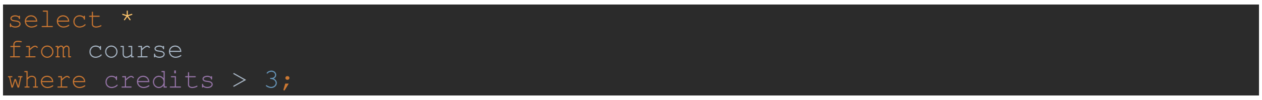
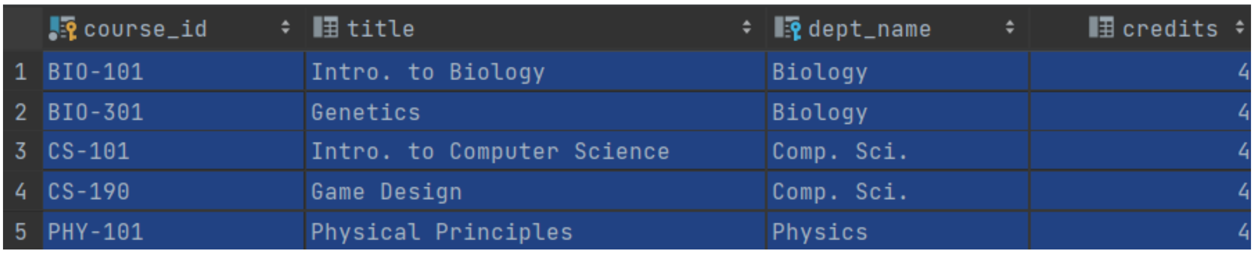
Laboratory work 4

1. Write the following queries in SQL, using the university schema:

a. Find all courses worth more than 3 credits;

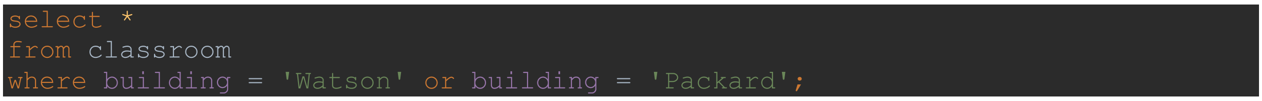


```
select *
from course
where credits > 3;
```

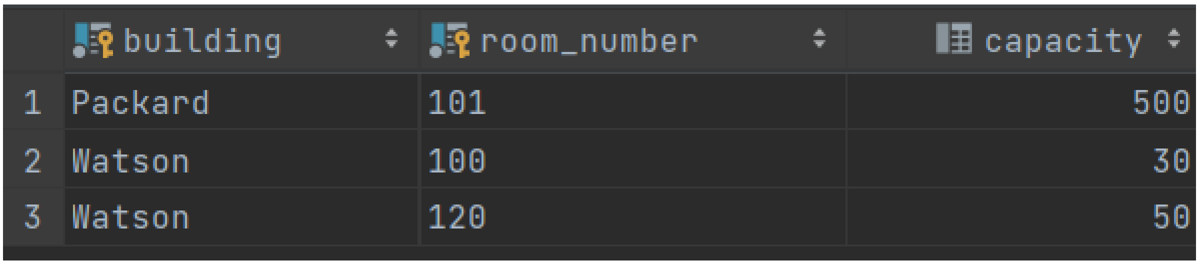


	course_id	title	dept_name	credits
1	BIO-101	Intro. to Biology	Biology	4
2	BIO-301	Genetics	Biology	4
3	CS-101	Intro. to Computer Science	Comp. Sci.	4
4	CS-190	Game Design	Comp. Sci.	4
5	PHY-101	Physical Principles	Physics	4

b. Find all classrooms situated either in ‘Watson’ or ‘Packard’ buildings;

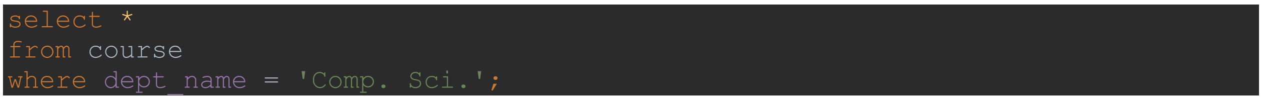


```
select *
from classroom
where building = 'Watson' or building = 'Packard';
```

