Database Systems — CSci 4380
Midterm Exam #2
March 31, 2016

RCS ID: ______________ @rpi.edu **Name:** ____________________

RIN # : ______________

Rules. The exam is 110 minutes for a total of 100 points. Open book and notes. Do not use any electronic tools including your computer. Work alone. You **cannot** talk to anyone in class, or share notes or thoughts.

Question 1. Write the following queries using SQL using the data model below.

Elections(eid, year, type, state, party)
Candidates(cname, eid, party)
Issues(issuename, type, description)
CandidatePositions(cname, eid, issuename, position, importance)
Voters(voterid, lname, fname, gender, age, street, state, city, zip)
Donations(id, voterid, amount, currency, date, cname, eid)

(a) **(12 points)** Due to a recent purge, candidate `RobotRick` has dropped out of the `general` election in year `2030`, but endorsed another candidate named `Tammy` for the same election.

Change all donations for candidate `RobotRick` to candidate `Tammy` for the same election. Then, delete all candidate positions for `RobotRick` (regardless of election).

(b) **(14 points)** Return the first and last name of voters who either has made at least one political donation to a candidate from party `BirdParty` (anytime) or live with another voter who made a donation to `BirdPerson` in year `2030`.

(c) **(14 points)** Return candidates running in an election in year `2030` on a `pro portals` (position and issue) platform and has received at least one donation of `10,000 dollars` or more from an alien voter (aliens have null values for zip codes).

(d) **(14 points)** Return the name of the candidates who have run for at least `three` elections (of any kind) and held a `pro zombie` position at least once and they never held a `con zombie` position.

(e) **(14 points)** Find candidates running for an election in year 2030 who have received the top 3 largest amount of donations per capita in 2030 (total donation amount given to the candidate divided by the number of unique voters for this year). Return the name of the candidate, per capita donation amount for the candidate and total donation amount. Break ties any way you wish.

Question 2 (16 points). For this question, you can use a single query, or you can piece together multiple queries, inserts and auxiliary tables for this question. You do not have to put them inside a procedure block and you do not need to drop your auxiliary tables.

Find issues that come up in **every** local election in the database for state `New Troy` with at least one candidate in the pro side and one candidate in the con side of the issue. Return the name of the issues.

Question 3 (16 points). You are given the following table definitions and instances. For each operation, show the changes to the tables by directly drawing on the tables. Provide a short sentence of why these tuples were changed or not changed right below the query.

```
CREATE TABLE bo (
    id INT PRIMARY KEY, name CHAR(2) );

CREATE TABLE so (
    id     INT PRIMARY KEY
    , did  INT FOREIGN KEY REFERENCES do(id)
      ON DELETE CASCADE ON UPDATE SET NULL ) ;

CREATE TABLE to (
    , bid   INT
    , sid   INT
    , PRIMARY KEY(bid, sid)
    , FOREIGN KEY (bid) REFERENCES bo(id)
      ON UPDATE CASCADE
    , FOREIGN KEY (sid) REFERENCES so(id)
      ON UPDATE CASCADE ON DELETE CASCADE);
```

```
CREATE TABLE do (
    id     INT PRIMARY KEY
    , bid  INT NOT NULL FOREIGN KEY
      REFERENCES bo(id) ON UPDATE CASCADE ) ;

CREATE TRIGGER toins BEFORE INSERT ON to
FOR EACH ROW
REFERENCING NEW ROW AS NEW
DECLARE
   c int ;
BEGIN
   SELECT count(*) INTO c FROM bo WHERE id = NEW.bid ;
   IF c = 0 THEN
      INSERT INTO b(id) VALUES(NEW.bid);
   END IF ;
END ;
```

(a) `DELETE FROM bo WHERE bo.name = 'dc';`

id	name
1	da
2	db
3	dc

(bo)

id	bid
1	2
2	2
3	1
4	2

(do)

id	did
1	2
2	3
3	3
4	4

(so)

bid	sid
2	3
2	4
3	2
3	4

(to)

(b) `DELETE FROM do WHERE do.bid = 1;`

id	name
1	da
2	db
3	dc

(bo)

id	bid
1	2
2	2
3	1
4	2

(do)

id	did
1	2
2	3
3	3
4	4

(so)

bid	sid
2	3
2	4
3	2
3	4

(to)

(c)

```
INSERT INTO to
SELECT max(bo.id),max(so.id) FROM bo,so;
```

id	name
1	da
2	db
3	dc

(bo)

id	bid
1	2
2	2
3	1
4	2

(do)

id	did
1	2
2	3
3	3
4	4

(so)

bid	sid
2	3
2	4
3	2
3	4

(to)

(d) `INSERT INTO to VALUES(4,2);`

id	name
1	da
2	db
3	dc

(bo)

id	bid
1	2
2	2
3	1
4	2

(do)

id	did
1	2
2	3
3	3
4	4

(so)

bid	sid
2	3
2	4
3	2
3	4

(to)

This page is left blank for scratch work, random thoughts and pictures!

Data model to be used in Exam #2

Note: The primary keys of each relation are underlined.

Elections(eid, year, type, state, party)

> Stores main information about elections. Type is one of: `'local'`,`'general'`, or `'local-party'`.
>
> If election is `'general'`, state and party are both empty (null value). For `'local'` and `'local-party'` elections state must be given.
>
> For `'local-party'` elections, party must also be given. These are elections in which various candidates from the same party compete. In local or general elections, candidates from different parties compete.

Candidates(cname, eid, party)

> Stores the names of the candidates, the id of the election they are running in (from Elections relation) and the party they are running for in this election. Obviously, the data model allows for candidates to run for different parties in different elections.

Issues(issuename, type, description)

> Stores political issues. Each issue has a name, e.g. `'time travel'`, `'cloning'`, `'thought control'`,`'transdimensional portal control'`, and a type e.g. `'health'`, `'portals'` and a longer description.

CandidatePositions(cname, eid, issuename, position, importance)

> Stores the position a candidate takes for a specific election. Cname is the name of a candidate from Candidates relation, eid is the id of an election, and issuename is the name of an issue from PoliticalIssues. Position is one of `'pro'` or `'con'`.
>
> Importance is a value between 1 and 10, 1 is the least important issue for the candidate and 10 is the most important. An issue may not even show up in this relation, in which case its importance is assumed to be zero.

Voters(voterid, lname, fname, gender, age, street, state, city, zip)

> Stores information for registered voters. Each voter is given a single voter id.

Donations(id, voterid, amount, currency, date, cname, eid)

> Stores the donations made by a specific voter given by their voter id, for a specific candidate in a specific election. The currency can be `'dollars'`, `'bitcoins'`, `'flurbo'`, etc.

Here is to democracy! Remember to vote for all elections you are invited to.