Homework 1

IMPORTANT

- **Plagiarism**: Homework may be discussed with other students, but all homework is to be completed individually.
- **Typeset** all of your answers whenever possible. Illegible handwriting may get no points, at the discretion of the graders.
- **Deposit hard copy** of your answers in class at 1:30pm on Tuesday, February 4th. For ease of grading, please
  - solve each of the 4 questions on a separate page. If you need more pages for one question, please staple them together.
  - Type the full info on each page: your name, Andrew ID, course #, Homework #, Question # on each of the 4 pages.
- **Late homeworks**: If you are turning your homework in late, please email it
  - to all TAs
  - with the subject line exactly 15-415 Homework Submission (HW 1)
  - and the count of slip-days you are using.

For your information:

- Graded out of 100 points; 4 questions total
- Rough time estimate: ≈6 hours (1-2 hours for each question)

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity-Relationship Diagram</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>SQL Tables from the ER Model</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Relational Algebra for Video Recommendation</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Relational Calculus</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Question 1: Entity-Relationship Diagram ................ [20 points]
Submit on separate page

Consider a database to store information about organizations on campus. The database has the following properties:

- Each Organization has a title, and an unique organization ID (oID).
- Students belong to zero or more organizations; we record start date for when they join the organization. For each student we also record the name, and a unique student ID (sID).
- Each student is either a part time or full time student. For part time students we record their minimum number of hours they must work per week. For full time students we record the number of credits they are taking.
- Each Faculty member has a name and a unique faculty ID (fID).
- Each organization must have at least one student member, and it must be advised by exactly one faculty member.
- Each organization must have one President who is a part time student (so he has enough free time to manage the organization) and each student cannot be the president of more than one organization. (The president does not need to be specified as a member of the organization.)

Given this description of the database and its constraints, we have created a mostly correct Entity-Relationship Diagram, shown in Figure [I]

(a) [10 points] Find and correct any mistakes in the given ER diagram. In addition, number and list them, like, e.g.
1. delete: arrow, from x to y
2. change to bold line: thin line, from z to w
3. change to bold box: entity e

(b) [5 points] There may also be some missing element(s). If any, add them to the picture, and list them, numbered. E.g.
1. add: attribute a, to entity e
2. add: bold line, arrow, from c to d.
3. add: weak entity, f, with attributes . . .

(c) [5 points] List and number all the bold lines and all the arrows that are in the final, corrected version of the diagram. E.g.
1. bold, arrow, from Organization to advises
2. (not bold), arrow, from x to y

Clarifications/Hints:

- List your assumptions, if any. We will accept all reasonable assumptions.
- For your drawing convenience, a PDF of the ER diagram is at http://www.cs.cmu.edu/~abeutel/TA/dbms.S14/HW1/q1.pdf
Figure 1: Almost correct ER diagram

Question 1 continues...
Solution:

(a) 1. Make line from Organization to President, bold with an arrow from Organization to President
   2. Make arrow from President to Part Time into an arrow (not bold) from Part Time to President
   3. Underline attribute fID for Faculty

(b) 1. Add attribute Start_date to Belongs To

(c) 1. Bold with an arrow from Organization to President
   2. Arrow no bold from Part Time to President
   3. Bold with an arrow from Organization to advises
   4. Bold with no arrow from Organization to Belongs To

Grading info:

-2 points if the student forgot to underline fID

Question 1 continues...
• Due to misunderstanding, no penalties in (c) for effectively skipping over relations in giving arrows (Organization to Faculty)

• -0.5 point for extra changes to the ER diagram that are incorrect, but don’t take off points for that mistake more than once.

• -2 in (a) if forgot to make line from Organization to President bold (or another -2 if forgot to add arrow appropriately)

• -3 for (b) if student knows that the date is missing but adds it incorrectly (such as to Student rather than belongs to)
Question 2: SQL Tables from the ER Model......... [25 points]
Submit on separate page

Consider a database for the courses of a university. We give a rough description below:

- Each Department has a unique numeric departmentID (for example computer science is 15 at CMU) and title.
- Each Course has a unique numeric CourseID, a title, a course capacity, and may be offered by one or more departments. Eg., intro calculus may have CourseID=0001, and offered by the physics, as well as the math departments.
- Each Section has a numeric SectionID, is taught by one or more teachers, meets once or more per week (at given days at given times), and is of a certain type (lecture, lab, recitation, etc.).

