Carnegie Mellon University
Department of Computer Science
15-415/615 - Database Applications
C. Faloutsos & A. Pavlo, Fall 2015

Homework 1 (by Dana Van Aken) - Solutions

Due: **hard copy, in class** at **3:00pm, on Monday, Sep. 21**

VERY IMPORTANT: Deposit **hard copy** of your answers, in class. For ease of grading, please

1. **Separate** your answers, on different page(s) for each question (staple additional pages, if needed).
2. **Type** the full info on **each** page: your **name**, **Andrew ID**, **course#**, **Homework#**, **Question#** on each of the 5 pages.

Reminders:

- *Plagiarism*: Homework is to be completed *individually*.
- *Typeset* all of your answers whenever possible. Illegible handwriting may get zero points, at the discretion of the graders.
- *Late homeworks*: in that case, 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; **5** questions total
- Rough time estimate: *approx. 6 hours* - 1 to 2 hours per question

Revision : 2015/09/27 21:18

Question	Points	Score
Entity-Relationship Diagram	25	
SQL Tables from the ER Model	15	
Relational Algebra for a Q & A Website	30	
Relational Tuple Calculus (RTC)	15	
Relational Domain Calculus (RDC)	15	
Total:	100	

Question 1: Entity-Relationship Diagram . . . . . [25 points]

GRADED BY: Dana Van Aken

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider a database to store information about a Sports Organization. The database has the following properties:

- Each league has a name and unique league ID (integer). The 'NFL' (National Football League) is an example of a sports league.
- A league may be affiliated with one or more teams.
- Every team is affiliated with exactly one league.
- We store the name, colors, and unique team ID (integer) for each team. For example, the name of Pittsburgh's football team is the 'Steelers' and their team colors are 'black & gold'.
- A team has exactly one home stadium. A stadium may be a home to zero or more teams. For example, the home stadium of the Pittsburgh's baseball team, the Pirates, is 'PNC Park'.
- For home stadiums, we record a city, name, and unique stadium ID (integer).
- A team has at least one player, and a player plays for exactly one team.
- A team has exactly one manager, and a manager manages exactly one team.
- Every player has a number and a position. For example, Pittsburgh Steelers player Ben Roethlisberger has the number 7 and his position is 'quarterback'.
- For managers, we record the dollar amount of their annual bonus.
- Both players and managers are types of employees.
- For each employee, we store a name, salary, and unique employee ID (integer).

Given this description of the database and its constraints, we have created a mostly correct Entity-Relationship Diagram, shown in Figure 1. This diagram is at this link - feel free to use it as a starting point.

