Database Management Systems (COP 5725)

(Spring 2018)

Instructor: Dr. Markus Schneider

TA: Matin Kheirkhahan

Exam 1 Solutions

Name:	
UFID:	
Email Address:	

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

____

Signature

For scoring use only:

	Maximum	Received
Question 1	25	
Question 2	40	
Question 3	35	
Total	100	

Question 1 [25 points]

Provide brief answers for the following questions.

(a) [4 points] What are *strong* and *weak* entity sets?

- Strong entity sets are those that exist autonomously and can be uniquely identified within an entity set by their key attributes.
- Weak entity sets are those that do not have sufficient attributes to form a key. Therefore, they are dependent in their existence from another *strong* entity and can be uniquely identified only in combination with the key of the *strong* entity.

(b) [4 points] Let R be a relation with schema $\mathbb{R}$, and let S be a relation with schema $\mathbb{S}$. Express the *division* (*quotient*) and *intersection* operations using basic Relational Algebra operations.

- Division: (Preconditions: $\mathbb{S} \subseteq \mathbb{R}$, $X = \mathbb{R} - \mathbb{S}$)
  $R \div S = \pi_x(R) - \pi_x((\pi_x(R) \times S) - R)$

- Intersection: (Precondition: $\mathbb{R} = \mathbb{S}$)
  $R \cap S = R - (R - S)$

(c) [6 points] Consider two database relations R and S. Argue whether the following statements are true or not. Evaluate each statement individually, and give a counter-example if a statement is incorrect ($⟗$ denotes the full outer join).

a. $\pi_A(R \cap S) = \pi_A(R) \cap \pi_A(S)$, where A is a common attribute of R and S.

- This statement is false: $R(A, B) = \{(1, 2)\}$; $S(A, B) = \{(1, 3)\}$
  $\pi_A(R \cap S) = \emptyset$
  $\pi_A(R) \cap \pi_A(S) = \{(1)\}$

b. $\sigma_C(R ⟗ S) = \sigma_C(R) ⟗ S$, where C is an attribute of R.

- This statement is false: Assume we have $R(A, B) = \{(1, 2), (3, 3)\}$, $S(B, C) = \{(2, 2)\}$.
  $\sigma_{A=3}(R ⟗ S) = \sigma_{A=3}(\{(1,2,2), (3, 3, null)\}) = \{(3, 3, null)\}$
  $\sigma_{A=3}(R) ⟗ S = \{(3, 3)\} ⟗ S = \{(3, 3, null), (null, 2, 2)\}$

(d) [6 points] Consider the following relation R:

```
R |   X    |  Y
----------------
  | 5725 | DBMS
  | 6610 |  ML
  | 7250 | MDA
```

Determine a Relational Algebra expression that creates the following new relation S:

```
S |  X1   |  Y   |   X2
-------------------------
  | 5725 | DBMS |  NULL
  | 6610 |  ML  |  6610
  | NULL | NULL |  7250
```

- $\rho_{R1(X1,Y)}(\sigma_{X<7000}(R))$
  $\rho_{R2(X2,Y)}(\sigma_{X>6\ 000}(R))$
  $S \leftarrow R1 ⟗ R2$

(e) [5 points]] Let $R(A, B)$ be a relation with $r > 0$ tuples. We assume that A and B have the *same data type*. For the Relational Algebra expression

$$\sigma_{A>B}(R) \cup \sigma_{A<B}(R)$$

determine the *minimum* and *maximum number of tuples* that the result relation can have in terms of r and s. In other words, we are interested in the number of tuples this Relational Algebra expression can have *at least* and *at most*. The numbers have to be given by using the two variables r and s. Please note that you have to give a precise explanation for your answer.