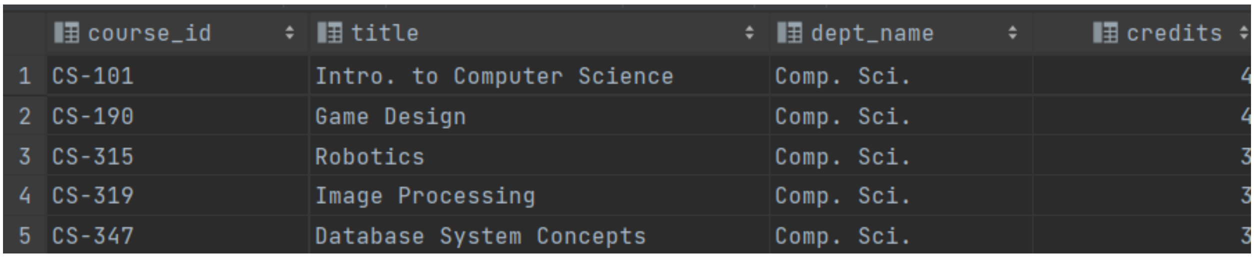


	building	room_number	capacity
1	Packard	101	500
2	Watson	100	30
3	Watson	120	50

c. Find all courses offered by the Computer Science department;

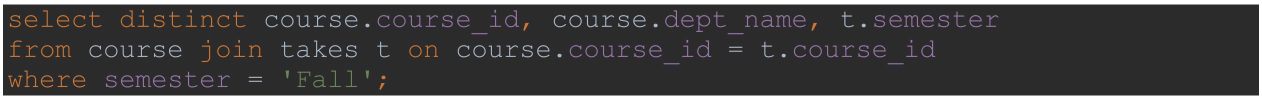


```
select *
from course
where dept_name = 'Comp. Sci.';
```

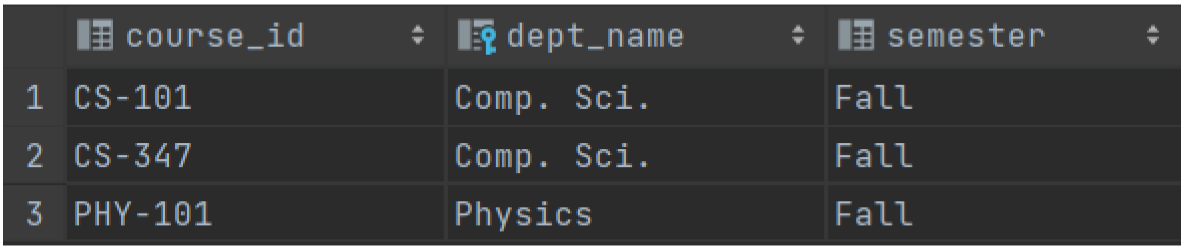


	course_id	title	dept_name	credits
1	CS-101	Intro. to Computer Science	Comp. Sci.	4
2	CS-190	Game Design	Comp. Sci.	4
3	CS-315	Robotics	Comp. Sci.	3
4	CS-319	Image Processing	Comp. Sci.	3
5	CS-347	Database System Concepts	Comp. Sci.	3

d. Find all courses offered during fall;

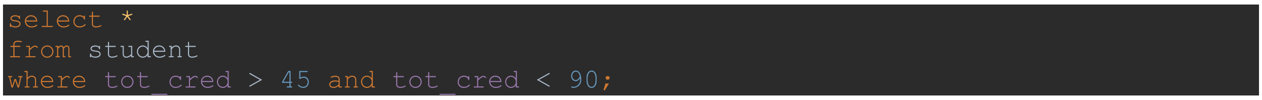


```
select distinct course.course_id, course.dept_name, t.semester
from course join takes t on course.course_id = t.course_id
where semester = 'Fall';
```

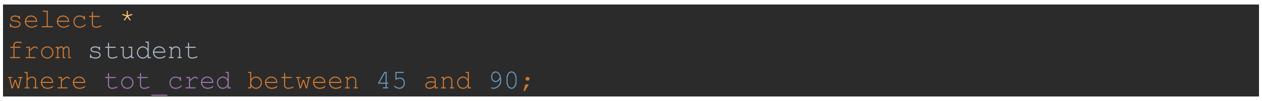


	course_id	dept_name	semester
1	CS-101	Comp. Sci.	Fall
2	CS-347	Comp. Sci.	Fall
3	PHY-101	Physics	Fall

