Database Systems — CSci 4380
Midterm Exam #1
February 8, 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(username, name, email, password, address)
FriendsWith(username, username2, sincewhen)
Landmarks(landmarkname, landmarktype, state, city, country, zip)
Segments(segmentid, startx, starty, endx, endy, landmarkname)
Events(eventid, username, eventtype, eventdate, starttime, endtime)
DataPoints(pointid, eventid, seqno, starttime, endtime, segmentid)

(a) Find the `name` and `email` of all users who have started an event at `Monument Square` (name of landmark) in `Troy, NY` (city and state) on `Nov 23, 2018` (event date) at exactly `8 A.M` (event start time). The starting segment for an event is listed in the data point with `seqno` 1 for that event.

Solutions:

$$\begin{aligned} A &= Users \bowtie Events \bowtie \Pi_{eventid,segmentid}(DataPoints) \bowtie Segments \bowtie Landmarks \\ B &= \Pi_{name,email}(\sigma_C(A)) \end{aligned}$$

where $C : city = Troy$ and $state = NY$ and $eventdate = 11 - 23 - 2018$ and $starttime = 08 : 00$ and $landmarkname = MonumentSquare$.

Note if we did not do the projection for DataPoints, we would have endtime that is not necessarily the same as the one for the event. However, we don't have to project out start time. The same can be achieved by renaming as well.

(b) Return the `username` of users who have `run` (eventtype) in a circle in at least one event (i.e. they passed through the same segment more than once in the same event).

Solutions:

$$\begin{aligned} R1(seqno2, eventid, segmentid) &= \Pi_{seqno,eventid,segmentid} DataPoints \\ R2 &= (\sigma_{eventtype=run} Events) \bowtie (\sigma_{seqno2<>seqno}(R1 \bowtie DataPoints)) \\ Result &= \Pi_{username} R2 \end{aligned}$$

Basically we are looking for a join of DataPoints with DataPoints for the same event, segment but different sequence number.

(c) Find the `username` and `email` of users who have no friends.

Solutions:

$$\begin{aligned} R(username) &= \Pi_{username2} FriendsWith \\ R2 &= \Pi_{username} Users - (\Pi_{username} FriendsWith \cup R) \\ Result &= Pi_{username,email}(R2 \bowtie Users) \end{aligned}$$

Question 2 (14+12=26 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) You are given the following new relation for group events, i.e. events multiple people take part in:

`GroupEvents(gid, eventname, eventdate, starttime, endtime, start_segmentid, username)`

For a given event id (`gid`), there is a single event name, date, starting segment id (`start_segmentid`), start and end times. However, each event can have multiple participating users (`username`). Also, for a given `start_segmentid, eventdate, starttime`, there is a unique event id (only a single group event on a given date, time and location).

(You can rename the attributes to save time writing as: `GE(gid, en, ed, st, et, ss, user)`)

(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) Functional dependences:
`gid` $\rightarrow$ `eventname, eventdate, starttime, endtime, start_segmentid`
`eventdate, starttime, start_segmentid` $\rightarrow$ `gid`

(2) Keys: `gid, username` and `eventdate, starttime, start_segmentid, username`

(3) Not in BCNF or 3NF. First functional dependency does not have superkey on the left, and eventname is not a prime attribute.

(4) `G1(gid, eventname, eventdate, starttime, endtime, start_segmentid)`, both F.D.s Keys: `gid` or `eventdate, starttime, start_segmentid`. In BCNF.
`G2(gid, username)`

(b) The following is a new relation for storing user achievements based on specific events:

```
Achievements(aid, username, eventid, atype, aname, priority)
       aid → eventid, atype, priority
       atype → aname, priority
       eventid → username
```

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

(1) Can an event have two different associated users?

(2) Is it possible for a user to obtain two different achievements of the same type?

(3) Is `aid, atype` a superkey?

(4) Is the above set of functional dependencies minimal or can it be simplified?

(5) Is this relation in 3NF?

Solutions:

(a) Can an event have two different users?
No, `eventid` implies `username`.

(b) Is it possible for the user to obtain two different achievements with the same name?
Yes, as long as they have different `aid` values. If atype determines an achivement, then name cannot be different given it is implied by the type. Both answers will be accepted as long as they are well explained.

(c) Is the above set of functional dependencies minimal or can it be simplified?
No, it is not minimal. We can simplify
`aid → eventid, atype, priority`
to
`aid → eventid, atype`

(d) Is `aid, atype` a superkey?
Yes, `aid` is a key, so `aid, atype` is a superkey.

