Database Systems — CSci 4380
Midterm Exam #1
February 22, 2016

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.

Elections(eid, year, type, state, party)
Candidates(cname, eid, party)
Issues(issuename, type, description)
CandidatePositions(cname, eid, issuename, position, importance)
Voters(voterid, lname, fname, gender, age, street, state, city, zip)
Donations(id, voterid, amount, currency, date, cname, eid)

(a) Return id, first and last name of voters who have donated at least `100` of currency `'bitcoins'` to a candidate running in an election taking place in year `2040` and the candidate is `'pro'` for the `'cloning'` issue in this election.

Solutions:

$$
\begin{aligned}
A &= \sigma_{\text{currency='bitcoins' and amount>=100}}\, Donations \\
B &= \sigma_{\text{issuename='cloning' and position='pro'}}\, CandidatePositions \\
C &= \sigma_{\text{year=2040}}\, Elections \\
D &= \Pi_{\text{voterid,lname,fname}}\, (A \bowtie B \bowtie C \bowtie Voters)
\end{aligned}
$$

(b) Find political issues that no candidate has ever attached an importance of `5` or higher according to CandidatePositions (regardless of whether `'pro'` or `'con'`). Return the name and type of the issues.

Solutions:

$$
\begin{aligned}
A &= \Pi_{\text{issuename}}\, (\sigma_{\text{importance }>=5}\, CandidatePositions) \\
B &= (\Pi_{\text{issuename}}\, Issues) - A \\
C &= \Pi_{\text{issuename,type}}\, (B \bowtie Issues)
\end{aligned}
$$

(c) Find candidates who held different positions (`'pro'` vs. `'con'`) on the same issue in two different elections. The importance of the issue should be at least `8` or higher in both elections. Return the candidate name and the issue name.

Solutions:

$$
\begin{aligned}
A(c1, e1, i1, p1, imp1) &= \sigma_{\text{importance >= 8}}\, CandidatePositions \\
B(c2, e2, i2, p2, imp2) &= A \\
C &= A \bowtie_{\text{c1=c2 and i1=i2 and p1 <> p2}}\, B \\
D &= \Pi_{c1,i1}\, C
\end{aligned}
$$

Question 2 (12+10+6=28 points). Suppose you are given the following additional 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:

SocialMediaAccounts(voterid, smedia, uname, uid, numfr, numfol, nummsgs)

uname → uid
uid → uname
smedia, uname, uid → voterid
smedia, uname → numfr, numfol, nummsgs

(i) According to this model, can a voter have multiple usernames in the same social media? Explain why or why not.

(ii) What are the key(s)?

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

(iv) Is this set of functional dependencies minimal or can it be simplified? Explain with a short sentence.

Solutions:

(i) Yes, the above functional dependencies do not imply that: voterid → username
Hence, it is possible to have multiple usernames with for the same voterid.

(ii) Keys: smedia, uname or smedia, uid

(iii) It is not in BCNF because the first two functional dependencies are not trivial and do not have a superkey on the left.
It is in 3NF because the first two functional dependencies have a prime attribute on the right, and the other two have a superkey on the left.

(iv) No, this is not trivial. We can simplify the third dependency as:
smedia, uname → voterid

or
smedia, uid → voterid

because (smedia, uid) closure remains the same after we do (all the attributes).

(b) You are given the following new relation for political messages:

PoliticalMessages(cname, eid, state, city, zip, issuename, position, msgtype, msgdate, msgtext)

cname, eid, state, city → msgtype
cname, eid, issuename, position → msgtype
msgtype, msgdate → msgtext
zip → state, city

Use 3NF decomposition to convert this relation to 3NF. For each decomposed relation, mark clearly the relevant functional dependencies. Also discuss whether each decomposed relation is in BCNF or not.

Solutions:

Key: cname, eid, zip, issuename, position, msgdate

R1(<u>cname, eid, state, city</u>, msgtype) cname, eid, state, city → msgtype

R2(<u>cname, eid, issuename, position</u>, msgtype) cname, eid, issuename, position → msgtype

R3(<u>msgtype, msgdate</u>, msgtext) msgtype, msgdate → msgtext

R4(<u>zip</u>, state, city) zip → state, city

R5(<u>cname, eid, zip, issuename, position, msgdate</u>)

All relations are in BCNF as well.

(c) You are given the following new relation for candidate travel:

Travel(cname, eid, state, city, date, time, speechopening, speechtext)

Assume that a candidate in a specific election can visit the same city in a given state many times, but only once on a given date. The time is fixed for each visit. The speechopening is same for each state and city for a given candidate in a given election. The speechtext may change from visit to visit.

