Database Systems, CSCI 4380-01
Homework # 3 Solutions
Due Thursday February 17, 2011 at 2 pm

Question 1 (10 points). Given the following functional dependencies and decomposition of $R(A, B, C, D, E, F)$ and

$\mathcal{F} = \{CD \rightarrow EA, A \rightarrow DB\}$

$R11(A, D)$, $R12(A, C, E)$, $R21(C, D, B)$, $R22(C, D, F)$

Use the chase algorithm to show whether this decomposition is lossless or not.

Answer:

A	B	C	D	E	F
a	b_1	c_1	d	e_1	f_1
a	b_2	c	d_2	e	f_2
a_3	b	c	d	e_3	f_3
a_4	b_4	c	d	e_4	f

$\xrightarrow{CD \rightarrow EA, row3}$

A	B	C	D	E	F
a	b_1	c_1	d	e_1	f_1
a	b_2	c	d_2	e	f_2
a_3	b	c	d	e_3	f_3
a_3	b_4	c	d	e_3	f

$\xrightarrow{A \rightarrow DB, row3}$

A	B	C	D	E	F
a	b_1	c_1	d	e_1	f_1
a	b_1	c	d	e	f_2
a_3	b	c	d	e_3	f_3
a_3	b	c	d	e_3	f

$\xrightarrow{CD \rightarrow EA, row3}$

A	B	C	D	E	F
a	b_1	c_1	d	e_1	f_1
a	b_1	c	d	e	f_2
a	b	c	d	e	f_3
a	b	c	d	e	f

There is a row4 without subscripts so we can stop. The result is that the decomposition is lossless as there is a row4 without subscripts.

Question 2 (20 points). Given the following functional dependencies and decomposition of $R(A, B, C, D, E, F)$ and

$\mathcal{F} = \{CD \rightarrow EA, A \rightarrow DB, CE \rightarrow D\}$

$R1(A, B, D)$ and $R2(A, C, E, F)$

Is the above decomposition dependency preserving? To do so, first find the projection of the above functional dependencies for each decomposed relation. Do not show all details of your work, only sufficient to show how you got the result. Then, check whether the union of these projected functional dependencies equal to $\{calF\}$.

Hint. To check whether two sets F_1 and F_2 of functional dependencies are equivalent, make sure all functional dependencies in F_1 are implied by F_2, and all functional dependencies in F_2 are implied by F_1.

Answer:

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

$R1(A, B, D)$, $\mathcal{F}_{R1} = \{A \rightarrow DB\}$;

$R2(A, C, E, F)$, $\mathcal{F}_{R2} = \{CE \rightarrow A, AC \rightarrow E\}$;

$R1 \cup R2 = (A, B, C, D, E, F)$;

$\mathcal{F}_{R1} \cup \mathcal{F}_{R2} = \{A \rightarrow DB, CE \rightarrow A, AC \rightarrow E\}$.

$CD^+ = \{C, D\}$, that means $CD \rightarrow CD$.

$A^+ = \{A, B, D\}$, that means $A \rightarrow BD$.

$CE^+ = \{A, C, E\}$, that means $CE \rightarrow A$.

Given these projected functional depedencies, we can see that the union of these functional dependencies do not imply the functional depedency $CD \rightarrow EA$. As such, we have clearly lost the functional dependency $CD \rightarrow EA$, thus this decomposition is not dependency preserving.

Question 3 (30 points). Create an E-R diagram for the following database. Show all your entities and relationships, participation constraints of the relationships, roles of relationships when appropriate.

Suppose you are creating an application for a community service (somewhat like Yelp, but broader) in Troy. The idea is that you want people to write reviews about businesses in Troy such as restaurants, car repair shop, etc. For businesses you will store information about the business that allows you to locate them, find out who owns them (if the information is available). Furthermore, you can input information about various locations in Troy, i.e. a street corner etc. where people can discuss crime in those locations, post incidents, etc. For places, you store street address. Furthermore, you can post information about people in Troy, such as your favorite land lord or the mayor. For people, you store their name and a descriptive field. The reviews have date, text, poster as well as a flag that indicates the tone of the review, from -3 to 3 (extremely negative to highly positive). People can respond to each other's reviews.

All the information is posted by people using the system. As such a system can get out of control, you want to have a way to identify good users from bad. To do so, you want to establish a mechanism for people to friend each other, vote for other's comments (good or bad). Responding to a review and voting are independent acts.

Answer: Figure 1 shows your TAs solution to Question 3 without the bonus part and Figure 2 shows my solution including the bonus. Note that my solution is geared towards how I would go about actually implementing this system. So, it may contain more detail than was asked for in the question. I used circle at the end of an arrow to show necessity (for rounded arrow).

