Database Systems — CSci 4380
Midterm Exam #1
September 27, 2018

SOLUTIONS

Question 1 (10*3=30 points). Write the following queries using relational algebra using the data model below. The model is described in detail in the back of the exam.

Users(email, password, name, street, state, city, zip, country, ccno)
Houses(id, label, description, street, state, city, zip, country, price, owneremail)
HouseAmenities(houseid, amenity)
Trips(id, fromdate, todate, isconfirmed, totalprice, customeremail, houseid)
Reviews(tripid, isclean, isgoodvalue, isgoodrenter, reviewtext)
Messages(id, senderemail, receiveremail, messagetext, sentdatetime, prevmid)

(a) Return the email and name of all users who are both renters (i.e. owners of houses) and customers (i.e. customers on trips).

Solutions:

$$R1 = \Pi_{email,name}(Users \bowtie_{email=owneremail} ((Trips \bowtie_{owneremail=customeremail} Houses)))$$

Alternatively, you can use intersection instead of join in some places.

(b) Find the id and label of all houses in Troy (city) NY (State) with price below $100 that have both the amenity `'Helipad'` and the amenity `'Batcave'`.

Solutions:

$$\begin{aligned} R1 &= \Pi_{houseid}(\sigma_{amenity='Helipad'} HouseAmenities) \\ R2 &= \Pi_{houseid}(\sigma_{amenity='Batcave'} HouseAmenities) \\ R3 &= (R1 \cap R2) \bowtie_{houseid=id} (\sigma_{price<100 \, and \, city='Troy' \, and \, state='NY'} Houses) \\ R4 &= \Pi_{id,label} R3 \end{aligned}$$

Alternatively, you can join R1 and R2 instead of intersecting.

(c) Return the email and name of all renters who only received 5 star reviews for their houses with respect to isgoodvalue attribute.

Solutions:

$$\begin{aligned} R1 &= \Pi_{houseid}((\sigma_{isgoodvalue=5} Reviews) \bowtie id = tripid Trips) \\ R2 &= \Pi_{owneremail}(R1 \bowtie Houses) \\ R3 &= \Pi_{houseid}((\sigma_{isgoodvalue<5} Reviews) \bowtie id = tripid Trips) \\ R4 &= \Pi_{owneremail}(R3 \bowtie Houses) \\ R5 &= \Pi_{email,name}((R2 - R4) \bowtie_{email=owneremail} Users) \end{aligned}$$

Note that the answer requires set subtraction and the two relations being subtracted must have owneremail only. You are trying to find owners who had at least one 5 star review and no lower review in any house or trip.

Question 2 (12+12=24 points). Suppose you are given the following relations to add to the data model in the appendix. Answer questions regarding each additional relation below.

(a) Given a trip, the customer may add other names of people staying in the house with him or her as other travelers.

`OtherTravelers(tripid, firstname, lastname, email, phone)`

Other travelers on a trip are uniquely identified by their email. The same person may take multiple trips as an "other traveler". An other traveler can have multiple phones listed in the database.

(1) Based on the above information, list all applicable functional dependencies.

(2) What are the key(s)?

(3) Is this relation in BCNF? 3NF? Explain why or why not.

(4) If it is not in BCNF, use BCNF decomposition to get relations that are in BCNF.

Solutions:

(1) Only one fd: tripid, email → firstname, lastname

(2) Key: tripid, email, phone

(3) Not in BCNF or 3NF as left hand side is not a superkey and the righthand is not composed of prime attributes.

(4) R1(tripid,email,firstname, lastname), Key: tripid,email, tripid, email → firstname, lastname

In BCNF
R2(tripid,email,phone) Key: all attributes, no f.d., in BCNF

Alternative answer. We also accept the answer:

email → firstname, lastname

Parts 2 and 3 remain the same for this f.d. However, BCNF decomposition returns: R1(email,firstname, lastname) and R2(tripid,email,phone).

(b) The following is a new relation for saving some houses in lists:

```
SavedHouses(houseid, saveddatetime, useremail, note, listname)
       houseid, useremail, listname → saveddatetime
       saveddatetime, useremail → houseid, useremail
       houseid, useremail → note
```

Answer each of the following with yes/no and write a sentence to explain your answer.

(1) Can a user save a house in multiple lists?

(2) Can a user save two different houses in the same datetime?

(3) Can a user have two different notes for the same house in two different lists?

(4) What are the keys? Is this relation in 3NF?

Solutions:

(a) Can a user save a house in multiple lists?
YES, listname is not implied by houses and datetime

(b) Can a user save two different houses in the same datetime?
No, houseid is implied user and saved datetime.

(c) Can a user have two different notes for the same house in two different lists?
No, note is implied by user and house, so does not change based on a list.

(d) What are the keys? Is this relation in 3NF?
Keys: houseid, useremail, listname or saveddatetime, useremail, listname
Not in 3NF, last f.d. does not have a super key on the left and the right hand side is not a prime attribute.