For example, we could have a course in computer science department (15), course ID 415 (databases), with section A (of type lecture).

More precisely, we have the following ER diagram:

(a) [15 points] Following the specification from the ER diagram, give the SQL commands to create the tables necessary in this database. (Use your best judgement for attribute types. We’ll accept all reasonable choices.) Include in your commands appropriately PRIMARY KEY, FOREIGN KEY, CASCADE, etc.. No need to use CHECK commands. Additionally, list and number all the constraints that your DDL statements can not enforce, or type ‘zero’, if no constraint is left un-enforced.

Question 2 continues...
**Solution:**

```sql
CREATE TABLE Departments (
    dID INTEGER,
    title CHAR(50),
    PRIMARY KEY (dID)
)

CREATE TABLE Courses (
    cID INTEGER,
    title CHAR(50),
    credits INTEGER,
    PRIMARY KEY (cID)
)

CREATE TABLE Teachers (
    tID INTEGER,
    name CHAR(50),
    office CHAR(20),
    PRIMARY KEY (tID)
)

CREATE TABLE Sections (
    sID INTEGER,
    cID INTEGER,
    type CHAR(20),
    capacity INTEGER,
    PRIMARY KEY (sID, cID),
    FOREIGN KEY (cID) REFERENCES Courses ON DELETE CASCADE
)

CREATE TABLE MeetingTimes (
    sID INTEGER,
    cID INTEGER,
    day INTEGER,
    startTime INTEGER,
    endTime INTEGER,
    PRIMARY KEY (sID, cID, day, startTime, endTime),
    FOREIGN KEY (sID, cID) REFERENCES Sections ON DELETE CASCADE
)

CREATE TABLE SectionTeachers (
    sID INTEGER,
    cID INTEGER,
    tID INTEGER,
    PRIMARY KEY (sID, cID, tID),

Question 2 continues...
FOREIGN KEY (sID, cID) REFERENCES Sections
FOREIGN KEY (tID) REFERENCES Teachers
)

CREATE TABLE DepartmentOfferings (  
dID INTEGER,  
cID INTEGER,  
PRIMARY KEY (dID, cID),  
FOREIGN KEY (dID) REFERENCES Departments  
FOREIGN KEY (cID) REFERENCES Courses
)

SQL, without CHECK commands, also cannot enforce certain participation constraints: That each Section has at least one teacher, that each Course belongs to at least one Department, that each Course has at least one Section, and each Section has to have at least one meeting time.

Grading info:

- -0.5 point for every incorrect or missing attribute.
- -1 point for every incorrect table.
- -1 point if the answer does not say what constraints can’t be enforced (-1 per part).
- -1 points if primary key is missing.
Consider now the following new ER diagram with slightly nonsensical labels (on purpose - the goal is to have you apply the rules of ER-to-tables, without relying on intuition):

```
CREATE TABLE Do_Re_Mi (  
    Fa INTEGER,  
    So INTEGER,  
    La INTEGER,  
    Ti INTEGER,  
    Do INTEGER,  
    PRIMARY KEY (So)  
  )

CREATE TABLE ABC (  
    def INTEGER,  
    ghi INTEGER,  
    OneTwoThree INTEGER,  
    So INTEGER NOT NULL,  
    PRIMARY KEY (ghi),  
    FOREIGN KEY (So) REFERENCES Do_Re_Mi ON DELETE CASCADE  
  )

CREATE TABLE Encore (  
)
```

(b) [10 points] Again, following the specification from the ER diagram, give the SQL commands to create the tables necessary in this database, and to enforce as many integrity constraints as you can.

- Consider all attributes to be integers.
- No need to use CHECK commands.
- List and number all the constraints that your DDL statements can not enforce, or type 'zero', if no constraint is left un-enforced.

**Solution:**

```
CREATE TABLE Do_Re_Mi (  
    Fa INTEGER,  
    So INTEGER,  
    La INTEGER,  
    Ti INTEGER,  
    Do INTEGER,  
    PRIMARY KEY (So)  
  )

CREATE TABLE ABC (  
    def INTEGER,  
    ghi INTEGER,  
    OneTwoThree INTEGER,  
    So INTEGER NOT NULL,  
    PRIMARY KEY (ghi),  
    FOREIGN KEY (So) REFERENCES Do_Re_Mi ON DELETE CASCADE  
  )