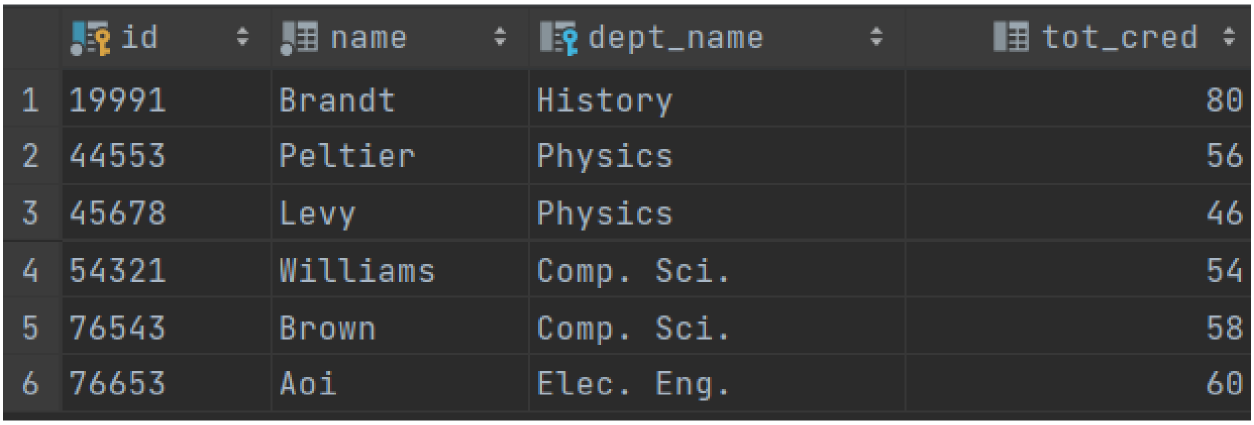
e. Find all students who have more than 45 credits but less than 90;



```
select *
from student
where tot_cred > 45 and tot_cred < 90;
```

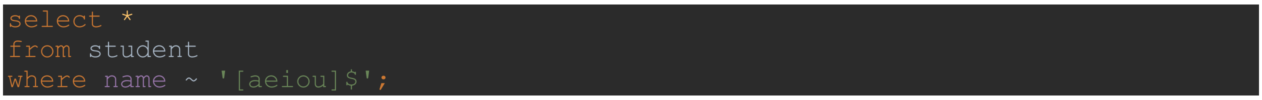


```
select *
from student
where tot_cred between 45 and 90;
```

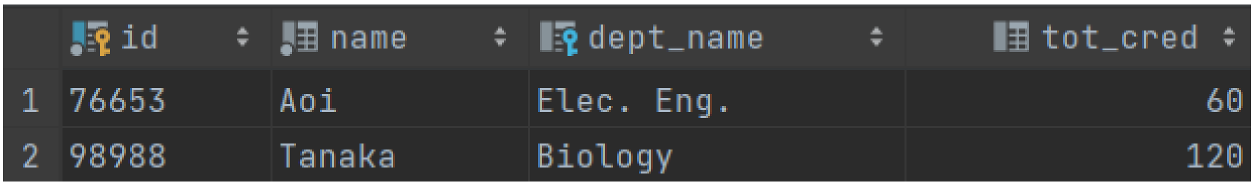


	id	name	dept_name	tot_cred
1	19991	Brandt	History	80
2	44553	Peltier	Physics	56
3	45678	Levy	Physics	46
4	54321	Williams	Comp. Sci.	54
5	76543	Brown	Comp. Sci.	58
6	76653	Aoi	Elec. Eng.	60

f. Find all students whose names end with vowels;

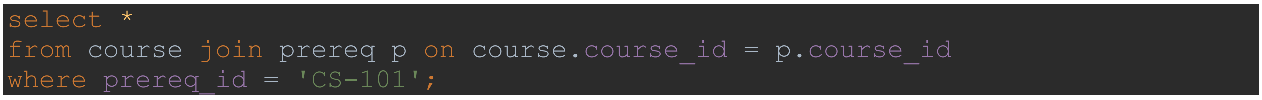


```
select *
from student
where name ~ '[aeiou]$';
```

