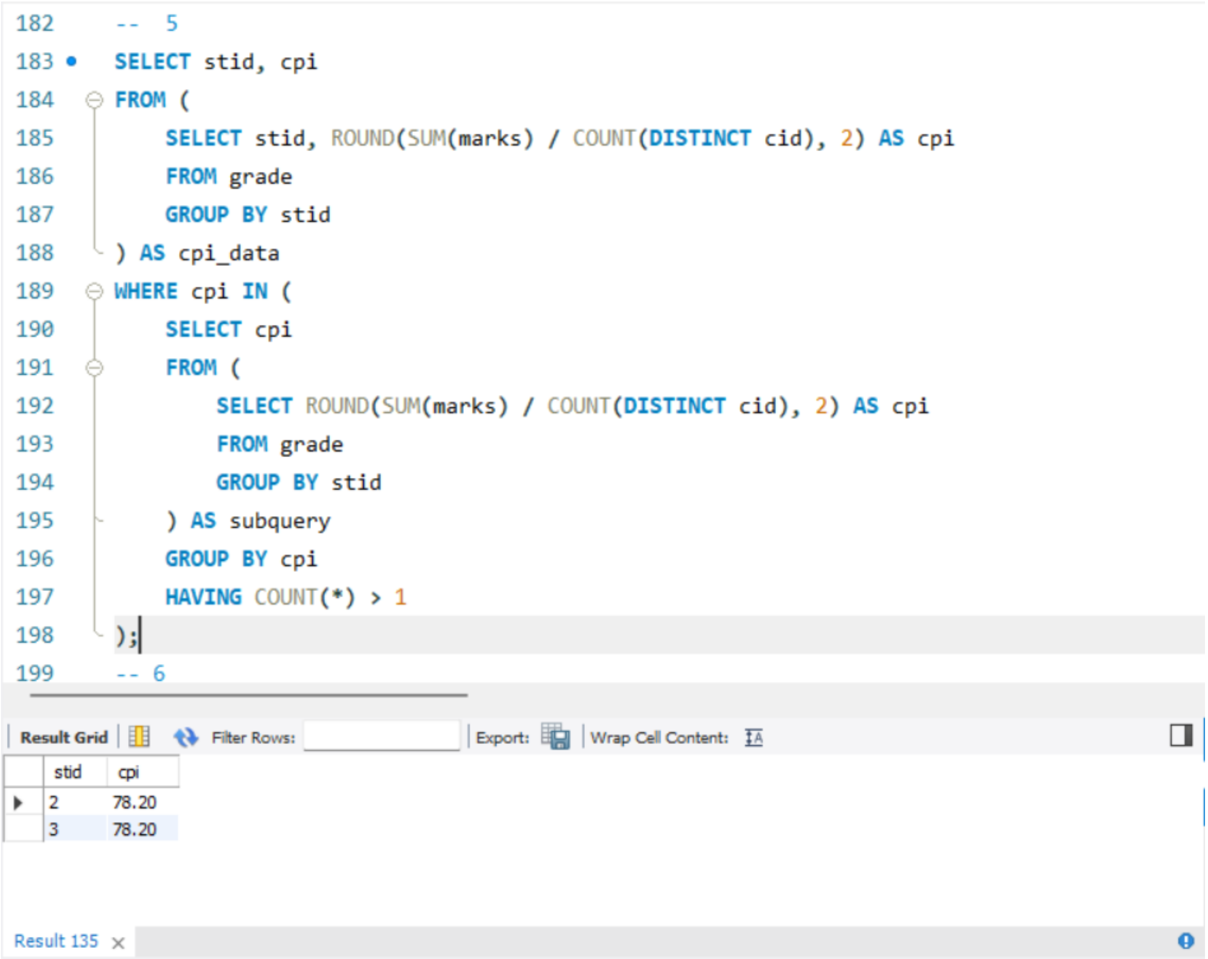


stid	cid	iname
1	101	Palak
1	202	Subrata
1	303	Vinay
1	404	Deepali
1	505	DS
2	101	Palak
2	202	Subrata
2	303	Vinay
2	404	Deepali
2	505	DS
3	101	Palak
3	202	Subrata
3	303	Vinay
3	404	Deepali
3	505	DS
4	101	Palak
4	202	Subrata
4	303	Vinay
4	404	Deepali
4	505	DS
5	101	Palak
5	303	Vinay
5	404	Deepali
5	505	DS
6	101	Palak
6	404	Deepali
6	505	DS
7	101	Palak
7	202	Subrata
7	303	Vinay
7	404	Deepali



7	505	DS
8	101	Palak
8	404	Deepali
8	505	DS
9	101	Palak
9	202	Subrata
9	303	Vinay
9	404	Deepali
10	101	Palak
10	202	Subrata
10	303	Vinay
10	404	Deepali

