

Bissenova Anara / 20BD / Lab3

```
create table classroom
  (building      varchar(15),
   room_number   varchar(7),
   capacity       numeric(4,0),
   primary key (building, room_number)
  );
```

```
create table department
  (dept_name     varchar(20),
   building      varchar(15),
   budget        numeric(12,2) check (budget > 0),
   primary key (dept_name)
  );
```

```
create table course
  (course_id     varchar(8),
   title         varchar(50),
   dept_name     varchar(20),
   credits       numeric(2,0) check (credits > 0),
   primary key (course_id),
   foreign key (dept_name) references department (dept_name)
     on delete set null
  );
```

```
create table instructor
  (ID            varchar(5),
   name          varchar(20) not null,
   dept_name     varchar(20),
   salary        numeric(8,2) check (salary > 29000),
   primary key (ID),
   foreign key (dept_name) references department (dept_name)
     on delete set null
  );
```

```
create table section
  (course_id     varchar(8),
```

```

        sec_id          varchar(8),
semester          varchar(6)
        check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
year              numeric(4,0) check (year > 1701 and year < 2100),
building          varchar(15),
room_number       varchar(7),
time_slot_id      varchar(4),
primary key (course_id, sec_id, semester, year),
foreign key (course_id) references course (course_id)
        on delete cascade,
foreign key (building, room_number) references classroom (building, room_number)
        on delete set null
);

```

create table teaches

```

(ID          varchar(5),
course_id    varchar(8),
sec_id       varchar(8),
semester     varchar(6),
year         numeric(4,0),
primary key (ID, course_id, sec_id, semester, year),
foreign key (course_id, sec_id, semester, year) references section (course_id, sec_id, semester,
year)
        on delete cascade,
foreign key (ID) references instructor (ID)
        on delete cascade
);

```

create table student

```

(ID          varchar(5),
name         varchar(20) not null,
dept_name    varchar(20),
tot_cred     numeric(3,0) check (tot_cred >= 0),
primary key (ID),
foreign key (dept_name) references department (dept_name)
        on delete set null
);

```

```

create table takes
  (ID          varchar(5),
   course_id   varchar(8),
   sec_id      varchar(8),
   semester    varchar(6),
   year        numeric(4,0),
   grade       varchar(2),
   primary key (ID, course_id, sec_id, semester, year),
   foreign key (course_id, sec_id, semester, year) references section (course_id, sec_id, semester,
year)
      on delete cascade,
   foreign key (ID) references student (ID)
      on delete cascade
  );

```

```

create table advisor
  (s_ID        varchar(5),
   i_ID        varchar(5),
   primary key (s_ID),
   foreign key (i_ID) references instructor (ID)
      on delete set null,
   foreign key (s_ID) references student (ID)
      on delete cascade
  );

```

```

create table time_slot
  (time_slot_id  varchar(4),
   day           varchar(1),
   start_hr      numeric(2) check (start_hr >= 0 and start_hr < 24),
   start_min     numeric(2) check (start_min >= 0 and start_min < 60),
   end_hr        numeric(2) check (end_hr >= 0 and end_hr < 24),
   end_min       numeric(2) check (end_min >= 0 and end_min < 60),
   primary key (time_slot_id, day, start_hr, start_min)
  );

```

```

create table prereq
  (course_id    varchar(8),
   prereq_id    varchar(8),

```

```

    primary key (course_id, prereq_id),
    foreign key (course_id) references course (course_id)
        on delete cascade,
    foreign key (prereq_id) references course (course_id)
);

```

```

delete from prereq;
delete from time_slot;
delete from advisor;
delete from takes;
delete from student;
delete from teaches;
delete from section;
delete from instructor;
delete from course;
delete from department;
delete from classroom;
insert into classroom values ('Packard', '101', '500');
insert into classroom values ('Painter', '514', '10');
insert into classroom values ('Taylor', '3128', '70');
insert into classroom values ('Watson', '100', '30');
insert into classroom values ('Watson', '120', '50');
insert into department values ('Biology', 'Watson', '90000');
insert into department values ('Comp. Sci.', 'Taylor', '100000');
insert into department values ('Elec. Eng.', 'Taylor', '85000');
insert into department values ('Finance', 'Painter', '120000');
insert into department values ('History', 'Painter', '50000');
insert into department values ('Music', 'Packard', '80000');
insert into department values ('Physics', 'Watson', '70000');
insert into course values ('BIO-101', 'Intro. to Biology', 'Biology', '4');
insert into course values ('BIO-301', 'Genetics', 'Biology', '4');
insert into course values ('BIO-399', 'Computational Biology', 'Biology', '3');
insert into course values ('CS-101', 'Intro. to Computer Science', 'Comp. Sci.', '4');
insert into course values ('CS-190', 'Game Design', 'Comp. Sci.', '4');
insert into course values ('CS-315', 'Robotics', 'Comp. Sci.', '3');
insert into course values ('CS-319', 'Image Processing', 'Comp. Sci.', '3');
insert into course values ('CS-347', 'Database System Concepts', 'Comp. Sci.', '3');
insert into course values ('EE-181', 'Intro. to Digital Systems', 'Elec. Eng.', '3');