Bonus (+10 points). You can also allow people to link places and people. For example, allow them to say that a person X in the system works for or owns a company Y. Note that users are going to determine the type of the relationships as well as the connecting entities.

Question 4 (10 points). Convert the following E-R model to the relational data model.

Answer:

$Project(\underline{id}, name, startDate, budget, departmenttId, managerId)$

departmentId points to Department(deptId)

managerId points to Staff(empId)

$Department(\underline{deptId}, name, officeNo, deanId, Buildingabr)$

deanId points to Staff(empId)

Buildingabr points to Building(abr)

$Staff(\underline{empId}, name, officeNo, phone, fax, datestarted, currentpositionname, startDate, Buildingabr)$

currentpositionname points to Position(name)

Buildingabr points to Buliding(abr)

$workin(\underline{departmentId}, \underline{employeeId})$

departmentId points to Department(deptId)

employeeId points to Staff(empId)

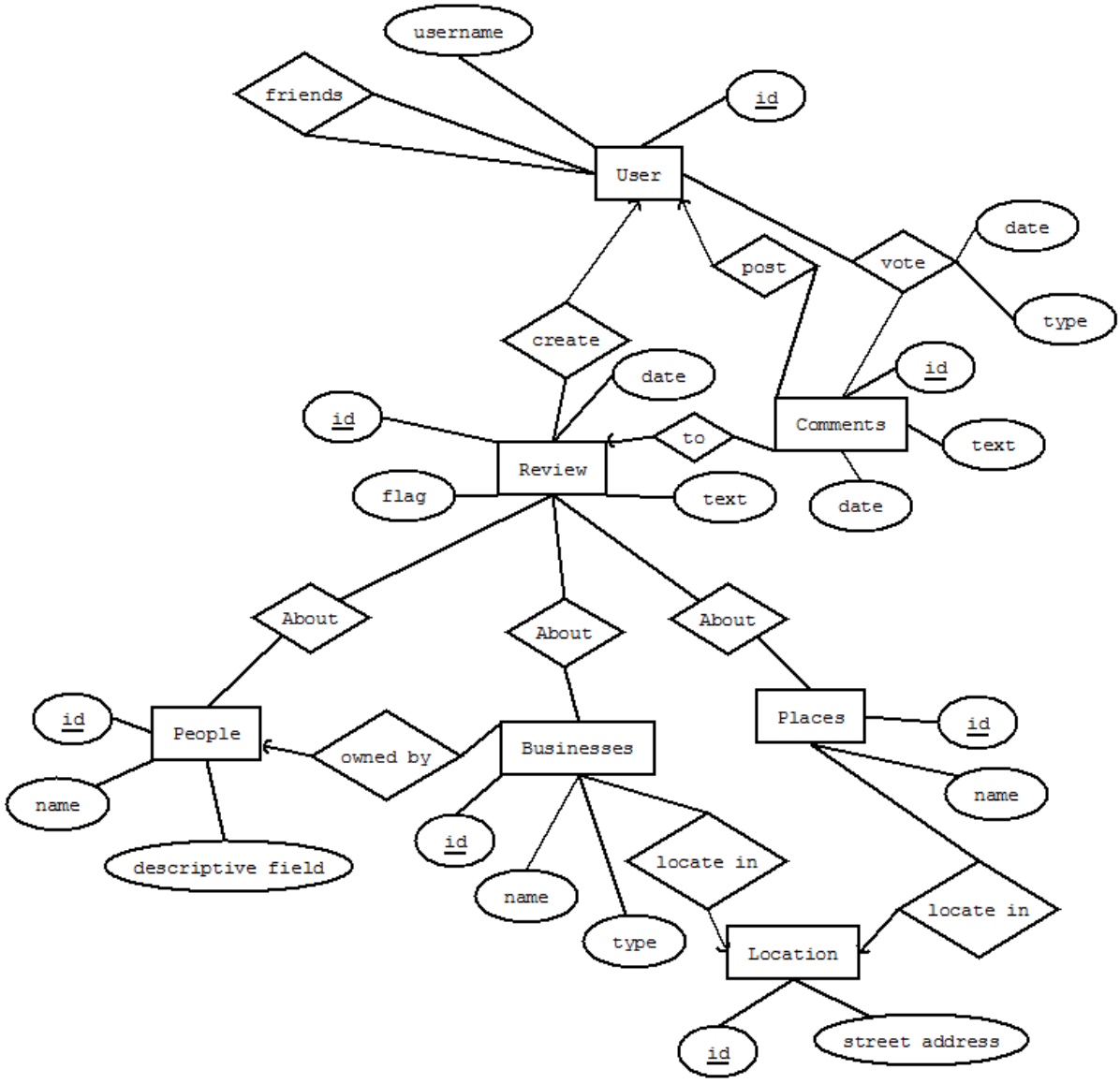

Figure 1: E-R Diagram for Question 3

$Building(\underline{abr}, name, street, state, city, zip)$

$PastPosition(\underline{id}, startYear, endYear)$

$Position(\underline{name})$

$heldposition(\underline{employeeId}, \underline{PastPositionid}.\underline{PastPositionname})$

employeeId points to Staff(empId)