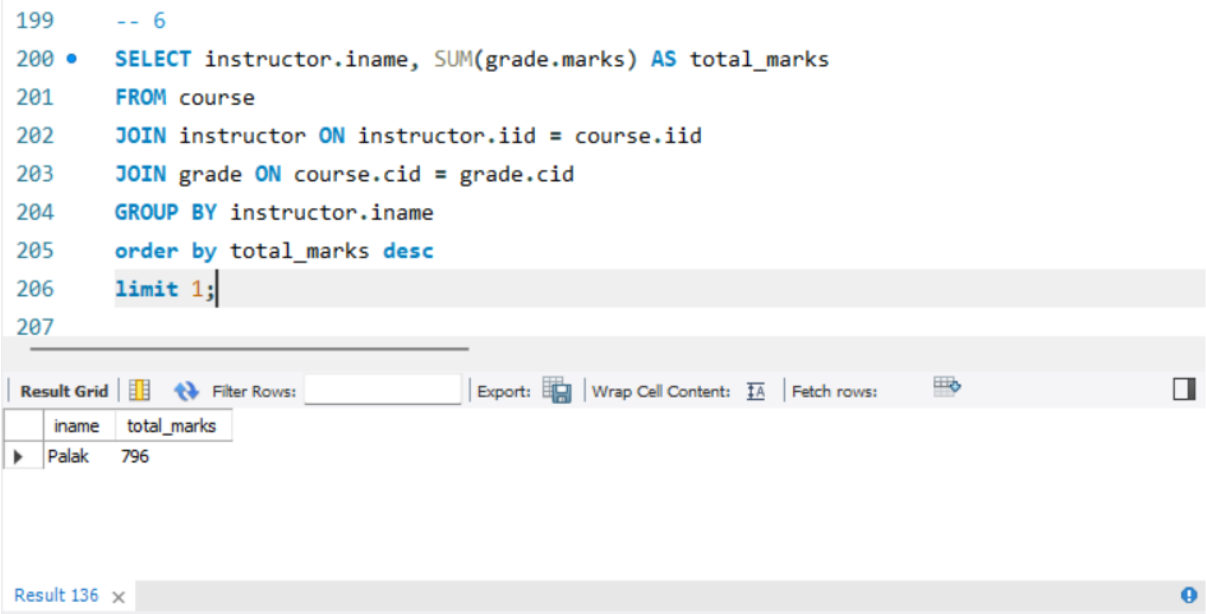
5. Students who are having the same CPI.



```sql
-- 5
SELECT stid, cpi
FROM (
    SELECT stid, ROUND(SUM(marks) / COUNT(DISTINCT cid), 2) AS cpi
    FROM grade
    GROUP BY stid
) AS cpi_data
WHERE cpi IN (
    SELECT cpi
    FROM (
        SELECT ROUND(SUM(marks) / COUNT(DISTINCT cid), 2) AS cpi
        FROM grade
        GROUP BY stid
    ) AS subquery
    GROUP BY cpi
    HAVING COUNT(*) > 1
);
-- 6
```

stid	cpi
2	78.20
3	78.20

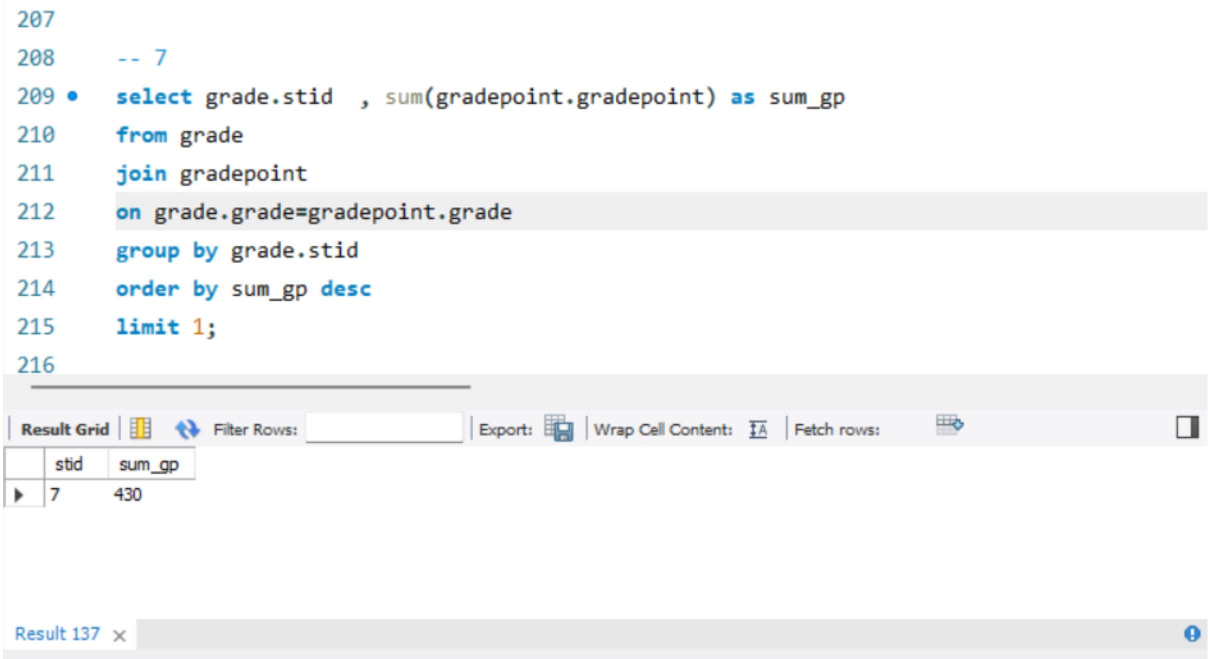
6. Most generous teacher(s) – highest mean GPA in course: (teacher name, course name).



```sql
-- 6
SELECT instructor.iname, SUM(grade.marks) AS total_marks
FROM course
JOIN instructor ON instructor.iid = course.iid
JOIN grade ON course.cid = grade.cid
GROUP BY instructor.iname
order by total_marks desc
limit 1;
```