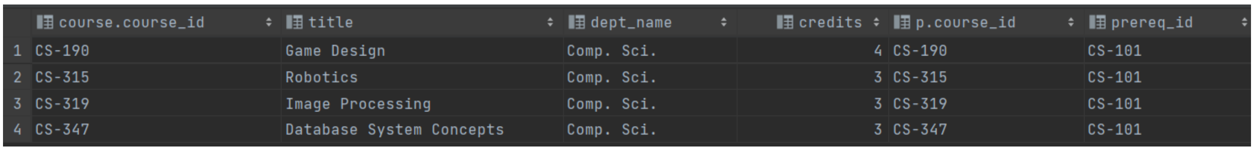


	id	name	dept_name	tot_cred
1	76653	Aoi	Elec. Eng.	60
2	98988	Tanaka	Biology	120

g. Find all courses which have course ‘CS-101’ as their prerequisite;



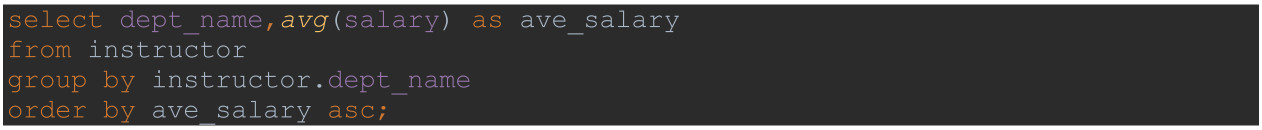
```
select *
from course join prereq p on course.course_id = p.course_id
where prereq_id = 'CS-101';
```



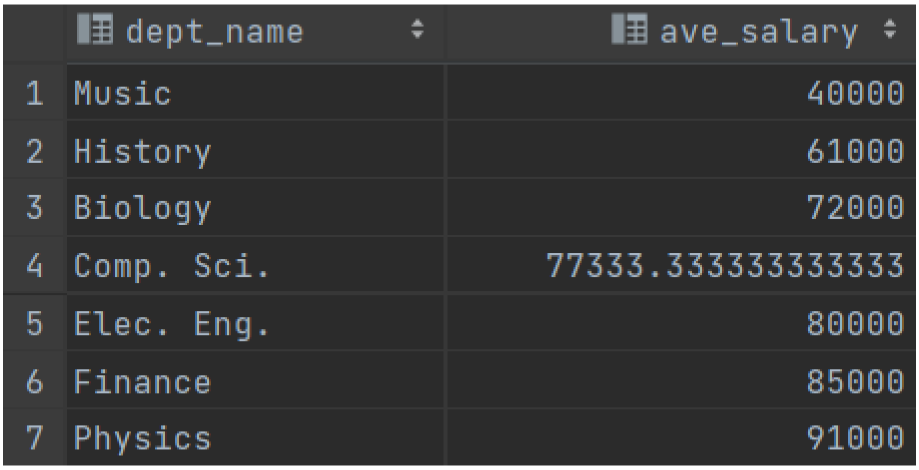
	course.course_id	title	dept_name	credits	p.course_id	prereq_id
1	CS-190	Game Design	Comp. Sci.	4	CS-190	CS-101
2	CS-315	Robotics	Comp. Sci.	3	CS-315	CS-101
3	CS-319	Image Processing	Comp. Sci.	3	CS-319	CS-101
4	CS-347	Database System Concepts	Comp. Sci.	3	CS-347	CS-101

2. Write the following queries in SQL, using the university schema:

a. For each department, find the average salary of instructors in that department and list them in ascending order. Assume that every department has at least one instructor;

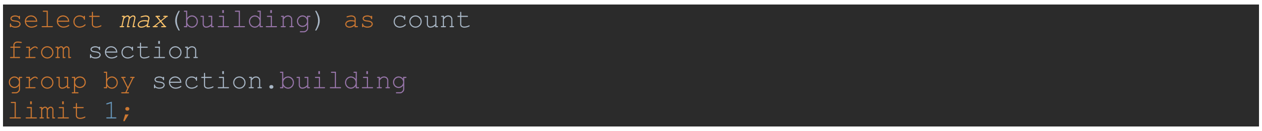


```
select dept_name,avg(salary) as ave_salary
from instructor
group by instructor.dept_name
order by ave_salary asc;
```

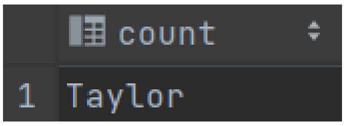


	dept_name	ave_salary
1	Music	40000
2	History	61000
3	Biology	72000
4	Comp. Sci.	77333.333333333333
5	Elec. Eng.	80000
6	Finance	85000
7	Physics	91000

b. Find the building where the biggest number of courses takes place;

