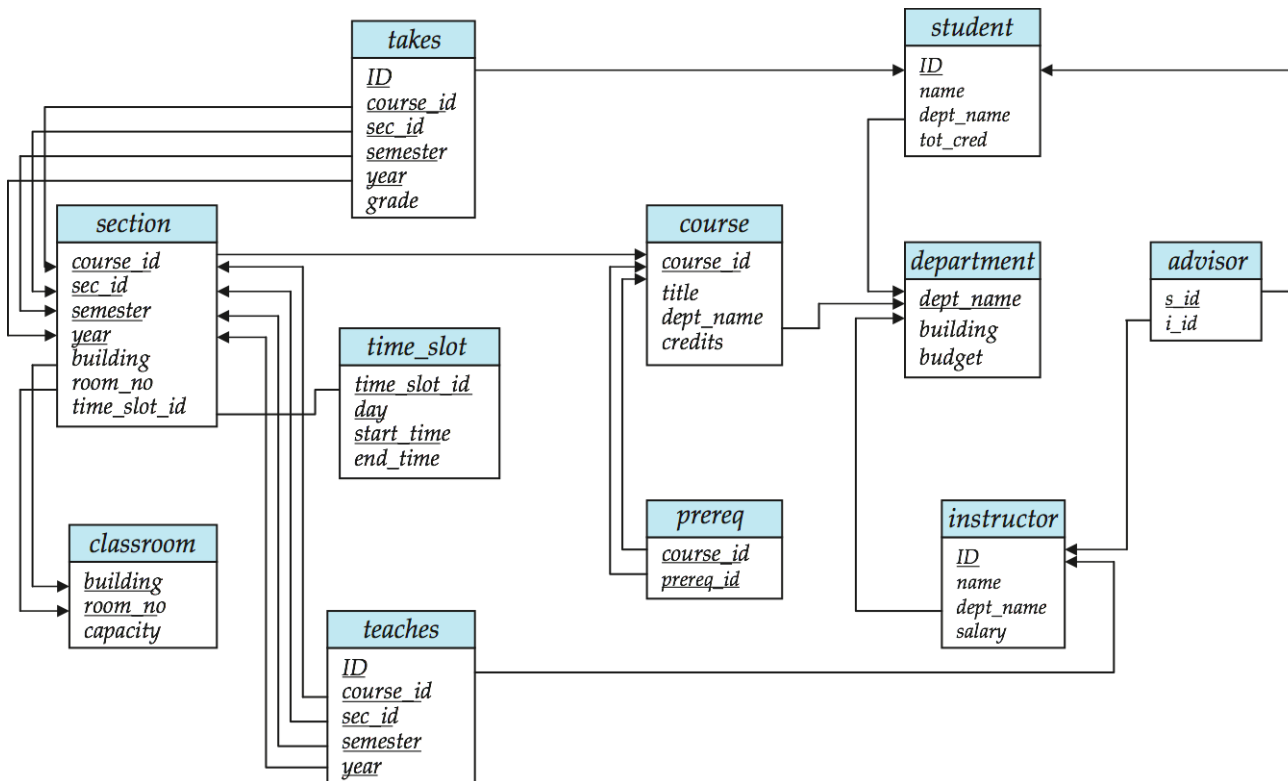


***** DON'T FORGET TO TURN THIS IN *****

Your Name:



- For each of the questions, think about how to write an SQL query. If having difficulties getting started, break the query into pieces by thinking of intermediate results that you would like to use.

0. Find student-instructor pairs s.t. the student has taken multiple courses with the instructor

The intermediate table that helps here is a table: T(s_id, i_id), where an entry here indicates that the student s_id took a course with instructor i_id. Once we have this table, the second part becomes easier.

```
with T as (
    select takes.ID as s_id, teaches.ID as i_id
    from takes, teaches
    where takes.course_id = teaches.course_id and takes.sec_id = teaches.sec_id and
    takes.semester = teaches.semester and takes.year = teaches.year
)
select s_id, i_id
from T
group by s_id, i_id
having count(*) >= 2;
```

If we need student and instructor names, we can do a further join:

```

with T as (
    select takes.ID as s_id, teaches.ID as i_id
    from takes, teaches
    where takes.course_id = teaches.course_id and takes.sec_id = teaches.sec_id and
    takes.semester = teaches.semester and takes.year = teaches.year
),
T2 as (
    select s_id, i_id
    from T
    group by s_id, i_id
    having count(*) >= 2
)
select student.name, instructor.name
from T2, instructor, student
where T2.s_id = student.ID and T2.i_id = instructor.id

```

1. Find students who took a course with the instructor they are advised by

All we need to do is “intersect” the T table above with the advisor table.

```

with T as ( ..... )

select * from T intersect select * from advisor;

```

If we need the student/instructor names, we can do a further join as above.

2. Find classrooms that have hosted most classes so far

```

with T3 as (
    select building, roomno, count(*) as num_classes
    from section
    group by building, roomno
)
select building, roomno
from T3
where T3.num_classes = (select max(num_classes) from T3);

```

3. Identify students who took a course but not one of its pre-req

There are usually quite a few ways to do most such queries. The following is one way to do it that we discussed in the class. There is also a way to do it using joins.

First, let’s write a query to find whether this is true for a specific student ID ‘X’ and a specific course_id ‘Y’. Consider the following query:

```

select prereq_id as c_id
from prereq
where course_id = 'Y'
except
select course_id as c_id
from takes
where ID = 'X';

```

This query finds the prerequisites for 'Y' and subtracts the courses taken by 'X' from it. So if the result is non-empty, then there is a prerequisite that the student hasn't taken (if the student has taken all pre-requisites of 'Y', then the result will be empty).

Now, we can use this to answer our overall query, by using the above query as a effectively a subroutine:

```

select *
from takes
where not exists (
    select prereq_id as c_id
    from prereq
    where course_id = takes.course_id
    except
    select course_id as c_id
    from takes
    where ID = takes.ID
);

```

This is probably not the most efficient way to do this, but hopefully the database query optimizer can figure out how to execute it efficiently.

4. Find classrooms that were over capacity, i.e., had more students enrolled than the capacity

We need a table that lists the number of students per classroom per section.

```

with T4 as (
    select course_id, sec_id, semester, year, building, roomno, count(*) as num_enrolled
    from section natural join takes
    group by course_id, sec_id, semester, year, building, roomno
)
select *
from T4 natural join classroom
where T4.num_enrolled > classroom.capacity;

```