CSE 444 Midterm

Spring 2010

April 30, 2010

Name: ____________________________________

Question	Points	Score
1	40	
2	15	
3	25	
4	20	
Total:	100	

This exam is an open book exam. You have 50 minutes; budget time carefully. Intermediate steps are rarely required but often useful for partial credit. Good luck!

1 SQL

1. (40 points)

   Consider the following social network database:

   ```
   Person(pid, name)
   Relationship(pid1, pid2, type)
   ```

   Where:

   - `Person.pid` is a key.
   - `Relationship.pid1` and `Relationship.pid2` are foreign keys to `Person`.
   - `Relationship.type` is either 'friend' or 'enemy'.

   Keep in mind that `Relationship` is not symmetric: if `p1` is a friend of `p2`, that does not mean `p2` is a friend of `p1`. It is not transitive either: if `p1` is a fried of `p2` who is a friend of `p3`, it doesn't mean `p1` is a fried of `p3`.

   (a) (12 points) A *second degree friend* is the friend of a friend[1]. Write a SQL query that computes for each person the total number of their second degree friends. Your query should return answers of the form: `pid, name, count`. Cryptic hint: "not every person has friends, but you have to count *everyone*'s second degree friends".

Solution:

```
select x.pid, x.name, count(distinct z.pid2)
from Person x left outer join Relationship y
              left outer join Relationship z
     on x.pid = y.pid1 and y.pid2 = z.pid1
    and y.type = 'friend' and z.type = 'friend'
group by x.pid, x.name
```

[1] A second degree friend can also be a first degree friend.

```
Person(pid, name)
Relationship(pid1, pid2, type)
```

(b) (12 points) Write a SQL query that returns all persons who have at least 12 common friends with “Mary”. Your query should return answers of the form: `pid1, pid2, name`, where `pid1` is Mary’s `pid` and `pid2` is that of a person who has 12 or more common friends (meaning there are at least 12 persons `p` such that `pid1` and `p` are friends, and `pid2` and `p` are friends). If there are multiple people called Mary, then you will report each of them.

Solution:

```
select x.pid, y.pid, y.name
from Person x, Person y, Relationship u, Relationship v
where x.pid = u.pid1 and y.pid = v.pid1 and u.pid2 = v.pid2
  and u.type = ’friend’ and v.type = ’friend
  and x.name = ’Mary’
group by x.pid, y.pid, y.name
having count(*) >= 12
```

(c) (12 points) Fred says: ”my enemies’ enemies are my friends”. Prove that Fred is wrong: write a query that returns all Fred’s enemies’ enemies that are not his friends. Your query should return answers of the form: `pid1, pid2`, where `pid1` is Fred’s `pid` and `pid2` represents an enemy’s enemy that is not Fred’s friend.

Solution:

```
select x.pid, z.pid2
from Person x, Relationship y, Relationship z
where x.name = ’Fred’
  and x.pid = y.pid1 and y.pid2 = z.pid1
  and y.type = ’enemy’ and z.type = ’enemy’
  and not exists (select * from Relationship u
                  where u.type = ’friend’
                    and u.pid1 = y.pid1 and u.pid2 = z.pid2)
```

```
Person(pid, name)
Relationship(pid1, pid2, type)
```

(d) (4 points) Your social network database has increased to more than 1M people. You are now running a business, and have two data-intensive applications accessing the database:

- Customer support: your customers call up, and ask questions about their friends and their friends' friends. (They don't ask about enemies).
- Mail advertising: once per week you send out mail to people who are part of groups of 5 or more mutual friends.

To assist in the development of these applications, and perhaps to speed them up, you decide to define two views:

```
CREATE VIEW  FriendsOfFriends(pid1, pid3)
             /* used by customer support */


CREATE VIEW  GroupsOfFive(pid1, pid2, pid3, pid4, pid5)
            /* used by mail advertising */
```

The applications will use these two views instead of, or in addition to the base tables of your database. For each view you have one or two choices: declare it a *virtual view* or a *materialized view*.

Indicate for each of the two views if you would declare it virtual, or materialized. Think about the requirements of the application when making your decision. You do not have to justify your answer.

Should `FriendsOfFriends` be materialized or virtual ?

