Midterm Exam #1 (Version B)
CS 122A
Spring 2018

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>Relational Design Theory</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Modeling Terms</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>E-R Modeling</td>
<td>30</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: Relational Design Theory (25 points)

Answer each of the following questions about relational DB design and functional dependencies:

(5 pts) Consider a relation \( R (v, w, x, y, z) \) with FDs: \( w \rightarrow v, w \rightarrow x, vx \rightarrow y, x \rightarrow z \).

a. Compute the attribute closure \((x^+)\) of the attribute \( x \): \( x, Z \)

b. Circle the highest normal form that \( R \) satisfies: \( 1NF \quad 2NF \quad 3NF \quad BCNF \)

(10 pts) Assuming that a (possibly different) table \( R (v, w, x, y, z) \) currently contains the following data values, answer the following questions about \( R \)'s set of functional dependencies.

<table>
<thead>
<tr>
<th>v</th>
<th>w</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1934</td>
<td>Daisy</td>
<td>Duck</td>
<td>Duckburg</td>
<td>F</td>
</tr>
<tr>
<td>1926</td>
<td>Winnie</td>
<td>DaPooh</td>
<td>AcreWood</td>
<td>M</td>
</tr>
<tr>
<td>1937</td>
<td>Snow</td>
<td>White</td>
<td>Forest</td>
<td>F</td>
</tr>
<tr>
<td>1928</td>
<td>Mickey</td>
<td>Mouse</td>
<td>Burbank</td>
<td>M</td>
</tr>
<tr>
<td>1934</td>
<td>Donald</td>
<td>Duck</td>
<td>Duckburg</td>
<td>M</td>
</tr>
<tr>
<td>1973</td>
<td>Little</td>
<td>John</td>
<td>Sherwood</td>
<td>M</td>
</tr>
</tbody>
</table>

a. Does the functional dependency \( v \rightarrow w \) currently hold in \( R \)? \( \text{YES} \quad \text{NO} \)

b. Does the functional dependency \( (v,w) \rightarrow x \) currently hold in \( R \)? \( \text{YES} \quad \text{NO} \)

c. Does the functional dependency \( x \rightarrow y \) currently hold in \( R \)? \( \text{YES} \quad \text{NO} \)

d. Does the functional dependency \( (v,x) \rightarrow z \) currently hold in \( R \)? \( \text{YES} \quad \text{NO} \)

e. Does the functional dependency \( x \rightarrow w \) currently hold in \( R \)? \( \text{YES} \quad \text{NO} \)

(10 pts) Consider a table \( R (t, u, v, w, x, y, z) \) with FDs: \( t \rightarrow v, \quad t \rightarrow u, \quad y \rightarrow x, \quad w \rightarrow z \).

a. List the candidate key(s) for \( R \): \( twy \)

b. Circle the highest normal form that \( R \) satisfies: \( 1NF \quad 2NF \quad 3NF \quad BCNF \)

c. Normalize \( R \) into a lossless-join BCNF design (\( R1, R2, \ldots \)):

\begin{align*}
R1 (t, u, v) & \quad \text{Alternative:} \quad R1 (t, u), R2(t, v), \\
R2 (y, x) & \quad R3 (y, x), \\
R3 (w, z) & \quad R4 (w, z), \\
R4 (t, w, y) & \quad R5 (t, w, y)
\end{align*}
d. Is your BCNF design dependency-preserving? YES NO

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

(10 pts) You’ve just been hired as a consultant for Orange County’s best TV show production company, ESPM! They are filming a new TV series called Late Nights with Databases, and they need your help to set up a database for their business. ESPM also has other shows and is hoping to use your new database for all of their show management. Here’s what their IT manager has to say:

- Each staff member has a unique staff id, a name, and a home address that includes a street address, city, state, and zipcode.
- Each staff member can be either a cast member or a producer. Cast members star in shows, while producers work on show productions. Each producer has a specialty. Each cast member has an hourly_rate and a monthly_salary calculated from his/her hourly_rate.
- Each show has a title and a first_air_date (together as a combination that uniquely identify it) and a show_description that includes both a show_time and a genre.
- When ESPM decides to keep the Late Nights with Databases show on the air, a new season is scheduled. A season is associated with a particular show. Each season of a show has a cost, a season number, and a duration of the season. There are multiple seasons for a successful show (e.g., this might be the 7th season of a show).
- Each season of a show includes one or more cast members, each of whom contributes a certain number of hours to the season in which his/her dialogue is present according to the story line.
- The ESPM production company profits from advertising. They have multiple clients around the globe. Each client has a unique client id, a business name, a business type, and a business address.
- When a client wants to advertise their business on a TV show, a contract is created among a producer, the client, and a particular show. Each contract states a duration, cost_per_minute, and number of occurrences.