Minimum number of possible tuples in the result:

$\forall\ t \in R : t.A = t.B \rightarrow$ Result: 0 [$t.A$ = value of attribute A in tuple t]

Maximum number of possible tuples in the result:

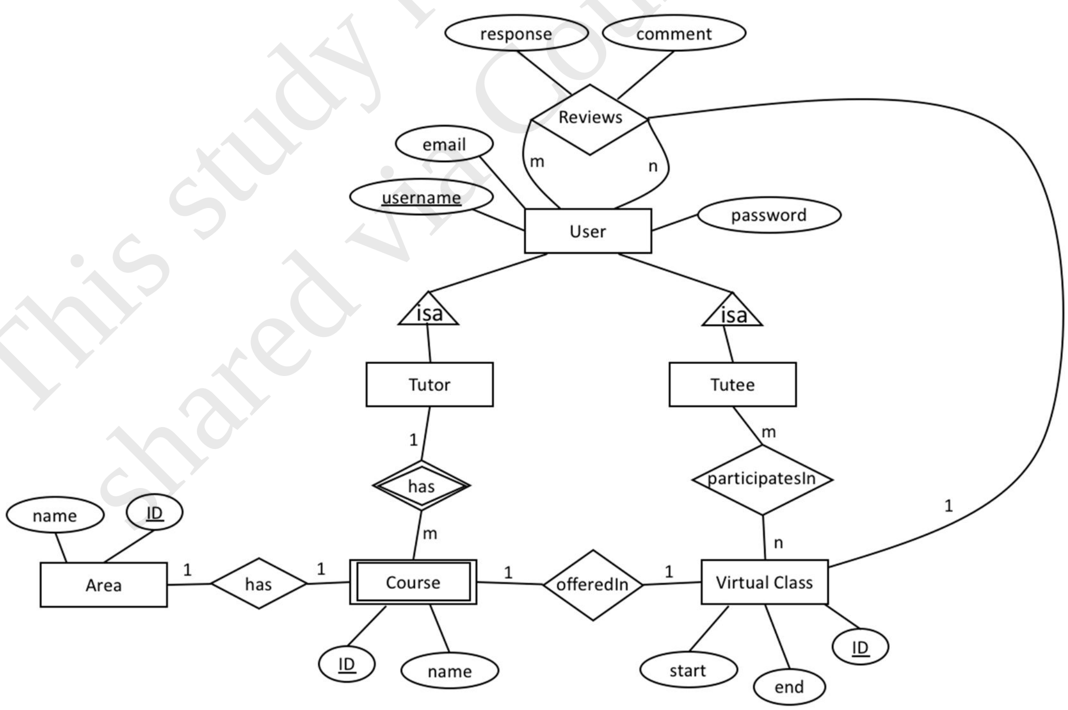
$\forall\ t \in R : t.A \neq t.B$ (that is, $t.A > t.B \vee t.A < t.B$) $\rightarrow$ Result: r

Question 2 [40 points]

For the following ER design questions, identify the primary key of each entity set (if the primary key is not specified by requirement, use your best knowledge to add a key or use existing attributes). For each relationship, state the cardinalities (1:1, 1:m, m:n) between the entities participating in a relationship.

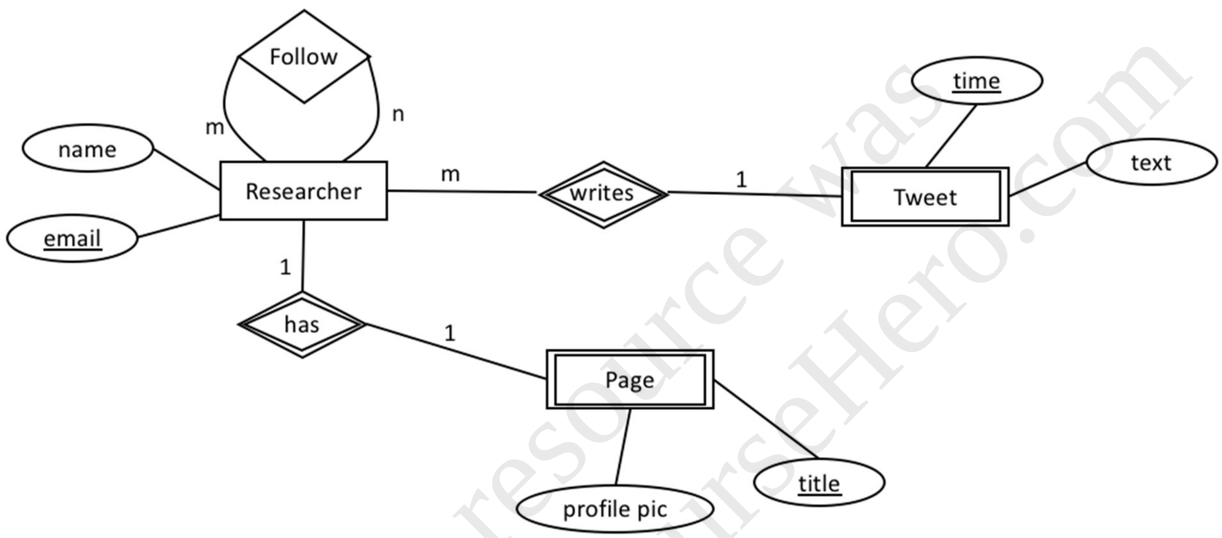
(a) [15 points] Design an ER diagram for the following Online Tutoring System.