iname	total_marks
Palak	796

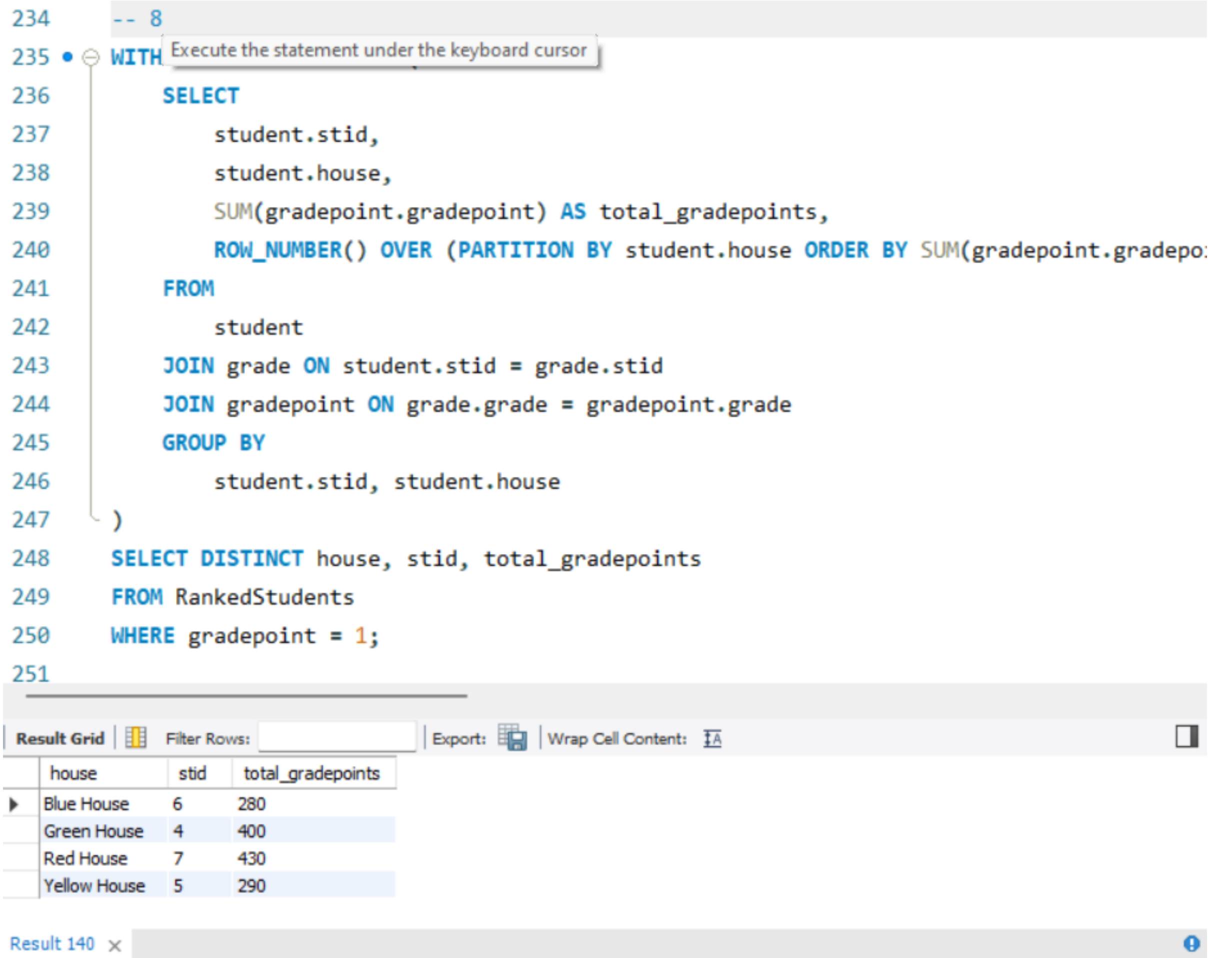
7. Student(s) with overall highest GPA: (student name, GPA).



```sql
-- 7
select grade.stid  , sum(gradepoint.gradepoint) as sum_gp
from grade
join gradepoint
on grade.grade=gradepoint.grade
group by grade.stid
order by sum_gp desc
limit 1;
```

stid	sum_gp
7	430

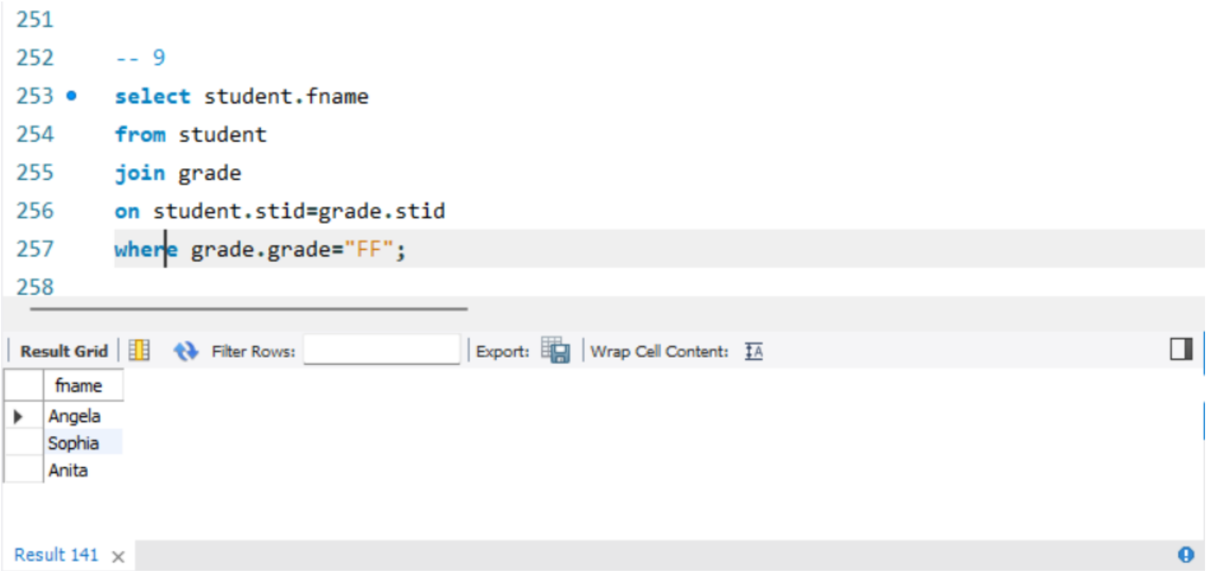
8. Student(s) with highest GPA in each house: (student name, GPA).



```sql
-- 8
WITH
    SELECT
        student.stid,
        student.house,
        SUM(gradepoint.gradepoint) AS total_gradepoints,
        ROW_NUMBER() OVER (PARTITION BY student.house ORDER BY SUM(gradepoint.gradepo
    FROM
        student
    JOIN grade ON student.stid = grade.stid
    JOIN gradepoint ON grade.grade = gradepoint.grade
    GROUP BY
        student.stid, student.house
)
SELECT DISTINCT house, stid, total_gradepoints
FROM RankedStudents
WHERE gradepoint = 1;
```

