Carnegie Mellon University
15-415 - Database Applications
Fall 2009, Faloutsos
Assignment 1: Data Modeling

Due: 9/8, 1:30pm, in class - hard-copy please

Reminders

- Weight: **5%** of the homework grade.
- Out of **100** points.
- Lead TA: B. Aditya Prakash
- Rough time-estimates: 2∼4 hours. (30-60mins for each of Questions 1, 2 and 3; 1-2hrs for Question 4)
- Please **type** your answers. Illegible handwriting may get no points, at the discretion of the grader. Only drawings may be hand-drawn, as long as they are neat and legible.

Remember that:

- There could be more than one correct answer. We shall accept them all.
- Whenever you are making an assumption, please state it clearly.

Question 1: Votes Database [10 points]

We want to design a database for keeping track of the votes taken in the U.S. Senate. For each senator, we want to record the name (assume it is unique), the name of his/her state, and whether the senator is the junior or senior senator for the state. For each bill we want to record the (unique) BillName (eg., 'Air Quality Act'), the DateOfVote on the bill, whether the bill PassedOrFailed (domain is *YES,NO*), as well as the one or more senators that sponsored the bill. Finally, we want to record how each senator voted on each bill (domain is *Yes, No, Abstain, Absent*).
The following constraints hold:

- A senator votes for every bill (and, obviously, only once per bill).
- A senator can sponsor zero or more bills.
- There are 1 or more sponsors for each bill.

Please answer the following questions:

- *Q1.1*: Draw an ER diagram for this database. Make sure to indicate primary keys, cardinality constraints, weak entities (if any), and participation constraints. List any assumptions you make in the process. **[5 points]**

- *Q1.2*: Translate the ER diagram in *Q1.1* into relational database tables (i.e. give the SQL DDL statements). Make sure that the translation captures key constraints (primary keys and foreign keys if applicable) and participation constraints in the ER diagram. Identify constraints, if any, that you are not able to capture. **[5 points]**

Question 2: Garage Database [20 points]

We want to design a database for a local garage. For each customer, we want to record the (unique) name, the customer address, and the contact phone number. For each vehicle, we want to record the unique vehicle's identification number (VIN), and the vehicle's make, model and year. For each repair job we want to record the description of the job done (maximum 200 chars), the date, and the total dollar cost. A repair job may involve zero or more parts (like, e.g., "windshield wipers", "battery", etc.) For each part we want to record its (unique) part number, the part name and its cost. In addition, note that:

- Each vehicle may have 1 or more repair jobs.
- Each customer may be the primary owner of 1 or more vehicles.
- Every vehicle has only one primary owner (we ignore co-owners)
- No vehicle can have more than one repair job in any given day.

Please answer the following questions:

- *Q2.1*: Draw an ER diagram for this database. Make sure to indicate primary keys, cardinality constraints, weak entities (if any), and participation constraints. List any assumptions you make in the process. **[10 points]**
- *Q2.2*: Translate the ER diagram in *Q2.1* into relational database tables (i.e. give the SQL DDL statements). Make sure that the translation captures key constraints (primary keys and foreign keys if applicable) and participation constraints in the ER diagram. Identify constraints, if any, that you are not able to capture. **[10 points]**

Question 3: Social Network [20 points]

Inspired by Facebook, we want to design a social networking site, *MyFace*. And as you might have guessed, it needs a database for storing all the information. For every user, we want to record the name, the unique username and address. We also want to support the following functions, and record the necessary information:

- People make friends with other people. Thus a user may have zero or more friends.
- People write postings on "walls". A person may write zero or more postings. Note that a user may write postings on his/her own wall, as well as on other people's walls. For each posting we want to record the author, the owner of the wall on which it appears, and the timestamp.
- For users that opt-in to a functionality like `dopplr.com`, we want to record their "appearances", whenever they choose to upload their position (say, through their mobile phone). For each "appearance" of a user, we want to record the (x, y) co-ordinates

and the timestamp. Notice that multiple users can be at the same position at the same time (e.g. during a lecture in a class room). Also, a user may visit the same position more than once (e.g., his/her home location)

Please answer the following questions:

- *Q3.1*: Draw an ER diagram for this database. Make sure to indicate primary keys, cardinality constraints, weak entities (if any), and participation constraints. List any assumptions you make in the process. **[10 points]**
- *Q3.2*: Translate the ER diagram in *Q3.1* into relational database tables (i.e. give the SQL DDL statements). Make sure that the translation captures key constraints (primary keys and foreign keys if applicable) and participation constraints in the ER diagram. Identify constraints, if any, that you are not able to capture. **[10 points]**

Question 4: Movie Ratings [50 points]

This question is on a movie ratings database. Download and install SQLite3 from `http://www.sqlite.org`.

Warmup

Follow the documentation and load the sample database at:

`http://www.cs.cmu.edu/~christos/courses/dbms-F09/hws/hw1/15415-hw1.db`

It has a table `recommendation` which is like the table of the *Netflix* competition: people rate movies, with ratings from 1-5 (1 for 'I hate it!', to 5, for 'I love it!'). As a sanity check that you have the correct database, running the following command at a Unix/Linux/Cygwin prompt

```
your-machine% sqlite3 15415-hw1.db 'select count(*) from recommendation'
```

should return

```
14
```

We want to write SQL queries to do the following:

- Query1: Return all movies with a rating of 5 from at least one reviewer.
- Query2: Return all the reviewers who rated 'Gone with the wind'.

Larger CSV file

A bigger raw comma separated value (csv) file is given here:

`http://www.cs.cmu.edu/~christos/courses/dbms-F09/hws/hw1/movie_ratings.csv`

It is a subset from the Netflix-competition dataset (If you are curious, the official Netflix USD 1 Million prize dataset is at `http://www.netflixprize.com//index`). The dataset is anonymized, hence customers will be represented by a random, integer id. We want to write queries to do the following:

- Query3: Return the count of reviews where the rating is 5.
- Query4: Return the count of reviewers who gave a rating of 1 to 'Gone with the wind'.

Life without SQL

Finally, in your favorite language (Python/Perl/Ruby/Java/C++ etc.) write code to do both queries above (Query3 and Query4) on the csv data file directly. Notice: the end-of-line convention is the DOS one (CR LF).

Deliverables Checklist

Please hand in the following (in hard copy):

Q4.1 The SQL query for Query1. [**5 points**]
Q4.2 The SQL query for Query2. [**5 points**]
Q4.3 The output of running Query1 in SQLite on the sample database. [**5 points**]
Q4.4 The output of running Query2 in SQLite on the sample database. [**5 points**]
Q4.5 The SQL query for Query3. [**5 points**]
Q4.6 The SQL query for Query4. [**5 points**]
Q4.7 The output of running Query3 on the csv file after loading it in SQLite. [**5 points**]
Q4.8 The output of running Query4 on the csv file after loading it in SQLite. [**5 points**]
Q4.9 Hard copy of your python/perl/etc code for doing Query3 on the raw csv file directly. [**5 points**]
Q4.10 Hard copy of your python/perl/etc code for doing Query4 on the raw csv file directly. [**5 points**]

Hints

For loading the csv file,

- Again, the end-of-line convention follows the DOS format (CR LF).
- Use the `.import` and `.mode csv` commands of sqlite3
- or check the tutorial at
  `http://my.opera.com/cookyjar/blog/2009/04/20/importing-csv-data-file`
- Again as a sanity check, the command

  ```
  your-machine% wc -l movie_ratings.csv
  ```

  should return

  ```
  10000 movie_ratings.csv
  ```