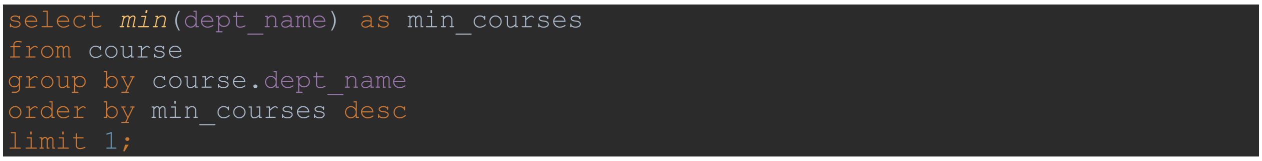


```
select max(building) as count
from section
group by section.building
limit 1;
```

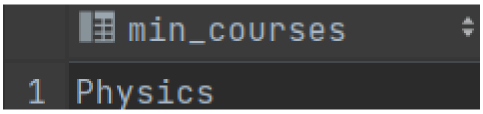


	count
1	Taylor

c. Find the department with the lowest number of courses offered;

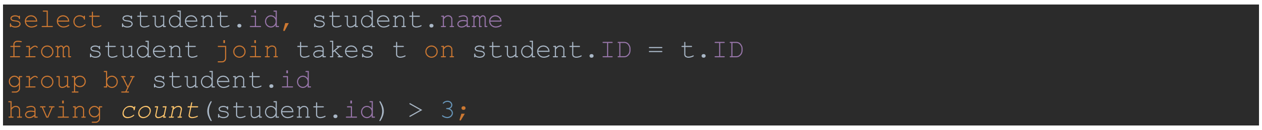


```
select min(dept_name) as min_courses
from course
group by course.dept_name
order by min_courses desc
limit 1;
```

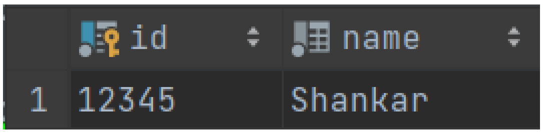


	min_courses
1	Physics

d. Find the ID and name of each student who has taken more than 3 courses from the Computer Science department;

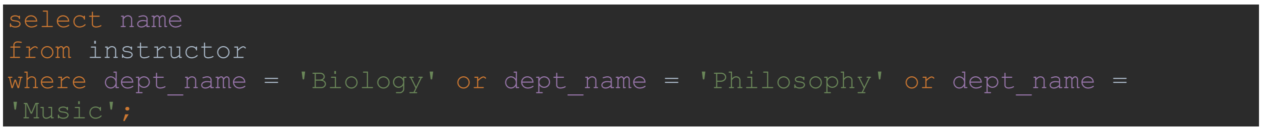


```
select student.id, student.name
from student join takes t on student.ID = t.ID
group by student.id
having count(student.id) > 3;
```

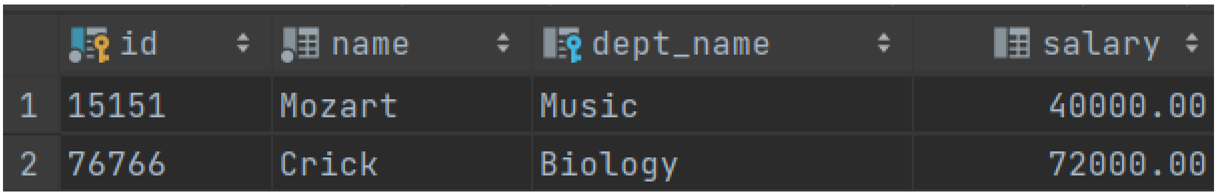


	id	name
1	12345	Shankar

e. Find all instructors who work either in Biology, Philosophy, or Music departments;

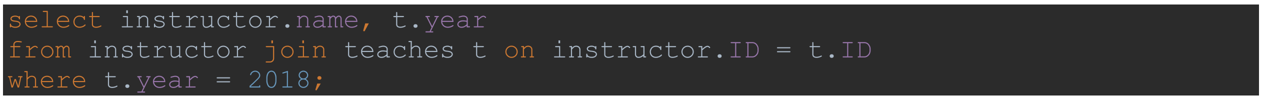


```
select name
from instructor
where dept_name = 'Biology' or dept_name = 'Philosophy' or dept_name =
'Music';
```