house	stid	total_gradepoints
Blue House	6	280
Green House	4	400
Red House	7	430
Yellow House	5	290

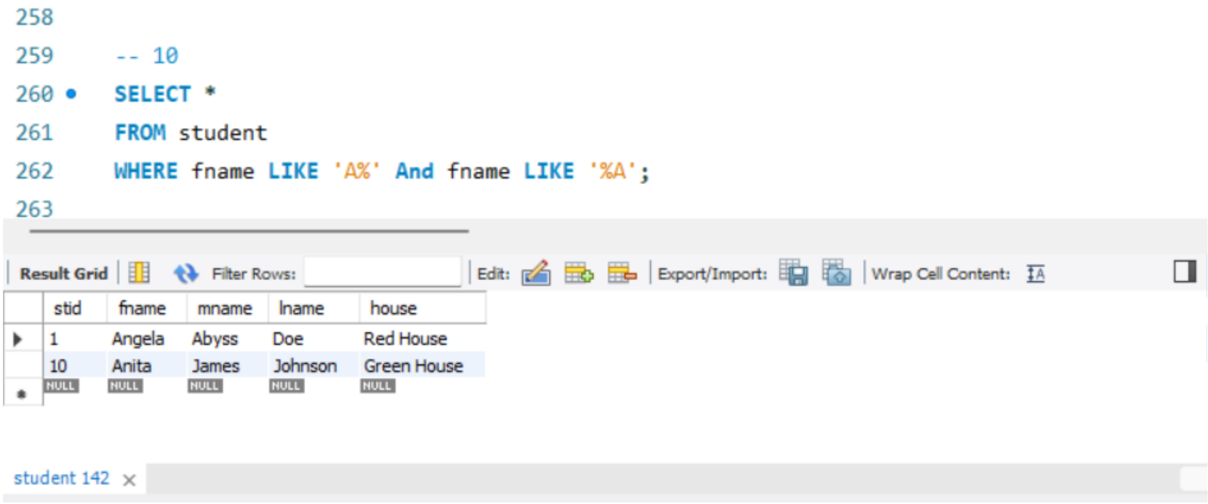
9. Show the name of students who are having FF grades.



```
-- 9
select student.fname
from student
join grade
on student.stid=grade.stid
where grade.grade="FF";
```

fname
Angela
Sophia
Anita

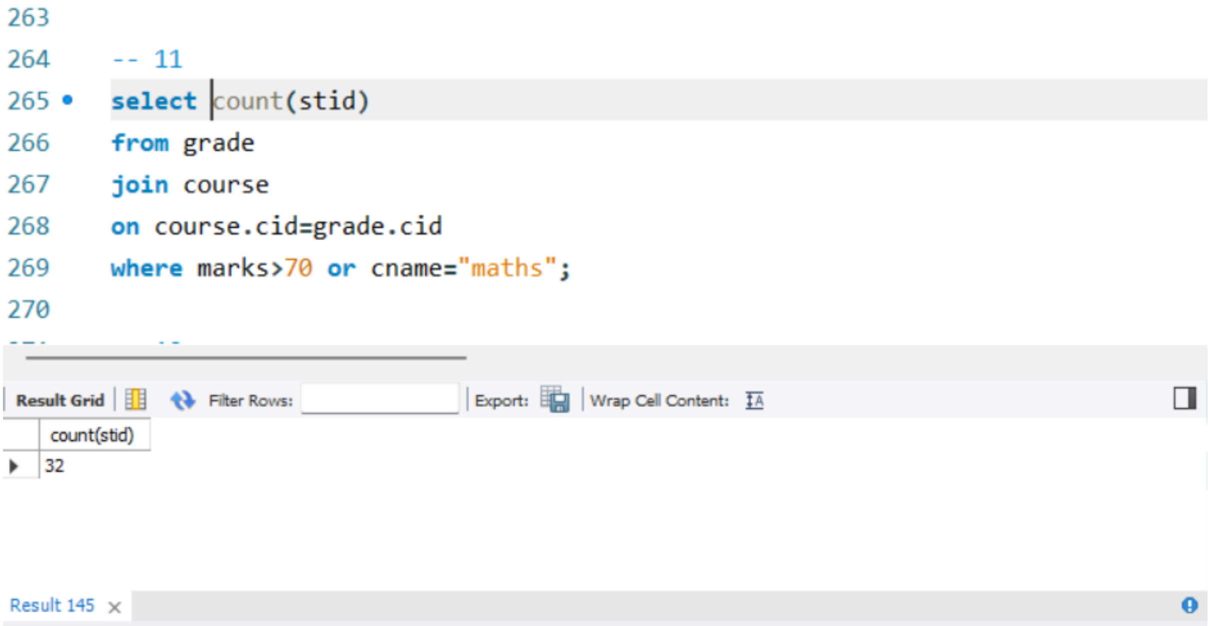
10. The students which are having name starting or ending alphabet is “A”.



```
-- 10
SELECT *
FROM student
WHERE fname LIKE 'A%' And fname LIKE '%A';
```

