3.12 Write the SQL statements using the university schema to perform the following operations:

a. Create a new course "CS-001", titled "Weekly Seminar", with 0 credits.

b. Create a section of this course in Fall 2017, with *sec_id* of 1, and with the location of this section not yet specified.

c. Enroll every student in the Comp. Sci. department in the above section.

d. Delete enrollments in the above section where the student's ID is 12345.

e. Delete the course CS-001. What will happen if you run this **delete** statement without first deleting offerings (sections) of this course?

f. Delete all *takes* tuples corresponding to any section of any course with the word "advanced" as a part of the title; ignore case when matching the word with the title.

a.

```
insert into course(course_id,title,dept_name,credits)
    values('CS-001','Weekly Seminar',null,0)
```

b.

```
insert into section(course_id,sec_id,semester,year,building,room_number
,time_slot_id)
    values('CS-001',1,'Fall',2017,null,null,null)
```

c.

```
insert into takes(ID,course_id,sec_id,semester,year,grade)
    select ID,'CS-001',1,'Fall',2017,null
    from student
    where dept_name='Comp.Sci.'
```

d.

```
delete from takes
    where ID=12345 and course_id='CS-
001' and sec_id=1 and semester='Fall' and year='2017'
```

e.

```
delete from course
    where course_id='CS-001'
```

如果没有先删除 section 中的这门课程，则会破坏外码的完整性约束。正确的删除方式应如下所示：

```
delete from teaches
    where course_id='CS-101'
delete from takes
    where course_id='CS-101'
delete from section
    where course_id='CS-101'
delete from prereq
    where course_id='CS-101'
delete from course
    where course_id='CS-101'
```

f.

```
delete from takes
    where course_id in
    (select course_id
    from course
    where title like '%advanced%')
```

3.16 Consider the employee database of Figure 3.19, where the primary keys are underlined. Give an expression in SQL for each of the following queries.

a. Find ID and name of each employee who lives in the same city as the location of the company for which the employee works.

b. Find ID and name of each employee who lives in the same city and on the same street as does her or his manager.

c. Find ID and name of each employee who earns more than the average salary of all employees of her or his company.

d. Find the company that has the smallest payroll.

a.

```
select employee.ID,employee.person_name
from employee,works,company
where employee.ID=works.ID and works.company_name=company.company_name
and employee.city=company.city
```

b.

```
select e1.ID,e2.person_name
from employee as e1,employee as e2,manages as m
where e1.ID=m.ID and m.manager_id=e2.ID and e1.street=e2.street and e1.
city=e2.city
```

c.

```
select e.ID,e.name
from employee as e,works as w1
where e.ID=w1.ID and w1.salary>(select avg(w2.salary)
                                from works as w2
                                where w1.company_name=w2.company_name)
```

d.

```
select company_name
from works
group by company_name
having sum(salary)<=all(select sum(salary)
                        from works
                        group by company_name)
```