Database Systems, CSCI 4380-01
Homework # 2 Answers

Question 1 (10 points each). For the following relations: first find the keys, then discuss whether it is in 3NF and/or BCNF with a short explanation of why or why not.

(a) $R(A, B, C, D, E, F)$, $\mathcal{F} = \{ABC \rightarrow DEF, CE \rightarrow A\}$

Answer. Keys: ABC, BCE

This is not in BCNF because CE is not a superkey and $CE \rightarrow A$ is not a trivial functional dependency.

This is in 3NF because ABC is a superkey for the first fd and A is a prime attribute for the second fd.

(b) $R(A, B, C, D, E, F)$, $\mathcal{F} = \{AB \rightarrow CDE, ABE \rightarrow F\}$

Answer. Keys: AB

This is in BCNF and 3NF because both fds have a superkey on the left.

(c) $R(A, B, C, D, E, F)$, $\mathcal{F} = \{AB \rightarrow DE, CD \rightarrow A\}$

Answer. Keys: ABCF, BCDF

Not in BCNF or 3NF because $AB \rightarrow DE$ is not trivial, does not have a superkey on the left and does not have all prime attributes on the right. If a relation is not in 3NF, it is not BCNF either.

(d) $R(A, B, C, D, E, F)$, $\mathcal{F} = \{BCD \rightarrow AEF, AB \rightarrow C\}$

Answer. Keys: BCD, ABD

Not in BCNF because none of the fds are trivial and the second fd does not have a superkey on the left.

It is in 3NF because the first fd has a superkey on the left and the second fd has a prime attribute (C) on the right.

Question 2 (10 points). You are given the following set $\mathcal{F}$ of functional dependencies. Convert this to a minimal basis. Show your work at each step by providing sufficient detail. You do not have to show all unsuccessful steps (what you tried to remove and decided cannot be removed), but you must show what needs to be removed and why.

$R(A, B, C, D, E, F, G)$ with $\mathcal{F} = \{BC \rightarrow ADE, ABD \rightarrow BEF, E \rightarrow A, ABC \rightarrow G\}$

Answer.

- First, convert to a basis by decomposing the right hand side.

$$\begin{aligned} \mathcal{F} \;=\; & \{BC \rightarrow A, BC \rightarrow D, BC \rightarrow E, ABD \rightarrow B, ABD \rightarrow E, ABD \rightarrow F, \\ & E \rightarrow A, ABC \rightarrow G\} \end{aligned}$$

- Second, remove trivial functional dependencies, i.e. $ABD \rightarrow B$.

$$\mathcal{F}2 = \{BC \rightarrow A, BC \rightarrow D, BC \rightarrow E, ABD \rightarrow E, ABD \rightarrow F, E \rightarrow A, ABC \rightarrow G\}$$

- Third, check if we can remove $ABD \rightarrow E$ to get:

$$\mathcal{F}21 = \{BC \rightarrow A, BC \rightarrow D, BC \rightarrow E, ABD \rightarrow F, E \rightarrow A, ABC \rightarrow G\}$$

  Given $\mathcal{F}2$, $ABD^+ = \{A, B, D, E, F\}$, given $\mathcal{F}21$, $ABD^+ = \{A, B, D, F\}$. Since they are not equal, we cannot.

- Check if we can remove $BC \rightarrow E$ from $\mathcal{F}2$ to get:

$$\mathcal{F}2 = \{BC \rightarrow A, BC \rightarrow D, BC \rightarrow E, ABD \rightarrow E, ABD \rightarrow F, E \rightarrow A, ABC \rightarrow G\}$$
$$\mathcal{F}3 = \{BC \rightarrow A, BC \rightarrow D, ABD \rightarrow E, ABD \rightarrow F, E \rightarrow A, ABC \rightarrow G\}$$

  Given $\mathcal{F}2$, $BC^+ = \{B, C, A, D, E, F, G\}$, given $\mathcal{F}3$, $BC^+ = \{B, C, A, D, E, F, G\}$. Since they are equal, we can remove it.

- We cannot remove any other functional dependencies fully, so we have:

$$\mathcal{F}3 = \{BC \rightarrow A, BC \rightarrow D, ABD \rightarrow E, ABD \rightarrow F, E \rightarrow A, ABC \rightarrow G\}$$

  Now, we check whether we can simplify the left hand side of any functional dependencies.