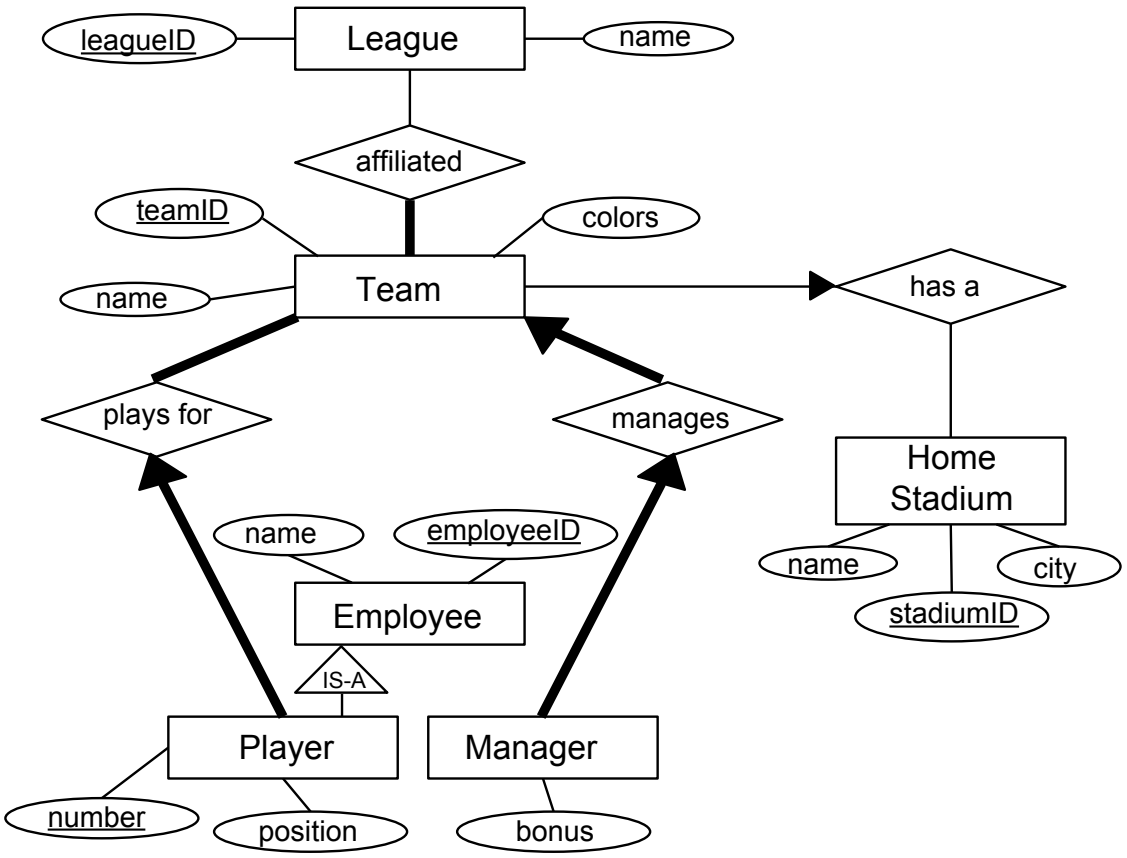


Figure 1: Almost correct ER diagram

(a) **[10 points]** Find and correct any mistakes in the given ER diagram. Specifically, 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

Solution:

1. Change the thin line from `League` to `Team` to bold.
2. Change the bold line from `Team` to `League` to a bold arrow.
3. Change the thin arrow from `Team` to `HomeStadium` to a bold arrow.
4. Change the direction of the bold arrow between the m̂anages relationship and the `Team` entity to point towards the `Manager` entity.
5. Delete the underline from the `number` attribute for entity `Player`.

(b) **[5 points]** There may also be some missing element(s). If none, say 'none' - otherwise, 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 ....

Solution:

1. Add a line from `Manager` to the ISA node.
2. Add attribute `salary` to entity `Employee`.

(c) **[10 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**, line, from `Department` to `Employee`
2. thin, **arrow**, from x to y