stid	fname	mname	lname	house
1	Angela	Abyss	Doe	Red House
10	Anita	James	Johnson	Green House
NULL	NULL	NULL	NULL	NULL

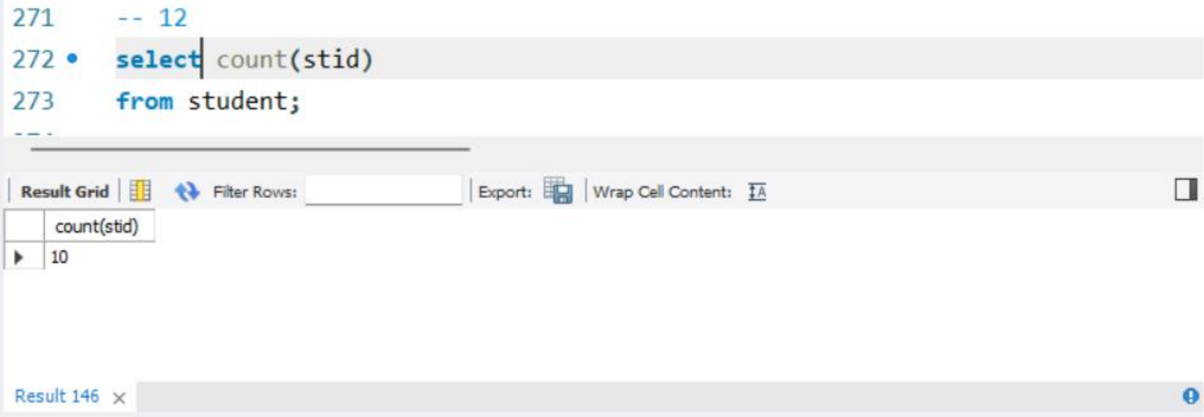
11. Show the number of students who are having marks either more than 70 or are enrolled for the subject of Maths.



```sql
-- 11
select count(stid)
from grade
join course
on course.cid=grade.cid
where marks>70 or cname="maths";
```