  We check if we can change $ABC \rightarrow G$ to $BC \rightarrow G$ to get:

$$\mathcal{F}4 = \{BC \rightarrow A, BC \rightarrow D, ABD \rightarrow E, ABD \rightarrow F, E \rightarrow A, BC \rightarrow G\}$$

  Given $\mathcal{F}3$, we have $BC^+ = \{B, C, A, D, E, F, G\}$. Given $\mathcal{F}4$, we have $BC^+ = \{B, C, A, D, E, F, G\}$. Since they are equal, we can make the change.

  There is no other possible change.

The minimal basis is therefore given by $\mathcal{F}4$.

Question 3 (10 points each). You are given the following relation:

$R(A, B, C, D, E, F)$, $\mathcal{F} = \{A \rightarrow C, CE \rightarrow D, CD \rightarrow A, DF \rightarrow A\}$

Find whether the following decompositions of this relation are lossless or not using the Chase algorithm:

(a) Decomposition 1: $R1(A,C,E,D)$, $R2(B,E,F)$, $R3(A,E,F)$

Answer.

A	B	C	D	E	F
a	b1	c	d	e	f1
a2	b	c2	d2	e	f
a	b3	c3	d3	e	f

Apply $A \rightarrow C$ to get:

A	B	C	D	E	F
a	b1	c	d	e	f1
a2	b	c2	d2	e	f
a	b3	c	d3	e	f

Apply $CE \rightarrow D$ to get:

A	B	C	D	E	F
a	b1	c	d	e	f1
a2	b	c2	d2	e	f
a	b3	c	d	e	f

We cannot apply any more rules and there is no row without a subscript. As a result, this decomposition is lossy.

(b) Decomposition 2: $R1(A,B,C,D)$, $R2(A,D,E)$, $R3(B,C,E,F)$

Answer.

A	B	C	D	E	F
a	b	c	d	e1	f1
a	b2	c2	d	e	f2
a3	b	c	d3	e	f

Apply $A \rightarrow C$ to get:

A	B	C	D	E	F
a	b	c	d	e1	f1
a	b2	c	d	e	f2
a3	b	c	d3	e	f

Apply $CE \rightarrow D$ to get:

A	B	C	D	E	F
a	b	c	d	e1	f1
a	b2	c	d	e	f2
a3	b	c	d	e	f

Apply $CD \rightarrow A$ to get:

A	B	C	D	E	F
a	b	c	d	e1	f1
a	b2	c	d	e	f2
a	b	c	d	e	f

Since last row has no subscript, this decomposition is lossless.

Question 4 (20 points). You are given the following relation:

$R(A, B, C, D, E, F)$, $\mathcal{F} = \{A \rightarrow BC, C \rightarrow D, D \rightarrow E\}$

Convert it to BCNF using the BCNF decomposition, starting with the functional dependency: $C \rightarrow D$.

In each step, find the projection of the functional dependencies for the decomposed relations and check whether the resulting relations are in BCNF or not. Continue until all relations are in BCNF.

When finished, find whether the resulting decomposition is dependency preseving or not (take the union of all projected functional dependencies and check if it is equivalent to the original set above).

Answer 1 (following the method I taught in class).

$R(A, B, C, D, E, F)$, $\mathcal{F} = \{A \rightarrow BC, C \rightarrow D, D \rightarrow E\}$

Key: AF, decompose by removing $C \rightarrow D$

$R1(C, D)$, $\mathcal{F}1 = \{C \rightarrow D\}$, Key: C, in BCNF
$R2(A, B, C, E, F)$, $\mathcal{F}2 = \{A \rightarrow BCE\}$, Key: AF, not in BCNF

R2 is not in BCNF. Take out $A \rightarrow BCE$ to decompose R2.