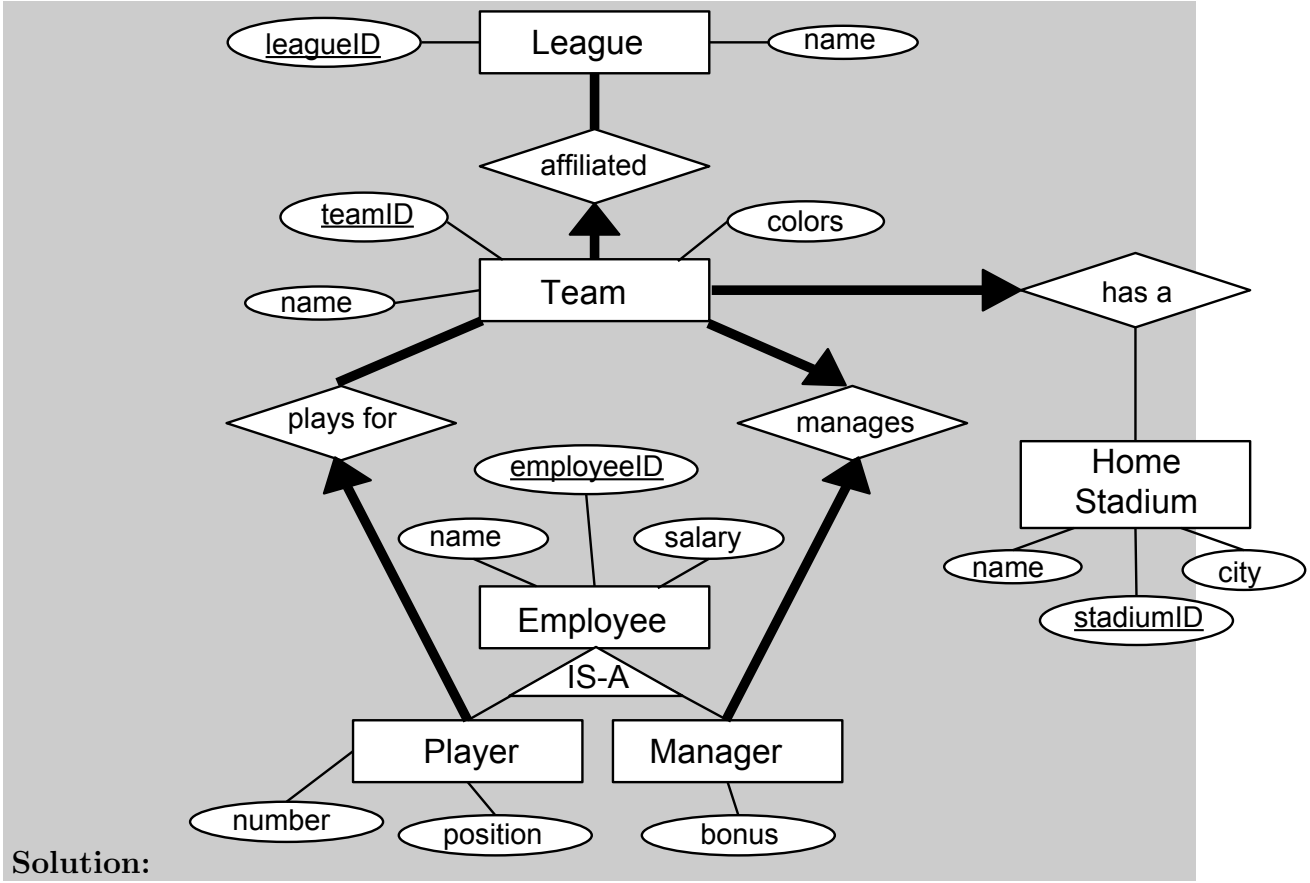
Solution:

1. Bold line from `League` to `Team`
2. Bold arrow from `Team` to `League`
3. Bold arrow from `Team` to `HomeStadium`
4. Bold line from `Team` to `Player`
5. Bold arrow from `Player` to `Team`
6. Bold arrow from `Team` to `Manager`
7. Bold arrow from `Manager` to `Team`

Clarifications/Hints:

- List your assumptions, if any. We will accept all reasonable assumptions.

Solution:



Grading info:

- *(a)*
  - *-3 points for forgetting to delete the underline from the number attribute for entity Player*
  - *-2 points for changing the number attribute to a partial key*
  - *-1 for incorrectly changing or forgetting to change the incorrect cardinalities*
  - *-1 points for making extra changes to the ER diagram that are incorrect, but this is a one-time penalty*
  - *No penalty for changing "Home Stadium" to "Stadium"*
- *(b)*
  - *-2 points for forgetting to add salary attribute to Employee or adding it to the wrong entity*
  - *-3 points for forgetting to connect Manager to the ISA node (Manager is a subclass of the Employee entity)*
- *(c)*
  - *-1 for each missing or incorrect arrow or bold line*
  - *No penalty for including thin lines*
- *No penalty for mixing up changes/additions/deletions in parts (a) and (b)*
- *Note: subclasses in class hierarchies inherit all attributes defined by the superclass(es). In this example, the Manager and Player entities are subclasses of the Employee entity and inherit the unique employeeID as their key, as well as the salary and name attributes. Managers and Players are uniquely identified by their employeeIDs and are therefore not weak entities.*

