15-415/Fall'11
Homework #1

Due: Tuesday, September 28, 2011 at 11:59:59pm (Handin electronically into the course AFS space)

1. Restaurant Database

   Inspired by Yelp, we want to design a database for maintaining restaurant reviews. The database should contain details about each restaurant: name (unique), type of cuisine, address, phone number and if it has a drive through. We also would want to capture the details of each customer: name (unique), address, age and gender. The customers can rate any restaurant on a scale of 1-10 and provide comments (text).

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

   [Q1.2] Based on the ER diagram, design a relational database to store the information. 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]

Write SQL queries/procedures/triggers to implement the following functionalities.

[Q1.3] Find the number of restaurants in Pittsburgh, PA (Drive- through vs dine-in) [2 points]

[Q1.4] Find the top-rated restaurants in Pittsburgh, PA. Top rated restaurant are defined as the ones with the highest average rating [3 points]

[Q1.5] List the top ten restaurants in each state that have a rating of 7 or more in 2009-2010 and which had at least 20 reviews [4 points]

[Q1.6] Inform a restaurant as soon as any customer rates it below three. How would you implement this feature? [5 points]

2. Patient Database

We want to design a database for maintaining patient records. For each patient, we want to store his details i.e. name, SSN (unique), phone number, address, age and gender. Every time he visits the hospital, the following information should be stored: date of visit, the doctor consulted, identified illness and medicines prescribed. We also want to store information about the doctor i.e. i.e. name, SSN (unique), phone number, address, age, gender and specialization.

In addition, please note that:
- A patient might be diagnosed with more than one illness.
- For each illness, multiple drugs might be prescribed.
- A doctor might be specialized in more than one area.

[Q1.1] Draw an ER diagram for the system. Make sure to indicate primary keys, cardinality constraints, weak entities (if any) and any constraints. List any assumptions you make in the process. [7 points]

[Q1.2] Based on the ER diagram, design a relational database to store the information. 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. [7 points]

[Q1.3] Write a SQL query to input the details about a patient. [2 points]

[Q1.4] Write a SQL query to input the details about a patient's hospital visit. [5 points]

[Q1.5] Find the number of people who were prescribed Flu shots so far in 2010. [3 points]

[Q1.6] Find the % of terminally ill patients in 2006. By definition, terminally ill patients are those who are prescribed a minimum of 10 different drugs and visit the hospital at least twice in any given month. [7 points]

[Q1.7] On a daily basis, the hospital manager wants view the list of patients who have has progressed into the "terminally ill" category. Design the functionality. [7 points]

[Q1.8] Concerned about his privacy, a patient wishes to have all his information deleted from the system. Write a SQL query to achieve that. [7 points]

[Q1.9] When the patient visits a doctor, the doctor would like to see the patients' history (i.e. his hospital visits, illness diagnosed and prescribed drugs) at one place. Write a view to achieve that. [7 points]

3. Twitter

If you are not familiar with Twitter, please check twitter.com. We will work with a fictitious, simplified, Twitter-like setting, with the following specifications:

  • Users post 'tweets', that is short pieces of text
  • They may tag their tweets with zero or more tags of their own choice. For example a user tweeting about the G20 summit may decide to use the tag 'G20' (prefixed by a 'sharp' sign: #G20, if we follow the convention imposed by the twitter site).
  • A user 'u' may follow zero or more other users, which means that their 'tweets' are visible to user 'u' when he/she logs in.

For the above setting, we will use the following schema:
  • Person (pname, city, street) - Assume the pname is unique
  • Follows (pname1, pname2) - Person pname1 follows person pname2
  • Tweets (tid, ttitle, ttext) - Tweet with tid has title ttitle and text ttext
  • PersonTweets (pname, tid, ts) - Person pname posted tweet tid at timestamp ts
  • TweetTag (tid, tagname) - Tweet tid had tagname in its list of tags.

We now want to extract some information from the database using the power of relational algebra. For each of the following questions

  (a) if they can be answered with relational algebra, give the corresponding expression or
  (b) if impossible, state so clearly, and justify briefly (1-2 lines)

Note: You can use any fundamental or derived operators shown in class or in the textbook.
        Also, feel free to create and use views. For example,
            PITT PEOPLE _ _city='Pittsburgh0(Person)
        defines a view PITT PEOPLE containing everyone in Person from Pittsburgh.

3.1 Find all the people (pname) who posted a tweet with tag 'Obama'. [2 points]

3.2 Find all the different, distinct tags ever used. [2 points]

3.3 Find all the tags 'Bob Smith' reads in the tweets of the people he follows. (i.e. Bob's reading interests) [2 points]

3.4 Find all the people (pname and city) who follow people who follow 'Ashton Kutcher (i.e. Second-level followers) [2 points]

3.5 Find all the people (pname) who could have potentially read a tweet about the IPhone 3GS, before its launch, i.e. all the people who follow people who posted a tweet with title 'New IPhone' or tag 'IPhone 3GS' before '00:00:00 06-19-2009'. [2 points]

3.6 Find all the people (pname and city) who follow at least everyone that 'Bob Smith' follows. [2 points]

3.7 Give relational tuple calculus (RTC) expressions for Questions 3.2 and 3.4. [3 points each]

3.8 Give relational domain calculus (RDC) expressions for Questions 3.5 and 3.6 [3 points each]