$R21(A, B, C, E)$, $\mathcal{F}21 = \{A \rightarrow BCE\}$, Key: A, in BCNF
$R22(A, F)$, $\mathcal{F}22 = \{\}$, Key: AF, in BCNF

The decomposition is complete. The resulting relations: $R1(C, D)$, $\mathcal{F}1 = \{C \rightarrow D\}$, Key: C, in BCNF
$R21(A, B, C, E)$, $\mathcal{F}21 = \{A \rightarrow BCE\}$, Key: A, in BCNF
$R22(A, F)$, $\mathcal{F}22 = \{\}$, Key: AF, in BCNF

Check for dependency preserving or not.

Original set: $\mathcal{F} = \{A \rightarrow BC, C \rightarrow D, D \rightarrow E\}$
Union of projections to decomposed relations: $\mathcal{F}d = \{A \rightarrow BCE, C \rightarrow D\}$

This is not decomposition preserving before given $\mathcal{F}d$, $D^+ = \{D\}$. As a result, $\mathcal{F}d$ does not imply $D \rightarrow E\}$.

Answer 2 (following the method in my course notes). Ok, as it turns out, I did not really follow the method in my course notes and the book properly in class. So, we will accept both answers. However, this method actually gives better results. So, I recommend you use this method in practice (life after this class).

Key: AF, decompose by removing $C \rightarrow D$

First find $C^+ = \{C, D, E\}$, hence use $C \rightarrow DE$ to decompose:

$R1(C, D, E)$, $\mathcal{F}1 = \{C \rightarrow D, D \rightarrow E\}$, Key: C, not in BCNF
$R2(A, B, C, F)$, $\mathcal{F}2 = \{A \rightarrow BC\}$, Key: AF, not in BCNF

Decompose R2 by removing $A \rightarrow BC$ to decompose R2.

$R21(A, B, C)$, $\mathcal{F}21 = \{A \rightarrow BC\}$, Key: A, in BCNF
$R22(A, F)$, $\mathcal{F}22 = \{\}$, Key: AF, in BCNF

Decompose R1 by removing $D \rightarrow E$ to get:

$R11(D, E)$, $\mathcal{F}11 = \{D \rightarrow E\}$, Key: D, in BCNF
$R12(C, D)$, $\mathcal{F}12 = \{C \rightarrow D\}$, Key: C, in BCNF

The decomposition is complete. The resulting relations:

$R11(D, E)$, $\mathcal{F}11 = \{D \rightarrow E\}$, Key: D, in BCNF
$R12(C, D)$, $\mathcal{F}12 = \{C \rightarrow D\}$, Key: C, in BCNF
$R21(A, B, C)$, $\mathcal{F}21 = \{A \rightarrow BC\}$, Key: A, in BCNF
$R22(A, F)$, $\mathcal{F}22 = \{\}$, Key: AF, in BCNF

The main distinction is that is we are taking out a functional dependency $X \rightarrow Y$, we first compute X^+. Then, in decomposition will have relations with attributes (a) X^+ (which includes Y as well as other attributes), and (b) all attributes in R except for $X^+ - X$ (so anything implied by X is removed, but X is not removed).

Check for dependency preserving or not.

It is easy to see that this is dependency preserving because the union of the functional depenedency sets for the decomposed relations is identical to the original set: $\mathcal{F} = \{A \rightarrow BC, C \rightarrow D, D \rightarrow E\}$

Question 5 (10 points). Convert the relation from Question 4 to 3NF using the 3NF decomposition. Show your work.

Answer. Key AF

$R(A, B, C, D, E, F)$, $\mathcal{F} = \{A \rightarrow BC, C \rightarrow D, D \rightarrow E\}$

$R1(A, B, C)$ $\{A \rightarrow BC\}$, both in 3NF and BCNF
$R2(C, D)$ $\{C \rightarrow D\}$, both in 3NF and BCNF
$R3(D, E)$ $\{D \rightarrow E\}$, both in 3NF and BCNF
$R4(A, F)$, $\{\}$, both in 3NF and BCNF

Question 6 (30 points). You are given the following relation. Create an ER diagram that represents all the requirements below precisely.

A note on drawing ER diagrams. You will find that the diagrams can get very large if entities have a lot of attributes. To avoid this, you can simply list the attributes for the entities inside the box for that entity. Remember to underline the key attributes.