Question 2: SQL Tables from the ER Model . . . . . [15 points]

GRADED BY: Jiaxi Xiong

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider a database for a Twitter-like organization. It records information about users, followers, and Tweets. The constraints are exactly as shown in Figure 2. Users and Tweets have unique identifiers as shown in the Figure, with binary relationships among them as illustrated. To clarify:

- The arrow from "Tweet" to "User", is thick.
- No other lines, boxes, or diamonds, are thick.

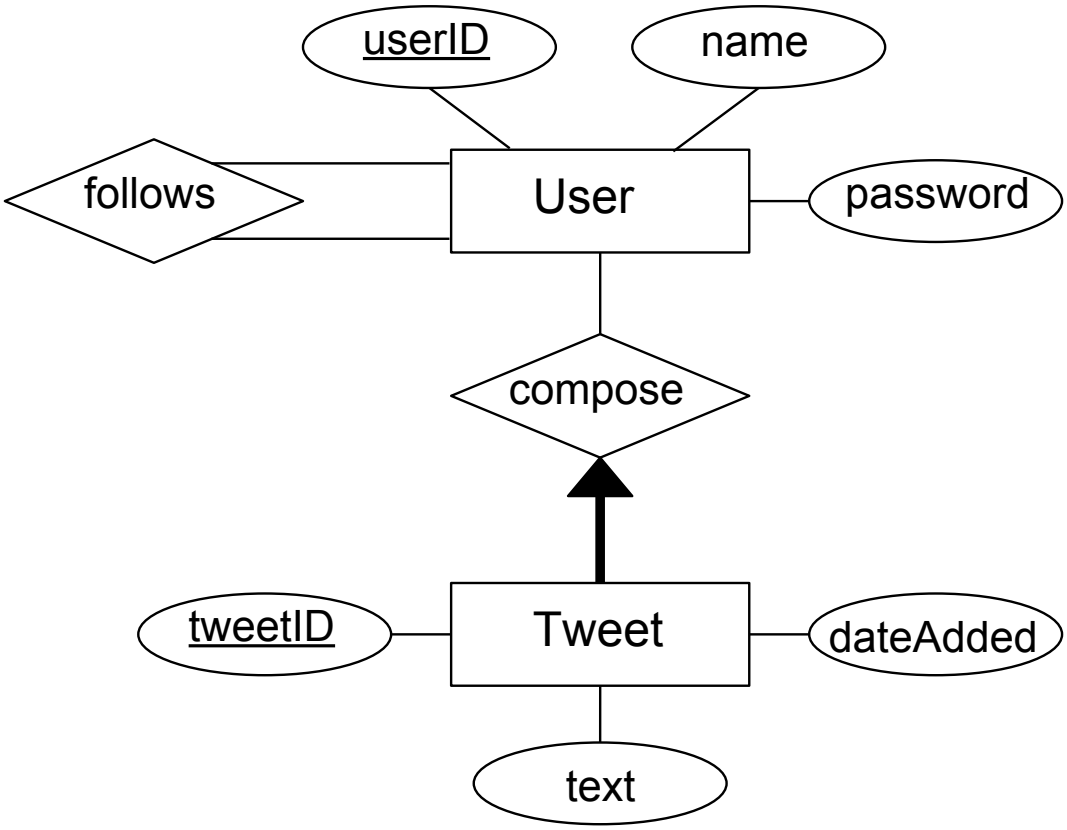


Figure 2: ER diagram for Twitter: turn to SQL tables

(a) **[15 points]** Give the DDL statements, that correspond to the above ER diagram.

