# Database Management Systems (COP 5725)

Spring 2014

Instructor:
Dr. Markus Schneider

TAs:
Soham Das
Yan Qiao
Yi Wang

## 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 | 20 | |
| Question 2 | 25 | |
| Question 3 | 30 | |
| Question 4 | 25 | |
| Total | 100 | |

## Question 1 (Knowledge Questions) [20 points]

Explain the following notions as precisely as possible in your own words.

a) What is the difference between a weak entity set and a strong entity set? [4 points]

Strong entity set: the entities exist autonomously and can be uniquely identified within an entity set by their key attributes;
Weak entity set: the entities are dependent in their existence from another, superior entity and uniquely identified only in combination with the key of a superior entity (The superior entity can be a strong entity or a weak entity.)

b) The Cartesian Product is one of the five basic relational algebra operators. The Division (Quotient) operator is a derived relational algebra operator. Is the statement *The Division operator is the opposite of the Cartesian Product operator* correct or incorrect? Please argue in either direction. [6 points]

Answer:

The statement is correct. Consider the following tables $R$, $S$, and $T$ given as

| $R$ | $C$ |
|---|---|
| | 4 |
| | 8 |

| $S$ | $D$ |
|---|---|
| | 3 |
| | 1 |
| | 7 |

| $T$ | $C$ | $D$ |
|---|---|---|
| | 4 | 3 |
| | 4 | 1 |
| | 4 | 7 |
| | 8 | 3 |
| | 8 | 1 |
| | 8 | 7 |

Obviously, $T = R \times S$ holds. *All* tuples of $R$ are concatenated with *all* tuples of $S$. Therefore, we obtain

| $T$ | $C$ | $D$ |
|---|---|---|
| | 4 | 3 |
| | 4 | 1 |
| | 4 | 7 |
| | 8 | 3 |
| | 8 | 1 |
| | 8 | 7 |

$T \div S =$

| $R$ | $C$ |
|---|---|
| | 4 |
| | 8 |

$T \div R =$

| $S$ | $D$ |
|---|---|
| | 3 |
| | 1 |
| | 7 |

c) What is the difference between an inner join and an outer join? [4 points]

Inner join: the result does not contain those tuples that did not find a partner
Outer joins: the result contains those tuples that did not find a partner. The result tuples are filled with null values

d) [6 points] Assume you have the following relation:

```
R | X |  Y
-----------
  | 1 | 'A'
  | 2 | 'B'
  | 3 | 'C'
```

Determine the Relational Algebra expression that creates a new relation S which looks as follows:

```
S | X1 | Y1  | X2 | Y2
-----------------------
  | 1  | 'A' | 1  | 'A'
  | 2  | 'B' | 2  | 'B'
  | 3  | 'C' | 3  | 'C'
```