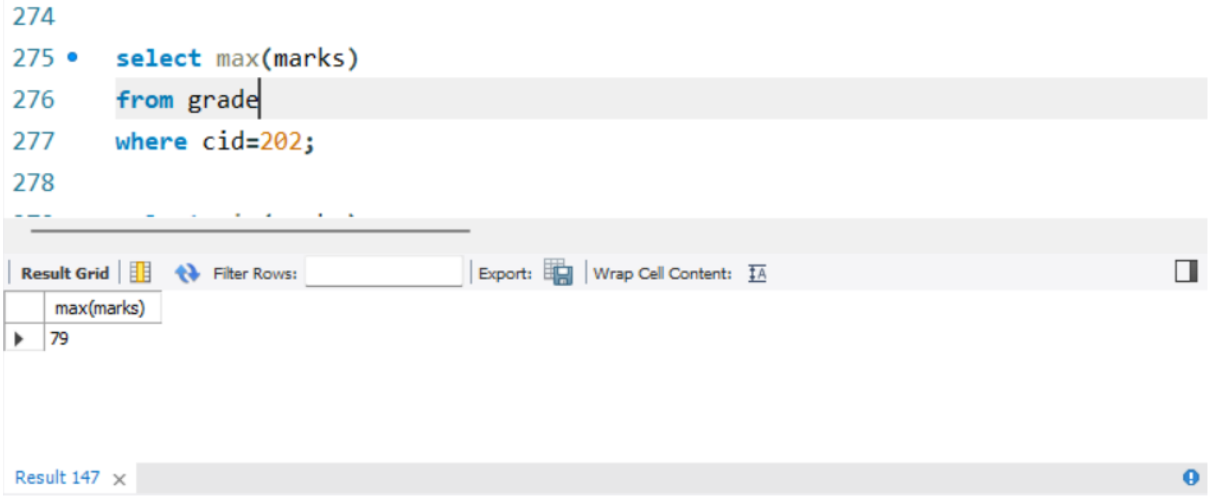
```
count(stid)
32
```

12. Show the use of AGGREGATE function.



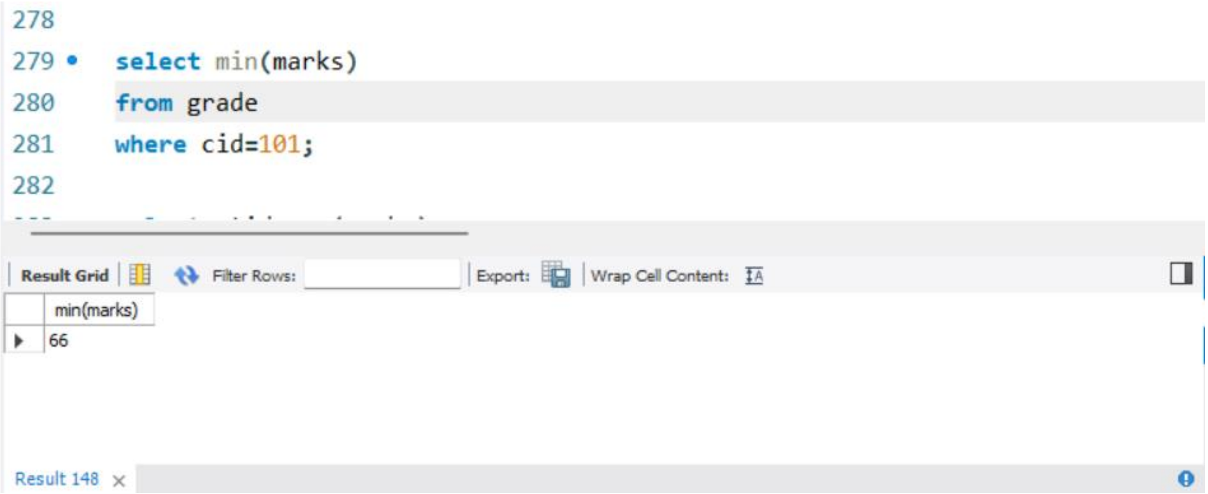
```sql
-- 12
select count(stid)
from student;
```

```
count(stid)
10
```



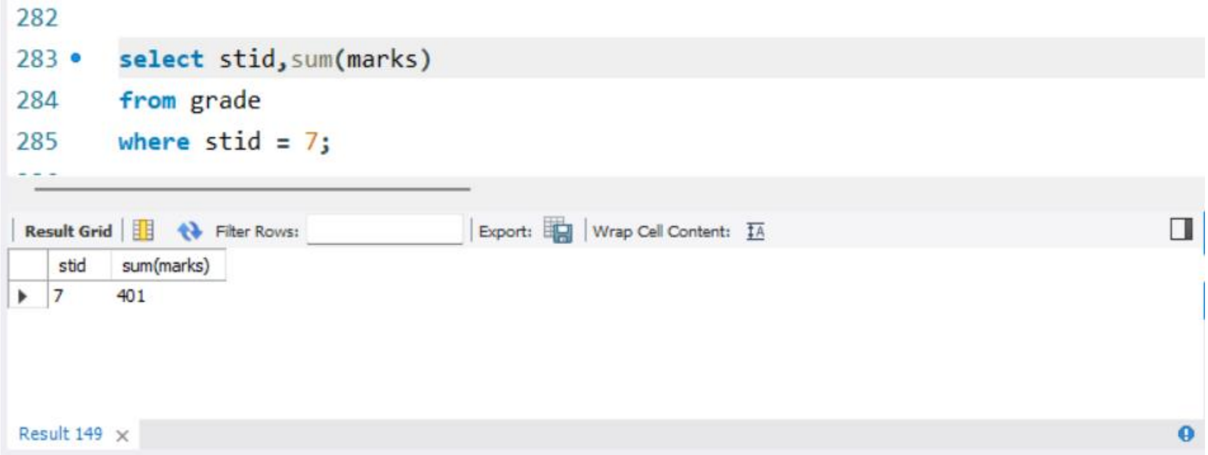
```sql
select max(marks)
from grade
where cid=202;
```

```
max(marks)
79
```



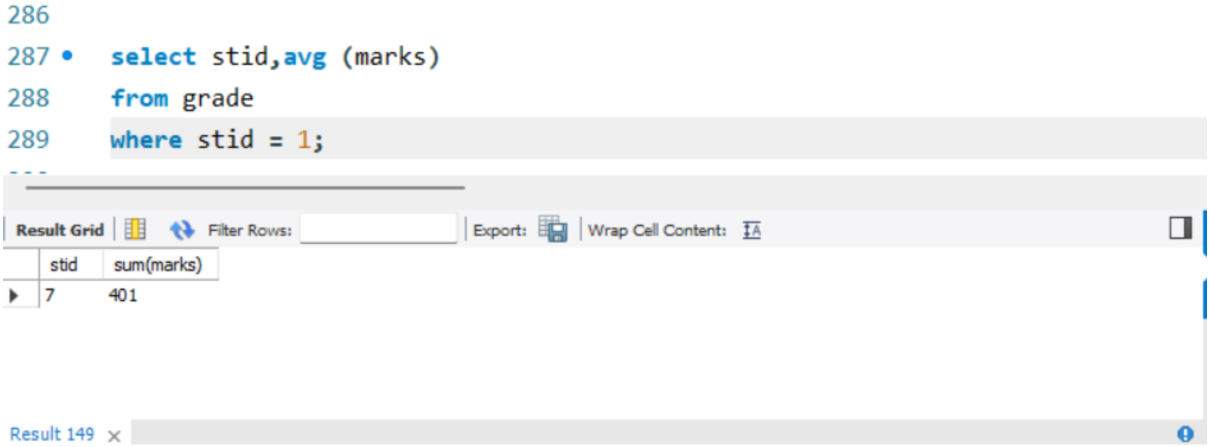
```sql
select min(marks)
from grade
where cid=101;
```

```
min(marks)
66
```



```sql
select stid,sum(marks)
from grade
where stid = 7;
```

