Carnegie Mellon University
Department of Computer Science
15-415/615 - Database Applications
C. Faloutsos & A. Pavlo, Spring 2014

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)

Revision : 2014/02/11 01:12

Question	Points	Score
Entity-Relationship Diagram	20	
SQL Tables from the ER Model	25	
Relational Algebra for Video Recommendation	30	
Relational Calculus	25	
Total:	100	

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 1.

(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`

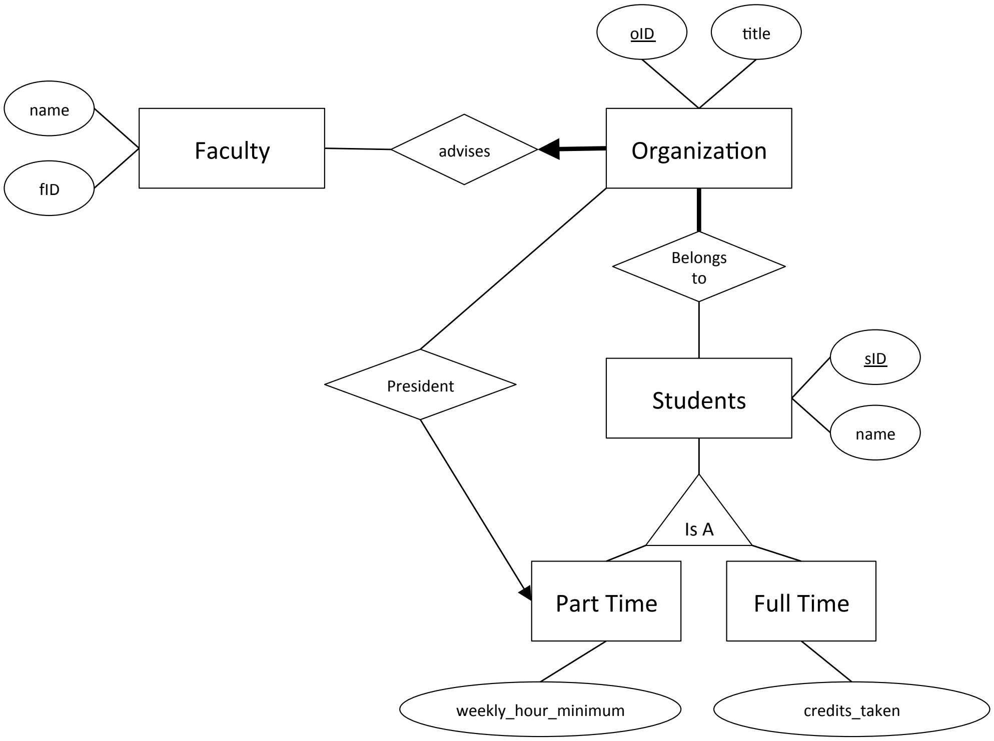
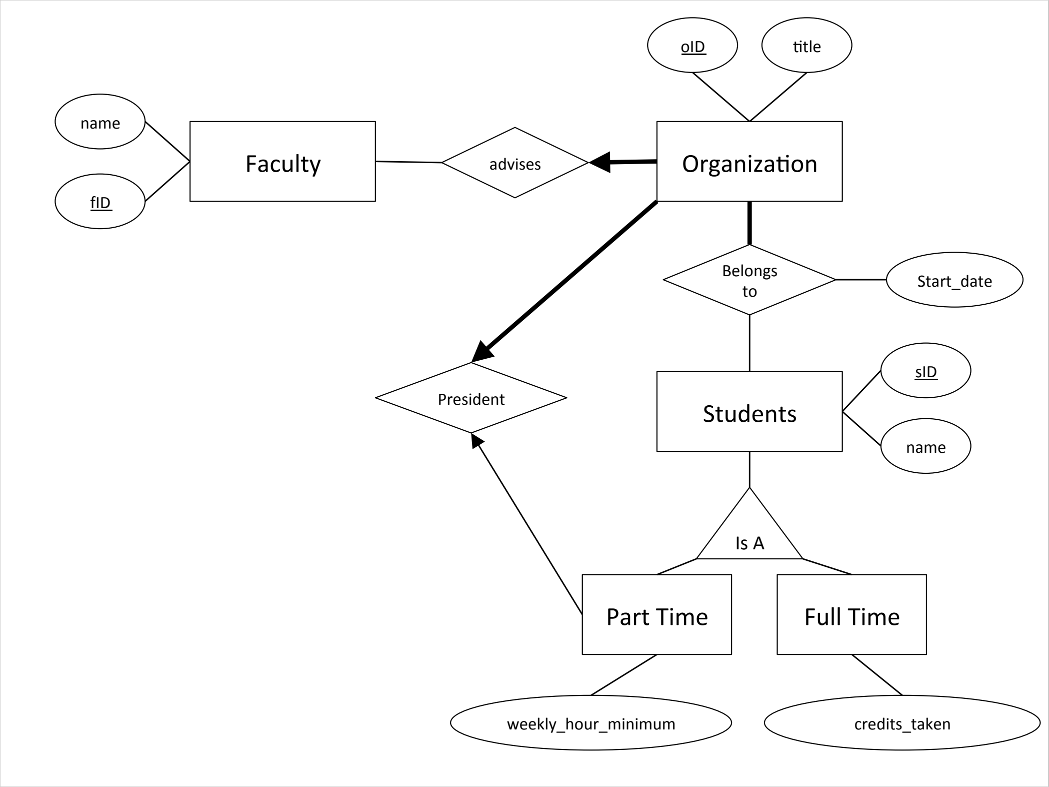


