Midterm Exam #1 Version B
CS 122A
Winter 2017

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

Instructions:
- The total time for the exam is 50 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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E-R Modeling</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Modeling Terms</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Relational Design Theory</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>E-R to Relational Translation</td>
<td>35</td>
</tr>
<tr>
<td>TOTAL</td>
<td>All</td>
<td>100</td>
</tr>
</tbody>
</table>

Question 1: E-R Modeling (30 points)
(30 pts) Consider the E-R model below and then indicate whether each of the statements that follow it are true (T) or false (F) by writing the appropriate letter in the space to the left of the statement.

- T  A pro may also be a lesson-giving teacher.
- F  A teacher may also be a lesson-taking student.
- F  A teacher’s fee may be different for each student.
- F  A pro can have contracts with multiple bars.
- F  Some musicians may be neither a pro nor a student nor a teacher.
- T  A student must play an instrument.
- T  An instrument can have multiple names.
- T  An instrument can have multiple colors.
- F  An instrument can have multiple teachers who teach it.
- T  An instrument can have multiple musicians who play it.
- F  A bar must have a contract with some pro.
- F  A teacher must teach at least one instrument.
- T  A musician may have a different skill level for each instrument played.
- T  Two musicians may have the same first and last names.
- F  A student can only take lessons from one teacher.

SCORE: _________
**Question 2: Modeling Terms (10 points)**

(10 pts) Consider the following description of the data involved in a new cloud-based photo management application you’ve been asked to develop a database design for:

- Users have a unique ssn (social security number), a name (consisting of a first name, last name, and middle initial), a unique email address, an age, and a password.
- Photos have a unique photo id, a date, a caption, and a set of search tags (e.g., “birthday”, “night”, “cat”, “music”).
- Every photo must have been taken by some user (who may have taken other photos as well).
- Credit cards have a cardholder name, a card type (e.g., “Visa”), a unique card number, an expiration date, and an expired flag that is true if the expiration date is in the past and false otherwise.
- Credit cards are owned by users and in some cases a card may be owned by several users.
- Users may like photos and can specify a rating (on a scale from -5 to +5) to show just how well they like them.

Match each of the modeling constructs in the left column below with their best-matching feature (drawn from the description above) in the right column below. Indicate your answer by writing the relevant description feature in the blank to the left of each modeling construct. *(Hint: You should end up using each feature once, and you might find it quicker to go down the list of features looking for constructs rather than going down the list of constructs looking for features.)*

<table>
<thead>
<tr>
<th>Modeling construct</th>
<th>Description feature</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong><strong>search_tags</strong></strong></td>
<td>Multivalued attribute</td>
</tr>
<tr>
<td><strong><strong>(photo_id, date)</strong></strong></td>
<td>Super key</td>
</tr>
<tr>
<td><strong><strong>expired_flag</strong></strong></td>
<td>Derived attribute</td>
</tr>
<tr>
<td><strong><strong>User</strong></strong></td>
<td>Entity</td>
</tr>
<tr>
<td><strong><strong>caption</strong></strong></td>
<td>Atomic attribute</td>
</tr>
<tr>
<td><strong><strong>user_name</strong></strong></td>
<td>Composite attribute</td>
</tr>
<tr>
<td><strong><strong>card_number</strong></strong></td>
<td>Primary key</td>
</tr>
<tr>
<td><strong><strong>email_address</strong></strong></td>
<td>Candidate key</td>
</tr>
<tr>
<td><strong><strong>rating</strong></strong></td>
<td>Relationship attribute</td>
</tr>
<tr>
<td><strong><strong>like</strong></strong></td>
<td>Relationship</td>
</tr>
</tbody>
</table>

**SCORE:** _________
Question 3: Relational Design Theory (25 points)

Consider a relational table with schema $S (a, b, c, d, e)$. Each of the sub-problems below gives a different list of FDs for $S$. For each sub-problem, list the candidate keys for $S$ based on the given FD list and then indicate which is the highest normal form – 1NF, 2NF, 3NF, or BCNF – that $S$ is in with the given FD list. If asked (in one case below), normalize the design by decomposing $S$ into several BCNF relations that have the lossless join and dependency-preserving properties. (Reminder: Be sure to include parentheses wherever you are indicating a composite key.)

(5 pts) $(a, b) \rightarrow c, (a, b) \rightarrow d, (a, b) \rightarrow e$

- Candidate key(s) for $S$: $(a, b)$
- Highest normal form: BCNF

(5 pts) $(a, b) \rightarrow e, (a, b) \rightarrow c, (a, d) \rightarrow c, b \rightarrow d, d \rightarrow b$

- Candidate key(s) for $S$: $(a, b), (a, d)$
- Highest normal form: 3NF ($b \rightarrow d$, and $b$ is not a candidate key)

(5 pts) $a \rightarrow b, b \rightarrow c, c \rightarrow d, d \rightarrow e$

- Candidate key(s) for $S$: $a$
- Highest normal form: 2NF (transitive dependency from $b \rightarrow c$ …)

(10 pts) $(a, b) \rightarrow c, d \rightarrow e$

- Candidate key(s) for $S$: $(a, b, d)$
- Highest normal form: 1NF
- Normalized BCNF design ($S_1, S_2, ...$):
  $S_1(a, b, c) 1$
  $S_2(d, e) 1$
  $S_3(a, b, d) 2$

SCORE: _________
Question 4: E-R to Relational Translation (35 points)

(35 pts) Using the “delta tables” approach to ISA relationship handling, 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 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.

```
CREATE TABLE Center(
  cid    INT,
  cname  VARCHAR(20),
  address VARCHAR(20),
  PRIMARY KEY (cid)
);

CREATE TABLE Business(
  bid    INT,
  bname  VARCHAR(20),
  phone  VARCHAR(10),
  cid    INT NOT NULL,
  PRIMARY KEY (bid),
  FOREIGN KEY (cid) REFERENCES Center(cid)
);

CREATE TABLE Eatery(
  bid    INT,
  cuisine VARCHAR(20),
  permitnum VARCHAR(10),
  PRIMARY KEY (bid),
  FOREIGN KEY (bid) REFERENCES Business(bid) ON DELETE CASCADE
);
```

SCORE: _________