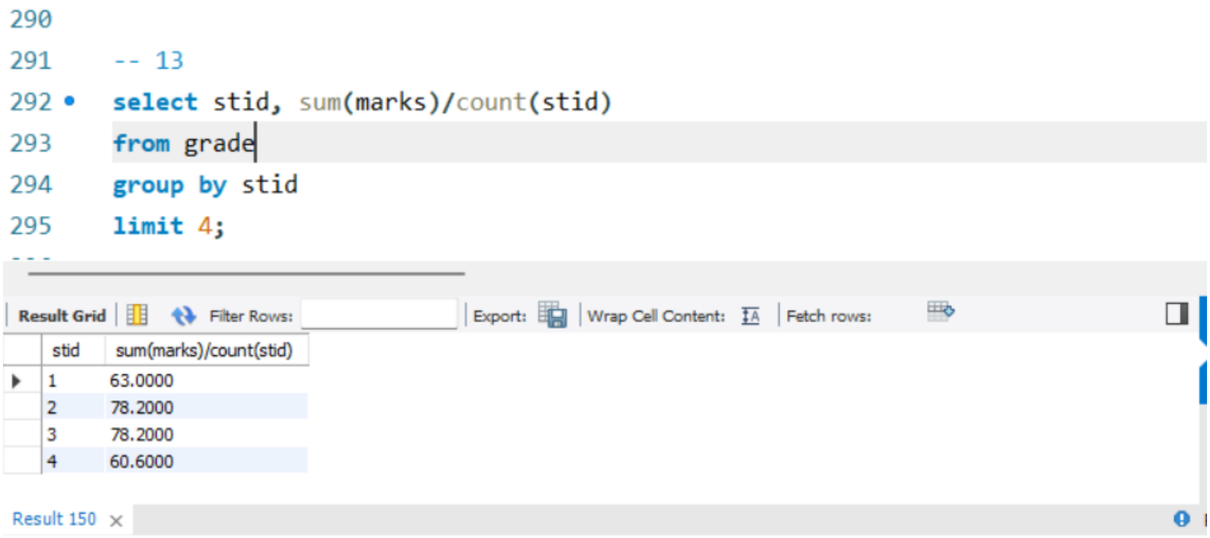
```
stid   sum(marks)
7      401
```



```sql
select stid,avg (marks)
from grade
where stid = 1;
```

```
stid   sum(marks)
7      401
```

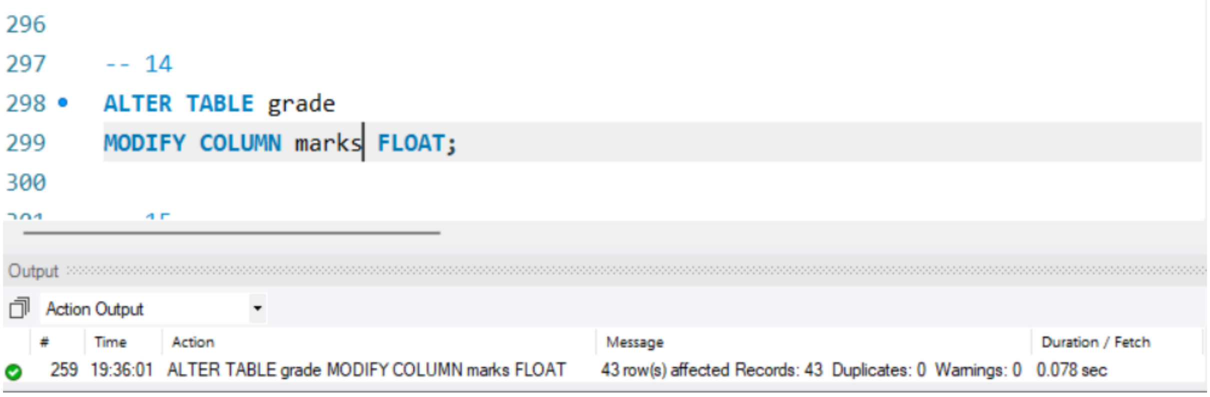
13. Show the list of top 5 students based on their 4 semester marks.



```
-- 13
select stid, sum(marks)/count(stid)
from grade
group by stid
limit 4;
```

stid	sum(marks)/count(stid)
1	63.0000
2	78.2000
3	78.2000
4	60.6000

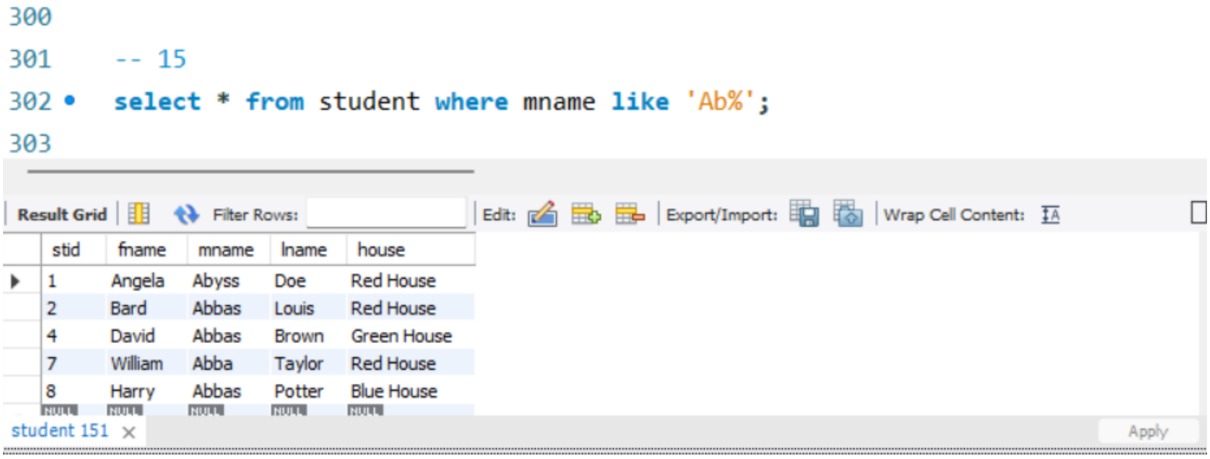
14. Change datatype of marks.



```
-- 14
ALTER TABLE grade
MODIFY COLUMN marks FLOAT;
```