Figure 1: Almost correct ER diagram

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*

- *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]

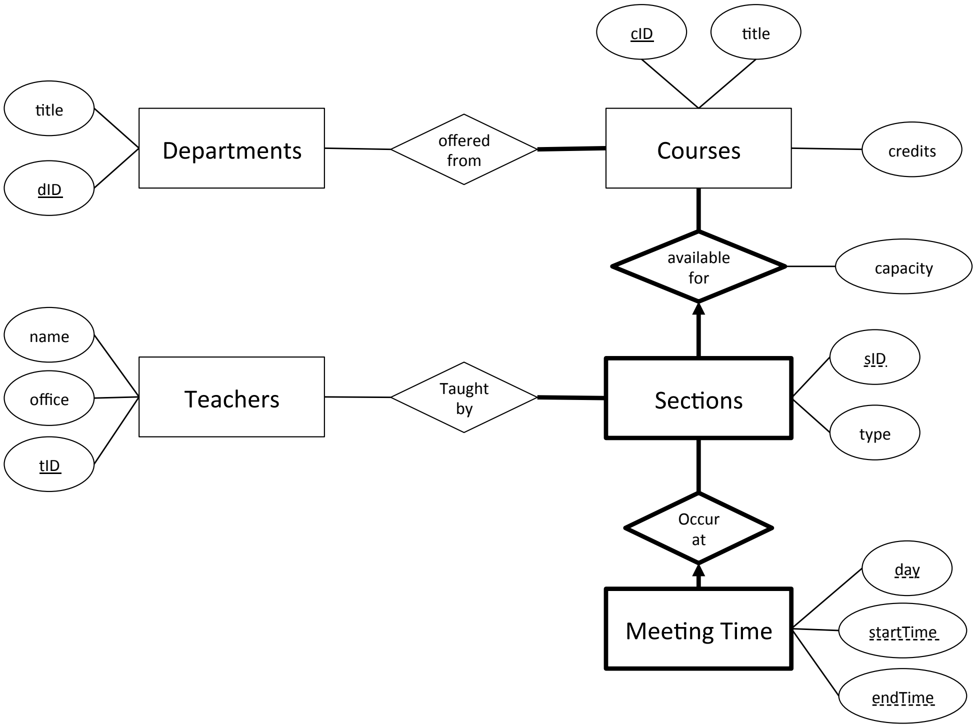
Submit on separate page

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:

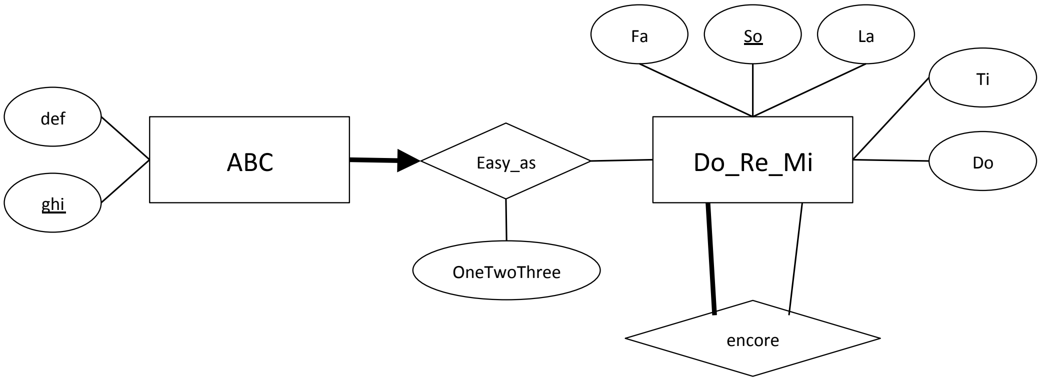
```
CREATE TABLE Departments (
    dID INTEGER,
    title CHAR(50),
    PRIMARY KEY (dID)
)

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,
    tID INTEGER,
    PRIMARY KEY (sID, tID),
    FOREIGN KEY (sID) 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 two participation constraints: That each Section has at least one teacher, and that each Course belongs to at least one Department.

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):



(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 (
    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

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.

`userID`	`name`
U1	Jack
U2	Jill
U3	John
U4	Jane
U5	Job
U6	Jay

Table 1: `Users`

`videoID`	`title`	`length`	`userID`
V1	Michelle Obama dunks on LeBron	18	U3
V2	Michael Sherman yells at Erin Andrews	46	U3
V3	Jack Gleeson on being famous	1707	U2
V4	The Wolf of Wall Street	10780	U2
V5	Transcendence Official Trailer	152	U2

Table 2: `Videos`

`userID`	`videoID`	`rating`
U3	V1	5
U3	V2	5
U3	V4	5
U2	V4	4
U1	V4	1
U6	V4	1
U5	V4	2
U4	V5	3

Table 3: `Ratings`

Given this database answer the following questions:

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

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 . . . .

Solution: Answer #1

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

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

Solution: Answer #3

(c) For the following command:

$$\texttt{Videos} \bowtie \texttt{Ratings}$$

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:

videoID	userID	title	length	rating
V1	U3	Michelle Obama dunks on LeBron	18	5
V2	U3	Michael Sherman yells at Erin Andrews	46	5
V4	U2	The Wolf of Wall Street	10780	4

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 didnt 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:

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

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:

videoID	userID	title	length	rating
V1	U3	Michelle Obama dunks on LeBron	18	5
V2	U3	Michael Sherman yells at Erin Andrews	46	5

(e) For the following command

$$\{\pi_{\texttt{userID,videoID}}(\texttt{Ratings})\} \div \{\pi_{\texttt{videoID}}(\sigma_{\texttt{userID}='U1'}(\texttt{Ratings}))\}$$

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:

userID
U1
U2
U3
U5
U6

(f) For the following command

$$\pi_{\texttt{R.userID, R1.userID}}\left(\rho_{\texttt{R}}(\texttt{Ratings}) \bowtie_{\texttt{R.videoID=R1.videoID}\wedge\texttt{R.rating=R1.rating}\wedge\texttt{R.userID>R1.userID}} \rho_{\texttt{R1}}(\texttt{Ratings})\right)$$

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

Solution: There are 2 columns: R.`userID`, R1.`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:

R.userID	R1.userID
U6	U1

Question 4: Relational Calculus .... [25 points]

Submit on separate page

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 $\{V | V \in \texttt{Videos} \wedge V.\texttt{length} < 20\}$