- There are two types of users: Tutors and tutees.
- One individual can be both a tutor and a tutee.
- Every user is identified by a user name. A user also has email and a password.
- Each course has an ID and a name. However, the ID alone is not sufficient to distinguish a course from the others, and it must be associated with its tutor.
- Courses are categorized into different areas. Each area has a unique ID and a name.
- Courses are offered in virtual classes.
- Virtual classes have a start date and an end date.
- A tutee might participate in multiple virtual classes.
- Users can write reviews for other individuals only if they have been related through any virtual class.
- A review includes a score, a comment from the reviewer, and a response from the other individual.



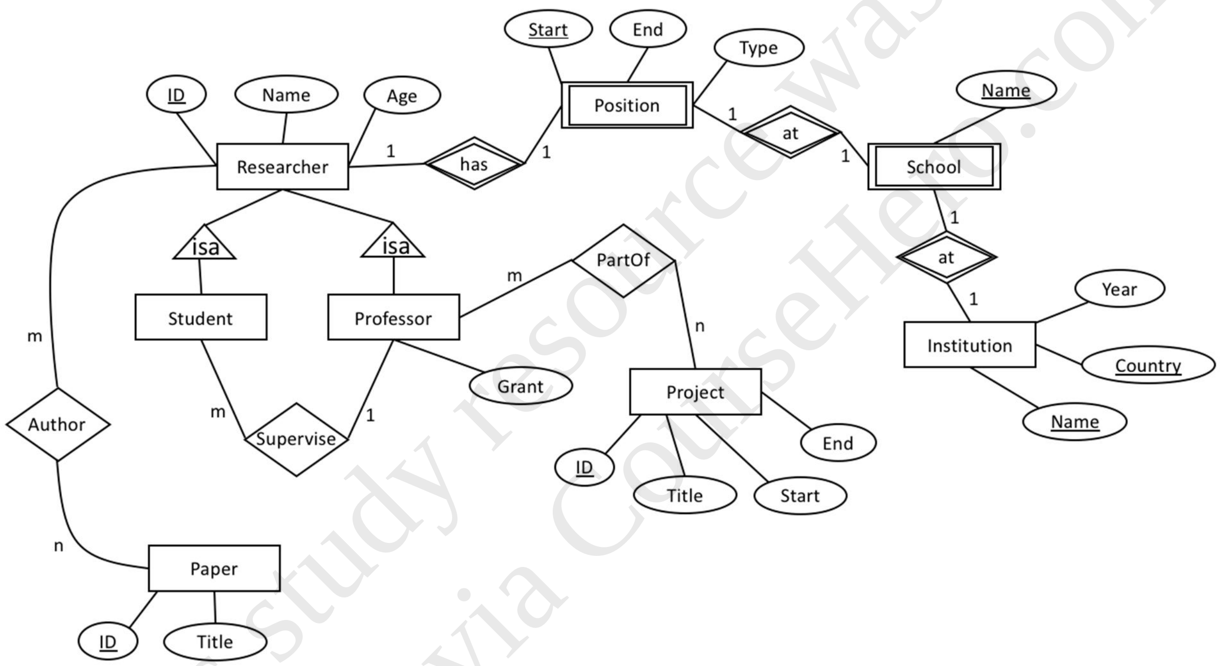
(b) [10 points] You are designing a simplified Twitter social network service, for which you should address the following requirements:

- Users have unique email addresses and you also record their names.
- A user can follow other users.
- Every user has a page, which has a title and (an address to) a profile picture.
- A user writes tweets.
- Each tweet has a text and time.



