SQL Assignment One

1. **Find the names of all the instructors from Biology department**
    a. SELECT instructor.name FROM instructor WHERE dept_name = 'Biology';

2. **Find the names of courses in Computer Science department which have 3 credits**
    a. SELECT course.title FROM course WHERE credits = 3;

3. **For the student with ID 30397, show all course_id and title of all courses registered for by the student**
    a.
```
SELECT c.course_id, c.title
FROM course c
JOIN takes t ON c.course_id = t.course_id
WHERE ID = '30397';
```

4. **As above, but show the total number of credits for such courses (taken by that student). -- Don't display the tot_creds value from the student table, you should use SQL aggregation on courses taken by the student.**
    a.
```
SELECT c.course_id, c.title, COUNT(c.course_id) AS sum
FROM course c
JOIN takes t ON c.course_id = t.course_id
WHERE t.ID = '30397'
GROUP BY c.title, c.course_id;
```

5. **Now display the total credits (over all courses) for each of the students having more than 85 in total credits, along with the ID of the student; -- don't bother about the name of the student. -- don't bother about students who have not registered**
    a.
```
SELECT t.id, SUM(c.credits)
FROM takes t
NATURAL JOIN course c
GROUP BY t.id
HAVING SUM(c.credits) > 85;
```

6. **Find the names of all students who have taken any course at the Languages department with the grade 'A+' (there should be no duplicate names)**

a.

```sql
SELECT DISTINCT s.name
FROM student s NATURAL JOIN takes t
WHERE dept_name = 'Languages' AND t.grade = 'A+'
ORDER BY s.name;
```

7. **Display the IDs of all instructors from the Marketing department who have never taught a course (interpret "taught" as "taught or is scheduled to teach")**

a.

```sql
select i.id
from instructor i
natural join department d
left join teaches t ON i.id = t.id
where dept_name = 'Marketing' AND t.course_id IS NULL;
```

8. **As above, but display the names of the instructors also, not just the IDs.**

a.

```sql
select i.id, i.name
from instructor i
natural join department d
left join teaches t ON i.id = t.id
where dept_name = 'Marketing' AND t.course_id IS NULL;
```

9. **Using the university schema, write an SQL query to find the number of students in each section in 2009. -- The result columns should be “course_id, sec_id, year, semester, num”, where the latter is the number. -- You do not need to output sections with 0 students.**

a.

```sql
SELECT course_id, sec_id, year, semester, COUNT(ID) AS num
FROM takes
WHERE year = 2009
GROUP BY course_id, sec_id, year, semester
HAVING COUNT(ID) > 0
ORDER BY course_id, sec_id, semester;
```

10. **Find the maximum and minimum enrollment across all sections, considering only sections that had some enrollment, don't worry about those that had no students taking that section. Tip: you can use a subquery in from or a with-clause to provide an intermediate table on (course_id,sec_id, semester,year,num) where num is the count of enrolled students.**

a.
```sql
select Max(num), Min(num)
from
(
  select course_id, sec_id, semester, year, count(sec_id) as num
  from takes
  group by course_id, sec_id, semester, year
);
```

11. **Find all sections that had the maximum enrollment (along with the enrollment). Tip: you can use a subquery in from or a with clause.**

a.
```sql
SELECT t.course_id, t.sec_id, t.year, t.semester, COUNT(t.ID) AS num
FROM takes t
GROUP BY t.course_id, t.sec_id, t.year, t.semester
HAVING COUNT(t.ID) = (
    SELECT MAX(num_count)
    FROM (
        SELECT course_id, sec_id, year, semester, COUNT(ID) AS num_count
        FROM takes
        GROUP BY course_id, sec_id, year, semester
    ) AS subquery
)
ORDER BY t.course_id, t.sec_id, t.semester;
```

12.
```sql
select Max(num), Min(num)
from
(
  select course_id, sec_id, semester, year, count(sec_id) as num
  from takes natural full outer join course
  group by course_id, sec_id, semester, year
);
```

13. **Find all courses that the instructor with id '19368' have taught**

a.

```
select t.id, t.course_id, t.sec_id, t.semester, t.year
from teaches t natural join instructor i
WHERE i.id = '19368';
```

14. Find instructors who have taught all the above courses. Hint: one option is to use "... not exists (... except ...)".

a.

```
select DISTINCT t.id
from teaches t
where t.course_id in
(
  select t.course_id
  from teaches t
  where id = '19368'
);
```

15. Insert each instructor as a student, with tot_creds = 0, in the same department

a.

```
insert into student
select id, name, dept_name, 0 as tot_cred
from instructor
where instructor.id in
(
  (
  select id from instructor)
  EXCEPT
  (
  select id from student)
);
```

16. Now delete all the newly added "students" above (note: already existing students who happened to have tot_creds = 0 should not get deleted)

a.

```
delete from student
where id in
(
  (
  select id
  from instructor)
  EXCEPT
  (
  select id
  from student
  where tot_cred != 0)
);
```