Homework #2 Solutions
due Monday September 27, 2010 at 2 pm

Database Systems, CSCI-4380-01

Each student must work on this homework alone.

Homework Description

Question 1. Suppose you are given the following set of functional dependencies for relation $R(A, B, C, D, E, F, G)$:

$$\mathcal{F} = \{AB \to C, A \to D, BC \to E, E \to F, AG \to B\}$$

Assume $\mathcal{F}$ forms a minimal basis. Answer all the questions below for this relation and $\mathcal{F}$.

1. Is this relation in BCNF or 3NF? If it is not, list all functional dependencies that violate 3NF and/or BCNF.

   R is in neither BCNF nor 3NF. It will suffice to show the functional dependencies which violate 3NF.

   Recall the definition of 3NF:
   For all functional dependencies $X \to A$ at least one of the following is true:

   - $X \to A$ is a trivial dependency.
   - X is a superkey.

- A is a prime attribute.

None of the functional dependencies is trivial. Any key of R must contain G, and thus must every superkey. The key of R is $\{A, G\}$ since $\{A, G\}^+$ contains all attributes of R. From these facts we determine that the functional dependencies which violate 3NF are $\{AB \to C, A \to D, BC \to E, E \to F\}$

2. If this relation is not in 3NF, use the 3NF decomposition to find a set of relations that are in 3NF.

   Using Algorithm 3.26 on page 103 of the text book gives a set of relations:

   $R1(A, B, C), \{AB \to C\}$; key: AB.

   $R2(A, D), \{A \to D\}$; key: A

   $R3(B, C, E), \{BC \to E\}$; key: BC

   $R4(E, F), \{E \to F\}$; key: E

   $R5(A, G, B), \{AG \to B\}$, key: AG

   All in 3NF

3. Is the following (random) decomposition lossless?

   $R1(A, B, C, D)$, $R2(B, E, F)$, $R3(A, B, G)$.

   Prove or disprove using the Chase method.

A	B	C	D	E	F	G
a	b	c	d	e_1	f_1	g_1
a_2	b	c_2	d_2	e	f	g_2
a	b	c_3	d_3	e_3	f_3	g

$\xrightarrow{AB \to C, row3}$

A	B	C	D	E	F	G
a	b	c	d	e_1	f_1	g_1
a_2	b	c_2	d_2	e	f	g_2
a	b	c	d_3	e_3	f_3	g

$\xrightarrow{BC \to E, row3}$

A	B	C	D	E	F	G
a	b	c	d	e_1	f_1	g_1
a_2	b	c_2	d_2	e	f	g_2
a	b	c	d_3	e_1	f_3	g

$\xrightarrow{E \to F, row3}$

A	B	C	D	E	F	G
a	b	c	d	e_1	f_1	g_1
a_2	b	c_2	d_2	e	f	g_2
a	b	c	d_3	e_1	f_1	g

$\xrightarrow{A \to D, row3}$

A	B	C	D	E	F	G
a	b	c	d	e_1	f_1	g_1
a_2	b	c_2	d_2	e	f	g_2
a	b	c	d	e_1	f_1	g

No further functional dependency applications result in a change in the tableu, so we are done. The result is that the decomposition is not lossless as there is no row without subscripts.

4. Is the decomposition above dependency preserving? Prove or disprove by first by finding the projection of $\mathcal{F}$ onto each of $R1, R2$ and $R3$, then by checking whether the union of these projections is equivalent to $\mathcal{F}$.

Using Algorithm 3.12 on page 82 of the text book we find the functional dependencies which hold in $R1$, $R2$, and $R3$.

$\mathcal{F}_{R1} = \{A \to D, AB \to C\}$; $\mathcal{F}_{R2} = \{E \to F\}$; $\mathcal{F}_{R3} = \{AG \to B\}$

Given these projected functional depedencies, we can see that the union of these functional dependencies do not imply the functional depedency $BC \to E$. As such, we have clearly lost the functional dependency $BC \to E$, thus this decomposition is not lossless.