CREATE TABLE Encore (  
)
```

Question 2 continues...
So_From INTEGER,
So_To INTEGER,
    PRIMARY KEY (So_From, So_To),
    FOREIGN KEY (So_From) REFERENCES Do_Re_Mi(So) ON DELETE CASCADE
    FOREIGN KEY (So_To) REFERENCES Do_Re_Mi(So) ON DELETE CASCADE
)

SQL, without CHECK commands, also cannot enforce the participation constraint for Do_Re_Mi in Encore.

Grading info:

- 0.5 point for every incorrect or missing attribute.
- 1 point for every incorrect table.
- 1 point if the answer does not say what constraints can’t be enforced (-1 per part).
- 1 points if primary key is missing.
Question 3: Relational Algebra for Video Recommendation [30 points]
Submit on separate page

Grading info: graded by Shen Wang (SW)

We discuss here a new database for recommending videos to users. We have the following tables:

- **Users**: Each user has a `userID` and name.
- **Videos**: Each video has a `videoID`, title, length (in seconds), and uploader (linked to a `UserID`).
- **Ratings**: Each rating is between a `userID` and `videoID` and is given a rating from 1 to 5 (as in 1 to 5 stars).

We give examples of these tables below.

<table>
<thead>
<tr>
<th>userID</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>Jack</td>
</tr>
<tr>
<td>U2</td>
<td>Jill</td>
</tr>
<tr>
<td>U3</td>
<td>John</td>
</tr>
<tr>
<td>U4</td>
<td>Jane</td>
</tr>
<tr>
<td>U5</td>
<td>Job</td>
</tr>
<tr>
<td>U6</td>
<td>Jay</td>
</tr>
</tbody>
</table>

Table 1: Users

<table>
<thead>
<tr>
<th>videoID</th>
<th>title</th>
<th>length</th>
<th>userID</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Michelle Obama dunks on LeBron</td>
<td>18</td>
<td>U3</td>
</tr>
<tr>
<td>V2</td>
<td>Michael Sherman yells at Erin Andrews</td>
<td>46</td>
<td>U3</td>
</tr>
<tr>
<td>V3</td>
<td>Jack Gleeson on being famous</td>
<td>1707</td>
<td>U2</td>
</tr>
<tr>
<td>V4</td>
<td>The Wolf of Wall Street</td>
<td>10780</td>
<td>U2</td>
</tr>
<tr>
<td>V5</td>
<td>Transcendence Official Trailer</td>
<td>152</td>
<td>U2</td>
</tr>
</tbody>
</table>

Table 2: Videos

Given this database answer the following questions:

(a) [3 points] Which of the following is the meaning of the command $\sigma_{\text{length}<20}(\text{Videos})$

1. Select all videos (`videoID`, `title`, `length`, and `userID`) with length under 20 seconds
2. Select the `length` of all videos that are less than 20 seconds
3. Select the `length` of each video and if the length is over 20 seconds round it down to 20 seconds
4. None of the above. The real answer is ..............

Question 3 continues...
### Solution: Answer #1

(b) [3 points] Which of the following commands selects the titles of all videos under 20 seconds long?

1. $\sigma_{\text{title}}(\pi_{\text{length}<20}(\text{Videos}))$
2. $\sigma_{\text{length}<20}(\pi_{\text{title}}(\text{Videos}))$
3. $\pi_{\text{title}}(\sigma_{\text{length}<20}(\text{Videos}))$
4. None of the above. The real answer is ..............

### Solution: Answer #3

(c) For the following command:

$$\text{Videos} \bowtie\bowtie \text{Ratings}$$

i. [1 point] How many, and which, are the columns (=attributes) of the answer?

**Solution:** There are 5 columns: videoID, userID, title, length, and rating.

ii. [2 points] How many tuples are in the results?

**Solution:** 3

iii. [3 points] Give, as a table, all of the tuples returned by the query.

**Solution:**

<table>
<thead>
<tr>
<th>videoID</th>
<th>userID</th>
<th>title</th>
<th>length</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>U3</td>
<td>Michelle Obama dunks on LeBron</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>V2</td>
<td>U3</td>
<td>Michael Sherman yells at Erin Andrews</td>
<td>46</td>
<td>5</td>
</tr>
<tr>
<td>V4</td>
<td>U2</td>
<td>The Wolf of Wall Street</td>
<td>10780</td>
<td>4</td>
</tr>
</tbody>
</table>

Question 3 continues...
Grading info:

- Since a typical problem 3 question first asks for the number of tuples and columns returned by the query and then asks for the entire query result, a wrong result will necessarily lead to a wrong number of results. Hence, to avoid double counting, the grading schema is specified as follows:
  - if the table is wrong and the number matches with the table, only the wrong table will lead to a loss of points.
  - if the table is correct but the number is wrong, full point of (ii) will be taken because of the confusion.
  - if the table is wrong and the number does not match with the table, then both of them will be considered wrong even if the number itself is correct.

- For problem 4, since $R.userId > R1.userId$ is ambiguous, we accept both tuples: $< R.userId : U1, R1.userId : U6 >$ and $< R.userId : U6, R1.userId : U1 >$

- For the values in the table, if the overall table seems correct but there exist some minor errors (most likely due to a copy-paste), in general no point will be taken. However, if the error could possibly lead to a confusion, such as wrong UserID, 0.5 point will be taken. If the overall table does not make much sense, then 1 tuple worths 1 point.

- Some students didn't list the columns but only write the number of columns for question (i)’s. This leads to a 0.5 point loss for the entire question 3 as long as the columns listed in the table are correct. However, if there is also no column in (iii) or the columns in (iii) is wrong, then for each problem that has missing/wrong columns, 0.5 point will be taken.

(d) For the following command:

$$\sigma_{rating > 4}(Videos \bowtie Ratings)$$

i. [1 point] How many, and which, are the columns of the answer?

**Solution:** There are 5 columns: videoID, userID, title, length, and rating.

ii. [2 points] How many tuples are in the results?

**Solution:** 2

iii. [3 points] Give, as a table, all of the tuples returned by the query.

**Solution:**

<table>
<thead>
<tr>
<th>videoID</th>
<th>userID</th>
<th>title</th>
<th>length</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>U3</td>
<td>Michelle Obama dunks on LeBron</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>V2</td>
<td>U3</td>
<td>Michael Sherman yells at Erin Andrews</td>
<td>46</td>
<td>5</td>
</tr>
</tbody>
</table>

Question 3 continues...
(e) For the following command
\[
\{\pi_{\text{userID}, \text{videoID}}(\text{Ratings})\} \div \{\pi_{\text{videoID}}(\sigma_{\text{userID} = 'U1'}(\text{Ratings}))\}
\]

i. [1 point] How many, and which, are the columns of the answer?
Solution: There is 1 column, userID.

ii. [2 points] How many tuples are in the results?
Solution: 5

iii. [3 points] Give, as a table, all of the tuples returned by the query.
Solution:

<table>
<thead>
<tr>
<th>userID</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
</tr>
<tr>
<td>U2</td>
</tr>
<tr>
<td>U3</td>
</tr>
<tr>
<td>U5</td>
</tr>
<tr>
<td>U6</td>
</tr>
</tbody>
</table>

(f) For the following command
\[
\pi_R.\text{userID}, R1.\text{userID} (R \bowtie_R.\text{videoID}=R1.\text{videoID} \land R.\text{rating}=R1.\text{rating} \land R.\text{userID}>R1.\text{userID}) \rho_{R1}(\text{Ratings})
\]

i. [1 point] How many, and which, are the columns of the answer?
Solution: There are 2 columns: R.\text{userID}, R1.\text{userID}.

ii. [2 points] How many tuples are in the results?
Solution: 1

iii. [3 points] Give, as a table, all of the tuples returned by the query.
Solution:

<table>
<thead>
<tr>
<th>R.\text{userID}</th>
<th>R1.\text{userID}</th>
</tr>
</thead>
<tbody>
<tr>
<td>U6</td>
<td>U1</td>
</tr>
</tbody>
</table>
Question 4: Relational Calculus .......................... [25 points]

Submit on separate page

Grading info: graded by Ming Zhong (MZ)

We will again use the video recommender database from the last question. We begin by looking at statements using tuple relational calculus.

(a) [2 points] Which of the following is the meaning of the statement \( \{ V \mid V \in \text{Videos} \land V.\text{length} < 20 \} \)

1. Select all videos tuples with \text{length} under 20 seconds
2. Select the \text{length} of all videos that are less than 20 seconds
3. Select the \text{length} of each video and if the length is over 20 seconds, round it down to 20 seconds
4. None of the above. The real answer is ..............