- Use proper data types (we'll accept all reasonable choices).
- Avoid syntax errors (we'll forgive missing semicolons).
- Specify your decisions with respect to `CASCADE` deletions.
- **Without** using `CHECK` statements, enforce as many as possible of the implied integrity constraints as you can.

Solution:

```
CREATE TABLE User (
    userID INTEGER,
    name CHAR(20),
    password CHAR(30),
    PRIMARY KEY (userID)
);
```

Solution:

```
CREATE TABLE Follows (
    follower INTEGER,
    followed INTEGER,
    PRIMARY KEY (follower, followed),
    FOREIGN KEY (follower) REFERENCES User ON DELETE CASCADE,
    FOREIGN KEY (followed) REFERENCES User ON DELETE CASCADE
);
```

Solution:

```
CREATE TABLE Tweet (
    tweetID INTEGER,
    text CHAR(140),
    userNumber INTEGER,
    dateAdded DATE,
    PRIMARY KEY (tweetID),
    FOREIGN KEY(userNumber) REFERENCES User ON DELETE CASCADE
);
```

Grading info:

- *(a) -5 points for missing a table*
- *(b) -1 point for missing primary key*
- *(c) -1 point for missing foreign key*
- *(d) -1 point for wrong ON DELETE actions*
- *(e) -1 point for wrong ON DELETE grammar*
- *(f) -1 point for wrong dateAdded type (we accept DATE, DATETIME, TIMESTAMP. But char and integer are not reasonable)*

Question 3: Relational Algebra for a Q & A Website. [30 points]

GRADED BY: Yujing Zhang

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

Consider the relations of a database for a `Question & Answer Website` as shown in Table 1. These relations describe a Q & A discussion board, where users can ask questions, post answers, and vote for the most helpful answers (as in StackExchange(TM)).

userID	name	reputation
U0	Judy	44
U1	Will	251
U2	Anne	11
U3	Gary	9
U4	Erika	200
U5	Mike	75

(a) `UserProfile` Table

questionID	userID	title
Q0	U1	"Making a one-to-many relation lazy"
Q1	U4	"Django = DatabaseError: No such table"
Q2	U5	"Inefficient database query inside a for loop"
Q3	U5	"Help with 15-415 homework..."

(b) `Question` Table: who posed what question

questionID	userID	vote_count
Q0	U3	46
Q1	U0	25
Q1	U2	131
Q2	U5	5
Q2	U3	21
Q2	U4	118
Q3	U5	45
Q3	U1	87

(c) `Answer` Table: who answered what question; how many people voted for that answer

Table 1: Relations of a database for a `Q & A Website`.

We have the following tables:

- `UserProfile`: For each user profile, we record the user's `name`, `reputation`, and unique `userID`. The `reputation` attribute measures the degree to which the community trusts a particular user. For example, a user can increase their reputation by posting helpful answers to questions.
- `Question`: For each question asked by a user, we record his or her `userID`, the question `title`, and unique `questionID`.
- `Answer`: We track all answers posted by users in the `Answer` table (see Table 1(c)). Each row stores the `userID` of the user that answered the question, the `questionID` of the question answered, and the `vote_count`, (i.e., the total number of users that voted that answer as being the most helpful). For example, the first row of Table 1(c) shows that user 'U4' (='Erika') answered question 'Q2' (='Inefficient database...') and, for her answer, she received a `vote_count` of 118.

Given this database instance, answer the following questions:

(a) **[2 points]** Which of the following is the meaning of the expression:

$\sigma_{\texttt{reputation} \geq 200}(\texttt{UserProfile})$

1. It lists the `userID` and `name` of all users with a `reputation` greater than or equal to 200.
2. It lists all `reputation`s that are greater than or equal to 200.
3. It lists all tuples in the `UserProfile` table ((`userID`, `name`, and `reputation`) with a `reputation` greater than or equal to 200.
4. None of the above. The real answer is .....

Solution: #3

(b) **[5 points]** We want to list the `title`s of questions asked by "top" users, (i.e., users with a `reputation` greater than 75). Which, if any, of the following expressions achieve that? Mark **all** valid expressions.