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. *(NOTE: You should use each feature ONLY 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>Alternatives</th>
<th>Description feature</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>business_address</em></td>
<td>Atomic attribute</td>
<td>cost_per_minute, staff_name</td>
</tr>
<tr>
<td>producer</td>
<td>Entity</td>
<td>cost_per_minute</td>
</tr>
<tr>
<td>staff_name</td>
<td>Inherited attribute</td>
<td>season_number</td>
</tr>
<tr>
<td>title, first_air_date</td>
<td>Candidate key</td>
<td>show_description</td>
</tr>
<tr>
<td>contract</td>
<td>Ternary relationship</td>
<td>business_address</td>
</tr>
<tr>
<td>client_id, business_name</td>
<td>Super key</td>
<td>title, first_air_date</td>
</tr>
<tr>
<td>monthly_salary</td>
<td>Derived attribute</td>
<td>monthly_salary</td>
</tr>
<tr>
<td>cost_per_minute</td>
<td>Relationship attribute</td>
<td>title, first_air_date</td>
</tr>
<tr>
<td>show_description</td>
<td>Composite attribute</td>
<td>producer</td>
</tr>
</tbody>
</table>
Question 3: E-R Modeling (30 points)

(30 pts) Based only on the E-R model shown at the bottom of this page, indicate whether each of the following statements are true (T) or false (F) by writing the appropriate letter in the space to the left of the statement.

__F__ Each coach has to train at least one coach.
__F__ An athlete that does not have at least one interest cannot be considered as a player.
__T__ Not every athlete needs to be coach or a player.
__F__ The name (tname) of a team combined with league name (lname) will uniquely identify the team in the database overall.
__F__ It is possible to have two teams with the same name in a given club only if they are for different sports.
__F__ Each player has to be trained by at least one coach.
__F__ A coach could have been trained by multiple coaches.
__F__ In a given league, a coach can participate with two teams that have the same name (tname) only in different years.
__T__ A coach can be trained by a coach who has the same rating as their rating.
__F__ The sport of a team can be derived from its name (tname) and its club name.
__F__ A coach can be his own trainer only if he is also a player.
__T__ It is possible for an athlete to have neither an interest list nor a rating.
__F__ A team has to use the same coach in order to participate in multiple leagues.
__F__ Each club has to support at least one team.
__F__ Only those teams that participate in a league need a club as their sponsor.
Question 4: E-R to Relational Translation (35 points)

(35 pts) Translate the following E-R schema into an appropriate set of SQL tables. As usual, avoid using more tables than necessary, 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. We’ve started the answer for you below, providing the full answer for Painter and a part of the answer for Painting. Finish filling in the details of Painting, adding any missing attributes and/or constraints, and then finish the the job by adding any additional CREATE TABLE statements. Note: Try to fit your answer on this page. Only if space on this page is not enough you can use the blank page at the end of this exam.

CREATE TABLE Painter (  
    name VARCHAR(20),  
    birthdate DATE,  
    nationality VARCHAR(30),  
    PRIMARY KEY(name)  
);
CREATE TABLE Painting(
    pname VARCHAR(20),
    subject VARCHAR(50),
    width DECIMAL(5,2),
    length DECIMAL(5,2),
    painter_name VARCHAR(20),
    buyer_name VARCHAR(20),
    price DECIMAL(8,2),
    Primary Key(pname,painter_name),
    Foreign Key(painter_name) References Painter(name) ON DELETE CASCADE,
    Foreign Key(buyyer_name) References Painter(name)
);  

Create Table Material(
    pname VARCHAR(20),
    painter_name VARCHAR(20),
    material_type VARCHAR(30),
    Primary Key(pname,painter_name,material_type),
    Foreign Key (pname,painter_name) References Painting(pname,painter_name)
);  

Create Table Inspire(
    inspirer VARCHAR(20),
    inspiree VARCHAR(20),
    Primary Key(inspirer,inspiree),
    Foreign Key(inspirer) References Painter(name),
    Foreign Key(inspiree) References Painter(name)
);