```

```

insert into course values ('FIN-201', 'Investment Banking', 'Finance', '3');
insert into course values ('HIS-351', 'World History', 'History', '3');
insert into course values ('MU-199', 'Music Video Production', 'Music', '3');
insert into course values ('PHY-101', 'Physical Principles', 'Physics', '4');
insert into instructor values ('10101', 'Srinivasan', 'Comp. Sci.', '65000');
insert into instructor values ('12121', 'Wu', 'Finance', '90000');
insert into instructor values ('15151', 'Mozart', 'Music', '40000');
insert into instructor values ('22222', 'Einstein', 'Physics', '95000');
insert into instructor values ('32343', 'El Said', 'History', '60000');
insert into instructor values ('33456', 'Gold', 'Physics', '87000');
insert into instructor values ('45565', 'Katz', 'Comp. Sci.', '75000');
insert into instructor values ('58583', 'Califieri', 'History', '62000');
insert into instructor values ('76543', 'Singh', 'Finance', '80000');
insert into instructor values ('76766', 'Crick', 'Biology', '72000');
insert into instructor values ('83821', 'Brandt', 'Comp. Sci.', '92000');
insert into instructor values ('98345', 'Kim', 'Elec. Eng.', '80000');
insert into section values ('BIO-101', '1', 'Summer', '2017', 'Painter', '514', 'B');
insert into section values ('BIO-301', '1', 'Summer', '2018', 'Painter', '514', 'A');
insert into section values ('CS-101', '1', 'Fall', '2017', 'Packard', '101', 'H');
insert into section values ('CS-101', '1', 'Spring', '2018', 'Packard', '101', 'F');
insert into section values ('CS-190', '1', 'Spring', '2017', 'Taylor', '3128', 'E');
insert into section values ('CS-190', '2', 'Spring', '2017', 'Taylor', '3128', 'A');
insert into section values ('CS-315', '1', 'Spring', '2018', 'Watson', '120', 'D');
insert into section values ('CS-319', '1', 'Spring', '2018', 'Watson', '100', 'B');
insert into section values ('CS-319', '2', 'Spring', '2018', 'Taylor', '3128', 'C');
insert into section values ('CS-347', '1', 'Fall', '2017', 'Taylor', '3128', 'A');
insert into section values ('EE-181', '1', 'Spring', '2017', 'Taylor', '3128', 'C');
insert into section values ('FIN-201', '1', 'Spring', '2018', 'Packard', '101', 'B');
insert into section values ('HIS-351', '1', 'Spring', '2018', 'Painter', '514', 'C');
insert into section values ('MU-199', '1', 'Spring', '2018', 'Packard', '101', 'D');
insert into section values ('PHY-101', '1', 'Fall', '2017', 'Watson', '100', 'A');
insert into teaches values ('10101', 'CS-101', '1', 'Fall', '2017');
insert into teaches values ('10101', 'CS-315', '1', 'Spring', '2018');
insert into teaches values ('10101', 'CS-347', '1', 'Fall', '2017');
insert into teaches values ('12121', 'FIN-201', '1', 'Spring', '2018');
insert into teaches values ('15151', 'MU-199', '1', 'Spring', '2018');
insert into teaches values ('22222', 'PHY-101', '1', 'Fall', '2017');
insert into teaches values ('32343', 'HIS-351', '1', 'Spring', '2018');