1. $\pi_{\texttt{title}}\big(\ \sigma_{\texttt{reputation}>75}(\texttt{UserProfile} \bowtie \texttt{Question})\ \big)$
2. $\sigma_{\texttt{reputation}>75}\big(\ \pi_{\texttt{title}}(\texttt{UserProfile} \bowtie \texttt{Question})\ \big)$
3. $\pi_{\texttt{title}}\big(\ \texttt{Question} \bowtie \sigma_{\texttt{reputation}>75}(\texttt{UserProfile})\ \big)$
4. $\pi_{\texttt{title}}\Big(\ \sigma_{\texttt{reputation}>75}\ \big(\texttt{UserProfile} \bowtie \pi_{\texttt{title,userID}}(\texttt{Question})\ \big)\ \Big)$
5. $\pi_{\texttt{title}}\Big(\ \sigma_{\texttt{reputation}>75}\big(\ \texttt{UserProfile} \bowtie \pi_{\texttt{title}}(\texttt{Question})\ \big)\ \Big)$

Solution: #1, #3, and #4

Grading info:

- *+1 point for each expression correctly listed (or not listed)*

(c) For the following expression:

$$\sigma_{\texttt{vote\_count}<25}(\texttt{Answer} \bowtie \texttt{Question})$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List `questionID`, `userID`, `title`, and `vote_count` of users who answered their own question and received a `vote_count` less than 25.

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: 4 columns: `questionID`, `userID`, `title`, and `vote_count`.

iii. **[3 points]** How many tuples are in the answer?

Solution: 1

iv. **[3 points]** List all the tuples in the answer, as a table.

Solution:

questionID	userID	title	vote_count
Q2	U5	"Inefficient database query inside a for loop"	5

Grading info:

- *ii. -1 for missing columns or having extra columns*
- *iii. -1 for being off by one on number of tuples; -3 for getting the wrong number of tuples by more than one*
- *iv. -1 for missing one tuple; -2 for performing wrong operation (e.g. $\times$ instead of $\bowtie$) or getting some columns and tuples wrong; -3 for all tuples and columns wrong*

(d) For the following expression:

$$\pi_{\texttt{userID,questionID}}(\texttt{Answer}) \div \pi_{\texttt{questionID}}\big(\sigma_{\texttt{userID}=\text{`U4'}}(\texttt{Answer})\big)$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List the `userID`s of users who have answered all of the same questions (and possibly more) as `userID` "U4".

ii. **[2 points]** How many, and which are the columns (= attributes) in the answer?

Solution: One column: `userID`.

iii. **[3 points]** How many tuples are in the answer?

Solution: 3

iv. **[3 points]** List all the tuples in the answer, as a table.

Solution:

userID
U3
U4
U5

Grading info:
- *ii. -1 for missing columns or having extra columns*
- *iii. -1 for being off by one on number of tuples; -3 for getting the wrong number of tuples by more than one*
- *iv. -1 for missing one tuple; -1 for missing column or having extra column; -2 for getting some columns and tuples wrong; -3 for all tuples and columns wrong*

(e) For the following expression:

$$\pi_{\texttt{C.userID}}\big(\rho_C(\texttt{UserProfile})\big) - \pi_{A.\texttt{userID}}\big(\rho_A(\texttt{Answer}) \bowtie_{A.\texttt{questionID}=B.\texttt{questionID} \wedge A.\texttt{vote\_count}<B.\texttt{vote\_count}} \rho_B(\texttt{Answer})\big)$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: Finds the "super-users", defined as users who have received the highest vote_count for every question they have answered, (including users who have never answered any questions). Equivalently, "super-users" are users who have never lost to any other user when answering a question.

ii. **[2 points]** How many, and which are the columns (= attributes) in the answer?

Solution: 1 columns: C.`userID`.

iii. **[3 points]** How many tuples are in the answer?