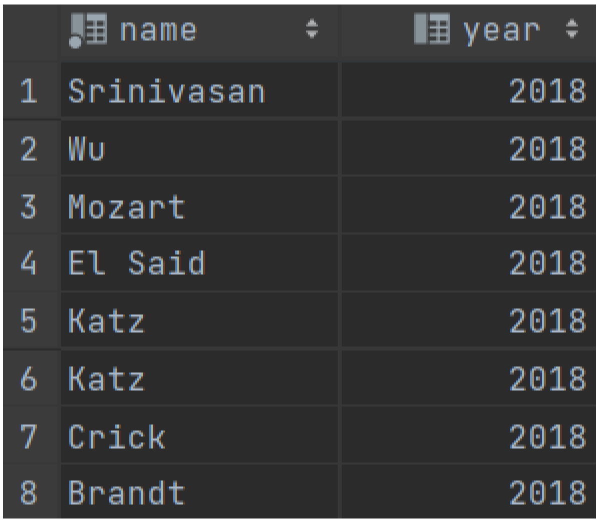


	id	name	dept_name	salary
1	15151	Mozart	Music	40000.00
2	76766	Crick	Biology	72000.00

f. Find all instructors who taught in the 2018 year but not in the 2017 year;



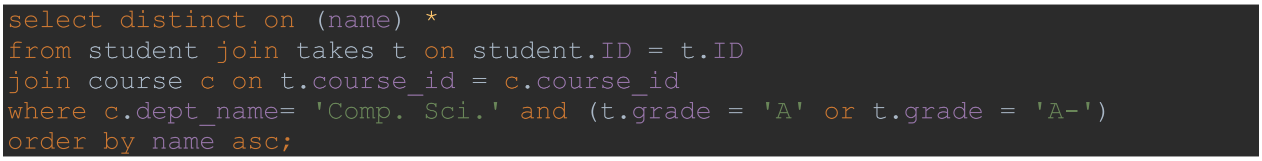
```
select instructor.name, t.year
from instructor join teaches t on instructor.ID = t.ID
where t.year = 2018;
```



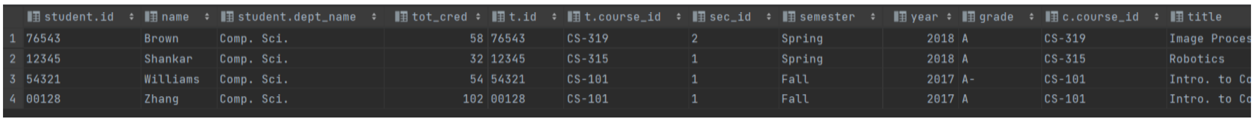
	name	year
1	Srinivasan	2018
2	Wu	2018
3	Mozart	2018
4	El Said	2018
5	Katz	2018
6	Katz	2018
7	Crick	2018
8	Brandt	2018

3. Write the following queries in SQL, using the university schema:

a. Find all students who have taken Comp. Sci. course and got an excellent grade (i.e., A, or A-) and sort them alphabetically;

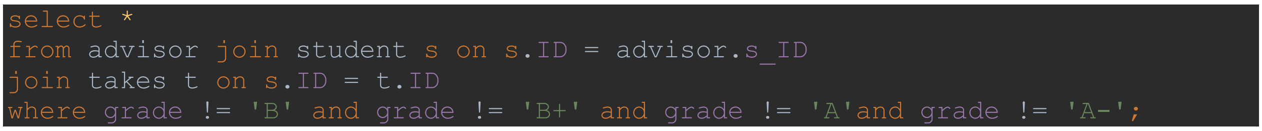


```
select distinct on (name) *
from student join takes t on student.ID = t.ID
join course c on t.course_id = c.course_id
where c.dept_name= 'Comp. Sci.' and (t.grade = 'A' or t.grade = 'A-')
order by name asc;
```

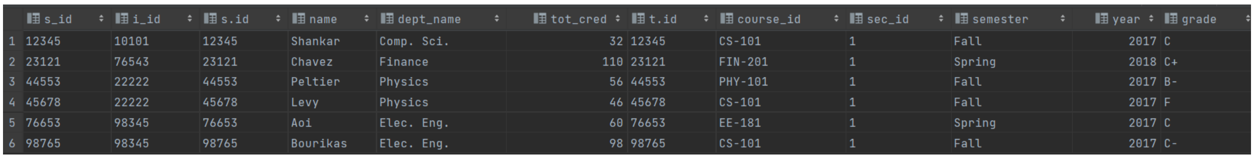


	student.id	name	student.dept_name	tot_cred	t.id	t.course_id	sec_id	semester	year	grade	c.course_id	title
1	76543	Brown	Comp. Sci.	58	76543	CS-319	2	Spring	2018	A	CS-319	Image Proces
2	12345	Shankar	Comp. Sci.	32	12345	CS-315	1	Spring	2018	A	CS-315	Robotics
3	54321	Williams	Comp. Sci.	54	54321	CS-101	1	Fall	2017	A-	CS-101	Intro. to Co
4	00128	Zhang	Comp. Sci.	102	00128	CS-101	1	Fall	2017	A	CS-101	Intro. to Co

b. Find all advisors of students who got grades lower than B on any class;