(d) **virtual**

Solution:

`For customer support accuracy is most important.`

Should `GroupsOfFive` be materialized or virtual ?

(d) **materialized**

Solution:

For this large computation, speed is valued more than accuracy.

2 E/R Diagrams and Database Design

2. (15 points)

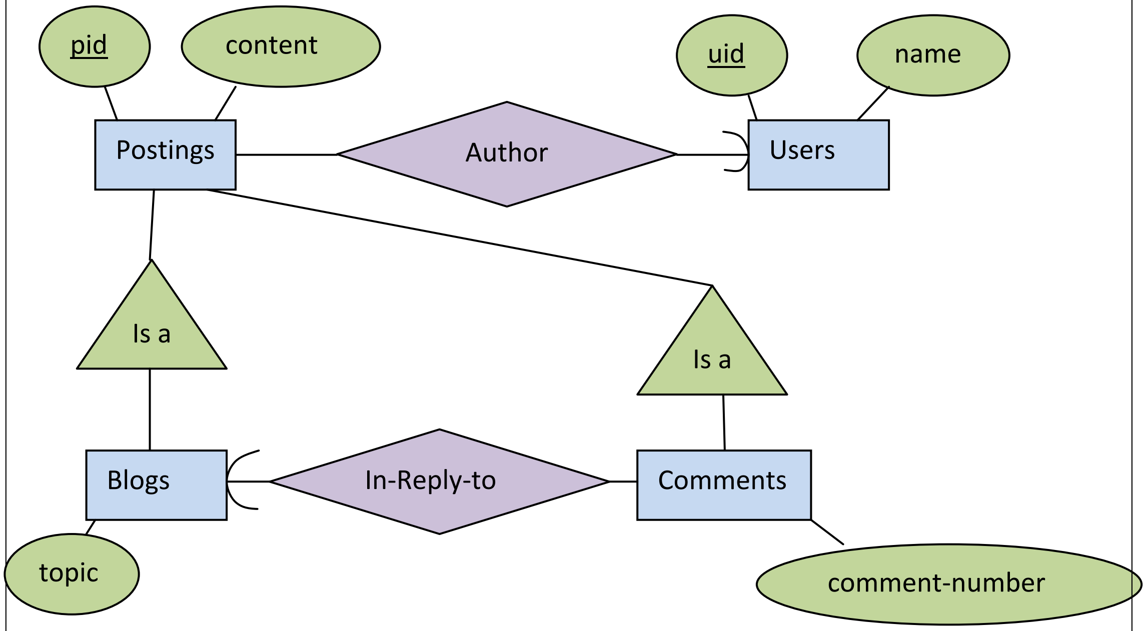
Consider a database with the following entity sets:

- `Users`: have an `uid` (10 characters) and `name` (up to 20 characters). `uid` is the key for `Users`.
- `Postings`: have an `pid` (10 characters) and `content` (text of 100 characters). `pid` is the key for `Postings`.
- `Blogs`: every blog is a `Posting` and has a `topic` attribute (up to 30 characters).
- `Comments`: every comment is a `Posting` and has a `comment-number` (an integer).

and the following relationships:

- There is an `Author` relationship connecting `Posting` to `User`.
- There is a `In-reply-to` relationship connecting `Comment` to `Blogs`.

(a) (5 points) Draw the E/R diagram for the database schema.



Solution:

(b) (10 points) Write the SQL statements for creating the tables that represent your E/R diagram. Indicate all keys and foreign keys.

Solution:

```
create table Users (
   uid char(10) primary key,
   name varchar(20)
)
create table Postings (
   pid char(10) primary key,
   uid char(10) not null references Users,
   content varchar(100)
)
create table Blogs (
    pid char(10) primary key references Postings,
    topic varchar(30)
)
create table Comments (
    pid char(10) primary key references Postings,
    comment-number int,
bid char(10) not null references Blogs(pid)
}
```

3 Conceptual Design

3. (25 points)

(a) (10 points) Consider a relation with attributes $R(A, B, C, D, E)$ that satisfies the following functional dependencies:

$$\begin{aligned} AC &\rightarrow B \\ BD &\rightarrow C \\ CE &\rightarrow D \\ DA &\rightarrow E \\ EB &\rightarrow A \end{aligned}$$

Find all the keys that contain the attribute A. Your answer should include only the keys that contain A, for example $ABCD$ (not necessarily a real answer), but not $BCDE$.