Question 3 (12 points). You are given the following relation: `R(A,B,C,D,E,F,G)` with the following set of functional dependencies:

`F = { AB → CE, BF → G, C → A}`

(a) List all the keys.

(b) Convert this relation to 3NF using 3NF decomposition. Show your work.

(c) For each resulting relation, show the key and state simply whether it is in BCNF or not.

Solutions:

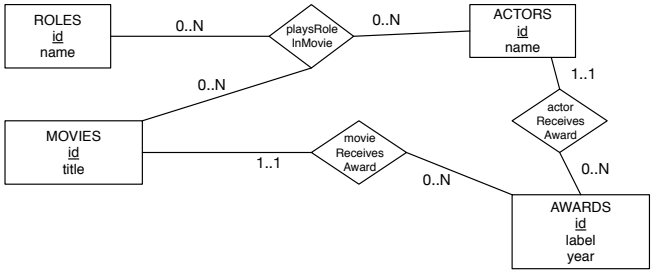
(a) Keys: ABDF, BCDF

(b) R1(A,B,C,E), AB → CE, C → A, keys: AB, BC, not in BCNF because of the second functional dependency.

R2(B,F,G), BF → G, Key: BF, in BCNF

R3(A,B,D,F), no f.d. key: all attributes. in BCNF (we need this for one of the keys, but B,C,D,F is a possible addition here).

Question 4 (6 points). You are given the following Entity-Relationship diagram. Based on this model, list all functional dependencies that this model implies.



Solutions:

actors: id → name
movies: id → title
roles: id → name
awards: id → label, year
awards.id → actors.id
awards.id → movies.id

Question 5 (12 points). Given the relation: R(A,B,C,D,E,F) and

$F = \{BC \rightarrow AE, DC \rightarrow F, AF \rightarrow D\}$

Show whether the following decomposition is lossless or not with the Chase algorithm. Show your work.

Decomposition: R1(A,B,E), R2(C,D,F), R3(A,D,F), R4(A,B,C,F)

Solutions:

A	B	C	D	E	F
a	b	c_1	d_1	e	f_1
a_2	b_2	c	d	e_2	f
a	b_3	c_3	d	e_3	f
a	b	c	d_4	e_4	f

Given AF $\rightarrow$ D, set d_4 to d.

A	B	C	D	E	F
a	b	c_1	d_1	e	f_1
a_2	b_2	c	d	e_2	f
a	b_3	c_3	d	e_3	f
a	b	c	d	e_4	f

Since we cannot simplify it further, this decomposition is lossy. There is no row without a subscript.

Question 6 (16 points). Create an Entity-Relationship diagram for the following database, capturing all the requirements below precisely. Make sure you list all the relevant attributes, underlining the keys. For each relationship, mark the participation constraints clearly (one-to-one, one-to-many or many-to-many). If you do not find a natural key for an entity, feel free to add an id attribute.

You are creating a database for storing data for a new social media company. In this database, you store users. Each user has an email, name, password, and a phone number. Emails are unique in the database.

Each user follows some users and is followed by some users. The following relationship is not symmetric; person A may follow person B while person B does not follow A. There are a number of known locations in the database. Each location has a unique coordinates (X,Y coordinates in the form of longitude and latitude values) and a label.

Each person generates posts. Each post has a datetime, text, URL and zero or more pictures. There can be at most one post by a specific user at a given datetime. Each picture may have a label and it may also have a location associated with it from the database. In addition, each picture may have zero or more rectangles drawn on them. Each rectangle is represented by its coordinates on the image (x1,y1,x2,y2) and another tagged user. Users may like or dislike posts of users. Users may also comment on other users' posts. Each comment has a sequence id (1,2,3,...) and text.

Solutions: Note: we will also accept id as key for location , like/dislike can be represented as two separate relationships. However, having a single relationship for like/dislike requires an additional attribute, without this attribute, it is impossible to describe whether the person liked or disliked the post.

Note that datetime being unique for a user implies a weak entity with datetime as key, depending on the users. Similarly, sequence id cannot be a key for comments, only if it is a weak entity. Be careful when using weak entities: there can be only one relationships that determines the weak entity (i.e. with double diamond). All other relationships should be regular. A common error is having comment text as an attribute of a commenting relationship. This makes it impossible to have a user comment on the same post more than once, hence it is not correct.