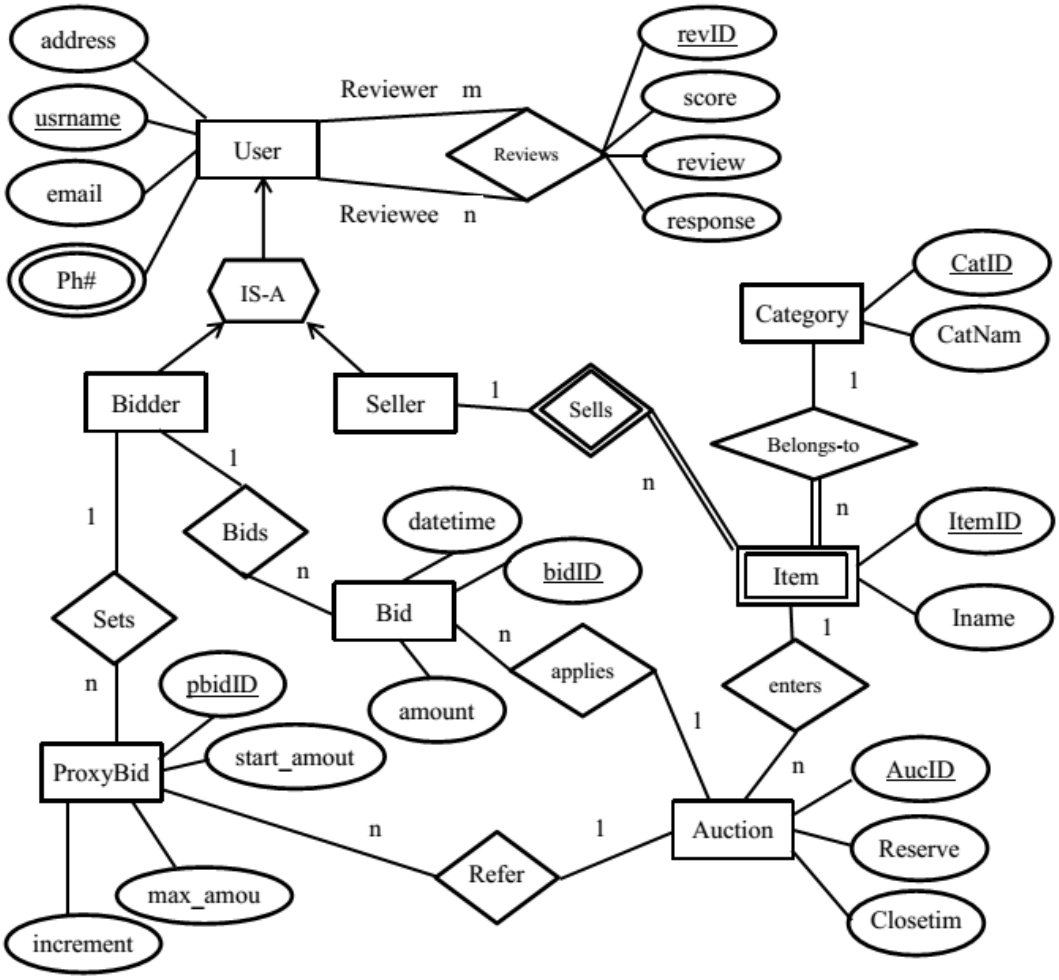
**Solution:**

$$\rho_S(\sigma_{X1=X2 \wedge Y1=Y2}(\rho_{X1 \leftarrow X, Y1 \leftarrow Y}(R) \times \rho_{X2 \leftarrow X, Y2 \leftarrow Y}(R)))$$

## Question 2 (ER Model) [25 points]

Based on the following requirements, design an ER diagram for the database of the following Online Auction System. For each entity set, mark clearly the primary key (If the primary key is not specified by requirement, use your best knowledge to add your own key or use existing attributes). For each relationship you identified, state the cardinalities (1:1, 1:m or m:n) on the entities participating in this relationship.

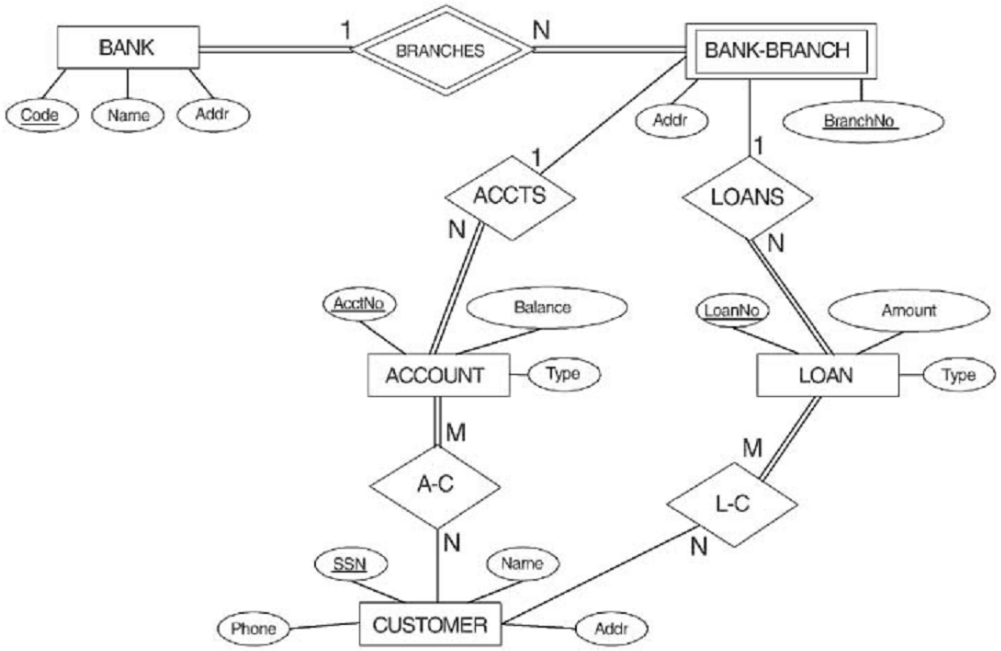
- There are two types of users: Bidders and Sellers
- Both types of users exist as users (identified by a username) that represent individuals. Individuals have email, address, and up to 3 contact phone numbers.
- One individual can exist as both bidder and seller.
- Items to be auctioned are categorized. Each category has a unique ID and a name. Each item also has an ID and a name. The item id alone is not enough to distinguish different items from each other; instead, it must be associated with its seller.
- Auctions can have a reserve price and an item can be auctioned more than once either by a different user or if the reserve price in previous auctions was not met.
- Regular bids are entered by bidders before auction close time.
- Bidder can also set proxy bids for an auction. Proxy bids are automatically incrementing bids that define a maximum amount, a starting amount and an increment over the max bid so that the amount can automatically be updated whenever a higher bid is entered. Every time a higher bid is entered the system parses through the proxy bids and enters regular bids depending on the proxy bids found for that item.
- Bidders and sellers can provide reviews of other individuals only if they have been related through any auction. Reviews include a score (min 0, max 5) and detailed comments from the individual that inputs the review plus a response from the individual who is targeted by the review.