#	Time	Action	Message	Duration / Fetch
259	19:36:01	ALTER TABLE grade MODIFY COLUMN marks FLOAT	43 row(s) affected Records: 43 Duplicates: 0 Warnings: 0	0.078 sec

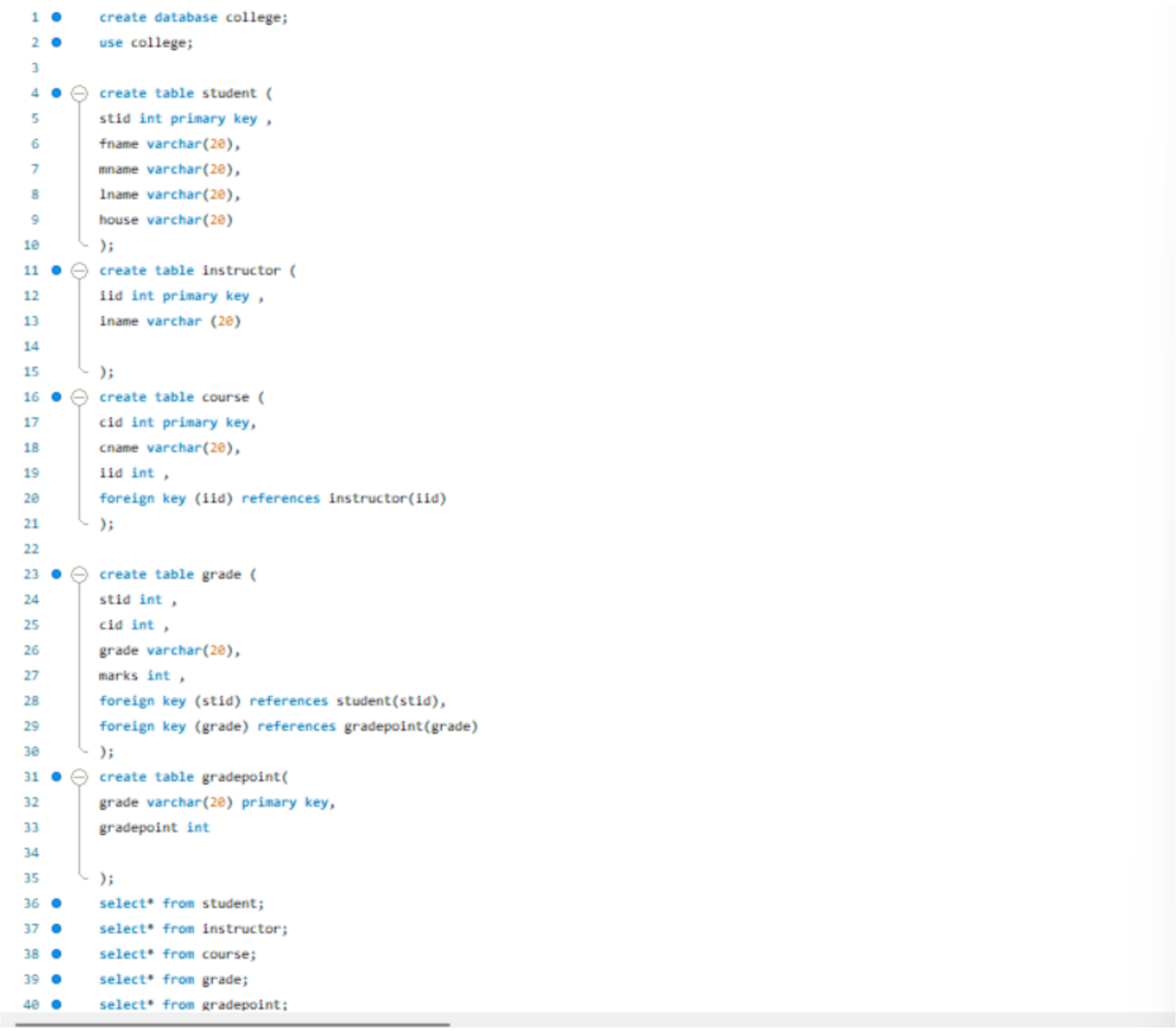
15. Check how many students are having the middle name starting with “AB”.



```
-- 15
select * from student where mname like 'Ab%';
```

stid	fname	mname	lname	house
1	Angela	Abyss	Doe	Red House
2	Bard	Abbas	Louis	Red House
4	David	Abbas	Brown	Green House
7	William	Abba	Taylor	Red House
8	Harry	Abbas	Potter	Blue House
NULL	NULL	NULL	NULL	NULL

16. Choose student id as primary key and use the same in other tables as a reference.



```sql
create database college;
use college;

create table student (
stid int primary key ,
fname varchar(20),
mname varchar(20),
lname varchar(20),
house varchar(20)
);
create table instructor (
iid int primary key ,
iname varchar (20)

);
create table course (
cid int primary key,
cname varchar(20),
iid int ,
foreign key (iid) references instructor(iid)
);

create table grade (
stid int ,
cid int ,
grade varchar(20),
marks int ,
foreign key (stid) references student(stid),
foreign key (grade) references gradepoint(grade)
);
create table gradepoint(
grade varchar(20) primary key,
gradepoint int

);
select* from student;
select* from instructor;
select* from course;
select* from grade;
select* from gradepoint;
```

THANK YOU

- - - - [END] - - - -