Midterm Exam (Version A)
CS 122A
Spring 2017

Max. Points: 100
(Please read the instructions carefully)

Instructions:
- The total time for the exam is 80 minutes; be sure to *budget your time* accordingly.
- The exam is closed book and closed notes but “open cheat sheet”.
- Read each question first, in its entirety, and then carefully answer each part of the question.
- If you don’t understand something, ask one of the exam proctors for clarification.
- If you still find ambiguities in a question, note the interpretation you are taking.

<table>
<thead>
<tr>
<th>QUESTION</th>
<th>TOPIC</th>
<th>POINTS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E-R Design</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Modeling Terms</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Relational Operators</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>FD Design Theory</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Relational Queries</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>E-R to Relational Translation</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>TOTAL</td>
<td>All</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
Question 1: E-R Design (25 points)

(25 pts) You have been hired by a new startup, VideoTweets, to help them with the data management aspects of a new online service. They plan to serve home-made videos to mobile users from a cluster of Linux servers. Your job is to design an E-R conceptual schema for their database. Draw your E-R design below, starting from the partial picture that we’ve provided. Be sure to capture all relevant entities, relationships, and attributes. Clearly label each relationship to indicate its cardinality constraints (e.g., 1:1, 1:N, N:M), participation constraints (using double lines where appropriate), and roles (if needed). Mark the primary key for each entity set (by underlining it). Here’s what you have been told by their lead application developer:

- Each user has a unique user id, a name, a unique e-mail address, and a birthdate.
- Users can become video sharing friends by “liking” one another. (Note: “Likes” may not be symmetric, i.e., one user might “like” another user but the opposite might not hold.)
- Users can choose to post videos, and their postdate should be remembered. Each video will be posted by one user, and every video must indeed have an associated user who posted it.
- Videos have a unique id, a caption, the date that it was recorded on, a thumbnail preview, a length, and a file name. Videos also have a set of tags (e.g., “dog”, “beach”, “music”) intended to help users when searching for videos of potential interest.
- To avoid use of the service by social outcasts, every user must be liked by at least one other user.
- Some videos are ads. Ad videos also have a sponsoring company name and a product name.

 SCORE: ____________
Question 2: Modeling Terms (5 points)

(5 pts) Refer again to the description of the VideoTweets data in Question 1. Match each of the modeling constructs in the left column below with their best-matching feature (from Question 1) in the right column below. Indicate each answer by writing the letter of the relevant description feature to the left of each modeling construct. (Hint: You should end up using each description feature once.)

<table>
<thead>
<tr>
<th>Modeling construct</th>
<th>Description feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>______ b ______</td>
<td>Atomic attribute</td>
</tr>
<tr>
<td>______ d ______</td>
<td>Multivalued attribute</td>
</tr>
<tr>
<td>______ a or e ____</td>
<td>Primary key</td>
</tr>
<tr>
<td>______ e or a ____</td>
<td>Candidate key</td>
</tr>
<tr>
<td>______ c ______</td>
<td>Super key</td>
</tr>
<tr>
<td>a. User.id</td>
<td></td>
</tr>
<tr>
<td>b. Post.postdate</td>
<td></td>
</tr>
<tr>
<td>c. (User.email, User.name)</td>
<td></td>
</tr>
<tr>
<td>d. Video.tags</td>
<td></td>
</tr>
<tr>
<td>e. User.email</td>
<td></td>
</tr>
</tbody>
</table>

Question 3: Relational Operators (10 points)

(1 pt each) Given two relations R and S, where R has \( N_R \) tuples, S has \( N_S \) tuples, and \( N_R > N_S > 0 \), what are the minimum and maximum possible result cardinalities for the following relational algebra queries expressed in terms of \( N_R \) and \( N_S \)?

\[
\begin{align*}
R \cup S: & \quad \text{min size: } N_R \quad \text{max size: } N_R + N_S \\
S - R: & \quad \text{min size: } 0 \quad \text{max size: } N_S \\
\sigma_{\text{attr}=\text{val}}(R): & \quad \text{min size: } 0 \quad \text{max size: } N_R \\
\Pi_{\text{attr1, attr2}}(R): & \quad \text{min size: } 1 \quad \text{max size: } N_R \\
R \bowtie S: & \quad \text{min size: } 0 \quad \text{max size: } N_S \times N_R \\
\end{align*}
\]

SCORE: ____________
Question 4: FD Design Theory (20 points)

Consider a relational table with schema T (u, v, w, x, y). Each of the sub-problems below is based on a different list of FDs for T. For each sub-problem, list the candidate keys for T based on its given FD list and indicate which is the highest normal form – 1NF, 2NF, 3NF, or BCNF – that T is currently in based on the given FD list. If asked (as in one case below), normalize the design into BCNF (else stop at 3NF) by decomposing T into several relations that have the lossless join and dependency-preserving properties. (Reminder: Be sure to use parentheses wherever you are indicating a composite key.) Last and also least, answer the 1-point question at the very end as well.