[Each entity set needs to have a primary key / partial key. Grading criteria:
* Missing keys/attributes: -1 each, -5 max
* Bad design: -1~10, depending on how bad it is.
* Missing relationship/entity: -2 each.
* Missing/incorrect cardinalities: -1 each, -5 max.]

## Question 3 (ER Model) [30 points]

Consider the following ER model in a bank database:




a) Transform the ER model into a corresponding database schema. Identify the primary keys and foreign keys. You need not include data types in the schema [14 points]

[Grading criteria: 2 points for each schema. incorrect key: -1. Incorrect attribute: -2]

Bank(code, name, addr)
BankBranch(branchNum, code, addr)
Customer(ssn, name, phone, addr)
Account(accNum, balance, accType, code, branchNum)
Loan(loanNum, type, amount, code, branchNum)
AccntsCust(accNum, ssn)
LoansCust(loanNum, ssn)

b) Using SQL, create the table BANK. Your answer should be recognized by oracle [4 points].

```
CREATE TABLE BANK(
      Code INTEGER PRIMARY KEY,
      Name VARCHAR(50),
      Addr VARCHAR(1000)
)
```

c) Is there any weak entity set? If so, give its name(s), partial key and identifying relationship. [6 points]

Weak entity set: Bank-Branch
Partial key: BranchNo
Identifying relationship: Branches

d) Suppose that every customer must have at least one account but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1000 loans. How does this show up on the (min, max) constraints? [6 points]
Write the answer in this form, i.e. [Relationship -> Entity1(min, max), Entity2(min, max)]

A-C -> ACCOUNT (1, *), CUSTOMER (1, *)
L-C -> CUSTOMER (0, 2), LOAN (1, *)
LOANS->BANK-BRANCH (0, 1000), LOAN (1, 1)

## Question 4 (Relational Algebra) [25 points]

We have 3 relations, all about movies: Columns in the primary key are underlined.

- Movies(<u>title, year</u>, director, country, rating, genre, gross, producer)
- Actors (<u>title, year, character name</u>, actor). (Title, Year) form a foreign key referencing Movies.
- Awards (<u>title, year, award</u>, result). (Title, Year) form a foreign key referencing Movies.

The values of award could be, e.g., Oscar, Golden Globe, Cesar, Palm, etc; the values of result are won or nominated. The primary key also indicates that if a movie won an award, the corresponding nomination entry is not recorded in the database. Express the following queries in relational algebra.

a. List actors who played more than one character in the same movie [5 points].

$\rho_{R1}$(Actors)
$\rho_{R2}$(Actors)
$\pi_{R1:actor}(\sigma_{R1:title=R2:title \wedge R1:year=R2:year \wedge R1:actor=R2:actor \wedge R1:cname!=R2:cname}(R1 \times R2))$

b. Find movies made in the 80s that won every award they were nominated for [5 points].

$\pi_{title;year}(\sigma_{year>1979 \wedge year<1990 \wedge result='won'}(Awards)$ -
$\pi_{title;year}(\sigma_{year>1979 \wedge year<1990 \wedge result='nominated'}(Awards)$

c. Find actors who only act in high grossing (more than $75 million) movies [5 points].

$\rho_{R1}(\sigma_{gross<75}(Movie \infty Actors))$
$\pi_{actors}(Actors) - \pi_{actors}(R1)$

d. Find the movie directed by Quentin Tarantino which has the highest rating [5 points].

$\rho_{R1}(\sigma_{director=' Quentin\ Tarantino'}(Movies))$
$\rho_{R2}(R1)$
$\rho_{R3(1->title;2->year)}(\sigma_{R1:rating<R2:rating}(R1 \times R2))$
$\pi_{sid}(R2 - R3)$

e. Find all pairs of movies (m1, m2) nominated for the same award, such that m1 has higher rating than m2, but m2 won the award [5 points].

$\rho_{R1}$(Movies ∞ Actors)

$\rho_{R2}$(Movies ∞Actors)

$\rho_{R1:title;R1:year;R2:title;R2:year}(\sigma_{R1:award=R2:award\wedge R1:rating>R2:rating\wedge R1:result='nominated'\wedge R2:result='won'}(R1 \times R2))$