(e) Is this relation in 3NF?
No, none of the functional dependencies have a superkey on the left or all prime attributes on the right.

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

`F = { AB → D, BC → E, E → C, BE → F}`

(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:

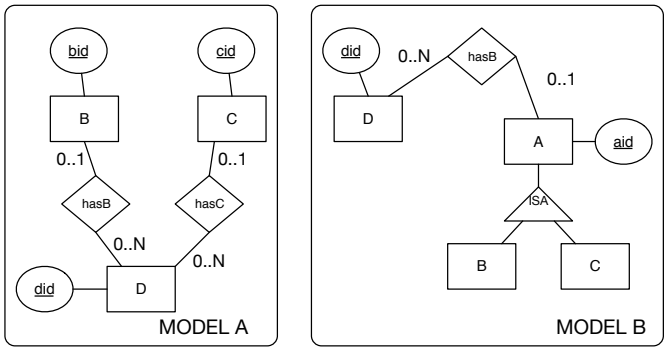
(a) Keys: ABC, ABE

(b) `R1(A,B,D) AB → D`
`R2(B,C,E) BC → E, E → C`
`R3(B,E,F) BE → F`
`R4(A,B,C)` (also we can use `R4(A,B,E)` instead, both are acceptable)

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

R1, Key: AB, in 3NF and BCNF
R2, Key: BC, BE, in 3NF but not in BCNF
R3, Key: BE, in 3NF and BCNF
R4, Key: ABC, in 3NF and BCNF

Question 4 (4 points). You are given the two E-R models on the right. Discuss whether model A and model B are the same, i.e. whether you can store the same information in them or not. If this depends on what type of hierarchy model B represents, you can also comment on it. Explain your reasoning.



Solutions:

They are not the same. In model A, we can store for a D a matching B and a matching C, two different attributes potentially referring to two different entities. In model B, we can only have a matching A, so even if A was both a B and a C it is still the same entity (while in A it can be different).

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

$F = \{ACD \rightarrow E, BC \rightarrow F, FD \rightarrow AE, C \rightarrow D\}$

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

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

Solutions:

A	B	C	D	E	F	G
a	b	c	d	e1	f1	g1
a2	b	c	d2	e	f2	g2
a	b3	c	d3	e3	f	g

Use $BC \rightarrow F$ to get:

A	B	C	D	E	F	G
a	b	c	d	e1	f1	g1
a2	b	c	d2	e	f1	g2
a	b3	c	d3	e3	f	g

$C \rightarrow D$ to get:

A	B	C	D	E	F	G
a	b	c	d	e1	f1	g1
a2	b	c	d	e	f1	g2
a	b3	c	d	e3	f	g

$FD \rightarrow AE$ to get:

A	B	C	D	E	F	G
a	b	c	d	e	f1	g1
a	b	c	d	e	f1	g2
a	b3	c	d	e3	f	g

$ACD \rightarrow E$ to get:

A	B	C	D	E	F	G
a	b	c	d	e	f1	g1
a	b	c	d	e	f1	g2
a	b3	c	d	e	f	g

Still there is no tuple without a subscript, so this decomposition is lossy.

Question 6 (16 points). Create an Entity-Relationship diagram for the following database. Make sure you list all the relevant attributes, underline the keys. For each relationship, mark the participation constraints clearly (one-to-one, one-to-many or many-to-many).

You are creating a database for storing information for a Netflix like service. The database stores movies with id, title, filename and TV shows with id, title. TVShows have episodes. Each episode has a corresponding TV show, a season id, an episode id, title, filename. Some episodes have a next episode (i.e. the episode that will automatically start showing once the user finishes watching the current episode!). The database also stores users, each user has an id, username, password. Users may watch zero or more movies, zero or more TV show episodes. For each movie or episode, the database stores a watch time value for each user, indicating how many minutes the user watched that movie or show. Finally, the database stores which movie appears similar to which other movie for a given user (to be able to make recommendations).

Solutions: Note that it is also possible to model episodes as a weak entity dependent on TVShows, with episodeid and seasonid as its keys.