PastPositionid points to PastPosition(id)

PastPositionname points to Position(name)

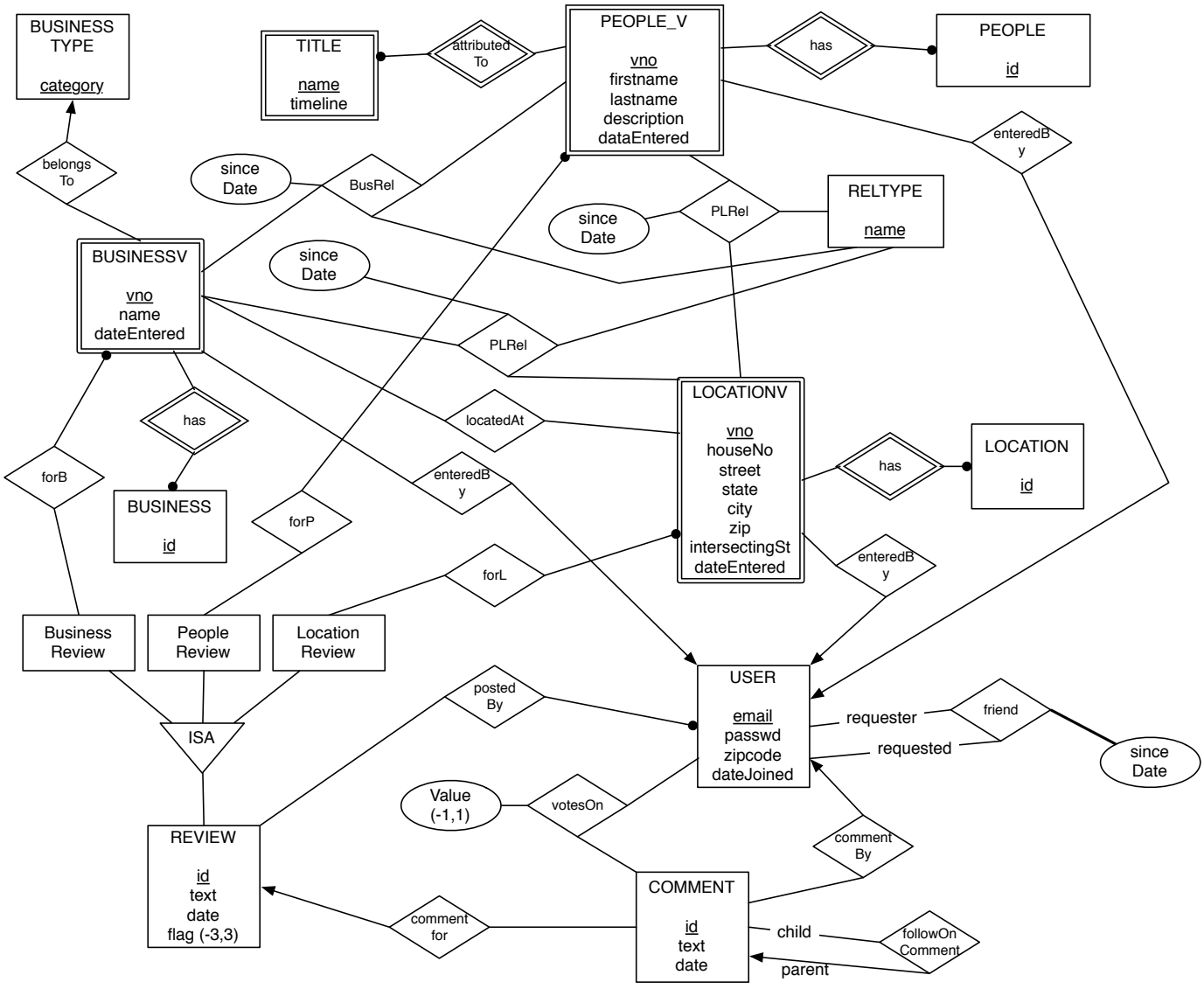

Figure 2: E-R Diagram for Question 3 with Bonus