Voter databases have become extremely important for election campaigns to find individuals who might vote for a candidate based on their past voting records and other issues they care about. Often campaign target specific neighborhooods and even individuals to contact, even customize their messages towards these individuals. Also, an amazing amount of personal information is collected and sold in these databases. A recently discovered security flaw exposed all sorts of extremely personal information about 191 million voters to the world. This shows you the scale of these databases and you can expect there are much larger ones out there.

Ok, you are hired by the campaign manager of BirdPerson (see on the left) to create one of these scary databases, with the hope that BirdPerson can win the next big elections.

Your database will store information about individual voters. First, you will store information for each person that is entered on voter registration as shown here:

http://www.elections.ny.gov/NYSBOE/download/voting/voteform.pdf

The information for a voter includes last and first name, middle initial, gender, birth date, phone, email, address for where the voter lives (street address, apt. number, city, state, zip) and address for where the voter receives mail, has the person voted before (Y/N) and which year. In addition, the voter is asked her DMV number or last 4 digits of SSN, or that the voter can indicate that she does not have either of these. Each voter can enroll in one party or none at all. You also store the occupation of the person, such as teacher or business owner, and the employer of the person.

An important part of each voter is their voting record. Basically, the database stores all the past elections that they voted in. Elections take place in a specific year and can be presidential, primaries, senate/congress or state level. Primaries are for a specific party. Other elections simply have a type. You do not know who the person voted for, but that they have voted.

Also, you will store political donations that they have made to a specific candidate for a specific election. Many donations can be made for a specific candidate and a person can donate to multiple candidates. For each donation, you need to store the dollar amount. You can see that donation data is public, for example at opensecrets.org.

You also want to store charitable donations made by the voter, to different organizations. You need to store the amount and date of each donation. Of course, it is possible to make multiple donations in a specific year.

Other information stored for voters include the names of magazines the person subscribes to and has subscribed to in the past.

You realize that you need to relate some of this information to each other for deeper analysis. So, you will store a number of issues that voters likely care for (immigration, education, gun control, tax cuts), for each issue you will store the name and description. Then, you will store whether a specific charitable organization stands for an issue and whether the standing is pro or con. You also want to know which candidates in a specific elections stood for a specific issue, again whether the candidate's position was pro or con on this issue.

Finally, to show off your extra skills (because you are from RPI after all), you will also store the various social media usernames for each voter, if they are known. You will mine as much information as you can from each social media outlet and store the following for each outlet: overall number of friends the voter has, number of messages per week, how frequently (percent of messages) the voter talks about the issues in the database and the date the analysis was performed.

BirdPerson thanks you for your contributions to his campaign!

Answer. The diagram is in the next page.

Note that the main entity, Voter has too many attributes to put in the ER diagram, it will just become too large. So I list them here first. The remaining attributes are in the ER diagram. Also, since voters may not have SSN or DMV, we will give them a separate voterid.

Entity Voters: <u>voterid</u>, lname, fname, mi, gender, bday, phone, email, street, aptnum, state, city, zip, mail_street, mail_aptnum, mail_state, mail_city, mail_zip, votedBefore?, votedWhen, dmvnum, ssn_last4, hasnodmv, hasnossn, occupation, employer.

Also note that technically you do not need votedBefore?, hasnodmv, hasnossn attributes because in an ideal world if any of the votedWhen, dmvnum, ssn_last4 value is empty (null), then these attributes are true. However, in real life, there is a lot of missing information. It would be good to know if the value is missing or that you know that it does not exist.

Combined all elections into a single entity for simplicity, but you can choose to put local elections as a separate entity or a subclass of regular elections.

Candidates were the trickiest part of the database. I wanted to identify the same person across different elections, but also represent how their personality seems to change with elections. So, I decided to create a new entity called election candidates who are for a specific election and a specific party in that election. This not necessarily a ternary relation, but if you had a ternary relation, it should be one-one-many.

The main limitation of this model is that it is possible for a person to have two election candidate personalities, which is not allowed in real life. One way to allow this is to use a weak entity for "Election Candidates", with an identifying relationship connecting to elections and a party. This would imply that there is exactly one candidate for each party and each election. This might be too limiting in some cases, so I decided to go with this version.