Solution: 3

iv. **[3 points]** List all the tuples in the answer, as a table.

Solution:

C.userID
U1
U2
U4

Grading info:
- *ii. -1 for missing columns or having extra columns*
- *iii. -1 for being off by one on number of tuples; -3 for getting the wrong number of tuples by more than one*
- *iv. -1 for missing one tuple; -2 for getting some columns and tuples wrong; -3 for all tuples and columns wrong*

Question 4: Relational Tuple Calculus (RTC) . . . . . [15 points]

GRADED BY: Jinliang Wei

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

We will again use the Q & A Website database from the last question (see Table 1).

(a) For the following RTC expression

$$\{t \mid \exists q \in \texttt{Question}\,(\, q.\texttt{userID} = \text{“}U5\text{”} \;\wedge\; q.\texttt{questionID} = t.\texttt{questionID}\,)\}$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List the `questionID` of all questions asked by the user with `userID` = "$U5$".

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: There is one column: `questionID`.

iii. **[2 points]** How many tuples are in the answer?

Solution: 2

iv. **[2 points]** List all the tuples in the answer, as a table.

Solution:

questionID
Q2
Q3

Grading info:

- *ii. -1 for wrong number of columns; -1 for wrong attribute names*
- *iii. -1 if number of tuples is off by 1; -2 if off by 2 or more*
- *iv. -1 if contains any extra column or missing any column; -1 for each extra or missing tuple (up to -2); no penalty for missing or wrong attribute names*

(b) For the following RTC expression

$$\begin{aligned}\{t \mid \exists p \in \texttt{UserProfile}, \exists a \in \texttt{Answer}\\ (\, & p.\texttt{userID} = a.\texttt{userID}\\ \wedge\; & p.\texttt{reputation} > 75\\ \wedge\; & a.\texttt{vote\_count} > 100\\ \wedge\; & t.\texttt{name} = p.\texttt{name}\\ \wedge\; & t.\texttt{questionID} = a.\texttt{questionID})\}\end{aligned}$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List the `name` and `questionID` of any "top" user, (i.e., a user with a `reputation` > 75), who received over 100 votes for his or her answer to the question.

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: Two columns: `name` and `questionID`.

iii. **[2 points]** How many tuples are in the answer?

Solution: 1

iv. **[2 points]** List all the tuples in the answer, as a table.

Solution:

name	questionID
Erika	Q2

Grading info:

- *ii. -1 for wrong number of columns; -1 for wrong attribute names*
- *iii. -1 if number of tuples is off by 1; -2 if off by 2 or more*
- *iv. -1 if contains any extra column or missing any column; -1 for each extra or missing tuple (up to -2); no penalty for missing or wrong attribute names*

(c) For the following RTC expression

$$\{t \mid \exists a1 \in \texttt{Answer}, \exists a2 \in \texttt{Answer}$$
$$(\ a1.\texttt{questionID} = a2.\texttt{questionID}$$
$$\wedge\ a1.\texttt{userID} > a2.\texttt{userID}$$
$$\wedge\ t.\texttt{userID1} = a1.\texttt{userID}$$
$$\wedge\ t.\texttt{userID2} = a2.\texttt{userID}\ )\}$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List the `userID`s of pairs of users that answered the same question. Eliminate self-pairs and mirror-pairs.

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: Two columns: `userID1`, `userID2`.

iii. **[2 points]** How many tuples are in the answer?

Solution: 5

iv. **[2 points]** Give, as a table, all of the tuples returned by the query.

Solution:

userID1	userID2
U2	U0
U5	U4
U5	U3
U4	U3
U5	U1

Grading info:

- *ii. -1 for wrong number of columns; -1 for wrong attribute names*
- *iii. -1 if number of tuples is off by 1; -2 if off by 2 or more*
- *iv. -1 if contains any extra column or missing any column; -1 for each extra or missing tuple (up to -2); no penalty for missing or wrong attribute names*

Question 5: Relational Domain Calculus (RDC) . . . . . [15 points]

GRADED BY: Jinliang Wei

On separate page, with '[course-id] [hw#] [question#] [andrew-id] [your-name]'

We will again use the Q & A Website database from question 3 (see Table 1).