1. Select all videos tuples 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 ....

Solution: Answer #1

(b) [**3 points**] Which of the following commands gives the tuples from `Users` who liked (gave at least 4 stars) to video `V4` (i.e., "The Wolf of Wall Street"):

1. $\{P | \exists U \in \texttt{Users}, \exists R \in \texttt{Ratings}(R.\texttt{userID} = U.\texttt{userID} \wedge P.\texttt{userID} = U.\texttt{userID} \wedge R.\texttt{videoID} = V4 \wedge P.\texttt{videoID} = V4 \wedge R.\texttt{rating} \geq 4\}$
2. $\{U | U \in \texttt{Users} \wedge \exists R \in \texttt{Ratings}(R.\texttt{userID} = U.\texttt{userID} \wedge R.\texttt{videoID} = V4 \wedge R.\texttt{rating} \geq 4\}$
3. $\{U | \exists R \in \texttt{Ratings}(R.\texttt{userID} = U.\texttt{userID} \wedge R.\texttt{videoID} = V4 \wedge R.\texttt{rating} \geq 4\}$
4. None of the above. The real answer is ....

Solution: Answer #2

(c) For the following command:

$$\begin{aligned}\{P | \exists U \in \texttt{Users}, \exists V1 \in \texttt{Videos}, \exists V2 \in \texttt{Videos}, \exists V3 \in \texttt{Videos}(\\ V1.\texttt{videoID} \neq V2.\texttt{videoID} \wedge V2.\texttt{videoID} \neq V3.\texttt{videoID} \wedge V1.\texttt{videoID} \neq V3.\texttt{videoID}\\ \wedge V1.\texttt{userID} = V2.\texttt{userID} \wedge V2.\texttt{userID} = V3.\texttt{userID}\\ \wedge U.\texttt{userID} = V3.\texttt{userID} \wedge V1.\texttt{userID} = P.\texttt{userID}\\ \wedge P.\texttt{username} = U.\texttt{name})\}\end{aligned}$$

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

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

Solution:

userID	username
U2	Jill

(**Big Hint:** Check out example *Q7* on Page 121 of the textbook.)

(d) For the following command:

$$\{U | U \in \texttt{Users} \wedge \exists V \in \texttt{Videos}(U.\texttt{userID} = V.\texttt{userID})\}$$

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.

Solution:

userID	name
U2	Jill
U3	John

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. $\{\langle N\rangle | \exists U, V, R(\langle U, N\rangle \in \texttt{Users} \wedge \langle V\rangle \in \texttt{Videos} \wedge \langle U, V, R\rangle \in \texttt{Ratings} \wedge R \geq 4 \wedge V =' V4')\}$
2. $\{\langle N\rangle | \exists U, V, T, L, R(\langle U, N\rangle \in \texttt{Users} \wedge \langle V, T, L, U\rangle \in \texttt{Videos} \wedge \langle U, V, R\rangle \in \texttt{Ratings} \wedge R \geq 4)\}$
3. $\{\langle N\rangle | \exists U, V, R(\langle U, N\rangle \in \texttt{Users} \wedge \langle U, V, R\rangle \in \texttt{Ratings} \wedge R \geq 4 \wedge V =' V4')\}$
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:

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

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:

`videoID`	`title`
V1	Michelle Obama dunks on LeBron
V2	Michael Sherman yells at Erin Andrews
V4	The Wolf of Wall Street

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.*

End of Homework 1