Question 2. Put the following relation $R(A, B, C, D, E, F)$ into 4NF.

$$\mathcal{F} = \{AB \to C, AC \twoheadrightarrow DE\}$$

Show your work.

First we check if R is already in 4NF. To do this we need to know the key (and thus any superkey). Any key must contain F as it does not appear in any functional dependency. We find that {A, B} can derive all attributes except D, E and F, so our key is {A, B, D, E, F}. We check the functional and multi-value dependencies to see if all left-hand-sides are superkeys. This is clearly not the case, so R is not in 4NF.

We choose to begin decomposition with the multi-value dependency $AC \twoheadrightarrow DE$. Using Algorithm 3.33 on page 111 of the text book, we decompose R into two relations, $R1$ and $R2$ as follows.

$R1(A, C, D, E), \{AC \twoheadrightarrow DE\}$; key: all attributes. In 4NF (the only mvd is trivial)

$R2(A, B, C, F), \{AB \to C\}$ key: ABF, not in 4NF

We then project the functional and multi-value dependencies onto $R1$ and $R2$ as done in previous questions. After projection we find $R1$ is a trivial multi-value dependency and is thus in 4NF. The functional dependency $AB \rightarrow C$ which holds in $R2$ still violates the conditions of 4NF. We must decompose $R2$ into $R21$ and $R22$ as follows.

$R21(A, B, C), \{AB \rightarrow C\}$; key AB

$R22(A, B, F)$; key all attributes

The functional dependency $AB \rightarrow C$ holds in $R21$ and is trivially fulfilled. Likewise in $R22$.

The complete decomposition of R is into three relations $R1$, $R21$, and $R22$ as defined above.

Beginning decomposition with the functional dependency $AB \rightarrow C$ may yield a different but equally correct result (note that we did not learn projection of multi-valued dependencies to decomposed relations, so we omit that step here).

Question 3. Create an ER model for the following application. The following describes the type of data that will be stored for this application.

You are designing an application for your favorite politician Kip Drordy. Kip wants to run for an important office, so he has some information on different issues on his website in the form of different web pages. However, Kip wants to use social networking to enable his supporters to register, make donations, talk to others online, vote on various polls to voice opinion and even organize face to face events in their local communities. Kip's success depends on how well this application works and enables people to feel connected. Ok, back to the application. Assume that Kip's website has a great deal of HTML content that is not stored in a database. This content lists all the opinions Kip has on different issues. Assume that the URL for these pages is fixed and does not change. So, you can point to a specific page using its URL as an identifier. The following is the data to be stored in the database.

Your database stores a list of issues, where each issue has a name, description and a URL it points to. There is also a list of user types, such as admin, pollster, user, staffer, etc. Furthermore, there is a list of user status types. The status is drived from the amount of online activity a user has. For now, you can assume status is a string. Your database should store information about users, their name, email, age, address, date joined, donations, friends. Users have a type as well as a status. For each friend, when the friendship was recorded is stored. Assume friendships are symmetric, A is friend of B means B is a friend of A. The donations have an amount and date. Furthermore, users have interests each of which is an issue. The system also allows users to create polls, where each poll contains a list of questions and a set of possible answers for each question. The creation data of a poll is also given. Different users may vote on multiple polls, but each user may vote on a poll once. Furthermore, users might post questions online for the candidate to answer. For each question, the user who posed the initial question and the creation date of the question is stored together with the text of the question. Each user may vote on questions to decide which questions are important to the community as a whole. So, the votes are also stored. Again, a user may only vote once. The date of a vote is also stored for any vote. Some questions may have answers from the candidate, which is a piece of text. Finally, users may create events with a specific title, date, time and location. Other users may RSVP for these events, in that case the date of their RSVP is stored.

Answer. See Figure 1 (Note that in this figure the rounded arrow is reversed, treat it as the usual rounded arrow).