Write down these requirements as functional dependencies, making sure that they are minimal.

Solutions:

cname, eid, state, city, date $\rightarrow$ time, speechtext

cname, eid, state, city $\rightarrow$ speechopening

Question 3 (10 points). Given the relation: R(A,B,C,D,E,F,G) $F = \{AB \rightarrow CE, DE \rightarrow F, C \rightarrow D\}$

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

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

Solutions:

A	B	C	D	E	F	G
a	b	c	d1	e1	f1	g1
a2	b2	c	d2	e	f2	g
a	b	c3	d	e3	f	g3

Given $AB \rightarrow CE$, we change c3 to c, e3 to e1

Given $C \rightarrow D$, we change d1 and d2 to d

Given $DE \rightarrow F$, we change f1 to f

The resulting table (see below) have no rows without a subscript, so this is lossy decomposition.

A	B	C	D	E	F	G
a	b	c	d	e1	f	g1
a2	b2	c	d	e	f2	g
a	b	c	d	e1	f	g3

Question 4 (22 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 regarding different exhibits of paintings, sculptures and other installations displayed at a museum. Each exhibit has a name, start and end date, featuring multiple art objects. The name for a given start day is unique. Each art object has an id, title, type, description, year and a location in the exhibit given by the name of the room, and x, y coordinates within the room. A single piece of art can be in multiple exhibits, but their location for a given exhibit is fixed. Each object of art can be by a single artist or multiple artists. For each artist, database stores name (unique), date of birth, and country. The museum has a number of curators, each has an id, name, biography, specialty, phone number and email. Each exhibit is curated by exactly one curator and a curator must always exist. Curators can curate multiple exhibits. The museum also shows movies associated with each exhibit. Each movie has a title and duration. Movies are made by a single artist or a single curator in the database. A movie can be associated with one or more exhibits and exhibits can have multiple movies.

Solutions:

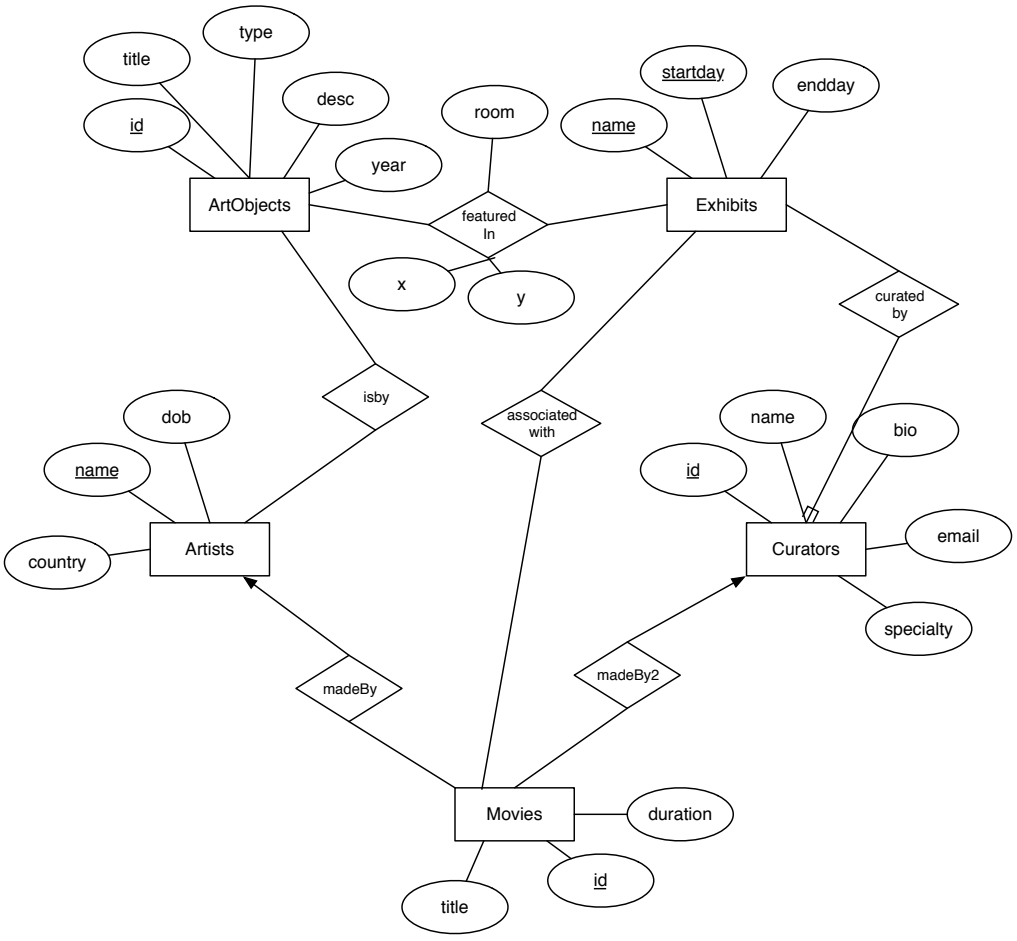
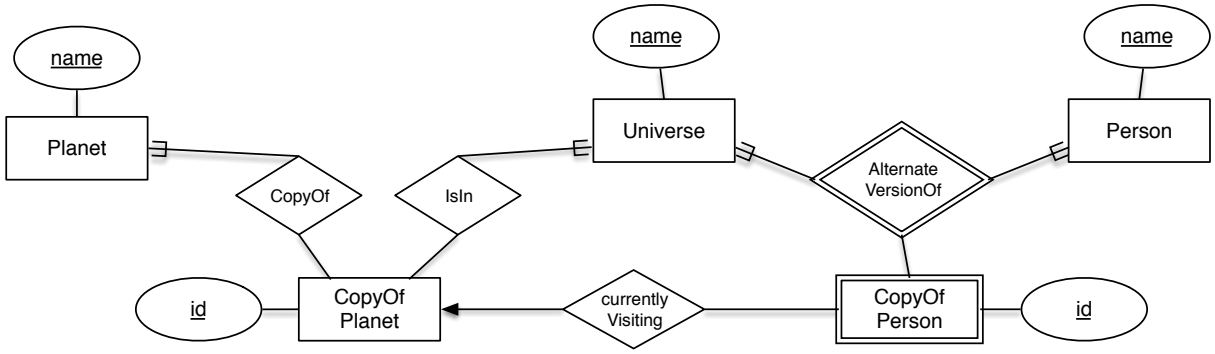


