CCT395 Fall 2011

PRACTICE EXAM

This exam is worth 40% of your grade and consists of 40 points.

Most of the questions on this exam refer to the hypothetical database for Elections Canada to track contributions to candidates for federal parliamentary elections. In other words, when people run for membership in the House of Commons and other people contribute money to their campaigns, those contributions are to be tracked to ensure compliance with appropriate legislation. To simplify things, this database only deals with federal parliamentary elections and only records donations to individual campaigns, i.e., specific people running for elections as members of the House of Commons from a particular electoral district. There is no way, to just donate money to *a party*. When analyzing the database, assume also names of political parties and electoral districts are unique within Canada: there cannot be two districts called "Trinity—Spadina" and there cannot be another "Nice Weather Party."

The database was designed in a hurry (the election is around the corner!) so the analyst may have made a few mistakes. Question 2 asks you to check for some of those mistakes. For section 1 and all questions in section 3, however, you should use the database **as it was implemented**.

```sql
create table party (
  party_id integer not null auto_increment,
  name varchar(50), -- e.g., "Nice Weather Party"
  primary key (party_id)
);

create table candidate (
  candidate_id integer not null auto_increment,
  name varchar(50), -- e.g., "Jane Awesome"
  address varchar(300),
  primary key (candidate_id)
);

-- Represents a specific electoral contest, e.g. the election of
-- a Member of Parliament from "Toronto Center" on May 2, 2011.
create table election (
  election_id integer not null auto_increment,
  date_ordered date, -- when the election was ordered, e.g. 2011-03-26
  polling_day date, -- the day of the vote, e.g. 2011-05-02
  district_name varchar(50), -- e.g. "Toronto Center"
  year_created year, -- the year the district was created
  primary key (election_id)
);

-- A "campaign" is a candidate's participation in an electoral contest
create table campaign (
  campaign_id integer not null auto_increment,
  candidate_id integer not null,
  election_id integer not null,
  party_id integer,
  primary key (campaign_id),
  foreign key (candidate_id) references candidate(candidate_id),
  foreign key (election_id) references election(election_id),
  foreign key (party_id) references party(party_id)
);

create table donor (
  donor_id integer not null auto_increment,
  name varchar(50),
  primary key (donor_id)
);

create table contribution (
  contribution_id integer not null auto_increment,
  donor_id integer not null,
  donor_sin integer not null, -- donor's social insurance number
  campaign_id integer not null,
  amount integer not null, -- in Canadian dollars
  contribution_date date not null,
  primary key (contribution_id),
  foreign key (donor_id) references donor(donor_id),
  foreign key (campaign_id) references campaign(campaign_id)
);
```

1. Below are six of the entities in the ER diagram that was used to design Up!'s database. Complete the diagram by drawing the **five** relationships that are implied by the primary and foreign keys in the table descriptions. Do not draw any other relationships – points will be subtracted for unnecessary relationships. Use crow's feet notation and make sure to mark cardinality. Indicate whether the relationship is mandatory or optional if this can be determined. **Label the relationships with what you believe they represent.** You do *not* need to fill in the attributes for the entities. (10 points.)

donor

party

candidate

contribution

campaign

election

2. Do any of the tables in the election database violate 2NF or 3NF? If so, indicate the specific functional dependencies that cause the violation, explain why they represent a violation and explain what would need to be changed to bring the tables to 3NF. If all tables do satisfy 3NF, describe what a 2NF and a 3NF violations would look like in the context of this database. Write your answer in the space below. (5 points)

(Your answer probably doesn't need to go past this point.)

3. Write SQL queries to perform the listed tasks. (5 points per query, 20 points total).

a) Get the total amount of money donated by "Mary Rich." (For full credit, consider the possibility that there are multiple donors with that name and make sure that your query shows separately the totals for each of them.)

b) List the top 10 candidates by the the largest number of donations to their campaigns. (Show the name of each candidate together with the number of of donations.) Make sure to list the number of donations, and **not** the total amount.

c) Get the name of the party with the largest number of candidates running in the May 2, 2011 election.

e) Create a new table, "audit" which will store information about audits conducted by Elections Canada. An audit should always be linked to a specific contribution. The table should always record the date when the audit was initiated and should record the date when the audit was completed *if* it has been completed. All other information about the audit is to be stored in a single long text field.

4. In this last part you will be one or two questions worth 5 points each. You will have between ½ and 1 page to answer each question. (Shorter answers are acceptable, as long as they answer the question fully.) Here are some *examples* of the questions that you might see. *The exam may include questions not listed below.*

What is "the relational model" and what are its main strengths and weaknesses?

What is "normalization" and what do we need it for?

What is a "storage engine" in MySQL and why is there a need or more than one?

What is the "SQL injection" attack and how can it be prevented?

What are the pros and cons of creating an index on a field?

What does XML have to do with relational databases?

What is the difference between "an ANSI join" and "a traditional join"? Which one is better and why?

What is the difference between the ER model and the relational model?

What is a "Cartesian product" in the context of SQL joins and how do we avoid it?

What is a "natural join" and why should one avoid it?

Suppose that you learn that your client has a database table with 50 fields. Should this be a source of concern? Why or why not?

Harrington's *Relational Database Design* says that 2NF requires that "[a]ll non-key attributes are functionally dependent on the entire primary key. " (p. 111). In this case, what are "non-key attributes"? Illustrate your answer with an example.

What are the benefits of using foreign key constraints in a relational database?

What is a "candidate key"? How is it different from a "primary key"? Provide an example.

MySQL's "InnoDB" storage engine enforces foreign key constraints. Some of the other storage engines ignore them. What could be the advantages of using a storage engine that does **not** support foreign key constraints?

Can a table have a foreign key to its own primary key? If not, explain why not. If yes, explain what this would achieve and whether this foreign key can be NOT NULL (and why).

What are "natural" primary keys and what are pros and cons of using them?

What's the difference between an "inner join" and an "outer join"? When would you want to use an outer join?

How can we use views to control access to data?

Harrington's *Relational Database Design* says that 2NF requires that "[a]ll non-key attributes are functionally dependent on the entire primary key" (p. 111). What does she mean by "functionally dependent"? Provide an example.

Harrington's *Relational Database Design* says: "True one-to-one relationships are very rare in business " (p. 66). Explain why this is the case.

While reviewing an ER diagram drawn by an inexperienced data modeller you notice that entities called "User" and "Book" are connected with an arrow that says "searches for." Why is this likely to be a mistake? What would be a possible *good* reason to have this relationship?

What is an "insertion anomaly"? Provide an example.

Why is it a bad idea to store user's passwords in your database? What shall you do instead?

Suppose that a table called "student" has fields "student_no" (the primary key), "utorid" and "last_name". The fields "student_no" and "utorid" functionally determine each other. They both also functionally determine "last_name". Is this a 3NF violation? Explain.

You are creating a table to store information about movies. What data type would you choose for the movie title? Explain the advantages and disadvantages of your choice (provide examples).

You are creating a table to store information about paintings in an art gallery. How would you represent the time when the painting was produced? Explain the advantages and disadvantages of your choice (providing examples).

Suppose that the City of Toronto has a database of parking spots in the city which has up-to-date information about fees and the hours during which parking is restricted. They want to make this information available in real-time to companies that would want to incorporate them into services such as Google Maps. How can they do this? Provide specific examples.