**CS157A Homework 4**

**Points: 25**

**Question 1-5 (2 point) Question 6-10 (3 points)**

**Part** **1**: SQL questions

The following queries are based on the campaign data (campaign-ca-2016.sql)

1. how many contributions are contained in the data?

* **select count(\*) from campaign**
* **180478**

1. show the min and maximum amounts of all contributions?

* **select min(contb\_receipt\_amt), max(contb\_receipt\_amt) from campaign;**
* **-10000**

1. list the (distinct) ids and names of all the candidates in the data. order by name?

* **select distinct cand\_nm, cand\_id from campaign order by cand\_nm;**

1. show the candidate name and number of contributions, for each candidate? order by number of contributions in descending order?

* **select cand\_nm, count(cand\_id) as tot\_contb from campaign group by cand\_nm order by tot\_contb desc;**

1. show the candidate name and average contribution amount for each candidate, looking at positive contributions only? Order by average amount in descending order?

* **select cand\_nm, avg(contb\_receipt\_amt) as avg\_contb\_amt from campaign where contb\_receipt\_amt > 0 group by cand\_nm order by avg\_contb\_amt desc;**

1. show the candidate name and the total amount received by each candidate. Order the output by total amount received.

* **select cand\_nm, count(contb\_receipt\_amt) as tot\_contb from campaign group by cand\_nm order by tot\_contb desc;**

**Part 2:**

Use the courses data (read the files courses-ddl.sql and courses-small.sql).

1. write an SQL query that gives the number of courses taken for every student in the student table.

* **select name, count(course\_id) from student left outer join takes group by name;**

1. For each instructor, show the instructor name and the number of sections that have been taught by that instructor. You do not need to include instructors who have never taught a section. List in order of decreasing number of sections taught.

* **select name, count(course\_id) as num\_secs from instructor natural join teaches natural join section group by name order by num\_secs desc;**

1. Give the number of semester/year combinations in which sections have been offered.

* **select count(\*) from (select count(course\_id), semester, year from course natural join section group by semester, year);**

**Part 3:**

The following queries are based on the campaign data (campaign-ca-2016.sql)

create table candidate (

cand\_id varchar(12) primary key, -- cand\_id

name varchar(40) -- cand\_nm

);

1. Write an insert statement to fill the candidate table from Campaign table. The candidate table, after it is filled, should contain one row for each candidate (there are 22 candidates in the campaign table). -- Hint: use the kind of insert statement in which you list the fields to be filled in parentheses after the table name

* **insert into candidate select distinct cand\_id, cand\_nm from campaign;**