```

```

insert into teaches values ('45565', 'CS-101', '1', 'Spring', '2018');
insert into teaches values ('45565', 'CS-319', '1', 'Spring', '2018');
insert into teaches values ('76766', 'BIO-101', '1', 'Summer', '2017');
insert into teaches values ('76766', 'BIO-301', '1', 'Summer', '2018');
insert into teaches values ('83821', 'CS-190', '1', 'Spring', '2017');
insert into teaches values ('83821', 'CS-190', '2', 'Spring', '2017');
insert into teaches values ('83821', 'CS-319', '2', 'Spring', '2018');
insert into teaches values ('98345', 'EE-181', '1', 'Spring', '2017');
insert into student values ('00128', 'Zhang', 'Comp. Sci.', '102');
insert into student values ('12345', 'Shankar', 'Comp. Sci.', '32');
insert into student values ('19991', 'Brandt', 'History', '80');
insert into student values ('23121', 'Chavez', 'Finance', '110');
insert into student values ('44553', 'Peltier', 'Physics', '56');
insert into student values ('45678', 'Levy', 'Physics', '46');
insert into student values ('54321', 'Williams', 'Comp. Sci.', '54');
insert into student values ('55739', 'Sanchez', 'Music', '38');
insert into student values ('70557', 'Snow', 'Physics', '0');
insert into student values ('76543', 'Brown', 'Comp. Sci.', '58');
insert into student values ('76653', 'Aoi', 'Elec. Eng.', '60');
insert into student values ('98765', 'Bourikas', 'Elec. Eng.', '98');
insert into student values ('98988', 'Tanaka', 'Biology', '120');
insert into takes values ('00128', 'CS-101', '1', 'Fall', '2017', 'A');
insert into takes values ('00128', 'CS-347', '1', 'Fall', '2017', 'A-');
insert into takes values ('12345', 'CS-101', '1', 'Fall', '2017', 'C');
insert into takes values ('12345', 'CS-190', '2', 'Spring', '2017', 'A');
insert into takes values ('12345', 'CS-315', '1', 'Spring', '2018', 'A');
insert into takes values ('12345', 'CS-347', '1', 'Fall', '2017', 'A');
insert into takes values ('19991', 'HIS-351', '1', 'Spring', '2018', 'B');
insert into takes values ('23121', 'FIN-201', '1', 'Spring', '2018', 'C+');
insert into takes values ('44553', 'PHY-101', '1', 'Fall', '2017', 'B-');
insert into takes values ('45678', 'CS-101', '1', 'Fall', '2017', 'F');
insert into takes values ('45678', 'CS-101', '1', 'Spring', '2018', 'B+');
insert into takes values ('45678', 'CS-319', '1', 'Spring', '2018', 'B');
insert into takes values ('54321', 'CS-101', '1', 'Fall', '2017', 'A-');
insert into takes values ('54321', 'CS-190', '2', 'Spring', '2017', 'B+');
insert into takes values ('55739', 'MU-199', '1', 'Spring', '2018', 'A-');
insert into takes values ('76543', 'CS-101', '1', 'Fall', '2017', 'A');
insert into takes values ('76543', 'CS-319', '2', 'Spring', '2018', 'A');

```

```

insert into takes values ('76653', 'EE-181', '1', 'Spring', '2017', 'C');
insert into takes values ('98765', 'CS-101', '1', 'Fall', '2017', 'C-');
insert into takes values ('98765', 'CS-315', '1', 'Spring', '2018', 'B');
insert into takes values ('98988', 'BIO-101', '1', 'Summer', '2017', 'A');
insert into takes values ('98988', 'BIO-301', '1', 'Summer', '2018', null);
insert into advisor values ('00128', '45565');
insert into advisor values ('12345', '10101');
insert into advisor values ('23121', '76543');
insert into advisor values ('44553', '22222');
insert into advisor values ('45678', '22222');
insert into advisor values ('76543', '45565');
insert into advisor values ('76653', '98345');
insert into advisor values ('98765', '98345');
insert into advisor values ('98988', '76766');
insert into time_slot values ('A', 'M', '8', '0', '8', '50');
insert into time_slot values ('A', 'W', '8', '0', '8', '50');
insert into time_slot values ('A', 'F', '8', '0', '8', '50');
insert into time_slot values ('B', 'M', '9', '0', '9', '50');
insert into time_slot values ('B', 'W', '9', '0', '9', '50');
insert into time_slot values ('B', 'F', '9', '0', '9', '50');
insert into time_slot values ('C', 'M', '11', '0', '11', '50');
insert into time_slot values ('C', 'W', '11', '0', '11', '50');
insert into time_slot values ('C', 'F', '11', '0', '11', '50');
insert into time_slot values ('D', 'M', '13', '0', '13', '50');
insert into time_slot values ('D', 'W', '13', '0', '13', '50');
insert into time_slot values ('D', 'F', '13', '0', '13', '50');
insert into time_slot values ('E', 'T', '10', '30', '11', '45 ');
insert into time_slot values ('E', 'R', '10', '30', '11', '45 ');
insert into time_slot values ('F', 'T', '14', '30', '15', '45 ');
insert into time_slot values ('F', 'R', '14', '30', '15', '45 ');
insert into time_slot values ('G', 'M', '16', '0', '16', '50');
insert into time_slot values ('G', 'W', '16', '0', '16', '50');
insert into time_slot values ('G', 'F', '16', '0', '16', '50');
insert into time_slot values ('H', 'W', '10', '0', '12', '30');
insert into prereq values ('BIO-301', 'BIO-101');
insert into prereq values ('BIO-399', 'BIO-101');
insert into prereq values ('CS-190', 'CS-101');
insert into prereq values ('CS-315', 'CS-101');