(c) [15 points] One of the main functionalities of academia is research. Draw an ER diagram which covers information on researchers, institutions, and collaborations among researchers, and includes the following requirements.

- For each researcher, we store his/her name, age, and current position.
- A researcher can either be employed as a professor or a student.
- Every student is supervised by exactly one professor.
- Each institution has a name, country, and start year.
- Each institution has names of its schools. A school belongs to exactly one institution.
- Researchers author papers. The titles of papers should be also stored.
- For each professor, we maintain information on what projects (title, start and end dates) he or she is involved in. We also store the total amount of grant money he or she was the main recipient.



Question 3 [35 points]

In an online multiplayer game, players register with a username and create characters to fight with others in an island. For this purpose, they choose a hero and a name for their character. Characters can be used in islands as long as they are not eliminated.

`Heroes(`<u>`hero`</u>`, type, race, numItems, size)`

`Characters(`<u>`name`</u>`, hero, username, created)`

`Islands(`<u>`name`</u>`, date)`

`Outcomes(`<u>`character, island`</u>`, result)`

- The relation `Heroes` stores unique names (`hero`), the type ('strength', 'agility', and 'intelligence'), the `race` ('Human', 'Orc', 'NightElf', and 'Undead'), number of items the hero can carry, and `size` of the hero (ranging from 1 to 10).
- The relation `Character` records the character's name, a hero, and the time it was created.
- The relation `Islands` maintains the name and date of an island created for a game.
- The relation `Outcomes` gives the result ('`won`', '`survived`', and '`eliminated`') for each character in each island.

Write Relational Algebra expressions for the following queries.

(a) [5 points] Find the name and lifetimes of characters that were eliminated in islands.

- $R1 \leftarrow \sigma_{result='eliminated'}(Outcomes) \bowtie_{Outcomes.island=Islands.name} Islands$
  $\pi_{Characters.name,date-created}(Characters \bowtie_{Character.name=Outcomes.character} R1)$

(b) [5 points] Find players' usernames who chose heroes from all types.

- $\sigma_{username}((Characters \bowtie Heroes) \div \pi_{type}(Heroes))$

(c) [6 points] Find the year in which the last 'Undead' hero was chosen for a character.

- $R1 \leftarrow Character \bowtie \sigma_{race='Undead'}(Heroes)$
  $R2 \leftarrow \rho R2(R1)$
  $\pi_{created}(Characters) - \pi_{R1.created}(R1 \bowtie_{R1.created<R2.created} R2)$

(d) [6 points] Find players' usernames who have not lost any character.

- $R1 \leftarrow$
  $\pi_{username}(\sigma_{result='eliminated'} Outcomes \bowtie_{Outcomes.character=Character=name} Characters)$

  $\pi_{username}(Characters) - R1$

(e) [6 points] Find the largest size of heroes for each race.

- $R1 \leftarrow \pi_{race,size}(Heroes)$
  $R2 \leftarrow \rho R2(R1)$
  $\pi_{race,size}(Heroes) - (R1 \bowtie_{R1.race=R2.race \wedge R1.size<R2.size} R2)$

(f) [7 points] Find distinct pair of usernames who have at least one hero in common.

- $R1 \leftarrow \rho R1(Characters)$
  $R2 \leftarrow \rho R2(Characters)$

$R3 \leftarrow \rho R3_{u1,u2}(\pi_{R1.username,R2.username}(R1 \bowtie_{R1.hero=R2.hero \wedge R1.username \neq R2.username} R2))$

$R4 \leftarrow \rho R4(R3)$

$R5 \leftarrow \rho R5_{u1,u2}(\pi_{R3.u1,R3.u2}(R3 \bowtie_{R3.u1=R4.u2 \wedge R3.u2=R4.u1} R4))$

$R3 - R5$