Solution: Answer #1

(b) [3 points] Which of the following commands gives the tuples from \text{Users} who liked (gave at least 4 stars) to video \( V4 \) (i.e., “The Wolf of Wall Street”):

1. \( \{ P \mid \exists U \in \text{Users}, \exists R \in \text{Ratings} (R.\text{userID} = U.\text{userID} \land P.\text{userID} = U.\text{userID} \land R.\text{videoID} = V4 \land P.\text{videoID} = V4 \land R.\text{rating} \geq 4 ) \} \)
2. \( \{ U \mid U \in \text{Users} \land \exists R \in \text{Ratings} (R.\text{userID} = U.\text{userID} \land R.\text{videoID} = V4 \land R.\text{rating} \geq 4 ) \} \)
3. \( \{ U \mid \exists R \in \text{Ratings} (R.\text{userID} = U.\text{userID} \land R.\text{videoID} = V4 \land R.\text{rating} \geq 4 ) \} \)
4. None of the above. The real answer is ..............

Solution: Answer #2

(c) For the following command:

\( \{ P \mid \exists U \in \text{Users}, \exists V1 \in \text{Videos}, \exists V2 \in \text{Videos}, \exists V3 \in \text{Videos} (V1.\text{videoID} \neq V2.\text{videoID} \land V2.\text{videoID} \neq V3.\text{videoID} \land V1.\text{videoID} \neq V3.\text{videoID} \land V1.\text{userID} = V2.\text{userID} \land V2.\text{userID} = V3.\text{userID} \land U.\text{userID} = V3.\text{userID} \land V1.\text{userID} = P.\text{userID} \land P.\text{username} = U.\text{name} ) \} \)

i. [1 point] How many, and which, are the columns of the answer?
   Solution: There are 2 columns: \text{userID} and \text{username}.

ii. [2 points] How many tuples are in the results?
   Solution: 1

iii. [3 points] Give, as a table, all of the tuples returned by the query.
(d) For the following command:

\[
\{ U \mid U \in \text{Users} \land \exists V \in \text{Videos}(U.userID = V.userID) \}
\]

i. [1 point] How many, and which, are the columns of the answer?

**Solution:** There are 2 columns: `userID` and `name`.

ii. [2 points] How many tuples are in the results?

**Solution:** 2

iii. [3 points] Give, as a table, all of the tuples returned by the query.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>U2</td>
<td>Jill</td>
</tr>
<tr>
<td>U3</td>
<td>John</td>
</tr>
</tbody>
</table>
Next we will look at commands that use domain relational calculus.

(e) [2 points] Which of the following commands gives the names of the users who liked (gave at least 4 stars) to videoID = ‘V4’ (ie., “The Wolf of Wall Street”)?

1. \{\langle N \rangle | \exists U, V, R (\langle U, N \rangle \in Users \land \langle V \rangle \in Videos \land \langle U, V, R \rangle \in Ratings \land R \geq 4 \land V =\text{‘V4’})\}

2. \{\langle N \rangle | \exists U, V, T, L, R (\langle U, N \rangle \in Users \land \langle V, T, L, U \rangle \in Videos \land \langle U, V, R \rangle \in Ratings \land R \geq 4)\}

3. \{\langle N \rangle | \exists U, V, R (\langle U, N \rangle \in Users \land \langle U, V, R \rangle \in Ratings \land R \geq 4 \land V =\text{‘V4’})\}

4. None of the above. The real answer is . . . . . . . . . . . . . . . .

Solution: Answer #3
Note, Answer #2 is incorrect because it requires that the same user uploaded the video and did the rating.

(f) For the following command:

\{\langle V, T \rangle | \exists L, U (\langle V, T, L, U \rangle \in Videos \land \exists U2, R ((\langle U2, V, R \rangle \in Ratings \land R = 5))\}

i. [1 point] How many, and which, are the columns of the answer?

Solution: There are 2 columns: videoID and title.

ii. [2 points] How many tuples are in the results?

Solution: 3

iii. [3 points] Give, as a table, all of the tuples returned by the query.

Solution:

<table>
<thead>
<tr>
<th>videoID</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Michelle Obama dunks on LeBron</td>
</tr>
<tr>
<td>V2</td>
<td>Michael Sherman yells at Erin Andrews</td>
</tr>
<tr>
<td>V4</td>
<td>The Wolf of Wall Street</td>
</tr>
</tbody>
</table>

Grading info:

- 1 point for every incorrect column name.
- Full points are deducted if there is any missing/wrong entry in the result. Also, if the table is correct but the number of entries is wrong, full point of (ii) will be taken.