(a) For the following RDC expression
$\{\langle u\rangle \mid \exists q, \exists v(\langle q, u, v\rangle \in \texttt{Answer} \wedge v < 50)\}$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List the `userID`s of users that have received less than 50 votes for at least one of the questions that they answered.

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: One column: `userID`.

iii. **[2 points]** How many tuples are in the answer?

Solution: 3

iv. **[2 points]** List all the tuples in the answer, as a table.

Solution:

userID
U0
U3
U5

Grading info:

- *ii. -1 for wrong number of columns; -1 for wrong attribute names*
- *iii. -1 if number of tuples is off by 1; -2 if off by 2 or more*
- *iv. -1 if contains any extra column or missing any column; -1 for each extra or missing tuple (up to -2); no penalty for missing or wrong attribute names; -1 for redundant tuples*

(b) For the following RDC expression:

$$\{\langle n1, n2\rangle \mid \exists q, \exists u1, \exists v1, \exists n1, \exists r1, \exists u2, \exists v2, \exists n2, \exists r2\ (\\ \langle q, u1, v1\rangle \in \texttt{Answer} \\ \wedge \langle q, u2, v2\rangle \in \texttt{Answer} \\ \wedge \langle u1, n1, r1\rangle \in \texttt{UserProfile} \\ \wedge \langle u2, n2, r2\rangle \in \texttt{UserProfile} \\ \wedge u1 > u2\ )\}$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: List the `name`s of pairs of users that answered the same question. Again, no self-pairs and mirror-pairs.

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: Two columns: n1.`name` and n2.`name`.

iii. **[2 points]** How many tuples are in the answer?

Solution: 5

iv. **[2 points]** List all the tuples in the answer, as a table.

Solution: The same as in RTC question (c) except that we list the user `name`s instead of `userID`s:

n1.name	n2.name
Anne	Judy
Mike	Erika
Mike	Gary
Erika	Gary
Mike	Will

Grading info:

- *ii. -1 for wrong number of columns; -1 for wrong attribute names*
- *iii. -1 if number of tuples is off by 1; -2 if off by 2 or more*
- *iv. -1 if contains any extra column or missing any column; -1 for each extra or missing tuple (up to -2); no penalty for missing or wrong attribute names*

(c) For the following RDC expression:

$$\left\{ \langle q, u1, v1 \rangle \;\middle|\; \left(\langle q, u1, v1 \rangle \in \texttt{Answer} \right. \right. \\ \left. \left. \wedge\ \forall u2 \left(\left(\exists v2 (\langle q, u2, v2 \rangle \in \texttt{Answer}) \ \Rightarrow\ (v1 \geqslant v2) \right) \right) \right) \right\}$$

i. **[0 points]** *Optional*: describe in English what the expression does

Solution: Lists (`questionID`, `userID`, `vote_count`) tuples for the highest-voted answer for each question that has at least one answer.

ii. **[1 point]** How many, and which are the columns (= attributes) in the answer?

Solution: Three columns: q.`questionID`, u1.`userID`, and v1.`vote_count`.

iii. **[2 points]** How many tuples are in the answer?

Solution: 4

iv. **[2 points]** List all the tuples in the answer, as a table.

Solution:

q.questionID	u1.userID	v1.vote_count
Q0	U3	46
Q1	U2	131
Q2	U4	118
Q3	U1	87

Grading info:

- *ii. -1 for wrong number of columns; -1 for wrong attribute names*
- *iii. -1 if number of tuples is off by 1; -2 if off by 2 or more*
- *iv. -1 if contains any extra column or missing any column; -1 for each extra or missing tuple (up to -2); no penalty for missing or wrong attribute names*