Figure 1: ER Diagram for Question 4

Question 5 (10 points). Given the ER diagram below.



(a) For each questions below, circle YES or NO according to the above model:

NO Should each CopyOfPerson be currently visiting a CopyOfPlanet?

YES Can two copies of the same person (CopyOfPerson for the same person) visit the same CopyOfPlanet?

YES Is there always a Planet for each CopyOfPlanet?

YES Is there always a Universe entity for each CopyOfPerson?

YES Is it possible to store more than one CopyOfPlanet entity of the same planet and the same Universe (i.e. two copies of Planet XYZ from Universe Z)?

YES Is it possible to store more than one CopyOfPerson entity of the same person and the same universe (i.e. two copies of Morty from Universe Z)?

(b) Complete the following relational data model to capture the above ER diagram:

Planet(<u>name</u>)

Universe(<u>name</u>)

Person(<u>name</u>)

CopyOfPlanet(<u>id</u>, planetname, universename)

CopyOfPerson(<u>id, universename, personname</u>, visitingCopyOfPlanet)

Solutions:

Data model to be used in Exam #1

Note: The keys of each relation are underlined.

Elections(<u>eid</u>, year, type, state, party)

Stores main information about elections. Type is one of: `'local'`,`'general'`, or `'local-party'`.

If election is `'general'`, state and party are both empty (null value). For `'local'` and `'local-party'` elections state must be given.

For `'local-party'` elections, party must also be given. These are elections in which various candidates from the same party compete. In local or general elections, candidates from different parties compete.

Candidates(<u>cname, eid</u>, party)

Stores the names of the candidates, the id of the election they are running in (from Elections relation) and the party they are running for in this election. Obviously, the data model allows for candidates to run for different parties in different elections.

Issues(<u>issuename</u>, type, description)

Stores political issues. Each issue has a name, e.g. `'time travel'`, `'cloning'`, `'thought control'`, `'transdimensional portal control'`, and a type e.g. `'health'`, `'portals'` and a longer description.

CandidatePositions(<u>cname, eid, issuename</u>, position, importance)

Stores the position a candidate takes for a specific election. Cname is the name of a candidate from Candidates relation, eid is the id of an election, and issuename is the name of an issue from PoliticalIssues. Position is one of `'pro'` or `'con'`.

Importance is a value between 1 and 10, 1 is the least important issue for the candidate and 10 is the most important. An issue may not even show up in this relation, in which case its importance is assumed to be zero.

Voters(<u>voterid</u>, lname, fname, gender, age, street, state, city, zip)

Stores information for registered voters. Each voter is given a single voter id.

Donations(<u>id</u>, voterid, amount, currency, date, cname, eid)

Stores the donations made by a specific voter given by their voter id, for a specific candidate in a specific election. The currency can be `'dollars'`, `'bitcoins'`, `'flurbo'`, etc.