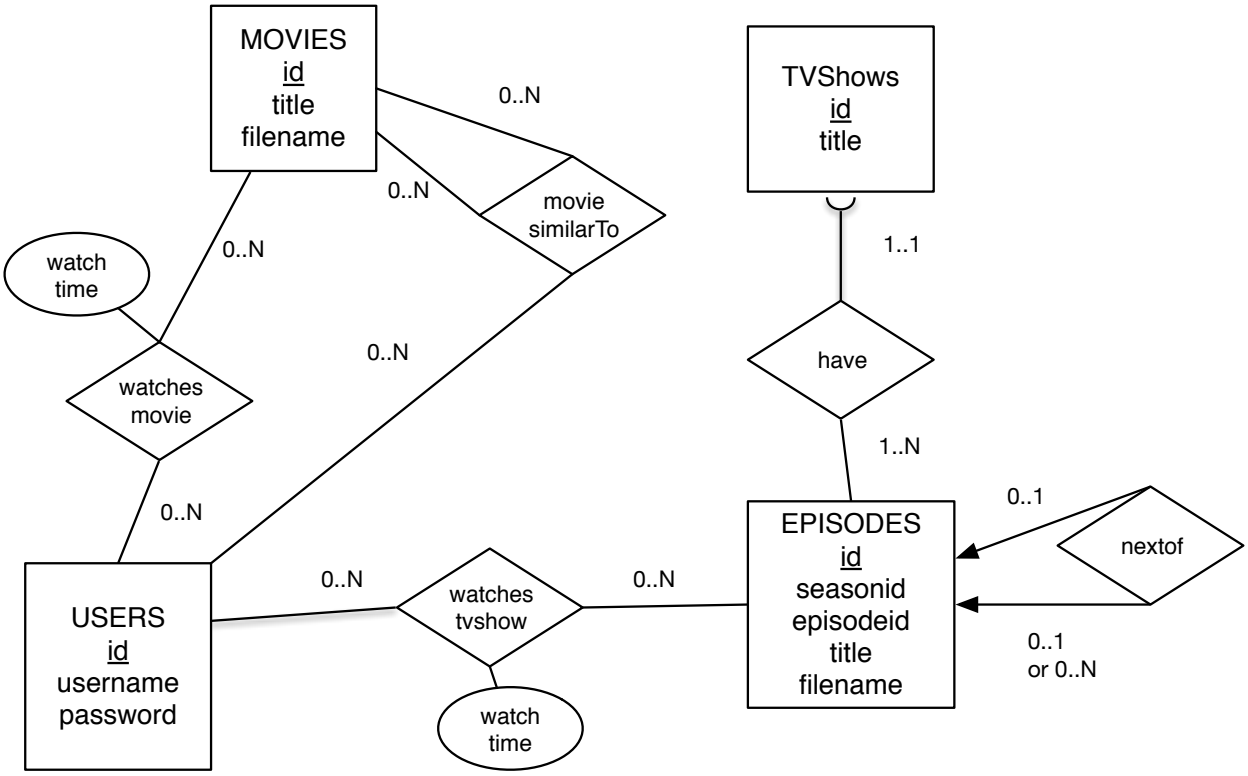


Figure 1: ER Diagram for Question 6

Use this page for scratch work only.

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 where people run or go on bicycle trips. The keys of each relation are underlined.

Users(<u>username</u>, name, email, password, address)

Stores information about the users in the system.

FriendsWith(<u>username, username2</u>, sincewhen)

Both `username` and `username2` refers to `username` in `Users`. Friendship is mutual, but we don't store both pairs (A,B and B,A). The first user (`username`) refers to the user who initiated the friendship request. `sincewhen` is the data the friendship was established.

Landmarks(<u>landmarkname</u>, landmarktype, state, city, country, zip)

Each landmark has a name, type (building, monument, etc.) and address.

Segments(<u>segmentid</u>, startx, starty, endx, endy, landmarkname)

A segment is a road segment where `startx, starty, endx, endy` refer to starting and end longitude and latitude. Each segment may have an associated landmark, referenced with `landmarkname`. If there is no landmark, landmarkname is equal to NULL.

Events(<u>eventid</u>, username, eventtype, eventdate, starttime, endtime)

Each event is created by a user, has a type (such as running and cycling), a specific date (day, month, year) and start and end times given in the 24 hour form, i.e. 16:00.

DataPoints(<u>pointid</u>, eventid, seqno, starttime, endtime, segmentid)

A data point is a point on someone's run or cycling event. Seq no 1 is the starting data point. Each event will have at least one associated data point by default. Each data point has a segment associated with it.
You can connect the segments associated with data points for a specific event based on sequence number (seqno) where the endx,endy of one associated segment should be the startx, starty of the next associated segment. This would show the full route someone took in this event.