Solution:

$$\begin{aligned} AB+ &= AB \\ AC+ &= ABC \\ AD+ &= ADE \\ AE+ &= AE \\ ABD+ &= ABCDE \\ ABE+ &= ABE \\ ACD+ &= ABCDE \\ ACE+ &= ABCDE \\ ADE+ &= ADE \end{aligned}$$

Keys that contain A are: ABD, ACD, ACE.

(b) (10 points) Decompose in BCNF relation $R(A, B, C, D, E)$ that satisfies the following functional dependencies. Show your steps, and show the keys in the decomposed relations.

$$\begin{array}{rcl} E & \rightarrow & C \\ BD & \rightarrow & E \end{array}$$

Solution: Step1: In $ABCDE$: $E+ = EC$. Decompose into $\underline{E}C$ and $EABD$

Step 2: In $EABD$: $BD+ = BDE$. Decompose into $\underline{BD}E$ and BDA

Final decomposition is: $\underline{E}C$, $\underline{BD}E$ and BDA.

(c) (5 points) Consider two relations $R(A, B, C)$ and $S(D, E, F)$ satisfying the following functional dependencies:

$$\begin{aligned} A &\rightarrow B \\ E &\rightarrow F \end{aligned}$$

and consider the following views:

```
CREATE VIEW V1(A,B,C) AS
  SELECT *
  FROM R
  WHERE C = 29

CREATE VIEW V2(A,B,E,F) AS
  SELECT DISTINCT R.A, R.B, S.E, S.F
  FROM R, S
  WHERE R.A = S.E
```

Indicate all functional dependencies that hold in each view. It suffices if you show a minimal set of functional dependencies, i.e. one that implies all other functional dependencies that hold on that view.

Solution: In $V1$:

$$\begin{aligned} A &\rightarrow B \\ B &\rightarrow C \end{aligned}$$

In $V2$:

$$\begin{aligned} A &\rightarrow B \\ E &\rightarrow F \\ A &\rightarrow E \\ E &\rightarrow A \end{aligned}$$

4 Transactions

4. (20 points)

   (a) (3 points) For each of the following statements indicate whether it is true or false:

   - The main reason why transactions were invented was to improve the database performance. True or false ?

     (a) **FALSE**

   - A banking application that manages customer accounts should perform all updates to the accounts from within transactions with ACID properties. True or false ?

     (a) **TRUE**

   - A Facbook application that allows you to write on your Wall (so that other friends can see what you are up to) should update the wall content in the database from within a transaction with ACID properties. True or false ?

     (a) **FALSE**

(b) (13 points) Consider the content of the following **undo log**:

```
LSN1   <START T1>
LSN2   <T1 X 5>
LSN3   <START T2>
LSN4   <T1 Y 7>
LSN5   <T2 X 9>
LSN6   <START T3>
LSN7   <T3 Z 11>
LSN8   <COMMIT T1>
LSN9   <START CKPT(T2,T3)>
LSN10  <T2 X 13>
LSN11  <T3 Y 15>
       *C*R*A*S*H*
```

- Show how far back in the recovery manager needs to read the log. Write below the earliest LSN that the recovery manager reads.

(b) **LSN3**

- Show below the actions of the recovery manager during recovery:

Solution:

$$\begin{aligned} Y &= 15 \\ X &= 13 \\ Z &= 11 \\ X &= 9 \end{aligned}$$

- What is the value of X at the end of the recovery ?

(b) **9**

(c) (4 points) Consider a **redo log**. For each of the questions below indicate whether it is true of false. You do not need to justify your answer.

- When a trasaction wants to commit, it first needs to wait until all its pages have been written to disk. True or false ?

  (c) **false**

- If the last checkpoint in the log has not been completed, then it cannot be used during recovery. True or false ?

  (c) **true**

- If the system crashes while a transaction T is active, then none of the records written by T have been written to disk. True or false ?

  (c) **true**

- If transactions $T1, T2, T3$ are active when a checkpoint starts, then the checkpoint cannot end before all three transactions commit. True or false ?

  (c) **false**