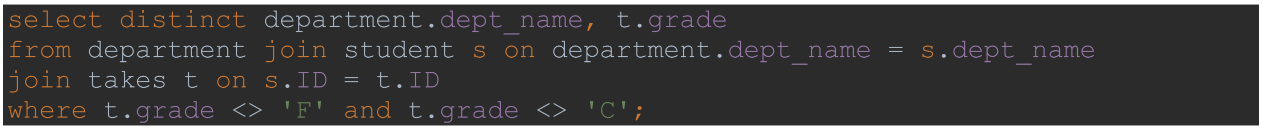


```
select *
from advisor join student s on s.ID = advisor.s_ID
join takes t on s.ID = t.ID
where grade != 'B' and grade != 'B+' and grade != 'A'and grade != 'A-';
```

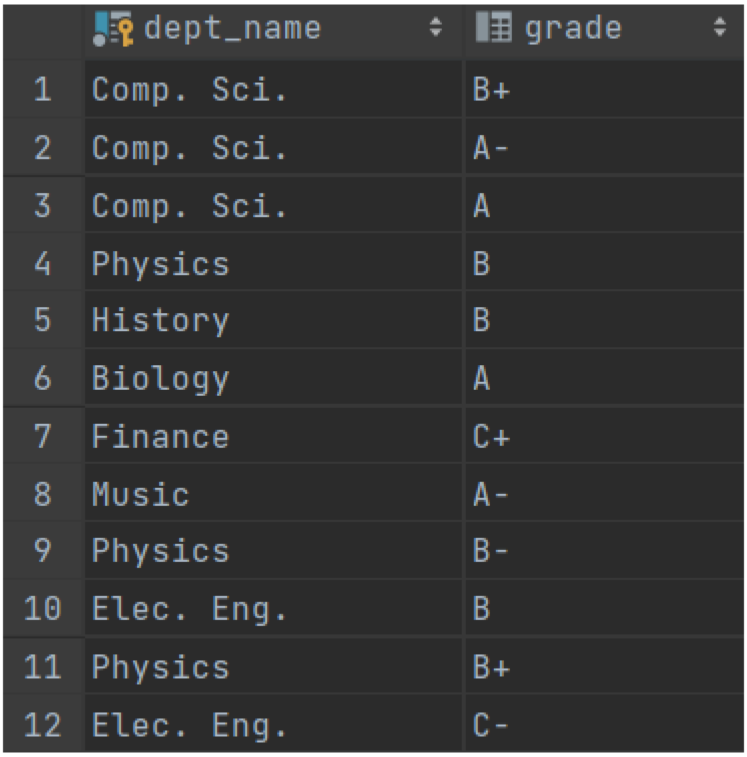


	s_id	i_id	s.id	name	dept_name	tot_cred	t.id	course_id	sec_id	semester	year	grade
1	12345	10101	12345	Shankar	Comp. Sci.	32	12345	CS-101	1	Fall	2017	C
2	23121	76543	23121	Chavez	Finance	110	23121	FIN-201	1	Spring	2018	C+
3	44553	22222	44553	Peltier	Physics	56	44553	PHY-101	1	Fall	2017	B-
4	45678	22222	45678	Levy	Physics	46	45678	CS-101	1	Fall	2017	F
5	76653	98345	76653	Aoi	Elec. Eng.	60	76653	EE-181	1	Spring	2017	C
6	98765	98345	98765	Bourikas	Elec. Eng.	98	98765	CS-101	1	Fall	2017	C-

c. Find all departments whose students have never gotten an F or C grade;