```

```
insert into prereq values ('CS-319', 'CS-101');
insert into prereq values ('CS-347', 'CS-101');
insert into prereq values ('EE-181', 'PHY-101');
```

"LAB3-TASK1:

- a. Find all courses worth more than 3 credits;
- b. Find all classrooms situated either in **Watson** or **Packard** buildings;
- c. Find all courses offered by the Computer Science department;
- d. Find all courses offered during fall;
- e. Find all students who have more than 45 credits but less than 90;
- f. Find all students whose names end with vowels;
- g. Find all courses which have course **CS-101** as their prerequisite;"

```
select * from course where credits > 3;
select * from classroom where building = 'Watson' or building = 'Packard';
select title, course_id from course where dept_name = 'Comp. Sci.';
select course_id, title from course where course_id in (select course_id from section where semester =
'Fall');
select * from student where tot_cred between 45 and 90;
select name from student where name like ('%a') or name like ('%e') or name like ('%u') or name like
('%o') or name like ('%i') or name like ('%y');
select course_id, title from course where course_id in (select course_id from prereq where prereq_id =
'CS-101');
```

"LAB3-TASK2:

- 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;
- b. Find the building where the biggest number of courses takes place;
- c. Find the department with the lowest number of courses offered;
- d. Find the ID and name of each student who has taken more than 3 courses from the Computer Science department;
- e. Find all instructors who work either in Biology, Philosophy, or Music departments;
- f. Find all instructors who taught in the 2018 year but not in the 2017 year;"

```
select dept_name, avg(salary) as avg_salary
from instructor group by dept_name order by avg_salary;
```



```
select department.building, count(department.building) from department, course
where course.dept_name = department.dept_name group by department.building
having count(department.building) = (select max(num) from (select count(department.building) as num
from course, department where department.dept_name = course.dept_name group by department.building) as
tbl1);
```

```
select dept_name, count(dept_name) from course group by dept_name
having count(dept_name) = (select min(num) from (select count(course.dept_name) as num
from course group by course.dept_name) as tbl1);
```

```
select student.id, name from student, takes, course
where student.ID = takes.ID and course.course_id = takes.course_id and course.dept_name = 'Comp. Sci.'
and credits > 3;
```

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

```
select teaches.ID, name from instructor, teaches
where teaches.ID = instructor.ID and year = '2018' and instructor.ID not in (select distinct
teaches.ID from teaches where year = '2017');
```

"LAB3 - TASK3:

- a. Find all students who have taken Comp. Sci. course and got an excellent grade (i.e., A, or A-) and sort them alphabetically;
- b. Find all advisors of students who got grades lower than B on any class;
- c. Find all departments whose students have never gotten an F or C grade;
- d. Find all instructors who have never given an A grade in any of the courses they taught;
- e. Find all courses offered in the morning hours (i.e., courses ending before 13:00)"

```
select distinct name from student, takes
where student.ID = takes.ID and course_id like 'CS%' and (grade = 'A' or grade = 'A-')
order by name;
```

```
select distinct name from advisor, takes, instructor
```

```
where s_ID = takes.ID and i_ID = instructor.id and (grade = 'C' or grade = 'B-' or grade = 'C+' or grade = 'C-' or grade = 'F');
```

```
select distinct dept_name from student, takes  
where student.ID = takes.ID and takes.ID not in  
(select distinct takes.id from takes where grade = 'F' or grade = 'C' or grade is NULL);
```

```
select distinct instructor.ID, name from instructor, teaches, takes  
where instructor.ID = teaches.ID and takes.course_id = teaches.course_id and takes.sec_id =  
teaches.sec_id and takes.year = teaches.year  
and grade != 'A' ;
```

```
select distinct title from section, time_slot, course  
where section.time_slot_id = time_slot.time_slot_id and course.course_id = section.course_id and  
end_hr < 13;
```