(5 pts)  (u, v) → y,  (u, v) → w,  v → x,  x → v,  (u, x) → w
- Candidate key(s) for T:
  (u,v), (u,x)
- Highest normal form:
  3NF

(5 pts)  (u, v) → w,  x → y
- Candidate key(s) for T:
  (u,v,x)
- Highest normal form:
  1NF

(7 pts)  (w, x) → y,  u → v,  v → x,  v → w, candidate key is u.
- Highest normal form:
  2NF
- Normalized BCNF or 3NF design:
  (u,v) ,  (v,w,x) ,  (w,x,y)

(2 pts)  Suppose that T contains a tuple (10, "Chen", 2017-05-09, "Database Systems", "CS"). If x → y is a functional dependency on T, give an example of another tuple that, if inserted into T, would violate this constraint.


(1 pt)  Give the largest possible superkey for T:
  (u,v,w,x,y)

SCORE: _______________
Question 5: E-R to Relational Translation (20 points)

(20 pts) Translate the following E-R schema into an appropriate set of SQL tables. Avoid having more tables than needed, and be sure that your translated design – expressed as a series of CREATE TABLE statements in SQL – includes any/all appropriate (i) primary keys, (ii) unique keys, (iii) NOT NULL constraints, (iv) FOREIGN KEY constraints, and (v) ON DELETE options.

```sql
CREATE TABLE Building(
    bldgid INTEGER, manager VARCHAR(35),
    address VARCHAR(60),
    PRIMARY KEY (bldgid)
);
CREATE TABLE Apartment(
    unit INTEGER, bedrooms INTEGER,
    baths INTEGER, bldgid INTEGER,
    floor INTEGER,
    PRIMARY KEY (bldgid, unit),
    FOREIGN KEY (bldgid) REFERENCES Building(bldgid) ON DELETE CASCADE
);
CREATE TABLE Tenant(
    tid INTEGER, name VARCHAR(35),
    phone VARCHAR(20),
    PRIMARY KEY tid
);
CREATE TABLE Rent(
    tid INTEGER, bldgid INTEGER,
    unit INTEGER, price DECIMAL(6,2),
    PRIMARY KEY (tid, bldgid, unit),
    FOREIGN KEY (tid) REFERENCES Tenant (tid) ON DELETE CASCADE,
    FOREIGN KEY (bldgid, unit) REFERENCES Apartment (bldgid, unit) ON DELETE CASCADE
);
```
Question 6: Relational Queries (20 points)

Write each of the following queries in the indicated relational query language against the Hoofers Sailing Club schema that we’ve been using in the lectures. Be sure to read each problem carefully!

Sailors(sid, sname, rating, age) Reserves(sid, bid, day) Boats(bid, bname, color)

(5 pts) Using the relational algebra, print the names and colors of boats that have been reserved by a sailor named Smith or by a sailor named Smythe:

\[ \pi \text{bname, color}((\text{Boats} \bowtie \text{Reserves}) \bowtie (\sigma \text{sname} = 'Smith' \lor \text{sname} = 'Smythe' (\text{Sailors}))) \]

(5 pts) Using the relational calculus, print the names and colors of boats that have been reserved both by a sailor named Lee and by a sailor named Li:

\[ \{t(\text{name}, \text{color}) \mid \exists \ b \in \text{Boats}(b.\text{bname}=t.\text{name} \land b.\text{color}=t.\text{color} \land \\
\exists \ r1 \in \text{Reserves}(r1.\text{bid}=b.\text{bid} \land \exists \ s1 \in \text{Sailors}(s1.\text{sid}=r1.\text{sid} \land s.\text{sname}='Lee')) \land \\
\exists \ r2 \in \text{Reserves}(r2.\text{bid}=b.\text{bid} \land \exists \ s2 \in \text{Sailors}(s2.\text{sid}=r1.\text{sid} \land s.\text{sname}='Li'))\} \]

(5 pts) Using the relational algebra, print the ids of boats that have been reserved by one or more sailors named Smith but no sailors named Jones.

\[ \pi \text{bid} (\text{Reserves} \bowtie (\sigma \text{sname} = 'Smith'(\text{Sailors}))) - \pi \text{bid} (\text{Reserves} \bowtie (\sigma \text{sname} = 'Jones'(\text{Sailors}))) \]

(5 pts) Using English, state the query to which the following would be the relational calculus answer:

\[ \{b | b \in \text{Boats} \land (\forall s \in \text{Sailors} ((s.\text{rating} = 10) \Rightarrow (\exists r \in \text{Reserves} ((r.\text{sid} = s.\text{sid}) \land (r.\text{bid} = b.\text{bid})))) \} \]

Find the boats that have been reserved by all of the sailors whose rating is 10.