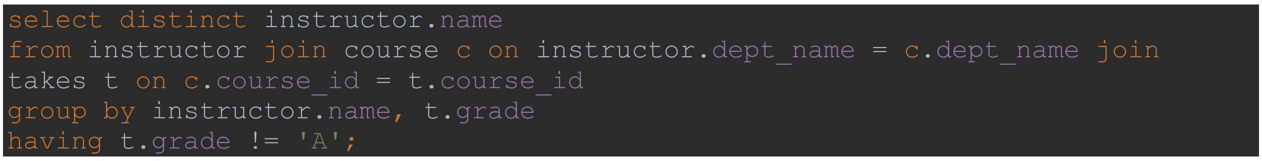


```
select distinct department.dept_name, t.grade
from department join student s on department.dept_name = s.dept_name
join takes t on s.ID = t.ID
where t.grade <> 'F' and t.grade <> 'C';
```

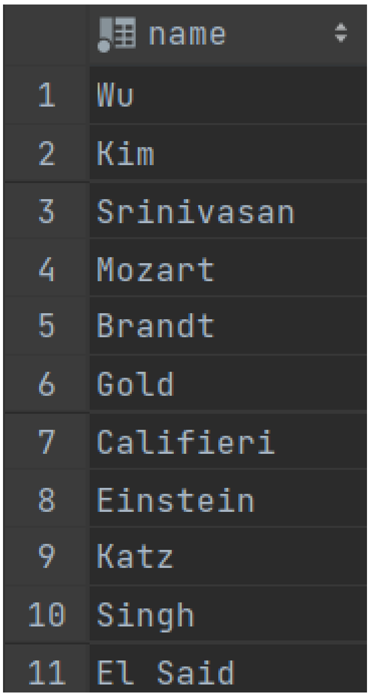


	dept_name	grade
1	Comp. Sci.	B+
2	Comp. Sci.	A-
3	Comp. Sci.	A
4	Physics	B
5	History	B
6	Biology	A
7	Finance	C+
8	Music	A-
9	Physics	B-
10	Elec. Eng.	B
11	Physics	B+
12	Elec. Eng.	C-

d. Find all instructors who have never given an A grade in any of the courses they taught;

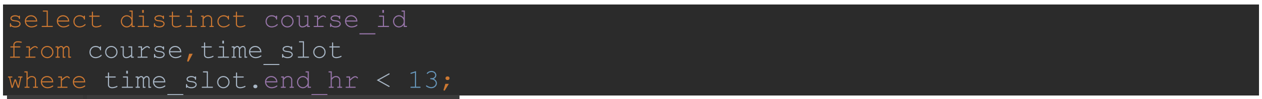


```
select distinct instructor.name
from instructor join course c on instructor.dept_name = c.dept_name join
takes t on c.course_id = t.course_id
group by instructor.name, t.grade
having t.grade != 'A';
```

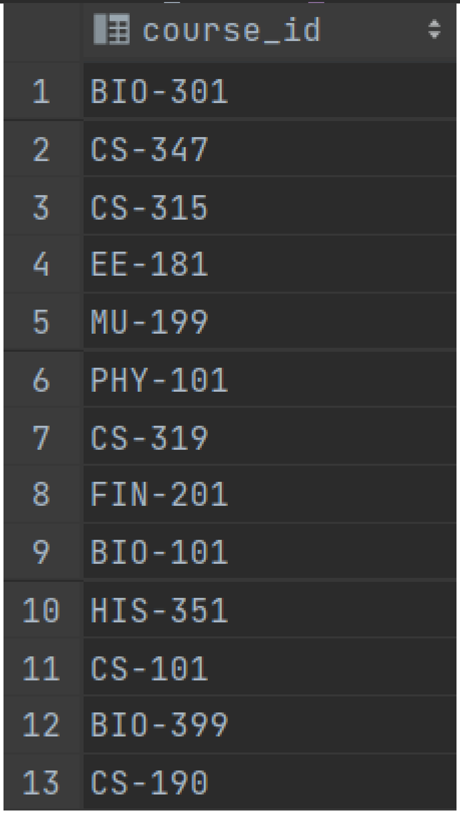


	name
1	Wu
2	Kim
3	Srinivasan
4	Mozart
5	Brandt
6	Gold
7	Califieri
8	Einstein
9	Katz
10	Singh
11	El Said

e. Find all courses offered in the morning hours (i.e., courses ending before 13:00)



```
select distinct course_id
from course,time_slot
where time_slot.end_hr < 13;
```



	course_id
1	BIO-301
2	CS-347
3	CS-315
4	EE-181
5	MU-199
6	PHY-101
7	CS-319
8	FIN-201
9	BIO-101
10	HIS-351
11	CS-101
12	BIO-399
13	CS-190