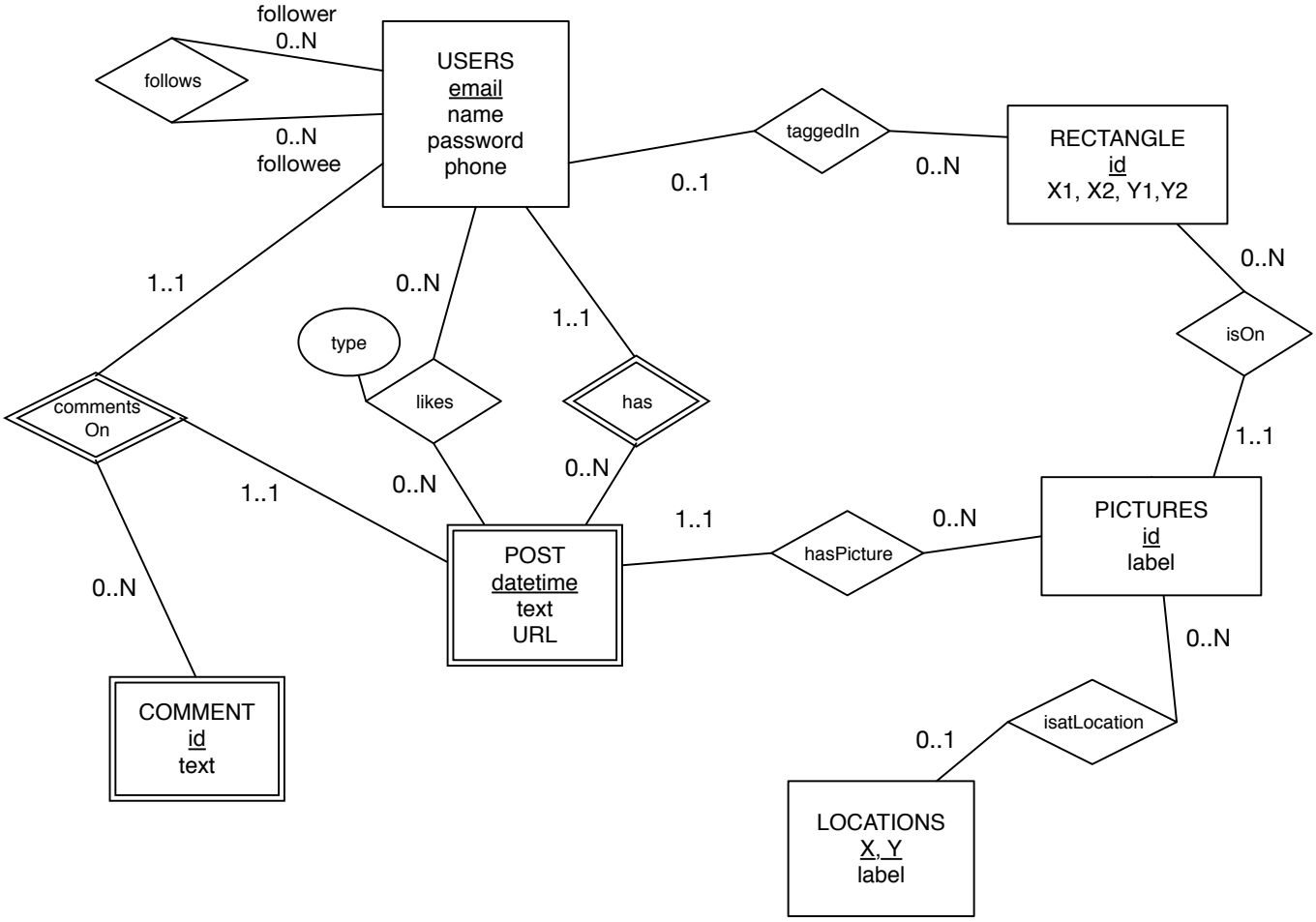


Figure 1: ER Diagram for Question 6

Use this page for scratch work only. Do not share your solutions or any drafts of your solutions with anyone.

Data model to be used in Exam #1

This is a data model based on the E-R problem from Homework#3, storing information about house rentals in a system similar to Airbnb. It is simplified in various ways to make it easier to user in an exam. Please read carefully. The keys of each relation are underlined.

Users(email, password, name, street, state, city, zip, country, ccno)

Stores information about all the users in the system, renters and customers including credit card number (ccno).

Houses(id, label, description, street, state, city, zip, country, price, owneremail)

Each house is listed by an owner and we store the email of the owner for each house. The daily price of a house is fixed for simplicity.

HouseAmenities(houseid, amenity)

This relation lists the various amenities a house has such as 'free wifi', 'garage parking', etc,

Trips(id, fromdate, todate, isconfirmed, totalprice, customeremail, houseid)

Each trip has a start and end date, whether it is confirmed or not (true/false value) and a total price for the trip. Each trip is by a user (customer), so his/her email is stored. This is the person paying for the trip. Each trip for a specific house, so the id of the house is also stored. (The renter can be deduced from the owner of the house).

Reviews(tripid, isclean, isgoodvalue, isgoodrenter, reviewtext)

We will only store the reviews of the houses in this database. There can only be one review for each trip, so we will use tripid as the key. Each review is for a specific house and its renter (stored in other relations) and contains 1-5 star values for isclean, isgoodvalue, isgoodrenter as well as a review text.

Messages(id, senderemail, receiveremail, messagetext, sentdatetime, prevmid)

Finally, we store messages simply from one user (sender) to another user (receiver) together with the text and datetime it was sent. The attribute `prevmid` can be null if this is an original email or store the id of the message that this message was sent in response to.

Use this page for scratch work only. Do not share your solutions or any drafts of your solutions with anyone.