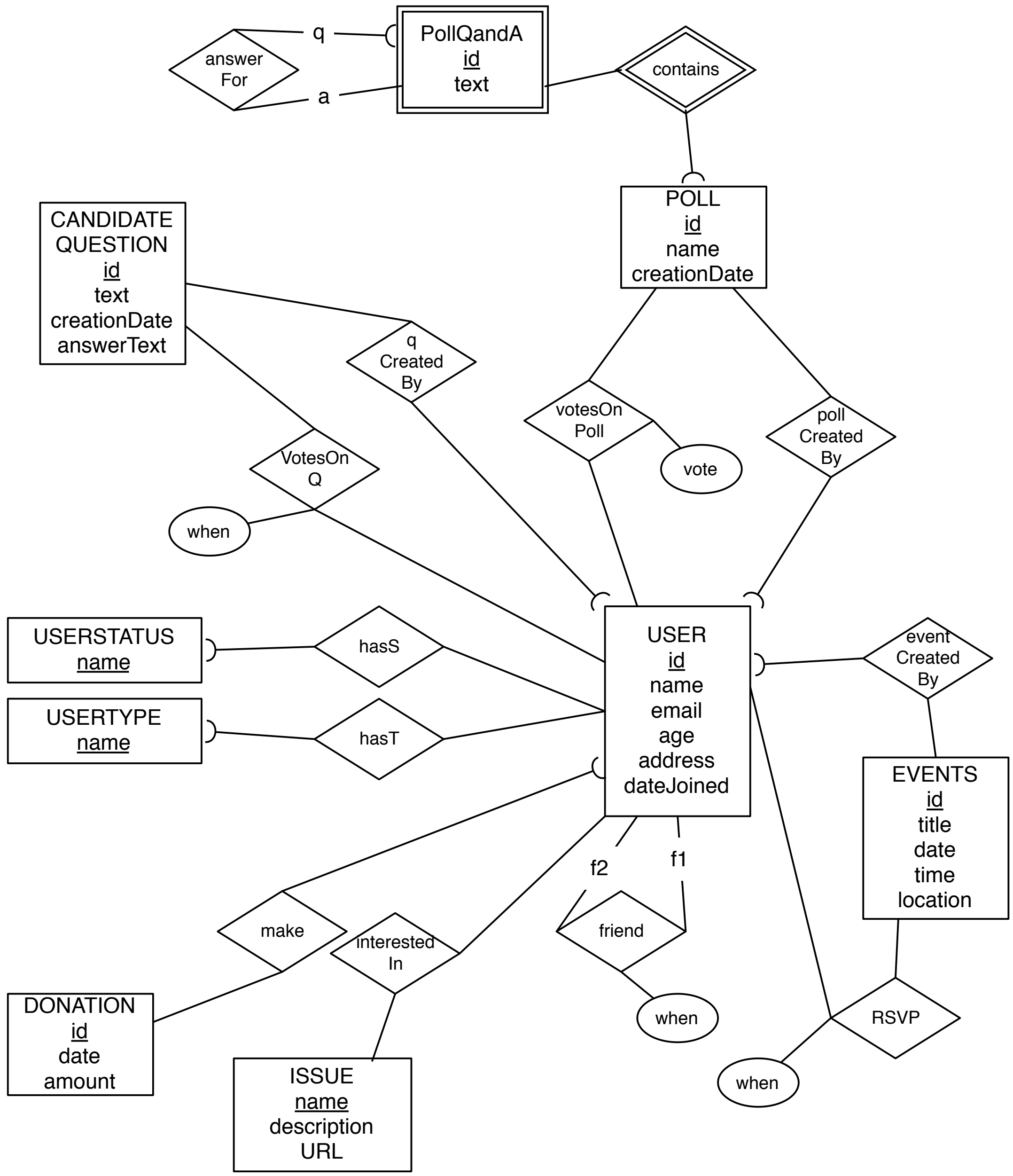

Figure 1: E-R Diagram for Question 3