

## 2. Formulate SQL queries to answer each of the following questions.

1. What are the names and total credits of students majoring in math? (Assume dept name refers to major.)

```
select name, tot_cred from student where dept_name='Math';
```

2. What are the titles and credit hours of courses taught by Professor Einstein?

```
select distinct(c.title), c.credits from instructor i join teaches using (ID) join
course c using (course_id) where i.name = 'Einstein';
```

3. What are the titles of courses taught in Purcell?

```
select distinct(title) from section join course using (course_id) where
building = 'Purcell';
```

4. Suppose some courses are on MW and some are on MWF. What are the titles of courses taught on Mondays in Smith Hall?

```
select distinct(title) from time_slot join section using (time_slot_id) join
course using (course_id) where building = 'Smith Hall' and day like '%M%';
```

5. What are the titles of courses that are prerequisites for 400-level CIS courses?

```
select distinct(p.title) from course c natural join prereq join course p on
prereq_id = p.course_id where c.course_id like 'CIS-4%';
```

6. What is the GPA of the student with ID=1? (Assume there are only five letter grades with values A=4, B=3, C=2, D=1, and F=0, and GPA is the sum of course credits × course grade value divided by total credits.)

```
select sum(credits * (case grade when 'F' then 0 else ascii('E') - ascii(grade) end)) /
sum(credits) as GPA from takes join course using(course_id) join student s
using (ID) where ID=1 group by ID;
```

7. What courses were taught in either Spring of 2014 or Fall of 2014?

```
select * from section where year=2014 and (semester='Spring' or semester='Fall');
```

8. What courses were taught in both Spring of 2014 and Fall of 2014?

```
select * from section s1 join section s2 using (course_id) where
s1.semester='Spring' and s2.semester='Fall' and
s1.year = 2014 and s2.year = 2014;
```

9. What are the average salaries of instructors in each department?

```
select dept_name, avg(salary) as 'average salaries' from instructor
group by dept_name;
```

10. How many open seats are there in section 010 of CISC437 in Spring 2015?

```
select capacity - count(ID) from takes natural join section join classroom using
(building, room_no) where sec_id = 10 and course_id = 'CIS-437' and
semester='Spring' and year=2015 group by course_id, sec_id, semester, year;
```

11. How many students does each instructor in computer science advise?

```
select i.name, count(s.ID) from instructor i join advisor a on i.ID=a.i_id
join student s on s.ID = a.s_id where i.dept_name='Comp. Sci.' group by i.ID;
```

12. What are the average GPAs for students by department?

```
select dept_name, avg(GPA) from
  (select ID, s.dept_name,
    sum(credits * (case grade when 'F' then 0 else ascii('E') - ascii(grade) end)) /
    sum(credits) as GPA from takes join course using(course_id) join student s
  using (ID) group by ID)
as X group by dept_name;
```

#### 4. On data consistency.

(a) Why are semester and year part of the primary key of the teaches, takes, and section relations? If they were not part of the key, how would that change the data that could be stored?

A: Because two different courses can have the same course\_id and sec\_id, but with different semester and year. It is the tuple (course\_id, sec\_id, semester, year) all together that determines a specific course. Therefore semester and year should be included to identify the course. If they were not part of the key, we probably need to replace them into a field called "when" as a foreign key to another new table which lists all the possible semesters and years for every course.

(b) Suppose we want to store information about university buildings (such as address, abbreviated name, full name, year built, etc.). How could we add this information without introducing any redundancy in the data? What would we have to change in the schema and foreign key constraints?

A: We need first to create a new table called "buildings" with all the interested fields (such as address, abbreviated name, full name, year built, etc.) in it, set "full name" as primary key. Insert all buildings' information values into that table, and then alter every table that has a "building" field, add a foreign key constraint to each "building" with a reference to a "full name" in particular item of "buildings" table.

(c) Find three cases in the university database diagram in which valid data can be stored that would be considered invalid in a real university. Give examples of data that is valid for the database but invalid in reality.

A: Three cases I find are:

- Student's department and his instructor's department are not the same. Example values:

Say one student has values (1, 'Wei', 'Elec. Eng.', 27), his instructor is (123, 'Fang', 'Finance', 10000).

- Total credits of a student can be inconsistent with the value of tot\_cred.

One example would be: One student has took only two courses, both of the two courses are of credit 4, but this student can still have a tot\_cred field with value 10 which is not equal to 8.

- One instructor can teach courses that have time slot overlap. One example:

A instructor with values (123, 'Fang', 'Finance', 10000), whose ID is pointed to two items in teaches: (123, 'FIS-100', 1, 'Fall', 2015) and (123, 'FIS-216', 1, 'Fall', 2015), and the two reference two section items with values ('FIS-100', 1, 'Fall', 2015, 'Alfred Lerner Hall', 101, 1) and ('FIS-216', 1, 'Fall', 2015, 'Alfred Lerner Hall', 102, 2) with their time slots overlapped: Say the two conflict time\_slot items have values (1, 'M', 9:00:00, 9:50:00) and (2, 'M', 9:30:00, 10:20:00).