Political donations are made for a specific candidate in a specific election. As the election candidate is now a unique thing, this information will be implied when we find who the candidate is. If it is possible to donate to a specific party only, not a specific candidate, then you would need to add a second relationship from political donations to election and parties. The question does not ask for this explicitly.

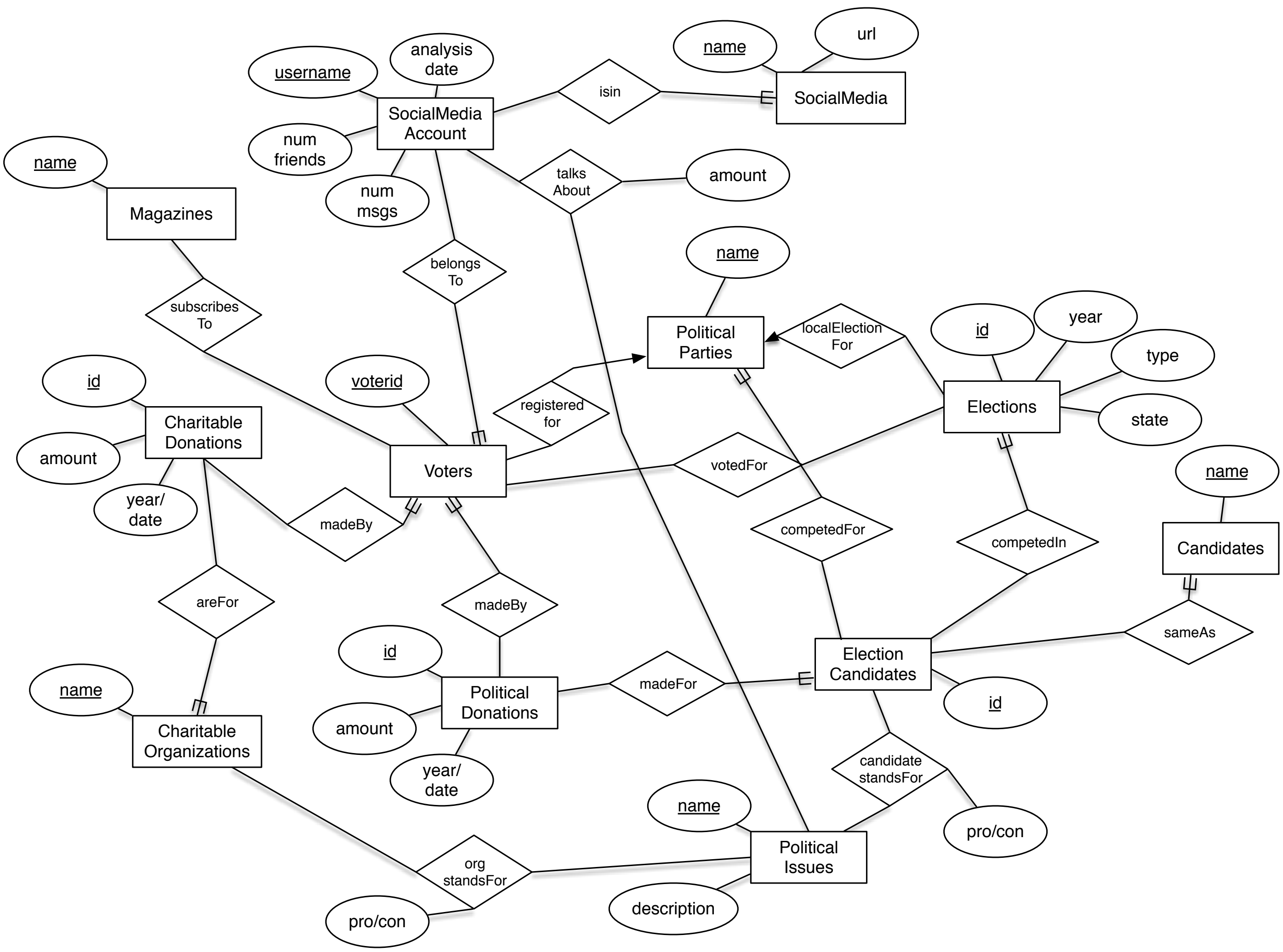

Figure 1: ER Diagram for HW#2 Question 6 Solution